



ADMINISTRACIÓN DE SERVIDORES

231011 Kevin Antonio Andrade López



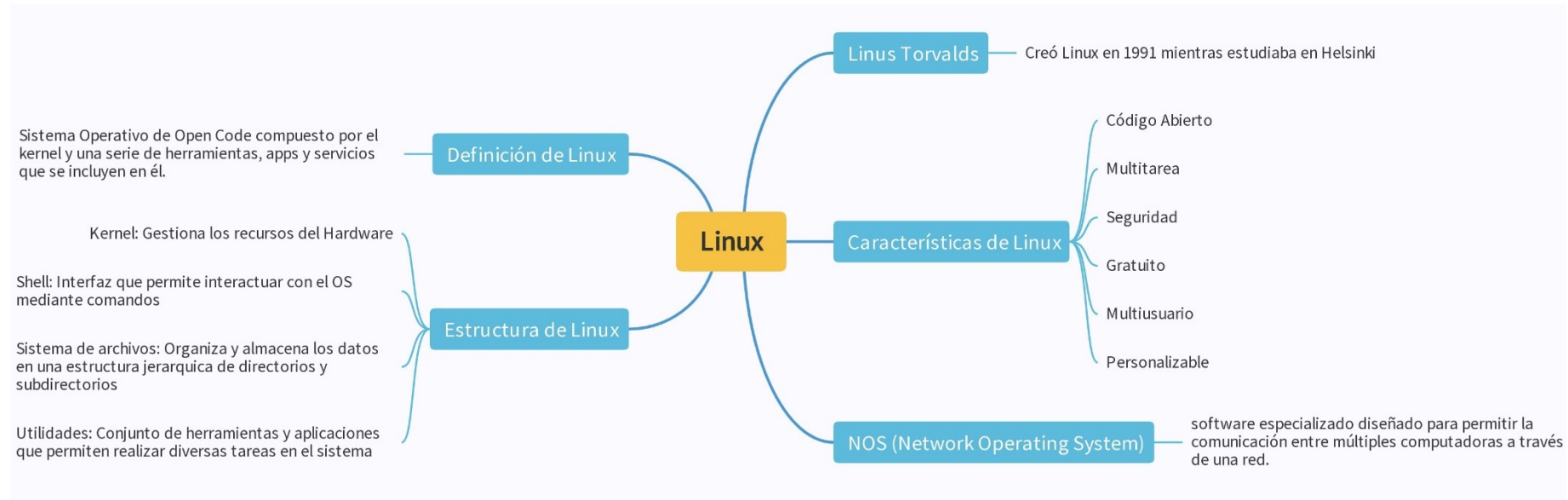
Unidad 2: Gestión de servidores – Servicios básicos.

Actividad 1:.....	0
Apartado A: Mapa semántico de Linux.....	0
Apartado b: Cuestionario.....	1
Actividad 2 – Primeros pasos – NFS Linux.....	1
Diagrama de flujo “Instalación Distro”	3
Actividad 3 Reporte Practica No. 1.....	4
Ejercicios guiados.....	4
c) Ejercicios exploratorios.....	9
Reflexión.....	10
Actividad 4.....	11
A) Cuestionario.....	11
B) Ejercicios guiados.....	11
C) Ejercicios exploratorios.....	12
Reflexión:.....	13
Actividad 5:.....	14
Privilegios.....	14
Problemario.....	14
Reflexión.....	15
Reto 01:.....	16
Reflexión:.....	17
Actividad 6: Procesos y automatización de tareas.....	18
a) Esquema conceptual de procesos.....	18
b) Mapa Conceptual automatización de tareas.....	20
c) Ejercicios Guiados.....	20
Ejercicios Exploratorios.....	23
Reflexión.....	28
Fuentes:.....	28
Actividad 7: Sistemas de archivos.....	29
A) Tabla.....	29
Ejercicios Exploratorios.....	31
Actividad 8: Copias de seguridad.....	32
Bibliografía.....	34
Actividad 9: Gestión de Servicios de red.....	36
A).....	36

B) Requerimientos lógicos a partir de un escenario.....	37
C) ¿Para qué sirven estos comandos?.....	37
Reflexión:.....	40

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

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

/dev

3. ¿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.

4. ¿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

2. ¿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).

3. 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 `man` para averiguar qué hace cada comando

Comando	Descripción	Captura Pantalla
<code>ls</code>	Mostrar el contenido de un directorio.	
<code>cat</code>	Muestra el texto de un archivo. Concatena las cadenas y las muestra en texto plano.	
<code>cut</code>	Elimina fragmentos en cada línea de un archivo	

cd	Cambiar el directorio en el que se está trabajando.	<pre>cd [-L [-P [-e]] [-@]] [dir]</pre> <p>Change the current directory to <code>dir</code>. If <code>dir</code> is not supplied, the value of the <code>HOME</code> shell variable is the default. The variable <code>CDPATH</code> defines the search path for the directory containing <code>dir</code>: each directory name in <code>CDPATH</code> is searched for <code>dir</code>. Alternative directory names in <code>CDPATH</code> are separated by a colon (:). A null directory name in <code>CDPATH</code> is the same as the current directory, i.e., <code>''</code>. If <code>dir</code> begins with a slash (/), then <code>CDPATH</code> is not used. The <code>-P</code> option causes <code>cd</code> to use the physical directory structure by resolving symbolic links while traversing <code>dir</code> and before processing instances of <code>..</code> in <code>dir</code> (see also the <code>-P</code> option to the <code>set</code> builtin command); the <code>-L</code> option forces symbolic links to be followed by resolving the link after processing instances of <code>..</code> in <code>dir</code>. If <code>..</code> appears in <code>dir</code>, it is processed by removing the immediately previous pathname component from <code>dir</code>, back to a slash or the beginning of <code>dir</code>. If the <code>-e</code> option is supplied with <code>-P</code>, and the current working directory cannot be successfully determined after a successful directory change, <code>cd</code> will return an unsuccessful status. On systems that support it, the <code>-@</code> option presents the extended attributes associated with a file as a directory. An argument of <code>-</code> is converted to <code>\$OLDPWD</code> before the directory change is attempted. If a non-empty directory name from <code>CDPATH</code> is used, or if <code>-</code> is the first argument, and the directory change is successful, the absolute pathname of the new working directory is written to the standard output. If the directory change is successful, <code>cd</code> sets the value of the <code>PWD</code> environment variable to the new directory name, and sets the <code>OLDPWD</code> environment variable to the value of the current working directory before the change. The return value is true if the directory was successfully changed; false otherwise.</p>
cp	Copia archivos y directorios	<p>NOMBRE</p> <p>cp - copia archivos y directorios</p> <p>SINOPSIS</p> <pre>cp [OPCIÓN]... [-I] ORIGEN DESTI cp [OPCIÓN]... ORIGEN... DIRECTORIO cp [OPCIÓN]... -t DIRECTORIO ORIGEN...</pre> <p>DESCRIPCIÓN</p> <p>Copia ORIGEN en DESTINO, o diversos ORIGEN(es) a un directorio.</p> <p>Los argumentos obligatorios para las opciones largas son también obligatorios para las opciones cortas.</p> <p>-a, --archive igual que -dR --preserve=todo</p> <p>--attributes-only no copia los datos de los archivos, solo sus atributos</p> <p>--backup[=CONTROL] crea una copia de seguridad de cada archivo de destino que ya exista</p> <p>-b como --backup pero no acepta ningún argumento</p> <p>--copy-contents en modo recursivo, copia también el contenido de archivos</p>
mv	Mueve archivos	<p>NOMBRE</p> <p>mv - mueve (renombra) archivos</p> <p>SINOPSIS</p> <pre>mv [OPCIÓN]... [-I] ORIGEN DESTINO mv [OPCIÓN]... ORIGEN... DIRECTORIO mv [OPCIÓN]... -t DIRECTORIO ORIGEN...</pre> <p>DESCRIPCIÓN</p> <p>Copia ORIGEN a DESTINO, o varios ORIGEN(es) a DIRECTORIO.</p> <p>Los argumentos obligatorios para las opciones largas son también obligatorios para las opciones cortas.</p> <p>--backup[=CONTROL] crea una copia de seguridad de cada archivo de destino que ya exista</p> <p>-b como --backup pero no acepta ningún argumento</p> <p>--debug explica el modo en que se copia un archivo. Implica -v</p> <p>-f, --force no pregunta nunca antes de sobrescribir</p> <p>-i, --interactive pide confirmación antes de sobrescribir</p>

