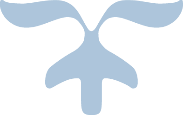


Administración de servidores

231011 Kevin Antonio Andrade López



universidad tecnologica de aguascalientes

# Unidad 2: Gestión de servidores – Servicios básicos.

[Unidad 2: Gestión de servidores – Servicios básicos. 0](#_Toc177455910)

[Actividad 1: 0](#_Toc177455911)

[Apartado A: Mapa semántico de Linux 0](#_Toc177455912)

[Apartado b: Cuestionario 1](#_Toc177455913)

[Actividad 2 – Primeros pasos – NFS Linux 1](#_Toc177455914)

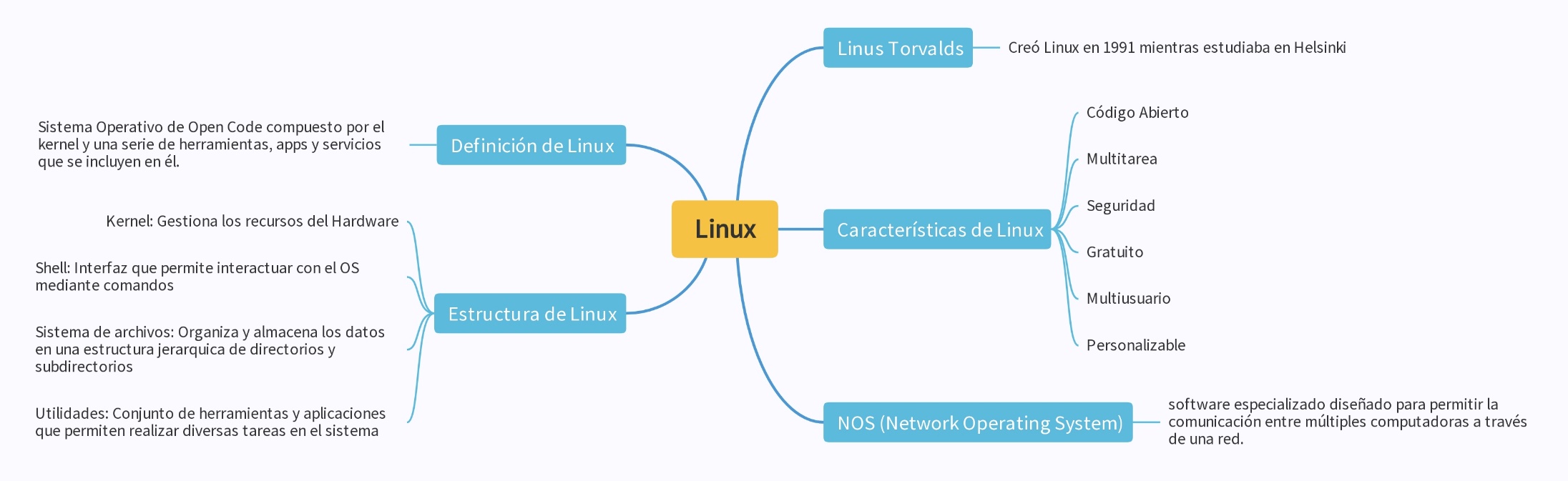
[Diagrama de flujo “Instalación Distro” 2](#_Toc177455915)

[Actividad 3 Reporte Practica No. 1 3](#_Toc177455916)

[Ejercicios guiados 3](#_Toc177455917)

# Actividad 1:

## Apartado A: Mapa semántico de Linux



Bibliografía:

*Definición de Linux*. (n.d.). https://www.redhat.com/es/topics/linux/what-is-linux

Celestyn. (2022, October 26). *Sistema operativo de red (NOS)*. TechEdu. https://techlib.net/techedu/sistema-operativo-de-red-nos/

colaboradores de Wikipedia. (2024, August 27). *Linus Torvalds*. Wikipedia, La Enciclopedia Libre. https://es.wikipedia.org/wiki/Linus\_Torvalds

## Apartado b: Cuestionario

1. ¿Cuál es el nombre que identificamos para acceder al escritorio de Linux?

Gestor de ventanas, ejemplos: GNOME, KDE Plasma, XFCE y LXDE

1. De la estructura jerárquica del sistema de archivo, ¿qué componente o carpeta permite el acceso al hardware?

/dev

1. ¿En qué se diferencia el distro de Linux al Kernel de Linux?

**Kernel de Linux:** Es el núcleo del sistema operativo, responsable de gestionar el hardware y proporcionar servicios básicos a las aplicaciones.

**Distribución de Linux (Distro):** Es un paquete completo que incluye el Kernel de Linux, junto con una colección de software, herramientas, y utilidades adicionales. Ejemplos de distribuciones son Ubuntu, Fedora, Debian y Arch Linux.

1. ¿Qué componente de Linux es común en todas las distribuciones del sistema operativo?

El Kernel de Linux es el componente común

El Kernel es la parte más importante de cada distro de Linux, es la conexión entre el hardware y el software, el encargado de repartir las tareas, la cantidad de recursos a los que se pueden acceder y de los que se pueden disponer. Para poder llegar a esto y siquiera poder manipularlo existe una carpeta con la ruta /dev que es la que mantiene los archivos necesarios de la comunicación entre el hardware y el software. La forma en la que se puede acceder a esta carpeta podría ser, ya sea por parte del Shell o incluso por un gestor de ventanas, que es el encargado de la parte gráfica al navegar entre las distintas carpetas, dándonos incluso acceso a varias de manera simultánea. Esta puede variar dependiendo de la distro que se haya usado, pero todas las distro llevan el mismo núcleo: el Kernel de Linux.

# Actividad 2 – Primeros pasos – NFS Linux

1. ¿Cuál es la importancia de File System para un S.O.?

El sistema de archivos es crucial para los S.O. ya que organizan y gestionan la forma en la que se almacenan y recuperan los datos en un dispositivo de almacenamiento. Básicamente se encargan de estructurar los archivos

1. ¿Qué es una partición lógica?

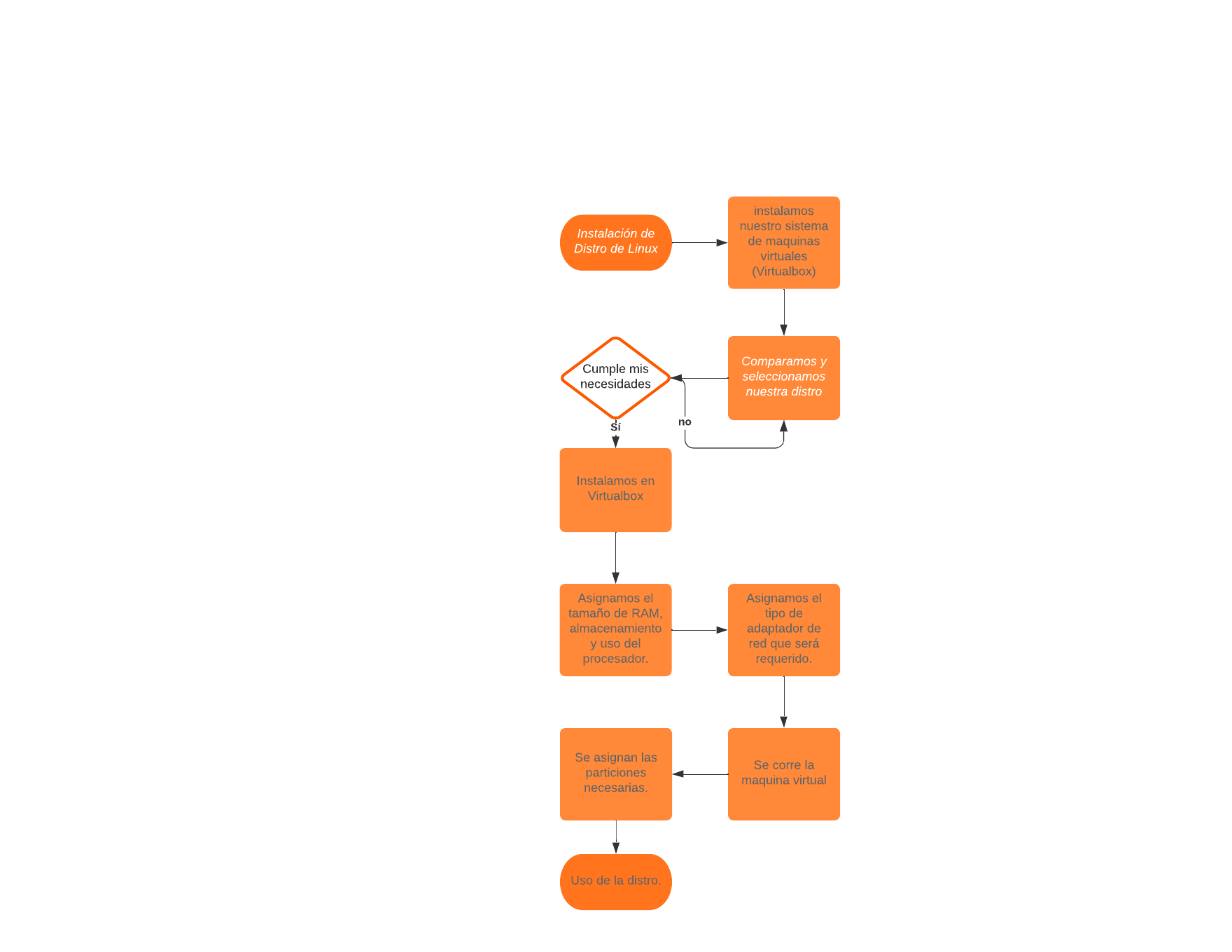
Una partición lógica es un tipo de partición que se encuentra dentro de una partición extendida en un disco duro:

