

Actividad 2

Primeros Comandos de Shell

Sistemas Operativos 1

Ingeniería en Desarrollo de Software

Tutor: Ortega Francisco


Alumno: Ortega Hernández Mayra Jacqueline

Fecha: 15 de Mayo 2023



INDICE

1.1 INTRODUCCION	3
2.1 DESCRIPCION	4
3.1 JUSTIFICACION	5
4.1 DESARROLLO	6
5.1 CONCLUSIÓN	12
6.1 REFERENCIAS	13



1.1 INTRODUCCION

El impacto que han tenido los sistemas y en especial el sistema operativo Ubuntu, dentro del área de informática ha sido de gran impacto, pero primero debemos conocer un poco sobre dicho sistema; para empezar Ubuntu es una distribución de Linux es el nombre que recibe una serie de sistemas operativos, Linux es el multiusuario, multitareas y multiplataforma, además puede funcionar en modo de consola para un consumo mínimo de recursos, al ser código libre podemos utilizarlo, copiarlo, modificarlo y redistribuirlo libremente para cualquier uso que queramos darle; Linux comenzó su andadura en el mundo del Software libre por el año 1980 con la idea de crear un sistema operativo, recibe el nombre de núcleo o kernel de este sistema operativo que son denominados comúnmente distros básicamente son el núcleo del sistema.

Como podemos observar en esta breve información, Ubuntu está orientado al usuario con una facilidad de uso y en mejora de la experiencia al ser ejecutado, además cuenta con paquetes de uso e instalación para facilitar las tareas a realizar.

2.1 DESCRIPCION

Ubuntu junto con los Comandos de Shell son una herramienta principal que se usa en su gran parte por la industria tecnológica gracias a este Software existen una gran multitud de aparatos que usamos en el día a día de nuestras vidas, además cada vez son más complejas en cuestión de los servicios que ofrecen son más delicados y especializados ya que es la base estable para todos los desarrollos innovadores de cualquier entorno ya sea virtual, en contenedores o incluso la nube. Lo mejor de esto es que cualquier persona puede ejecutar el software, estudiarlo, compartir y modificar algún código como cualquier tipo de alternativa, es un elemento esencial para algunas empresas y sectores mas grandes del mundo, desde crear un sitio web hasta compartir información hasta los dispositivos móviles que utilizamos como Android, cuenta con una gran cantidad de casos prácticos, funciones, distribuciones, dispositivos y sistemas de destino; esto va dependiendo en las necesidades y carga de trabajo que se necesite realizar ya que este Software nos ayudara al desarrollo de un nuevo interfaz grafico que nos facilitara las acciones para un determinado usuario. Por ello es importante reconocer la importancia de un Software tan complejo ya que en nuestra vida cotidiana usamos más de mil comandos diferentes sin darnos cuenta de lo mismo.

3.1 JUSTIFICACION

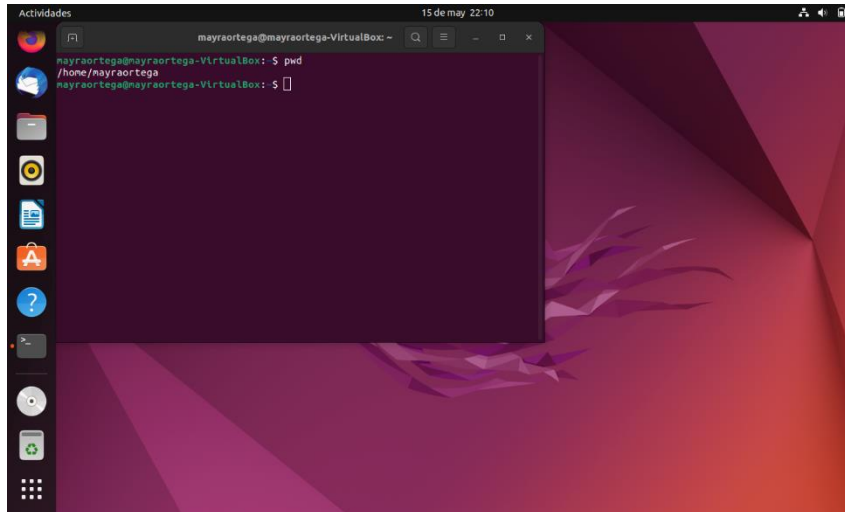
La importancia de Virtual box junto a Ubuntu es tan inmensa ya que es un software que sirve para hacer máquinas virtuales con instalaciones de sistemas operativos. Esto quiere decir que, si tienes un ordenador con Windows, GNU/Linux o incluso macOS, se puede crear una máquina virtual con cualquier otro sistema operativo así mismo se lograra utilizarlo dentro del que se desee usar. VirtualBox sirve principalmente para virtualizar sistemas operativos que no podamos o no queramos ejecutar de forma nativa en nuestro equipo. Esto servirá también para trabajar sobre esos sistemas operativos con un relativamente amplio abanico de posibilidades, pues una de las principales virtudes de una máquina virtual es el aislamiento que ello proporciona. Este tipo de Software es un apoyo en términos de compatibilidad ya que en algunos