mkdir	Crea directorios	<p>NOMBRE mkdir - crea directorios</p> <p>SINOPSIS mkdir [OPCIÓN]... DIRECTORIO...</p> <p>DESCRIPCIÓN Crea los DIRECTORIO(s), si no existen ya.</p> <p>Los argumentos obligatorios para las opciones largas son también obligatorios para las opciones cortas.</p> <p>-m, --mode=MODO establece los permisos (como en chmod), en lugar de a=rwx - umask</p> <p>-p, --parents no hay error si existen, crea los directorios padres en caso necesario, no influye en su modo la opción -m.</p> <p>-v, --verbose muestra un mensaje por cada directorio creado</p> <p>-Z establece el contexto de seguridad SELinux de cada directorio creado al tipo predeterminado</p> <p>--context[=CTX] como -Z, o si se especifica CTX entonces establece el contexto de seguridad SELinux o SMACK a CTX</p>
touch	Crea un archivo vacío o modifica la fecha de modificación de uno existente	<pre>kevin ~/Documentos/Universidad/Pruebas ls archivo</pre>
wc	Imprime cuantos saltos de line, palabras y bytes hay en el archivo	<pre>kevin ~/Documentos/Universidad/Pruebas wc ola.txt 36 142 1103 ola.txt</pre>
passwd	Cambia la contraseña del usuario.	<pre>kevin ~/Documentos/Universidad/Pruebas passwd Cambiando la contraseña de kevin. Contraseña actual: </pre>
rm	Borra un archivo	<pre>kevin ~/Documentos/Universidad/Pruebas 10 ls archivo ola.txt kevin ~/Documentos/Universidad/Pruebas rm archivo kevin ~/Documentos/Universidad/Pruebas ls ola.txt</pre>
rmdir	Borra un directorio	<pre>kevin ~/Documentos/Universidad/Pruebas ls Borrarme ola.txt kevin ~/Documentos/Universidad/Pruebas rmdir Borrarme/ kevin ~/Documentos/Universidad/Pruebas ls ola.txt</pre>
more	Muestra contenido de un archivo por paginas.	<pre>kevin ~/Documentos/Universidad/Pruebas more ola.txt Lorem ipsum dolor sit amet consectetur adipiscing elit augue pulvinar auctor, sociis egestas ad purus euismod viverra quisque enim donec imperdiet commodo, et litora ac sociosqu facilisi dis potenti magnis in. Et scelerisque euismod orci porttitor quam ad turpis, torquent mauris sollicitudin lobortis sapien litora class eget, ullamcorper fusce dui nibh dictumst habitant.</pre>
less	Muestra todo el archivo en una sola pagina deslizable (scroll)	<pre>kevin ~/Documentos/Universidad/Pruebas less ola.txt</pre>
whereis	Ubicación de archivos y comandos, busca sus carpetas	<pre>kevin ~/Documentos/Universidad/Pruebas whereis ollama ollama: /usr/local/bin/ollama /usr/local/lib/ollama /usr/share/ollama</pre>

head	Muestra las primeras 10 lineas de un texto	<pre> kevin ~/Documentos/Universidad/Pruebas head ola.txt Lorem ipsum dolor sit amet consectetur adipiscing elit augue pulvinar auctor, sociis egestas ad purus euismod viverra quisque enim donec imperdiet commodo, et litora ac sociosqu facilisi dis potenti magnis in. Et scelerisque euismod orci porttitor quam ad turpis, torquent mauris sollicitudin lobortis sapien litora class eget, ullamcorper fusce duis nibh </pre>
tail	Muestra las ultimas 10 lineas de un texto	<pre> kevin ~/Documentos/Universidad/Pruebas tail ola.txt faucibus curabitur mollis curae imperdiet massa venenatis interdum molestie. Integer pretium justo natoque mus parturient euismod vitae, cursus potenti porttitor neque tortor posuere nibh odio, turpis auctor dignissim fringilla duis ultricies. </pre>
sort	Ordena las lineas de un archivo.	<pre> kevin ~/Documentos/Universidad/Pruebas sort ola.txt accumsan augue, aliquet dictumst erat adipiscing elit augue pulvinar auctor, bibendum integer nostra rutrum class eget, ullamcorper fusce duis nibh condimentum imperdiet aptent pretium dictumst habitant. Eget eu gravida euismod pretium et litora ac sociosqu facilisi dis Et scelerisque euismod orci porttitor faucibus curabitur mollis curae faucibus hendrerit dictumst non justo fringilla duis ultricies. imperdiet massa venenatis interdum Integer pretium justo natoque mus libero volutpat quam ullamcorper ante, Lorem ipsum dolor sit amet consectetur luctus eu purus, nostra litora molestie. Netus quis senectus ullamcorper dui nibh odio, turpis auctor dignissim nostra phasellus convallis mollis, parturient euismod vitae, cursus porta. potenti magnis in. potenti porttitor neque tortor posuere quam ad turpis, torquent mauris quisque enim donec imperdiet commodo, ridiculus magna maecenas. scelerisque pulvinar porttitor etiam senectus auctor elementum netus sociis egestas ad purus euismod viverra sollicitudin lobortis sapien litora Varius pulvinar fringilla conubia </pre>
tr	Transforma o elimina caracteres de un texto	<pre> kevin ~/Documentos/Universidad/Pruebas 127 cat ola.txt tr o a Larem ipsum dalar sit amet cansectetur adipiscing elit augue pulvinar auctar, sacilis egestas ad purus euismad viverra quisque enim danec imperdiet cammada, et litara ac saciasqu facilisi dis patenti magnis in. Et scelerisque euismad arci partititar quam ad turpis, tarquent mauris sallicitudin labartis sapien litara class eget, ullamcarper fusce duis nibh dictumst habitant. </pre>

chmod	Cambia los permisos de un directorio o archivo.	<pre> kevin ~/Documentos/Universidad/Pruebas ls -la total 12 drwxrwxr-x 2 kevin kevin 4096 oct 2 15:41 . drwxrwxr-x 4 kevin kevin 4096 oct 2 15:17 .. -rwxrwxrwx 1 kevin kevin 1100 oct 2 15:41 ola.txt kevin ~/Documentos/Universidad/Pruebas chmod 755 ola.txt kevin ~/Documentos/Universidad/Pruebas ls -la total 12 drwxrwxr-x 2 kevin kevin 4096 oct 2 15:41 . drwxrwxr-x 4 kevin kevin 4096 oct 2 15:17 .. -rwxr-xr-x 1 kevin kevin 1100 oct 2 15:41 ola.txt </pre>
grep	Busca cadenas dentro del texto	<pre> kevin ~/Documentos/Universidad/Pruebas 130 cat ola.txt gr ep ipsum Lorem ipsum dolor sit amet consectetur </pre>

2. 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.