* **Funcionamiento**: En un disco duro, puedes tener un máximo de cuatro particiones primarias. Si necesitas más particiones, puedes crear una partición extendida, que actúa como un contenedor para particiones lógicas. Las particiones lógicas se crean dentro de esta partición extendida.
* **Propósito**: Las particiones lógicas permiten superar la limitación de cuatro particiones primarias en un disco. Esto es útil cuando necesitas organizar tu disco en más de cuatro secciones separadas, por ejemplo, para diferentes sistemas operativos, datos de usuario, y particiones de intercambio (swap).

1. Completa la siguiente tabla

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo** | **Sistema (funcionamiento, en donde surgió y por qué)** | **Funcionamiento** | **Tamaño** |
| **EXT2** | Surgió en 1993, desarrollado por Rémy Card para Linux.  Sistema de archivos sin journaling, soporta archivos grandes y nombres largos, ofreciendo mejor rendimiento y capacidad. | Diseñado para reemplazar el sistema de archivos extendido (EXT) original, ofreciendo mejor rendimiento y capacidad. | 2 tb |
| **EXT3** | Introducido en 2001, también desarrollado para Linux. Similar a EXT2 pero con journaling, lo que mejora la recuperación de datos tras fallos. | Añadir journaling a EXT2 para mejorar la integridad de datos y reducir el tiempo de recuperación. | 16 tb |
| **EXT4** | Lanzado en 2008, desarrollado por Theodore Ts’o y otros para Linux. Mejoras en rendimiento, escalabilidad y características como extents y verificación de suma de comprobación. | Mejorar las limitaciones de EXT3 en cuanto a tamaño de archivos y directorios, y rendimiento. | 1 eb |
| **SWAP** | Concepto general en sistemas operativos, ampliamente utilizado en Unix y Linux.  Espacio en disco utilizado como memoria virtual para ampliar la RAM disponible. | Permitir que el sistema operativo utilice espacio en disco como memoria adicional cuando la RAM está llena. | Depende de la RAM |
| **SMBFS** | Desarrollado en el contexto de la red SMB/CIFS de Microsoft. Sistema de archivos que permite montar recursos compartidos de Windows en Linux. | Facilitar la interoperatividad entre sistemas Windows y Linux, permitiendo el acceso a archivos compartidos. | Basado en Red |
| **VFAT** | Introducido por Microsoft en Windows 95. Variante de FAT que soporta nombres de archivos largos, compatible con Windows. | Proporcionar compatibilidad con sistemas Windows y permitir nombres de archivos largos. | Hasta 4 gb por archivo |
| **NFS** | Desarrollado por Sun Microsystems en 1984. Sistema de archivos de red que permite compartir archivos entre sistemas Unix/Linux. | Facilitar el acceso remoto a archivos y directorios a través de una red. | Hasta 256 TB |

## Diagrama de flujo “Instalación Distro”



# Actividad 3 Reporte Practica No. 1

## Ejercicios guiados

* 1. Use el comando para averiguar qué hace cada comando

man

|  |  |  |
| --- | --- | --- |
| Comando | Descripción | Captura Pantalla |
| ls | Mostrar el contenido de un directorio. |  |
| cat | Muestra el texto de un archivo. Concatena las cadenas y las muestra en texto plano. |  |
| cut | Elimina fragmentos en cada línea de un archivo |  |
| cd | Cambiar el directorio en el que se está trabajando. |  |
| cp | Copia archivos y directorios |  |
| mv | Mueve archivos |  |
| mkdir | Crea directorios |  |
| touch | Crea un archivo vacío o modifica la fecha de modificación de uno existente |  |
| wc | Imprime cuantos saltos de line, palabras y bytes hay en el archivo |  |
| passwd | Cambia la contraseña del usuario. |  |
| rm | Borra un archivo |  |
| rmdir | Borra un directorio |  |
| more | Muestra contenido de un archivo por paginas. |  |
| less | Muestra todo el archivo en una sola pagina deslizable (scroll) |  |
| whereis | Ubicación de archivos y comandos, busca sus carpetas |  |
| head | Muestra las primeras 10 lineas de un texto |  |
| tail | Muestra las ultimas 10 lineas de un texto |  |
| sort | Ordena las lineas de un archivo. |  |
| tr | Transforma o elimina caracteres de un texto |  |
| chmod | Cambia los permisos de un directorio o archivo. |  |
| grep | Busca cadenas dentro del texto |  |

Para que sirve el comando ls -lia

* + - ¿Qué infieres?

Sirve para listar los permisos de los archivos, la información y todos los archivos, incluyendo los ocultos.

* 1. Para que sirve el comando Locate

Sirve para encontrar documentos o archivos

* 1. Busque un archivo , utilizando el comando find. ¿Qué sintaxis uso y cuál es la ruta absoluta? Muestra sintaxis y Captura de Pantalla

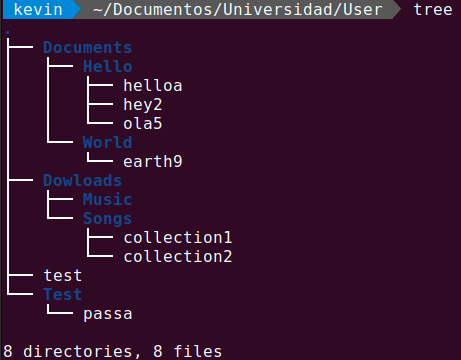


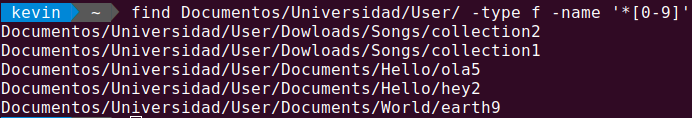
## c) Ejercicios exploratorios

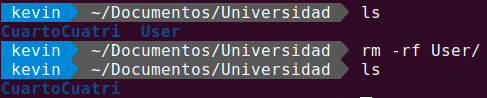
1. Hay un comando en la tabla anterior que no tiene una página man ¿Cuál es y por qué cree que el comando no tiene una página de manual?

Cd, porque es un comando que entra desde bash

2. Usando los comandos de la tabla de la sesión anterior, cree el siguiente árbol de archivos. Los nombres que comienzan con mayúscula son directorios y los que están en minúscula son archivos.