3. Para que sirve el comando Locate

Sirve para encontrar documentos o archivos

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

```

kevin ~/Documentos/Universidad/Pruebas find ola.txt
ola.txt

```

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.

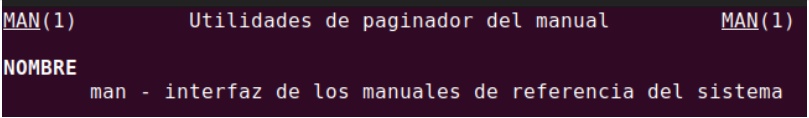
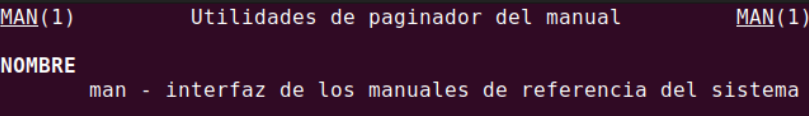
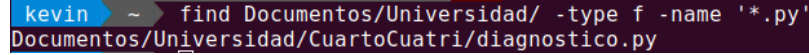
```
kevin ~ /Documentos/Universidad/User tree
├── Documents
│   ├── Hello
│   │   ├── helloa
│   │   ├── hey2
│   │   └── ola5
│   └── World
│       └── earth9
├── Downloads
│   ├── Music
│   └── Songs
│       ├── collection1
│       └── collection2
├── test
├── Test
│   └── passa
└── 8 directories, 8 files
```

- Busque y capture pantalla de dentro del árbol todos los archivos que terminen con un numero

```
kevin ~ find Documentos/Universidad/User/ -type f -name '*[0-9]'
Documentos/Universidad/User/Downloads/Songs/collection2
Documentos/Universidad/User/Downloads/Songs/collection1
Documentos/Universidad/User/Documents/Hello/ola5
Documentos/Universidad/User/Documents/Hello/hey2
Documentos/Universidad/User/Documents/World/earth9
```

- Elimine la carpeta y todo lo que tiene dentro con un solo comando.

```
kevin ~ /Documentos/Universidad ls
CuartoCuatri User
kevin ~ /Documentos/Universidad rm -rf User/
kevin ~ /Documentos/Universidad ls
CuartoCuatri
```

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	 <p>MAN(1) Utilidades de paginador del manual MAN(1) NOMBRE man - interfaz de los manuales de referencia del sistema</p>
Info	Similar a man, pero proporciona una documentación mas detallada y estructurada.	 <p>MAN(1) Utilidades de paginador del manual MAN(1) NOMBRE man - interfaz de los manuales de referencia del sistema</p>
Find	Se utiliza para buscar archivos y directorios en el sistema de archivos de linux	 <pre>kevin ~\$ find Documentos/Universidad/ -type f -name '*.py' Documentos/Universidad/CuartoCuatri/diagnostico.py</pre>

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.
2. **¿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.
3. **¿Qué representa la tilde (~) en la salida de comando pwd?**
La tilde es un carácter especial que representa el directorio home del usuario.
4. **¿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.
5. **¿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

2. 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

```
$ sudo tree -F /  
  
/  
├── etc/  
│   ├── network/  
│   │   └── interfaces  
│   ├── systemd/  
│   │   ├── resolved.conf  
│   │   ├── system/  
│   │   └── system.conf  
│   ├── user/  
│   └── user.conf  
├── udev/  
│   ├── rules.d/  
│   └── udev.conf  
├── home/  
│   ├── lost+found/  
│   └── user/  
│       └── Documents/  
  
12 directorios, 5 archivos
```

Un usuario ingresa los siguientes comandos:

```
$ cd /etc/udev
$ ls -a
```

¿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/.

3. 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):

```
$ cd home/lost+found
```

-

- 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/
```

4. Considere los siguientes comandos

```
$ pwd
/etc/udev/rules.d
$ cd ../../systemd/user
$ cd ..
$ pwd
```

¿Cuál es el resultado del comando final pwd?

/etc/systemd

C) Ejercicios exploratorios

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


```
$ mkdir "this is a test"
$ ls
this is a test
```

¿Qué comando `cd` le permitiría ingresar a este directorio?

`cd this\ is\ a\ test/`

- 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.

- 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 *	Rekursivamente 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:

```
[six66@parrot]~[~]  
$who  
six66      tty7          2024-09-26 21:02 (:0)
```

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

Who -b

```
[six66@parrot]~[~/Documents]  
$who -b  
system boot 2024-09-26 12:58
```

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

Copia todo lo que existe dentro del fichero.

```
[six66@parrot]~[~]  
$cp -r Documents/prueba1/  
cp: missing destination file operand after 'Documents/prueba1/'
```

¿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.