* Busque y capture pantalla de dentro del árbol todos los archivos que terminen con un numero
* Elimine la carpeta y todo lo que tiene dentro con un solo comando.



|  |  |  |
| --- | --- | --- |
| Comando | ¿para qué? | Captura de pantalla |
| Man | Se utiliza para acceder a la documentación de referencia de los comandos y programas instalados en el sistema |  |
| Info | Similar a man, pero proporciona una documentación mas detallada y estructurada. |  |
| Find | Se utiliza para buscar archivos y directorios en el sistema de archivos de linux |  |

## Reflexión

Comprender la utilidad de los comandos básicos de Linux es crucial para lograr un trabajo eficiente y efectivo en este sistema operativo. Familiarizarse con estos comandos no solo incrementa la productividad, sino que también optimiza el flujo de trabajo al permitir llevar a cabo tareas que, de otro modo, requerirían múltiples pasos si se usara una interfaz gráfica. La capacidad de ejecutar diversas tareas de manera rápida y directa mediante el uso de comandos es esencial para una gestión más eficaz del sistema y de los servidores. Además, el conocimiento de estos comandos proporciona una visión más profunda del funcionamiento interno del sistema, lo que nos permite abordar y resolver problemas de manera más efectiva. Dominar los comandos básicos de Linux no solo facilita la realización de tareas cotidianas, sino que también potencia nuestra comprensión general del sistema.

# Actividad 4

## A) Cuestionario

1. **¿Qué diferencia hay entre los nombres de archivos Linux y en Windows en cuanto a sufijos?**

En Linux, los sufijos en los nombres de archivos, como el .txt, no tienen un significado especial y están ahí solo para la comprensión humana, A diferencia de Windows, donde los sufijos pueden determinar el tipo de archivo y como se debe de manejar. Por ejemplo, en Linux, un archivo con el sufijo .txt podría contener cualquier tipo de dato, no necesariamente texto plano.

1. **¿Qué comando se utiliza en Linux para imprimir el directorio actual de trabajo?**

El comando pwd se utiliza en Linux para imprimir el directorio de trabajo actual.

1. **¿Qué representa la tilde (~) en la salida de comando pwd?**

La tilde es un carácter especial que representa el directorio home del usuario.

1. **¿Cuál es la diferencia principal entre una ruta absoluta y una ruta relativa en un sistema de archivos?**

La diferencia principal es que una ruta absoluta contiene cada paso de la ruta desde la parte superior del sistema de archivos (/) hasta el destino final, comenzando siempre con un/. En cambio, una ruta relativa es más corta y solo tiene significado en la relación a la ubicación actual del usuario en el sistema de archivos.

1. **¿Qué representan los directorios especiales (.) y (..) en el sistema de archivos?**

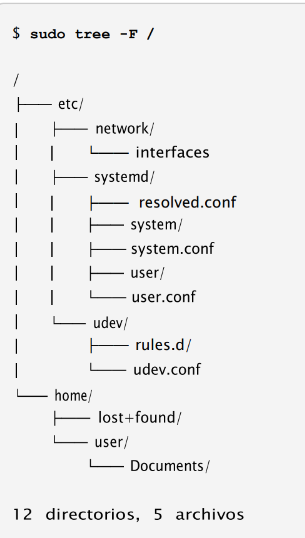
El directorio (.) representa la ubicación actual, mientras que el directorio (..) representa el directorio padre.

## B) Ejercicios guiados

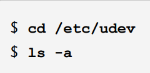
1. Para cada una de las siguientes rutas, identifique si es absoluta o relativa

|  |  |
| --- | --- |
| /home/user/Downloads | Absoluta |
| ../Reports | Relativa |
| /var | Absoluta |
| docs | Relativa |
| / | Absoluta |

1. Observe la siguiente estructura de archivos. Nota: Los directorios terminan con una barra inclinada (/) cuando se ejecuta tree con la opción -F. Necesitará privilegios elevados para ejecutarlo en el directorio principal root (/). El siguiente es un resultado de ejemplo y no es indicativo de una estructura de directorio completa. Úselo para responder las siguientes preguntas



Un usuario ingresa los siguientes comandos:



¿Cuál será la salida del comando ls -a?

La salida del comando ls -a en el directorio /etc/udev/ será

. .. rules.d

. representa el directorio actual

.. representa el directorio padre

rules.d es el archivo presente en /etc/udev/.

1. Ingrese el comando mas corto posible para cada una de las siguientes situaciones:

* Su ubicación actual es root (/). Ingrese el comando para navegar a lost+found dentro del directorio home (ejemplo):
* 
* Su ubicación actual es root (/). Ingrese el comando para navegar al directorio llamado: /etc/network/.

$ cd etc/network

* Su ubicación actual es /home/user/Documents/. Navegue al directorio llamado /etc/.

$ cd ../../home/user/

* Su ubicación actual es /etc/systemd/system/. Navegue al directorio llamado /home/user/.

$ cd ../../../home/user/

1. Considere los siguientes comandos

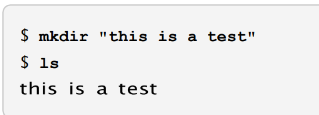


¿Cuál es el resultado del comando final pwd?

/etc/systemd

## C) Ejercicios exploratorios

1. Supongamos que un usuario ha ingresado los siguientes comandos:



¿Qué comando cd le permitiría ingresar a este directorio?

cd this\ is\ a\ test/

1. Intente esto nuevamente, pero después de escribir cd, presione la tecla TAB ¿Qué se muestra ahora en la solicitud?

cd this\ is\ a\ test/

Este es un ejemplo de \_autocompletado\_, la cual es una herramienta invaluable no solo para ahorrar tiempo, sino también para evitar errores ortográficos.

1. Intente crear un directorio cuyo nombre contenga un carácter \. Visualice el nombre del directorio con ls y elimine el directorio.

Mkdir “this/is/a/test”

## Reflexión:

Estos comandos son esenciales para la gestión y la navegación en el entorno del sistema. Por ejemplo, el comando pwd muestra el directorio en el que nos encontramos en ese momento, mientras que con cd podemos cambiar de un directorio a otro. Si queremos crear nuevos directorios, utilizamos mkdir, y para listar los archivos y directorios que existen en el sistema, empleamos ls. Estos comandos nos permiten realizar una amplia gama de tareas básicas en Linux, lo que hace que sea crucial conocerlos y saber usarlos correctamente.

Además, dominar estos comandos no solo mejora nuestra capacidad para realizar tareas, sino que también nos ahorra tiempo, ya que podemos ejecutar trabajos de manera más rápida y directa. Al familiarizarnos con ellos, conseguimos optimizar nuestra productividad y eficacia en el manejo del sistema.

En el contexto de la navegación por el sistema, es importante entender el concepto de rutas. Existen dos tipos principales de rutas en Linux: la ruta absoluta y la ruta relativa. Una ruta absoluta proporciona la ubicación completa de un archivo o directorio, comenzando desde la raíz del sistema de archivos. Esto es particularmente útil cuando necesitamos acceder a un archivo específico sin importar en qué directorio nos encontremos.

Por otro lado, las rutas relativas ofrecen solo la ubicación de un archivo o directorio en relación con el directorio actual en el que estamos trabajando. Esto significa que las rutas relativas son más convenientes cuando estamos navegando dentro de un mismo conjunto de directorios y queremos referirnos a otros archivos sin necesidad de especificar la ruta completa.

# Actividad 5:

## Privilegios.

|  |  |  |
| --- | --- | --- |
| Instrucción | ¿Qué hace? | ¿Cómo lo representas de otra forma? |
| chmod 0777 | Otorga permisos completos (lectura, escritura y  ejecución) para todos (usuario, grupo y todos) | chmod a=rwx |
| chmod 0666 \* | Intenta dar permisos de lectura y escritura a  todos los archivos seleccionados, sin ejecución. | chmod a=rw \* |
| chmod –r 0644 \* | Recursivamente da permisos de lectura/escritura al usuario y solo lectura a grupo y otros para todos los archivos y directorios internos de la  carpeta | chmod –r u=rw, go=r \* |
| chmod o=rwx \* | Otorga permisos de lectura, escritura y ejecución  a “otros” | chmod 007 \* |
| chmod go= \* | Elimina todos los permisos para grupo y otros en  todos los archivos seleccionados. | chmod g=,o= \* |
| chmod u=rx | Da permisos de lectura y ejecución al propietario,  quitando escritura. | chmod 500 |
| chmod =rwx, g=rw, o= \* |  |  |
| chmod=w | Da permisos de escritura a todos sin especificar  usuarios. | chmod a=w |

## Problemario

* Un administrador de servidores de Linux desea cambiar el permiso de todos los archivos de un directorio de forma recursiva. ¿Qué comando debe de emplear?

chmod -r 777 Directorio

* Deseo asignar todos los privilegios para todos a una carpeta llamada gerencia. ¿Qué debo de hacer? chmod 777 gerencia
* Deseo que solo el propietario pueda escribir, leer y que todos los demás puedan ejecutar una carpeta. chmod 711 fichero
* Deseo asignarle todos los privilegios al propietario de la carpeta. ¿qué instrucción debo de ejecutar? chmod 700 fichero
* Si deseo que nadie pueda ejecutar una carpeta. ¿Qué comando emplearía? chmod a-x fichero

## Reflexión

Aprendí cómo gestionar permisos de archivos y directorios, lo cual es esencial en la administración de sistemas. Entender cómo modificar estos permisos me permitirá garantizar la seguridad y el acceso adecuado en un entorno multiusuario. Primero, comprendí que, para cambiar los permisos de manera recursiva en un directorio, es decir, modificar los permisos de todos los archivos y subdirectorios dentro de él, el comando adecuado es ***chmod -r 777 Directorio***. Esto es útil cuando se necesita aplicar cambios masivos de permisos sin tener que hacerlo archivo por archivo. Este conocimiento sobre el comando es muy valioso, ya que me permitirá administrar mejor los permisos de archivos y carpetas, manteniendo control sobre quién puede hacer qué con los archivos, y asegurando que los recursos del servidor estén protegidos adecuadamente. Es importante tener en cuenta que el comando chmod -R 777 debe utilizarse con precaución, ya que al otorgar permisos completos a todos los usuarios, se corre el riesgo de comprometer la seguridad del sistema si se aplica de manera indiscriminada. Sin embargo, conocer esta herramienta y su funcionamiento es invaluable para escenarios donde se requiere un acceso amplio temporal o en entornos de desarrollo donde la colaboración abierta es necesaria.

El manejo eficiente de los permisos no solo me permite administrar mejor los archivos y directorios, sino que también me da el control sobre quién puede hacer qué con cada recurso. Este control es esencial para asegurar que los datos sensibles permanezcan protegidos, mientras se facilita el acceso necesario a los usuarios que requieren modificar o ejecutar ciertos archivos. Además, este conocimiento me ayudará a mitigar problemas de seguridad en el futuro, asegurando que los recursos del servidor estén bien protegidos y accesibles solo para aquellos con los permisos adecuados.

# Reto 01:

El administrador de un servidor Unix desea crear un archivo.

* Desde línea de comandos:

¿Qué instrucciones emplearías para lograrlo? Identifica al menos 4 maneras

diferentes.

-touch archivo

-echo “texto” > archivo

-cat > archivo

- >

-printf

* Y si no quiere usar línea de comandos, pero necesita que sea desde terminal. ¿Cómo lo haría? Identifica 3 maneras diferentes:

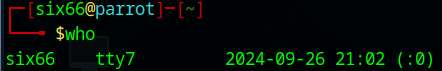
-Nano

-Vim

-mc

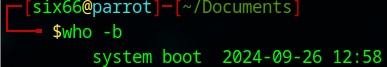
¿Para que usarías el comando who?

Para ver quien está loggeado:



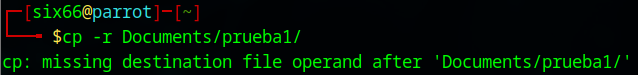
¿Y si deseo saber la fecha y hora en la que inició el sistema?

Who -b



¿Como interpreto el comando: cp -r /dir/subdir/ ?

Copia todo lo que existe dentro del fichero.



¿Cómo puedo copiar todos los archivos con extensión txt de la carpeta descargas a /directorio/subdirectorio/?

cp ~/descargas/\*.txt /directorio/subdirectorio/

¿Cómo identifico un archivo oculto en Linux? ¿y si deseo ocultarlo que debo hacer?

Por el punto que aparece al inicio de su nombre, y la extensión “.hidden”. Para ocultar un archivo solo hay que añadir un punto antes del nombre, esto evitará que aparezca en las carpetas gráficas y el comando ls. Pero con ls –a ya aparece.

## Reflexión:

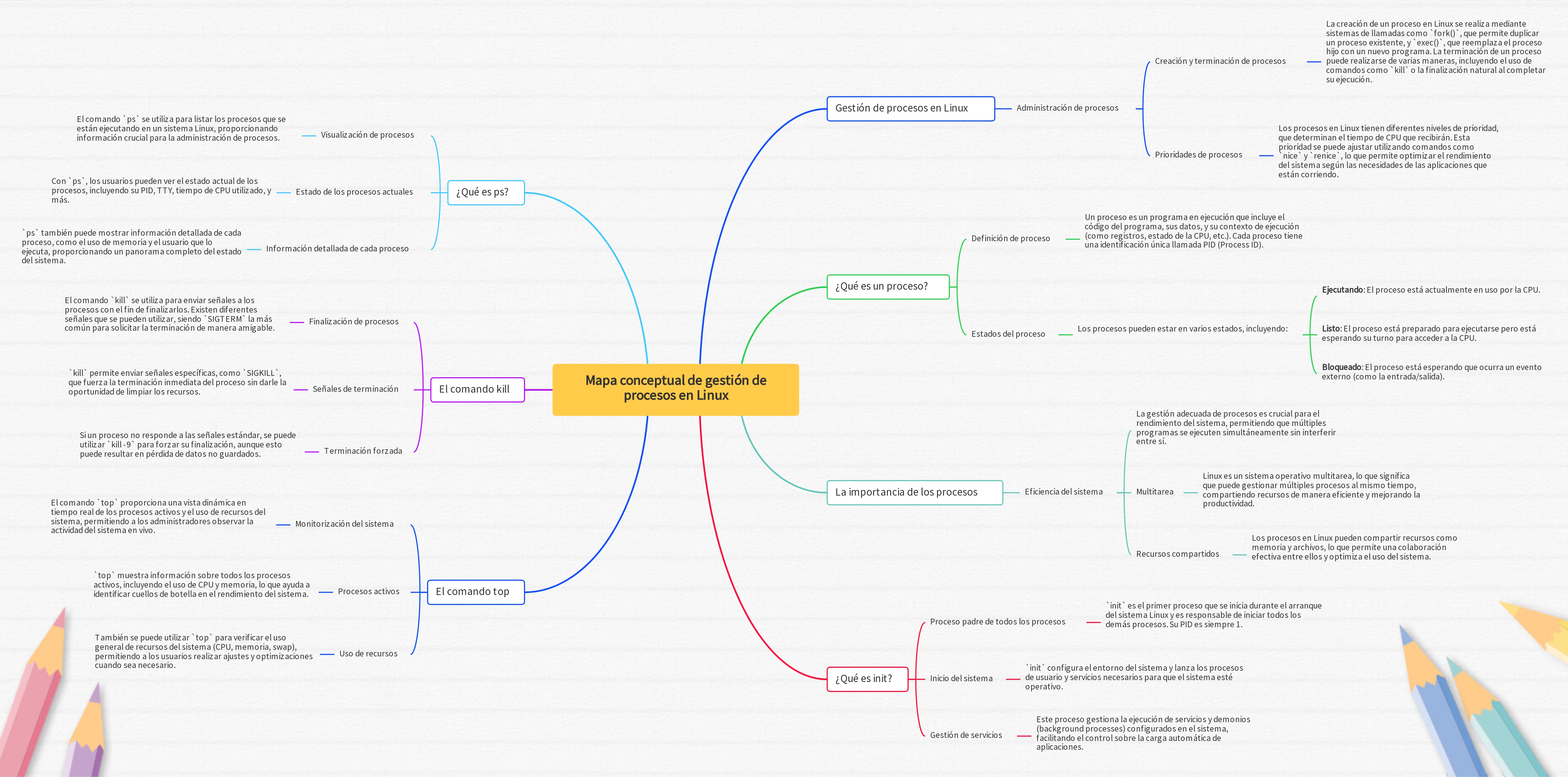
Este reto me enseñó lo flexible que puede ser el sistema para adaptarse a las preferencias del usuario. Con cada comando que investigué me di cuenta de que, aunque cumplen el mismo objetivo, funcionan de formas distintas y eso puede ayudar a la hora de crear archivos. Lo interesante es que incluso pueden adaptarse a las circunstancias; algunos comandos ayudan a crear scripts desde la línea de comandos y otros permiten la creación inmediata de un documento para usarse después. Si quisiéramos armar un script en un editor de texto, tenemos Nano y Vim. Nano ofrece una curvatura de aprendizaje suave, ideal para principiantes. Vim es más poderoso, pero también es muy complicado de entender al principio. Esto me enseña que, aunque los comandos son poderosos, a veces es mejor una interfaz visual para poder llevar un seguimiento de lo que se está escribiendo el script. El comando who me enseño lo sencillo que es obtener información importante sobre el sistema, como ver quien está conectado o cuando fue el último reinicio del sistema. Esto podría ser fundamental en un servidor ya que podríamos ver si algo ocasiona fallas. El manejo de comandos como cp nos permiten entender estructuras jerárquicas de los sistemas de archivos. Estos comandos son fundamentales a la hora de manejar archivos, ya que en muchos entornos no se tiene una interfaz gráfica a la cual hacer copy/paste. Este reto me ayudó a entender cuan poderoso y flexible es este sistema operativo, con herramientas casi para cualquier tarea y flexibles para cualquier solución. Nos ofrece tanto control como personalización, permitiéndonos trabajar de forma más efectiva.