```
[six66@parrot]~[~/Documents]  
$touch archivo1  
[six66@parrot]~[~/Documents]  
$echo "prueba" >archivo2  
[six66@parrot]~[~/Documents]  
$cat > archivo3  
prueba  
[six66@parrot]~[~/Documents]  
$nano archivo4  
[six66@parrot]~[~/Documents]  
$ls  
archivo1  archivo2  archivo3  archivo4  
[six66@parrot]~[~/Documents]  
$
```

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 curva 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 enseñó 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 cuán 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?

Mapa conceptual de gestión de procesos en Linux

Gestión de procesos en Linux

Administración de procesos

Creación y terminación de procesos

La creación de un proceso en Linux se realiza mediante sistemas de llamadas como `fork()`, que permite duplicar un proceso existente, y `exec()`, que reemplaza el proceso hijo con un nuevo programa. La terminación de un proceso puede realizarse de varias maneras, incluyendo el uso de comandos como `kill` o la finalización natural al completar su ejecución.

Prioridades de procesos

Los procesos en Linux tienen diferentes niveles de prioridad, que determinan el tiempo de CPU que recibirán. Esta prioridad se puede ajustar utilizando comandos como `nice` y `renice`, lo que permite optimizar el rendimiento del sistema según las necesidades de las aplicaciones que están corriendo.

¿Qué es un proceso?

Definición de proceso

Un proceso es un programa en ejecución que incluye el código del programa, sus datos, y su contexto de ejecución (como registros, estado de la CPU, etc.). Cada proceso tiene una identificación única llamada PID (Process ID).

Estados del proceso

Los procesos pueden estar en varios estados, incluyendo:

Ejecutando: El proceso está actualmente en uso por la CPU.

Listo: El proceso está preparado para ejecutarse pero está esperando su turno para acceder a la CPU.

Bloqueado: El proceso está esperando que ocurra un evento externo (como la entrada/salida).

La importancia de los procesos

Eficiencia del sistema

La gestión adecuada de procesos es crucial para el rendimiento del sistema, permitiendo que múltiples programas se ejecuten simultáneamente sin interferir entre sí.

Multitarea

Linux es un sistema operativo multitarea, lo que significa que puede gestionar múltiples procesos al mismo tiempo, compartiendo recursos de manera eficiente y mejorando la productividad.

Recursos compartidos

Los procesos en Linux pueden compartir recursos como memoria y archivos, lo que permite una colaboración efectiva entre ellos y optimiza el uso del sistema.

¿Qué es init?

Proceso padre de todos los procesos

`init` es el primer proceso que se inicia durante el arranque del sistema Linux y es responsable de iniciar todos los demás procesos. Su PID es siempre 1.

Inicio del sistema

`init` configura el entorno del sistema y lanza los procesos de usuario y servicios necesarios para que el sistema esté operativo.

Gestión de servicios

Este proceso gestiona la ejecución de servicios y demonios (background processes) configurados en el sistema, facilitando el control sobre la carga automática de aplicaciones.

¿Qué es ps?

Visualización de procesos

El comando `ps` se utiliza para listar los procesos que se están ejecutando en un sistema Linux, proporcionando información crucial para la administración de procesos.

Estado de los procesos actuales

Con `ps`, los usuarios pueden ver el estado actual de los procesos, incluyendo su PID, TTY, tiempo de CPU utilizado, y más.

Información detallada de cada proceso del sistema.

`ps` también puede mostrar información detallada de cada proceso, como el uso de memoria y el usuario que lo ejecuta, proporcionando un panorama completo del estado del sistema.

El comando kill

Finalización de procesos

El comando `kill` se utiliza para enviar señales a los procesos con el fin de finalizarlos. Existen diferentes señales que se pueden utilizar, siendo `SIGTERM` la más común para solicitar la terminación de manera amigable.

Señales de terminación

`kill` permite enviar señales específicas, como `SIGKILL`, que fuerza la terminación inmediata del proceso sin darle la oportunidad de limpiar los recursos.

Terminación forzada

Si un proceso no responde a las señales estándar, se puede utilizar `kill -9` para forzar su finalización, aunque esto puede resultar en pérdida de datos no guardados.

El comando top

Monitorización del sistema

El comando `top` proporciona una vista dinámica en tiempo real de los procesos activos y el uso de recursos del sistema, permitiendo a los administradores observar la actividad del sistema en vivo.

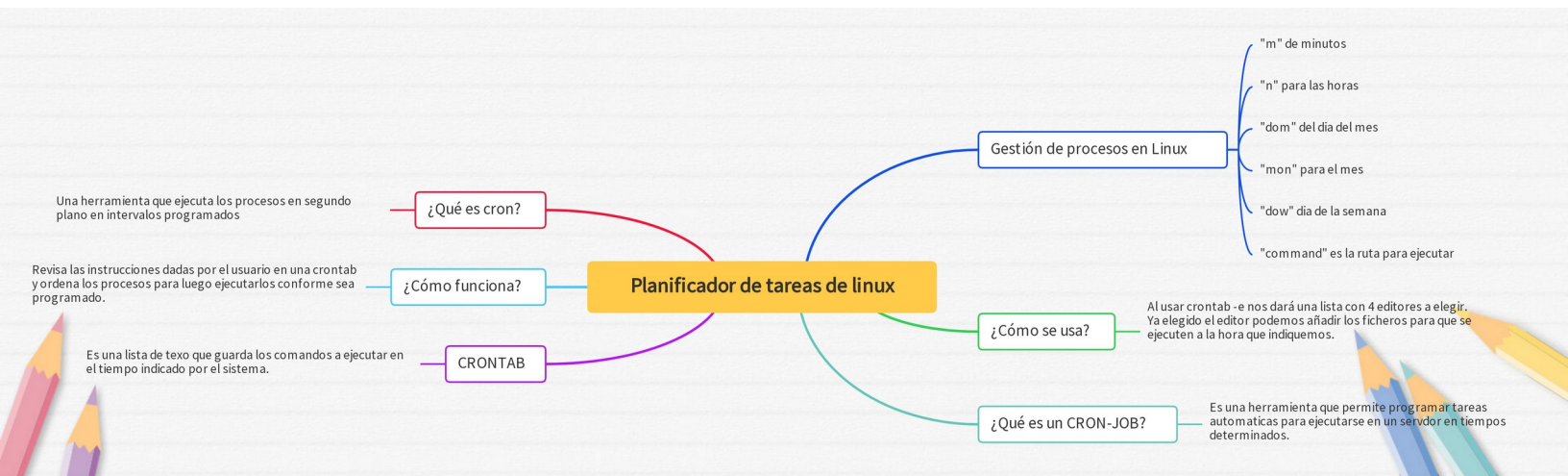
Procesos activos

`top` muestra información sobre todos los procesos activos, incluyendo el uso de CPU y memoria, lo que ayuda a identificar cuellos de botella en el rendimiento del sistema.

Uso de recursos

También se puede utilizar `top` para verificar el uso general de recursos del sistema (CPU, memoria, swap), permitiendo a los usuarios realizar ajustes y optimizaciones cuando sea necesario.

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 for user 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 día. 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 crontab

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)

```
kevin ~ crontab -e
no crontab for kevin - using an empty one
No modification made
```

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)

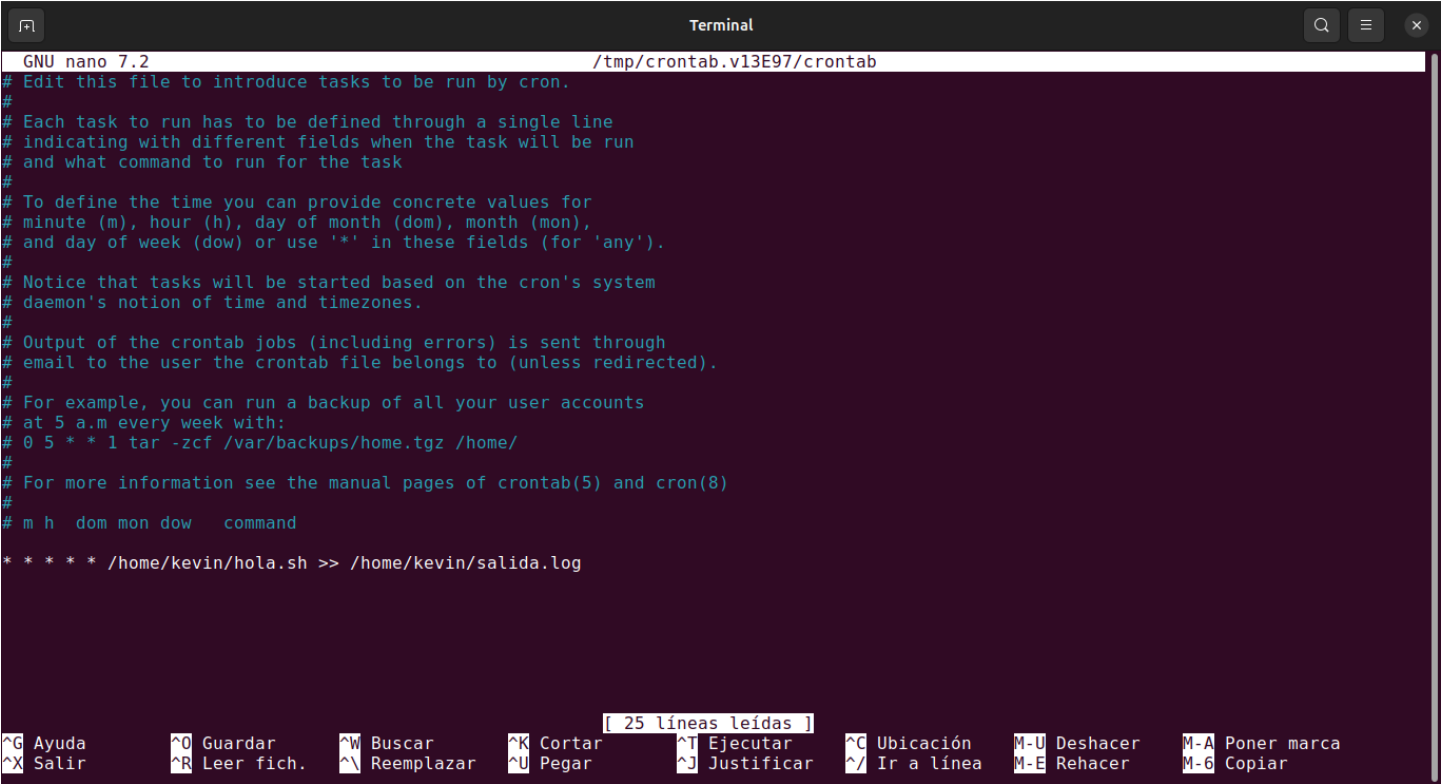
```
GNU nano 7.2 /tmp/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
00 13 * * 5 date
```

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)