# Actividad 6: Procesos y automatización de tareas

## a) Esquema conceptual de procesos.

Puntos claves

* Procesos
* Importancia
* comandos
* ps (funciones)
* Kill
* variables
* ¿Como identificamos un procesos¿qué función tiene un ls/proc?



## b) Mapa Conceptual automatización de tareas

## c) Ejercicios Guiados

¿Qué procesos ha iniciado el usuario carol?

887 top con 0.3 de uso en el cpu

¿Qué directorio virtual de /proc debe visitar para buscar datos sobre el comando top?

/proc/887/

¿Qué proceso se ejecutó primero?

Systemd por el numero bajo de ID

Complete la tabla

|  |  |  |
| --- | --- | --- |
| Información sobre… | Área de resumen. | Área de tareas |
| Memoria | SI | SI |
| Swap | SI | NO |
| PID | NO | SI |
| CPU time | SI | NO |
| Comandos | NO | SI. |

¿Que comando se usa para leer los siguientes registros binarios?

/var/log/wtmp

last -f /var/log/wtmp

/var/log/btmp

last -f var/log/btmp

/run/log/journal/2a7d9730cd3142f4b1520d6be631836/system.journal

journalctl –file /run/log/journal/2a7d9730cd3142f4b1520d6be631836/system.journal

En combinación con grep, ¿qué comandos usaría para averiguar la siguiente información?

¿Cuando se reinició por ultima vez el sistema?

last -x | grep reboot | tail -n 1

¿Qué discos duros están instalados (kern.log)?

grep -i "sd" /var/log/kern.log

¿Cuando ocurrió el ultimo inicio de sesión (auth.log)?

grep "session opened" /var/log/auth.log | tail -n 1

¿Cuáles dos comandos usarías para mostrar el kernel ring buffer?

Dmesg y journalctl -k

Indique a dónde pertenecen los siguientes mensajes de registro:

◦ Jul 10 13:37:39 debian dbus[303]: [system] Successfully activated service

'org.freedesktop.nm\_dispatcher'

|  |  |
| --- | --- |
| /var/log/auth.log |  |
| /var/log/kern.log |  |
| /var/log/syslog | Aquí |
| /var/log/messages |  |

* Jul 10 11:23:58 debian kernel: [ 1.923349] usbhid: USB HID

core driver

|  |  |
| --- | --- |
| /var/log/auth.log |  |
| /var/log/kern.log | aquí |
| /var/log/syslog |  |
| /var/log/messages |  |

* Jul 10 14:02:53 debian sudo: pam\_unix(sudo:session):

session opened foruser root by carol(uid=0)

|  |  |
| --- | --- |
| /var/log/auth.log | aquí |
| /var/log/kern.log |  |
| /var/log/syslog |  |
| /var/log/message |  |

* Jul 10 11:23:58 debian NetworkManager[322]: <info>

[1562750638.8672] NetworkManager (version 1.6.2) is starting…

|  |  |
| --- | --- |
| /var/log/auth.log |  |
| /var/log/kern.log |  |
| /var/log/syslog | Aquí |
| /var/log/messages |  |

* ¿Tiene información para consultar el comando journalctl sobre las siguientes

unidades?

|  |  |
| --- | --- |
| Unidad | Comando |
| ssh | journalctl -u ssh.service |
| networking | journalctl -u networking.service |
| rsyslog | journalctl -u rsyslog.service |
| cron | journalctl -u cron.service |

## Ejercicios Exploratorios

**Tarea 1**

Para cada uno de los siguientes atajos, indique la especificación de tiempo correspondiente (es decir, las cinco primeras columnas de un archivo de usuario):

|  |  |
| --- | --- |
| @hourly | Una vez por hora. 0 \* \* \* \* |
| @daily | Una vez al dia. 0 0 \* \* \* |
| @weekly | Una vez a la semana. 0 0 \* \* 0 |
| @monthly | Una vez al mes. 0 0 1 \* \* |
| @annually | Una vez al año. 0 0 1 1 \* |

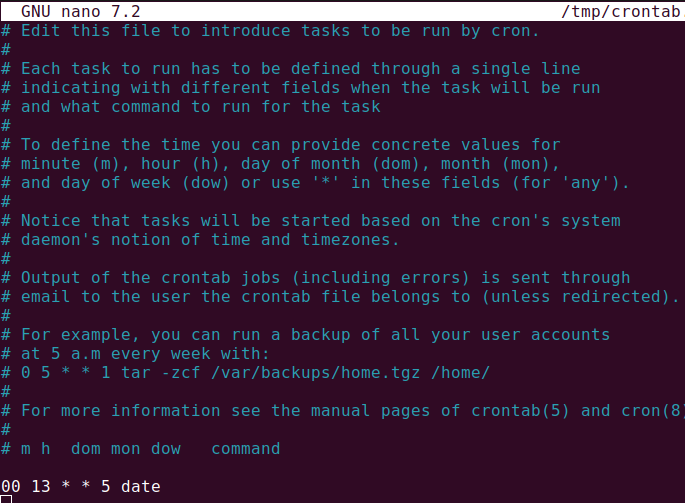
Explique el significado de las siguientes especificaciones de tiempo que se encuentran en un archivo contab

|  |  |
| --- | --- |
| 30 13 \* \* 1-5 | De lunes a viernes a las 13:30. |
| 00 09-18 \* \* \* | De 9 a 18 todos los días. |
| 30 08 1 1 \* | Cada primero de enero a las 8:30 am |
| 0,20,40 11 \* \* Sun | Todos los domingos a las 11:00, 11:20, 11:40 |
| 00 09 10-20 1-3 \* | Entre el 10 y el 20 de Enero a Marzo a las 9:00 |
| \*/20 \* \* \* \* | Cada 20 minutos. |

Asumiendo que usted está autorizado a programar trabajos con cron como un usuario ordinario ¿Qué comando usaría para crear su propio archivo crontab? (captura)

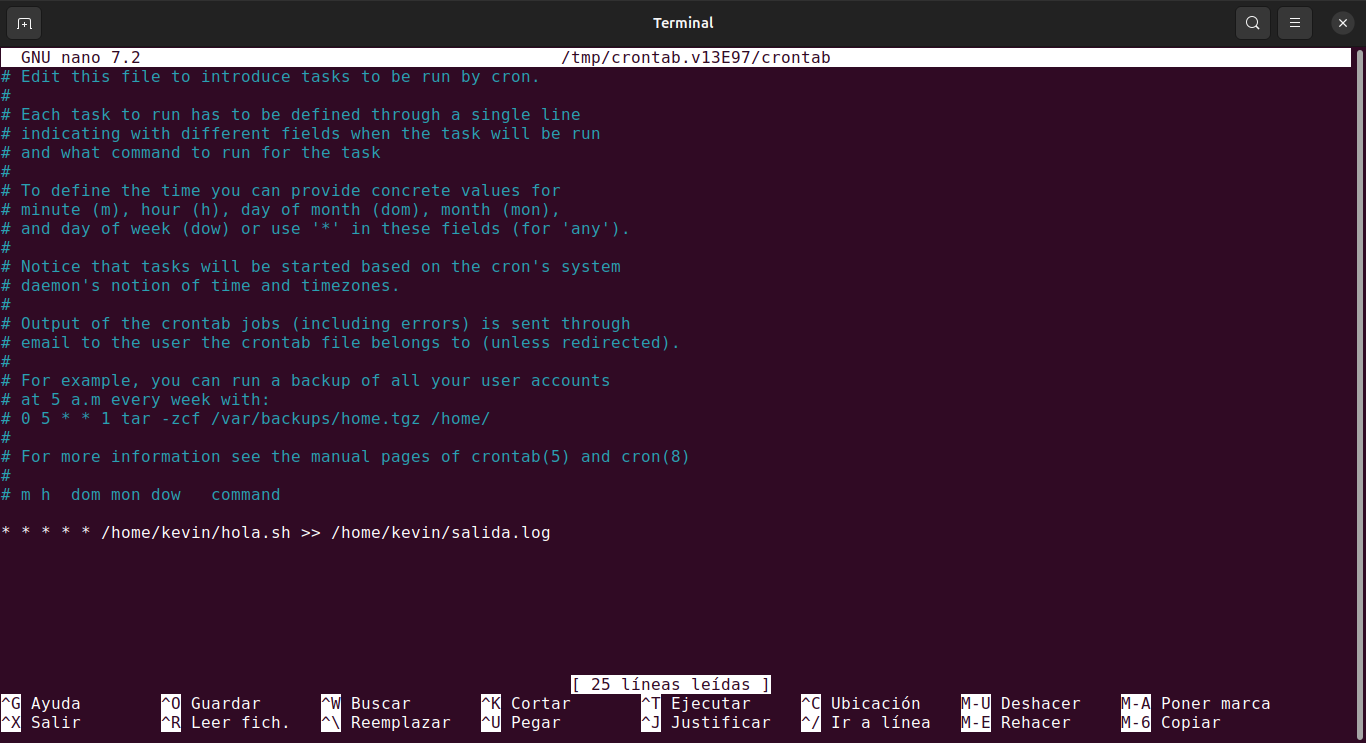
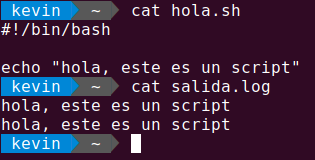


Cree un trabajo simple y programado que ejecute el comando date todos los viernes a la 01:00pm. ¿Dónde puede ver la salida de este trabajo? (captura)



Se envía por correo del usuario, es necesario tener mail o mailx (postfix)

Cree otro trabajo programado que ejecute el script hola.sh cada minuto, redirigiendo la salida al archivo salida.log en su directorio de origen para que sólo se le envíe el error estándar por correo electrónico. (captura)



|  |  |  |
| --- | --- | --- |
| HERRAMIENTA | FUNCIÓN | EN QUE CASOS LO USARÍAS |
| CRON | Programación periódica de tareas | Cuando se necesita programar tareas de forma recurrente |
| AT | Programación de una tarea única | Para ejecutar una tarea una sola vez. |
| ANSIBLE | Automatización de configuración y gestión de servers | Para manejar múltiples servidores de manera eficiente. |
| SYSTEMD | Sistemas de inicio y gestión de servicios. | Para iniciar, detener y supervisar servicios en linux |

Tarea 2

**Investiga mediante el manual de Linux el comando shutdown y sus parámetros**

**¿En que situaciones lo emplearías?**

Para reiniciar, apagar o programar el apagado de un servidor, esto con el fin de poder efectuar un cambio de alguna actualización.

**PROGRAMAR EL APAGADO DEL SERVIDOR MEDIANTE UN SCRIPT**

**Crea y Analiza el script**

Paso1 Para crear el script abrimos la terminal

Paso 2 Crear el archivo apagar.sh ejecutando el siguiente comando en la terminal .

**touch apagar.sh**

Paso 3 Una vez creado el archivo **lo abriremos ejecutando el siguiente comando en la termina de Linux**

**nano apagar.sh**

Paso 4 Una vez abierta copia este código

////////////////////////////////////////////////////////////////////////////////////////////////

**#!/bin/bash**

**clear**

**echo " \*\*\* SCRIPT PARA EL APAGADO DEL EQUIPO \*\*\* "**

**echo " SELECCIONA UNA OPCIÓN:"**

**echo " 1.-Apagar equipo ahora"**

**echo " 2.-Reiniciar equipo ahora"**

**echo " 3.-Asignar hora de apagado del equipo" echo "**

**4.-Apagar equipo a los xx minutos"**

**echo " 5.-Salir"**

**echo ""**

**read -p "OPCIÓN: " OPCION**

**case $OPCION in**

**sudo halt;;**

**sudo reboot;;**

**echo -n " ¿ A qué hora ?: "**

1. **read hora**

**sudo shutdown -h $hora;;**

**echo -n " ¿ En cuántos minutos se apagará el equipo?: "**

**read minutos**

**sudo shutdown -h $minutos;;**

**exit;;**

**\*) echo " OPCIÓN NO VÁLIDA "**

**exit 1;;**

**esac**

**exit 0**

**////////////////////////////////////////////////////////////////////////////**

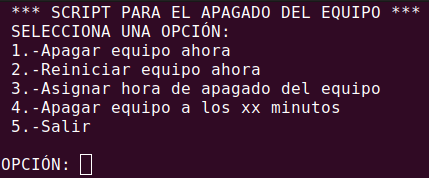
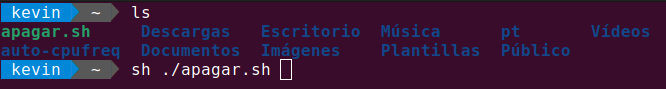
Una vez copiado el texto guardamos los cambios y cerramos el fichero. (Presiona CTRL + O para guardar los cambios y Para salir del editor, presiona CTRL + X.)

**Paso 5** Asigna permisos de ejecución sudo chmod +x apagar.sh

Una vez ejecutado el comando ya podemos usar el script. Para ello ejecutamos el siguiente comando en la terminal:

**sh ./apagar.sh**

Transcribe el script apoyándote de un editor para su elaboración y ejecuta el archivo apagado.sh (Captura )



**Análisis**

* ¿Por qué se eligieron estas opciones específicas para el script? ¿Existen otras opciones que podrían ser útiles, como hibernar o suspender el sistema?

Porque son las que cubren las necesidades más comunes.