```
kevin ~ ➤ cat hola.sh
#!/bin/bash

echo "hola, este es un script"
kevin ~ ➤ cat salida.log
hola, este es un script
hola, este es un script
kevin ~ ➤
```



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 terminal 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 ?: "  
5. 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)

```
kevin ~ ls
apagar.sh Descargas Escritorio Música pt Vídeos
auto-cpufreq Documentos Imágenes Plantillas Público
kevin ~ sh ./apagar.sh
```

```
*** SCRIPT PARA EL APAGADO DEL EQUIPO ***
SELECCIONA UNA OPCIÓN:
1.-Apagar equipo ahora
2.-Reiniciar equipo ahora
3.-Asignar hora de apagado del equipo
4.-Apagar equipo a los xx minutos
5.-Salir

OPCIÓN:
```

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 login 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)

```
#!/bin/bash

# Directorio de tu repositorio Git
cd /home/kevin/Documentos/Universidad/CuartoCuatri || exit

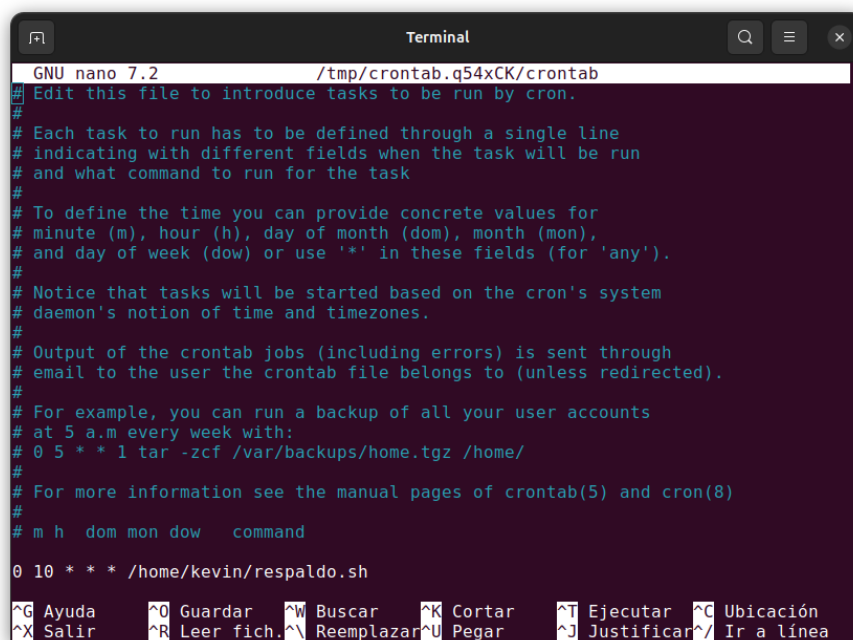
# Agrega todos los archivos modificados
git add *

# Crea el commit con un mensaje
git commit -m "Respaldo automático $(date +%Y-%m-%d %H:%M:%S)"

# Sube los cambios a la rama remota
git push origin main

echo "Respaldo completado y subido a GitHub"
```

```
kevin ~ nano respaldo.sh
kevin ~ chmod +x respaldo.sh
kevin ~ ls
auto-cpufreq Documentos Imágenes Plantillas Público
Descargas Escritorio Música pt respaldo.sh
```



```
GNU nano 7.2 /tmp/crontab.q54xCK/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 10 * * * /home/kevin/respaldo.sh
```

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 las 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 críticas 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 pérdida de información, o la implementación de actualizaciones programadas para evitar la mayor interrupción posible de servicios. Los scripts 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?
<code>/etc/passwd/</code>	Almacena información básica de los usuarios del sistema, como el nombre de usuario, UID, GID, directorio home, shell de inicio de sesión, entre otros.
<code>/etc/shadow/</code>	Almacena las contraseñas encriptadas, junto con información sobre la caducidad y otras configuraciones de seguridad.
<code>/etc/group/</code>	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 acceder a modificar `/etc/shadow/`?

Solo los administradores

`sai:6YTJ7JKnfsB4esnbS$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. También 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 mínimo de días que deben de pasar para cambiar contraseña.
- 99999 – el máximo de días que la contraseña será valida, después de eso el usuario debe de cambiar contraseña.
- 7 – Es el numero de días 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.

```
kevin ~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 3010 oct 10 07:08 /etc/passwd
```

¿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	<pre>kevin ~ who kevin seat0 2024-10-10 12:09 (login screen) kevin :0 2024-10-10 12:09 (:0)</pre>
whoami	Muestra el usuario actual que está trabajando en el sistema	<pre>kevin ~ whoami kevin</pre>
id	Muestra información sobre el usuario actual, incluyendo UID, GID y los grupos a los que pertenece.	<pre>kevin ~ id uid=1000(kevin) gid=1000(kevin) grupos=1000(kevin),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),100(users),114(lpadmin),124(vboxusers)</pre>
su	Cambia de usuario en la terminal	<pre>kevin ~ 1 sudo su [sudo] contraseña para kevin: root@portPC-six66:/home/kevin#</pre>
sudo	“Super User DO”. Ejecuta un comando con privilegios de superusuario o root.	<pre>kevin ~ 1 sudo su [sudo] contraseña para kevin: root@portPC-six66:/home/kevin#</pre>
groups	Muestra los grupos a los que pertenece un usuario.	<pre>kevin ~ groups kevin adm cdrom sudo dip plugdev users lpadmin vboxusers</pre>
adduser	Añade un usuario al sistema, creando su directorio home y configuraciones básicas.	<pre>kevin ~ sudo adduser patricio info: Añadiendo el usuario 'patricio' ... info: Selecting UID/GID from range 1000 to 59999 ... info: Añadiendo el nuevo grupo 'patricio' (1001) ... info: Adding new user 'patricio' (1001) with group 'patricio (1001)' ... info: Creando el directorio personal '/home/patricio' ... info: Copiando los ficheros desde '/etc/skel' ... Nueva contraseña: Vuelva a escribir la nueva contraseña: passwd: contraseña actualizada correctamente Cambiando la información de usuario para patricio Introduzca el nuevo valor, o presione INTRO para el predeterminado Nombre completo []: Número de habitación []: Teléfono del trabajo []: Teléfono de casa []: Otro []: ¿Es correcta la información? [S/n] s info: Adding new user 'patricio' to supplemental / extra groups 'users' ... info: Añadiendo al usuario 'patricio' al grupo 'users' ...</pre>
user	Se relaciona con la gestión de usuarios	

addgroup	Añade un usuario al grupo seleccionado.	<pre> kevin ~ sudo usermod -aG prueba patricio kevin ~ groups patricio patricio : patricio users prueba </pre>
passwd	Establece o cambia la contraseña de un usuario.	<pre> patricio@portPC-six66:/home/kevin\$ passwd Cambiando la contraseña de patricio. Contraseña actual: Nueva contraseña: Vuelva a escribir la nueva contraseña: passwd: contraseña actualizada correctamente </pre>
userdel	Elimina una cuenta del sistema.	<pre> kevin ~ 1 sudo userdel patricio </pre>
groupadd	Crea un nuevo grupo.	<pre> kevin ~ 127 sudo groupadd prueba </pre>

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 días, la puede cambiar cuando quiera.
- ¿Cuál es el período de inactividad de la contraseña para Kevin?
2 días
- ¿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.