* ¿Cuál es la función exacta de los comandos sudo halt, sudo reboot, y sudo shutdown -h? ¿Por qué se utiliza sudo?

halt detiene el sistema, reboot lo reinicia y shutdown lo apaga. Sudo es para que se tengan los permisos necesarios para poder ejecutar la acción.

* ¿Cómo se podría modificar el script para permitir que el usuario especifique un día específico para el apagado programado, además de la hora?

Se podría agregar un fragmento al código como el siguiente:

echo -n "¿En qué día (formato DD/MM)? "

read dia

echo -n "¿A qué hora (formato HH:MM)? "

read hora

fecha="$dia $hora"

sudo shutdown -h $fecha

* ¿Es necesario utilizar sudo para todas las opciones?

Para todos los que afectan el sistema si. Ya que podrían afectar a otros usuarios.

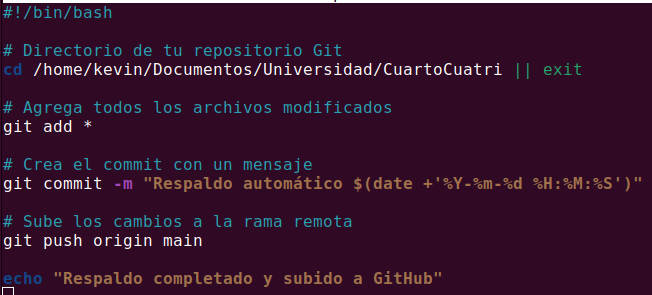
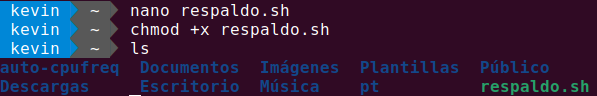
* ¿Sería útil agregar un registro de las acciones realizadas por el script, como la hora de apagado programada o los errores ocurridos?

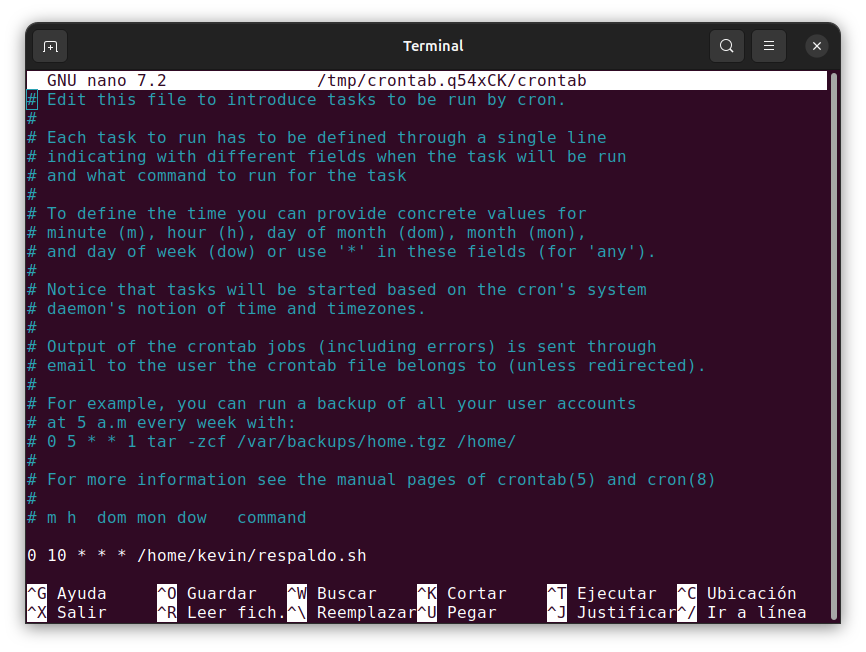
Tener un loggin permitiría llevar un seguimiento de las acciones realizadas y detectar problemas.

**Tarea 4**

Crea un script

Es momento que crees un script (apoyándote del ejemplo dado crea tu propio script donde apliques algún comando analizado en clase)





Lo que hace este script es un respaldo automático de los trabajos que suelo hacer en clase, para que no pueda olvidarlo o perder datos, automatizamos el proceso de respaldo.

## Reflexión

Automatizar tareas es una de las habilidades más importantes que un profesional puede tener, esto debido a que permite ahorrar tiempo en muchas de las labores repetitivas que suelen existir. Nos ayudan a ahorrar tiempo y esfuerzo, lo cual aumenta la eficacia de as labores, previene y reduce errores y permite un mejor manejo de recursos. Es muy importante que una persona encargada de la gestión y administración de servidores conozca la estructura de los scripts y la forma de implementarlos para automatizar labores. Una vez que se entiende como automatizar una tarea, las posibilidades son infinitas. No solo se pueden programar acciones simples, sino también procesos más complejos, como la recolección de datos, la actualización de sistemas o incluso la implementación de servicios. Aprender a usar herramientas como cron nos enseña a estructurar y organizar nuestro trabajo, dejando que tareas criticas se lleven a cabo sin necesidad de preocuparnos por ellas. Aunque configurar trabajos por medio de cron parecería ser una tarea muy complicada, es todo lo contrario, simplemente hay que comprender como funcionan los formatos de tiempo que la herramienta utiliza y listo. Cron nos recuerda la importancia de planificar nuestro trabajo técnico, como adelantarnos a las posibles necesidades futuras: la creación de un respaldo que ayude a evitar la perdida de información, o la implementación de actualizaciones programadas para evitar la mayor interrupción posible de servicios. Los script nos abren un abanico de posibilidades para automatizar comandos, es una habilidad clave a la hora de administrar entornos. Este aprendizaje es uno de los que van a marcar mi futuro, al permitirme mejorar mis capacidades y mis habilidades.

## Fuentes:

Gomez, E. E. P. (2024, 23 julio). *Tutorial de programación de Bash: Script de Shell de Linux y línea de comandos para principiantes*. freeCodeCamp.org. https://www.freecodecamp.org/espanol/news/tutorial-de-programacion-de-bash-script-de-shell-de-linux-y-linea-de-comandos-para-principiantes/

Vargas, D., & Vargas, D. (2023, 22 marzo). *Qué es la sintaxis Crontab: Crontab en Linux y ejemplos útiles*. Tutoriales Hostinger. https://www.hostinger.mx/tutoriales/sintaxis-crontab

# Actividad 7: Sistemas de archivos

## A) Tabla

|  |  |
| --- | --- |
| **Directorio** | **¿Cual es su función?** |
| **/etc/passwd/** | Almacena informacion básica de los usuarios del sistema, como el nombre de usuario, UID, GID, directorio home, shell de inicioi de sesión, entre otros. |
| **/etc/shadow/** | Almacena las contraseñas encriptadas, junto con informacion sobre la caducidad y otras configuraciones de seguridad. |
| **/etc/group/** | Contiene información de los grupos del sistema, incluyendo el nombre del grupo, GID, y la lista de usuarios que pertenecen a cada grupo. |

**2)** **¿En donde se almacenan las contraseñas?**

En /etc/shadow/

**3) ¿Qué pasa si digito cat /etc/shadow/?**

Sin permisos de root, no se puede visualizar, pero con permiso se pueden visualizar los usuarios y los hash de las contraseñas

**4) ¿Cualquier usuario puede accesar a modificar /etc/shadow/?**

Solo los administradores

**sai**:**$6$YTJ7JKnfsB4esnbS$5XvmYk2.GXVWhDo2TYGN2hCit**:19446:0:99999:7:::

* **sai** – User ID
* **$6** – El prefijo del algoritmo de encriptación.
* **$YTJ7JKnfsB4esnbS** – Conocido como salt, sirve para escoger la cantidad de bits que se usaran para encriptar la contraseña. Tambien evita que 2 usuarios con la misma contraseña usen el mismo hash de encriptamiento.
* **$5XvmYk2.GXVWhDo2TYGN2hCit** – La contraseña encriptada.
* 19446 – Es la ultima vez que se cambio la contraseña.
* 0 – El minimo de días que deben de pasar para cambiar contraseña.
* 99999 – el maximo de dias que la contraseña será valida, despues de eso el usuario debe de cambiar contraseña.
* 7 – Es el numero de dias antes de que expire la contraseña en la que se le avisará al usuario de cambiar contraseña.

¿Qué sucede si aplico ls -l /etc/passwd en mi terminal?

Vemos que solo tiene permisos de lectura en general, solo el propietario puede escribir.



¿En donde se almacenan las contraseñas?

En la carpeta /etc/shadow/

|  |  |  |
| --- | --- | --- |
| **Comando** | **¿Para qué sirve?** | **Ejemplo de uso** |
| **who** | Muestra los usuarios activos en el sistema |  |
| **whoami** | Muestra el usuario actual que está trabajando en el sistema |  |
| **id** | Muestra informacion sobre el usuario actual, incluyendo UID, GID y los grupos a los que pertenece. |  |
| **su** | Cambia de usuario en la terminal |  |
| **sudo** | “Super User DO”. Ejecuta un comando con prvilefios de superusuario o root. |  |
| **groups** | Muestra los grupos a los que pertenece un usuario. |  |
| **adduser** | Añade un usuario al sistema, creando su directorio home y configuraciones basicas. |  |
| **user** | Se relaciona con la gestion de usuarios |  |
| **addgroup** | Añade un usuario al grupo seleccionado. |  |
| **passwd** | Establece o cambia la contraseña de un usuario. |  |
| **userdel** | Elimina una cuenta del sistema. |  |
| **groupadd** | Crea un nuevo grupo. |  |

# Ejercicios Exploratorios

* ¿Cuál es el ID de usuario (UID) y el ID de grupo (GID) de root y

catherine?

Root: 0

catherine: 1030

* ¿Cuál es el nombre del grupo primario de Kevin? ¿Hay otros miembros en

este grupo?

1015, emma y grace.

* ¿Cuál shell está asignado para el mail? ¿Qué significa?

/sbin/nologin/ Significa que no es un usuario que pueda hacer login.

* ¿Quiénes son los miembros del grupo de app-developer?

Catherine, dave, christian.

* ¿Cuáles de estos miembros solos administradores del grupo y cuáles son

los miembros ordinarios?

Emma es administradora de db-admin. Los demás son usuarios regulares.

* ¿Cuál es la duración mínima de la contraseña para catherine ? ¿Y cuál es

la duración máxima de la contraseña?

0 dias, la puede cambiar cuando quiera.

* ¿Cuál es el período de inactividad de la contraseña para Kevin?

2 dias

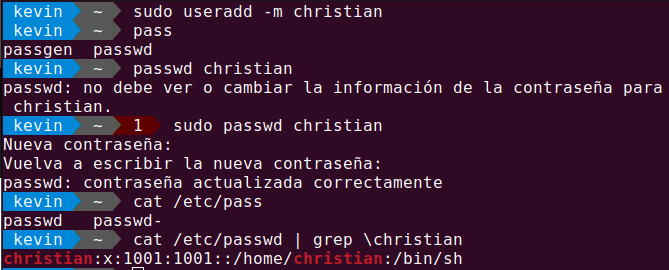
* ¿Cómo puede saber si una cuenta de usuario, que antes podía acceder al

sistema, ahora se encuentra bloqueada? Supongamos que su sistema

utiliza contraseñas en la sombra.

Generalmente tendrán un \* o un ! al inicio del hash de contraseña.

* Crea una cuenta de usuario llamada christian usando el comando useradd -m e identifica UID, GID y el shell.



* Identifica el nombre del grupo primario de christian. ¿Qué puedes deducir?

Además de que es el siguiente GID disponible, tambien es el ID del usuario, esto debido a que no hay otros grupos creados que hayan usado el GID antes.

* Usando el comando getent. Revisa la informacion de la constraseña de la cuenta del usuario christian.





* Añade el grupo editor a los grupos secundarios de christian, Supongamos que este grupo ya tiene a Emma, Dave y Frank como miembros ordinarios. ¿Cómo puedes verificar que hay administradores para este grupo?
* Ejecute el comando (ls -l /etc/passwd /etc/shadow /etc/gshadow) y describe la salida que imprime en terminos de permisos de archivo. ¿Cuál de estos cuatro archivos están en shadow por razones de seguridad? Supongamos que tu sistema utiliza contraseñas shadow.

# Actividad 8: Copias de seguridad.

¿Qué sucedería si no contamos con respaldo confiable?

Podríamos perder muchísima información de suma importancia para la empresa.

a) Describir las diferencias entre respaldo incremental

|  |  |  |
| --- | --- | --- |
| **Respaldo** | **Ventaja** | **Desventajas** |
| **Total** | -Respalda todos los datos, sin importar las modificaciones.  -Restauración rápida, ya que se tienen todos los datos en un solo respaldo. | -Toma mucho tiempo y espacio de almacenamiento.  -Mayor carga en el sistema al momento de realizarse. |
| **Incremental** | -Respalda solo los datos que han sido modificados desde el ultimo respaldo.  -Toma menos tiempo y espacio. | -Mayor riesgo de perdida de datos si un respaldo incremental falla. |
| **Diferencial** | -Respalda solo los datos modificados desde el ultimo respaldo total.  -Restauración más rapida que el incremental (se necesita solo el total y el diferencial más reciente). | -Toma más tiempo y espacio que un respaldo incremental, pero menos que uno total.  -Los respaldos pueden volverse grandes con el tiempo si no se hace un respaldo total. |

* ¿Con qué comando puedo realizar respaldos en linux?

Con cpio, tar o rsync

* Parámetros de CPIO.
  + -o: crea un archivo (salida).
  + -i: extrae archivos de una salida existente.
  + -v: modo detallado, muestra los archivos procesados.
  + -d: Crea directorios necesarios durante la extracción.
  + -t: Lista el contenido de un archivo sin extraer.
* Tar y sus flags.
  + -c: Crea un nuevo archivo tar.
  + -x: Extrae archivos de un tar.
  + -j: Comprime o descomprime utilizando gzip.
  + -v: Modo detallado, muestra los archivos procesados.
  + -f: Indica el archivo tar que se creará o utilizará.
  + -t: Lista el contenido de un archivo tar sin extraer.
    - Crear un archivo comprimido:

tar -czvf respaldo.tar.gz /directorio\_origen

* + - Extraer:

tar -xzvf respaldo.tar.gz

* ¿En dónde podemos respaldar?
  + Almacenamiento local, Remoto (rsync), en la nube, cinta u opticos.
* ¿Cómo los identifico en linux?
  + Con ls podemos ver los locales, con tar -tvf vemos el contenido sin extraer, con cpio -t listamos el contenido sin extraer.

|  |  |  |
| --- | --- | --- |
| **Instrucción** | **Qué hace?** | **Ejemplo** |
| **tar -cvf** | Crea un archivo tar sin comprimir, especificando el nombre del archivo de salida.   * -**c**  crea un nuevo archivo. * **-v**  muestra los archivos procesados. * **-f**  Especifica el nombre del archivo resultante. | Tar -cvf respaldo.tar /directorio |
| **tar -rvf** | Añade archivos o directorios a un archivo tar existente.   * **-r** Añadir los archivos a un tar existente. | Tar -rvf respaldo.tar /nuevodir/subdir |
| **tar -xvf** | Extrae los archivos del archivo.   * **-x** Extrae archivos. | Tar -xvf respaldo.tar |
| **tar -tvf** | Lista los archivos contenidos en un archivo tar sin extraerlos.   * **-t** muestra el contenido | Tar -tvf respaldo.tar |
| **tar -czf** | Crea un archivo tar comprimido con gzip.   * **-z** Comprime usando gzip. | Tar -czf respaldo.tar.gz /directorio |
| **tar -cpvzt** | Crea un archivo tar comprimido con gzip, manteniendo los permisos y usando modos detallado y verbose.   * **-p** Preserva los permisos de los archivos. * **-g** Crea un respaldo incremental. * **-N** Crea un respaldo diferencial | Tar -cpvzt respaldo.tar.gz /directorio |