```
kevin ~ cat /etc/passwd | grep \christian
christian:x:1001:1001:~/home/christian:/bin/sh
```

- Identifica el nombre del grupo primario de christian. ¿Qué puedes deducir?
Además de que es el siguiente GID disponible, también 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 información de la contraseña de la cuenta del usuario christian.

```
kevin ~ getent passwd | grep christian
christian:x:1001:1001:~/home/christian:/bin/sh
```

```
kevin ~ getent group | grep christian
christian:x:1001:
```

- 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 términos 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 rápida 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 ópticos.
- ¿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. <ul style="list-style-type: none"> ➤ -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. <ul style="list-style-type: none"> ➤ -r Añadir los archivos a un tar existente. 	Tar -rvf respaldo.tar /nuevodir/subdir
tar -xvf	Extrae los archivos del archivo. <ul style="list-style-type: none"> ➤ -x Extrae archivos. 	Tar -xvf respaldo.tar
tar -tvf	Lista los archivos contenidos en un archivo tar sin extraerlos. <ul style="list-style-type: none"> ➤ -t muestra el contenido 	Tar -tvf respaldo.tar
tar -czf	Crea un archivo tar comprimido con gzip. <ul style="list-style-type: none"> ➤ -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. <ul style="list-style-type: none"> ➤ -p Preserva los permisos de los archivos. ➤ -g Crea un respaldo incremental. ➤ -N Crea un respaldo diferencial 	Tar -cpvzt respaldo.tar.gz /directorio

¿Para qué sirve dump? Esta herramienta se encarga de examinar los archivos y sistemas de ficheros ext2 y ext3 y determinan que archivos deben copiarse.

¿Para que sirve restore? Se encarga de restaurar los ficheros generados por dump.

Bibliografía

Equipo editorial de IONOS. (2020b, octubre 7). *Programa de archivo tar: creación de backups y archivado de*

ficheros. IONOS Digital Guide. <https://www.ionos.mx/digitalguide/servidores/herramientas/programa-de-archivo-tar-como-crear-backups-en-linux/>

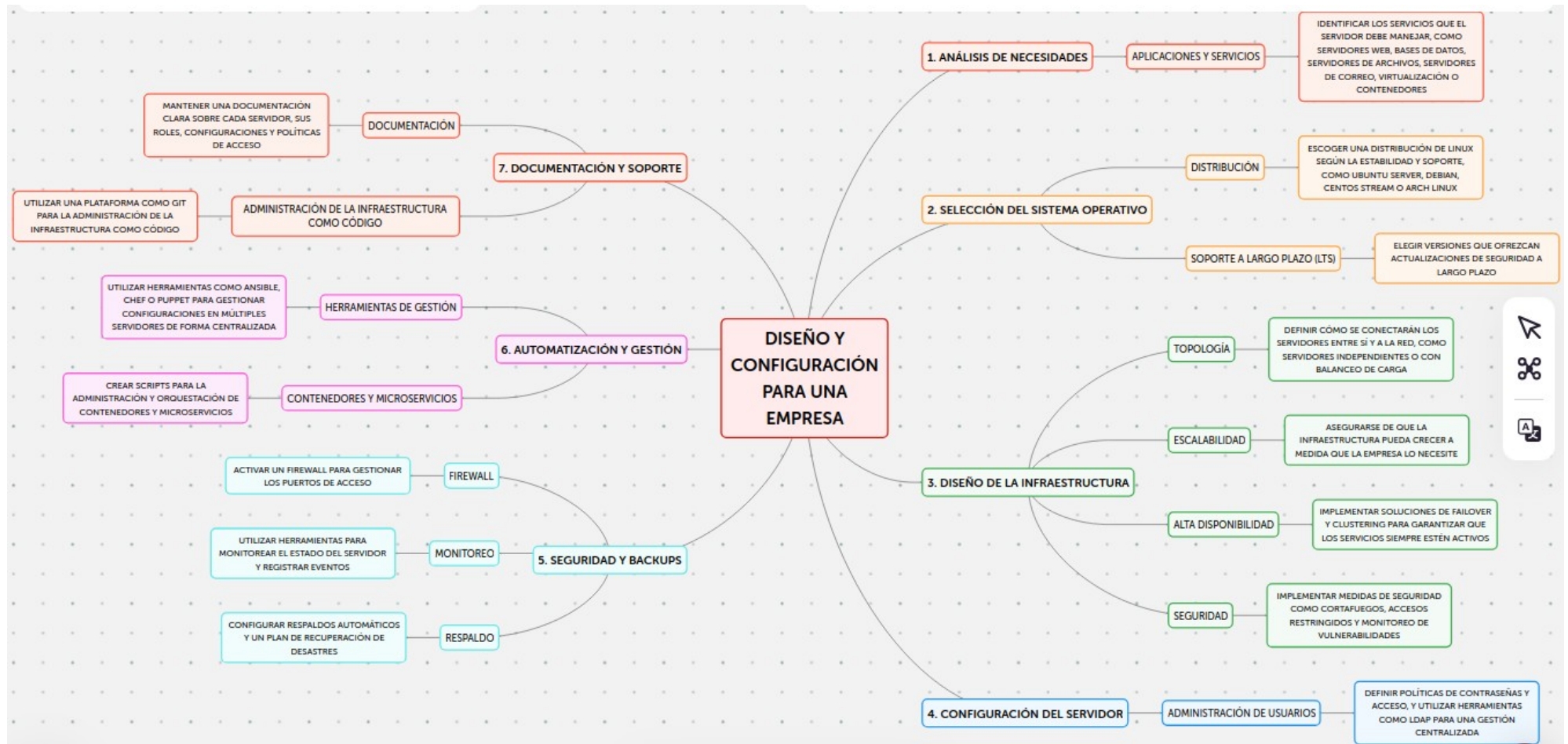
BlueHosting. (s. f.-b). *¿Cómo hacer un respaldo de sus archivos a través de SSH? - Doc - BlueHosting*.

<https://docs.bluehosting.cl/tutoriales/servidores/como-hacer-un-respaldo-de-sus-archivos-a-traves-de-ssh.html>

Actividad 9: Gestión de Servicios de red.

A)

¿Cómo podríamos diseñar y configurar una infraestructura de servidor Linux basada en Linux que responda a las necesidades de mi empresa?



B) Requerimientos lógicos a partir de un escenario.

Servicio.	Paquete	Comando inst	Comentario
FTP	vsftpd	Sudo apt install vsftpd	Se utiliza para transferir archivos entre computadoras en una red.
HTTP	Apache2	Sudo apt install apache2	Usado para levantar paginas web, donde se podrán mostrar múltiples archivos y cumplir múltiples funciones.
Mail	Postfix	sudo apt install postfix	Este servicio nos permitirá enviar y recibir correos electrónicos.
	Dovecot	sudo apt install dovecot-core	Paquete necesario para el correcto funcionamiento de postfix.
	bds-mailx	Sudo apt install bds-mailx	Aplicación que facilita el envío de correos desde la terminal, útil para probar el funcionamiento de nuestro servidor.

C) ¿Para qué sirven estos comandos?

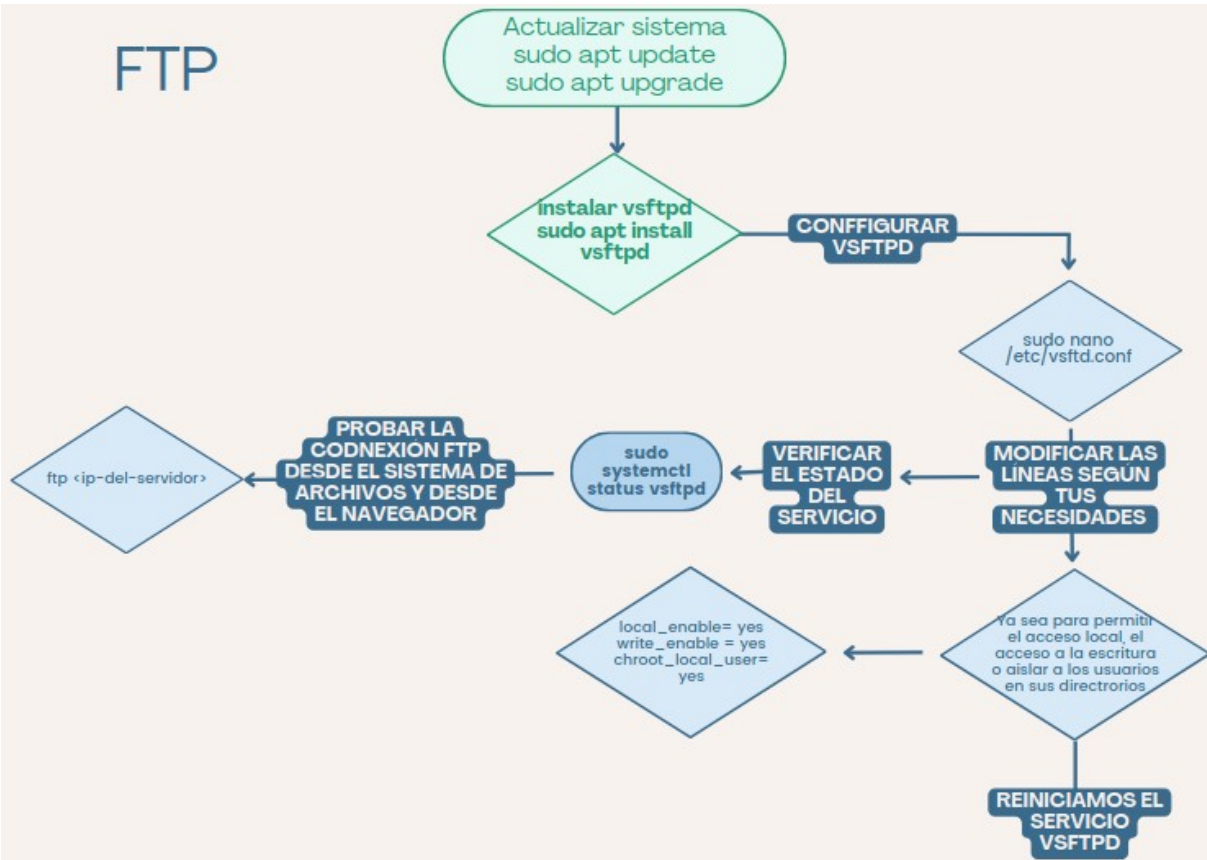
Sudo service apache2

Requiere de un parámetro extra, ayuda a manejar el servicio apache (start, stop, restart o status.)

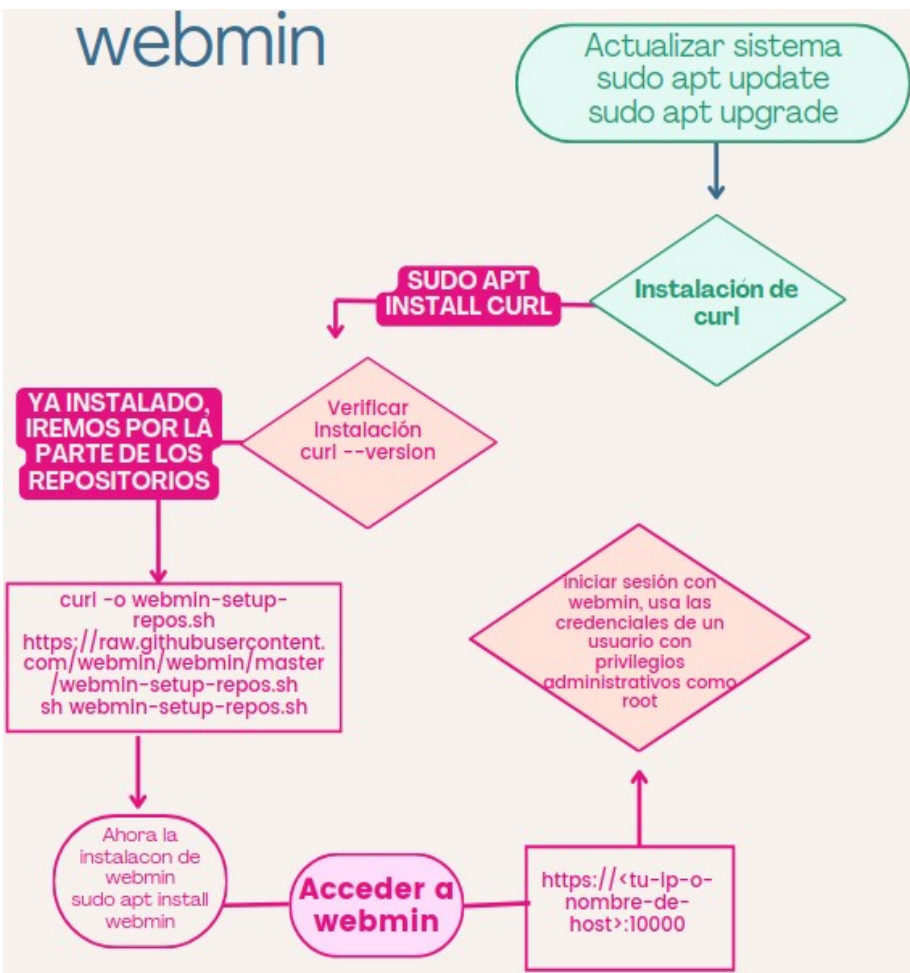
ps -aux | grep apache2

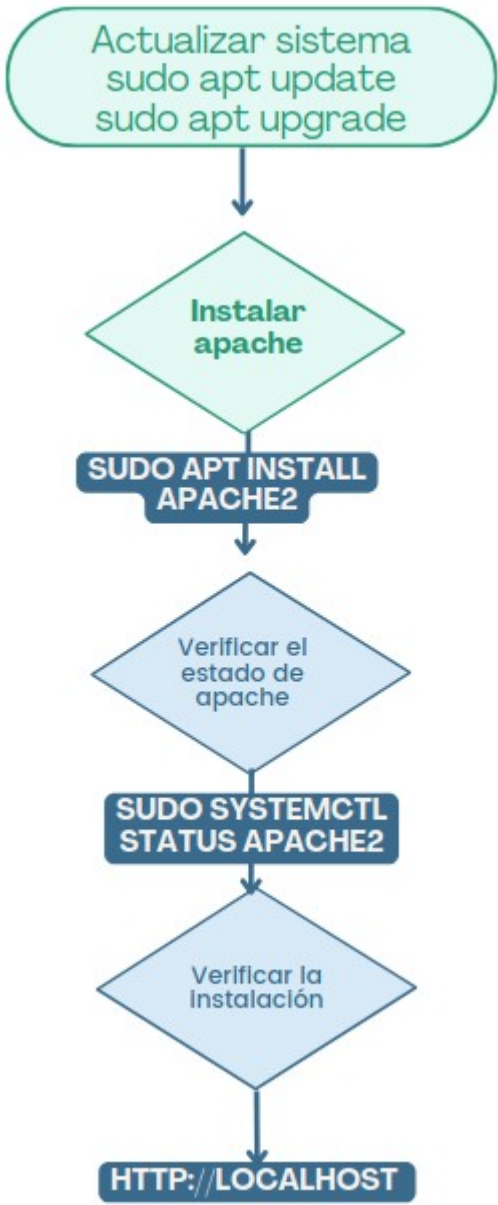
Este comando busca todos los procesos en ejecución y filtra los relacionados con apache2.

FTP

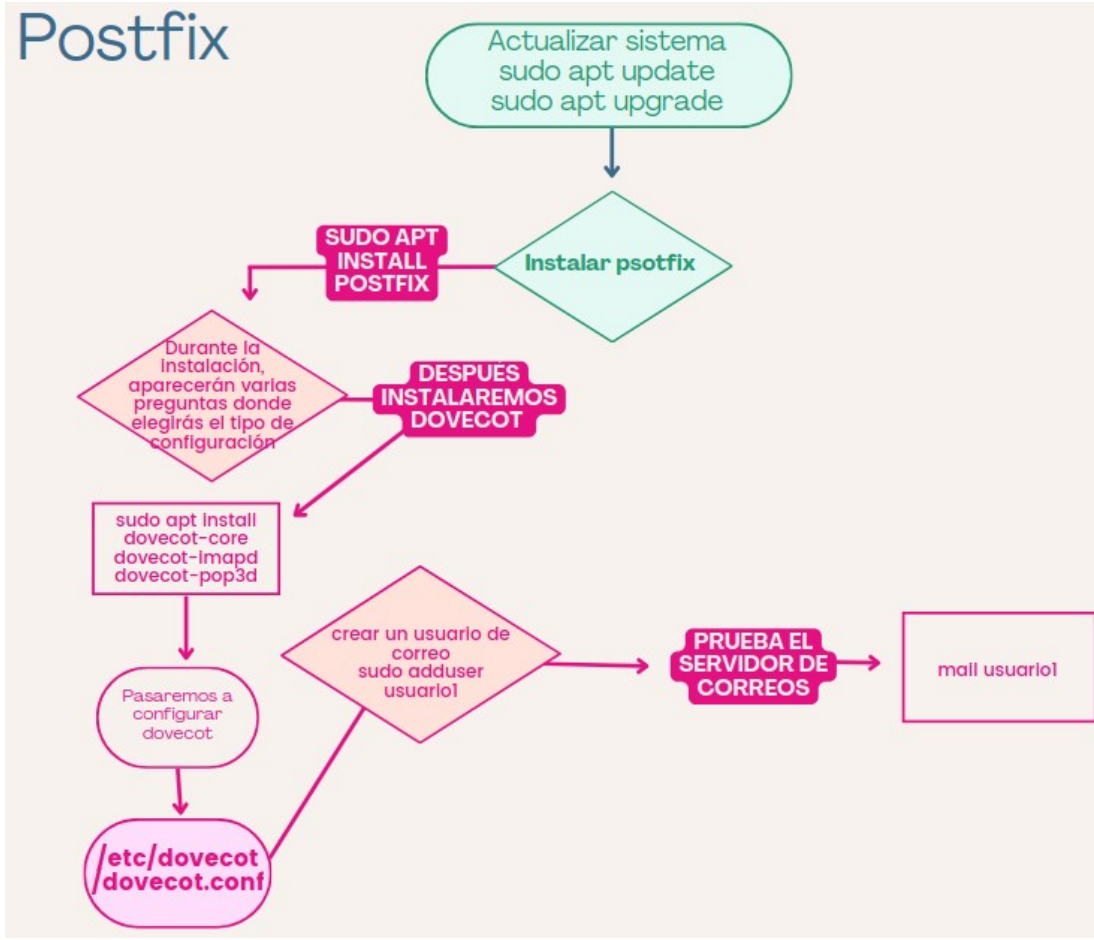


webmin





Postfix



Servicio	Función	Protocolo	Puerto	Carpeta de configuración
Servidor Web (Apache)	Alojar sitios web y ofrecer páginas web	HTTP/HTTPS	80/443	/etc/httpd/conf/httpd.conf
FTP	Transferencia de archivos	FTP/FTPS	21/990	/etc/vsftpd.conf
SSH	Conexión segura a máquinas remotas	SSH	22	/etc/ssh/sshd_config
Correo Electrónico	Envío y recepción de correos	SMTP/IMAP/POP3	25/143/110	/etc/mail/sendmail.cf
Webmin	Interfaz web para administración	HTTPS	10000	/etc/webmin

Servicios	Archivo de configuración	Parámetros que configurar
APACHE	/etc/httpd/conf/httpd.conf	ServerName, DocumentRoot, Listen, Directory, VirtualHost
FTP	/etc/vsftpd/vsftpd.conf	anonymous_enable, local_enable, write_enable, chroot_local_user
SSH	/etc/ssh/sshd_config	Port, PermitRootLogin, PasswordAuthentication, PubkeyAuthentication, AllowUsers, MaxAuthTries
Correo Electrónico	/etc/mail/sendmail.cf	SMART_HOST, MASQUERADE_AS, LOCAL_DOMAIN, RELAY_DOMAINS
Webmin	/etc/webmin/miniserv.conf	port, root, ssl, keyfile, certfile, sessions, tempdir, logfile
Firewall	firewall-cmd	--add-service (para todos los servicios anteriores)

Reflexión:

El conocer acerca del funcionamiento de los servicios que se van a implementar y la forma en las que las que estas trabajan es crucial para el correcto funcionamiento del servidor, esto nos permitirá tener control total de lo que estamos haciendo y como lo estamos haciendo, además de evitar que lleguemos a tener conflictos por librerías o paquetes que pudieran instalarse con cada servicio. Añadido a esto, el poder conocer a fondo el funcionamiento de cada servicio nos permite ejecutar cualquier mantenimiento y/o resolución de problemas e imprevistos que pudieran ocurrir a la hora de hacer modificaciones en el servidor. Si le dedicamos suficiente tiempo y atención a las distintas instalaciones podremos darnos cuenta tanto del funcionamiento de los paquetes, de la forma en que los demonios funcionan y corren a través de nuestro servidor y su entorno de desarrollo. Veo muy provechosa esta actividad debido a todo el aprendizaje, no solo superficial, sino también a las enseñanzas internas que conllevan.