programas determinados no nos permite ejecutar los procesos limitados por los recursos y abstracciones que se les han proporcionado. Cualquiera de los procesos no puede salir en ningún momento del entorno de este ‘ordenador virtual’. Aunque VirtualBox está enfocado el usuario doméstico, también es muy usada en entornos profesionales. Esto se debe a que soporta una gran cantidad de sistemas operativos.

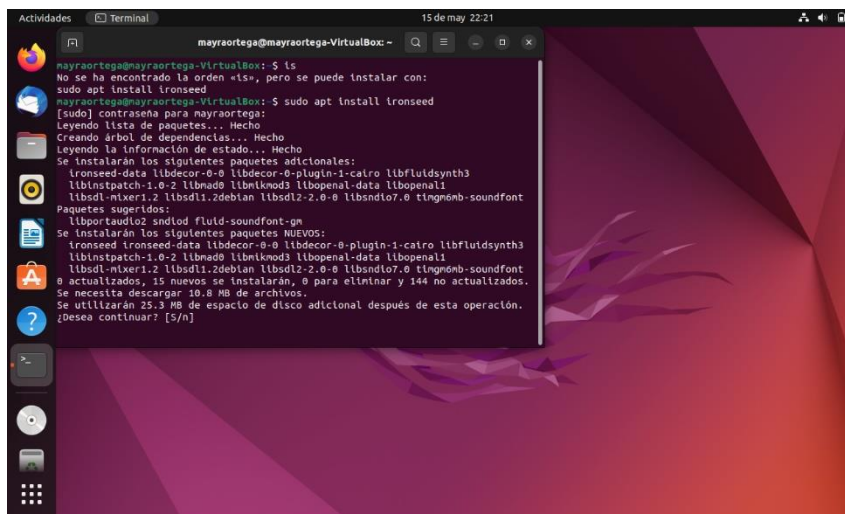
4.1 DESARROLLO

Para poder ejecutar un Software adecuadamente se necesita seguir una serie de comandos que nos ayudara a correr cualquier programa, es decir es un medio de comunicación con el sistema operativo ya que son las ordenes mediante textos y caracteres, por ello cada comando tiene una función diferente que nos llevara a un árbol de directorios estos mismos siguen un orden básico que nos permite realizar una variedad de tareas con el sistema operativo además se pueden utilizar para almacenar información sobre el entorno del sistema operativo, como la configuración del sistema, las rutas de acceso de los archivos, además crea y modifica archivos; mejor conocido como comunicación directa con el usuario y el sistema operativo. Aunque actualmente las interfaces graficas de usuario facilitan el trabajo cotidiano, además el sistema operativo solicita la entrada interpretando y gestionando cualquier resultado de salida procedente del sistema operativo.

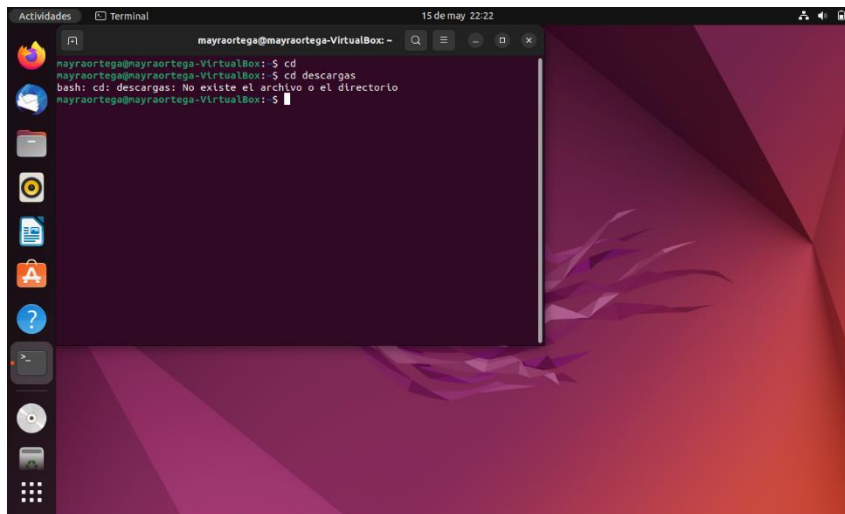
Por ello este documento pondremos en ejecución algunos comandos básicos de Shell que nos ayudan a crear y dirigirnos a diferentes rutas al igual dando una breve explicación sobre dicho comando.



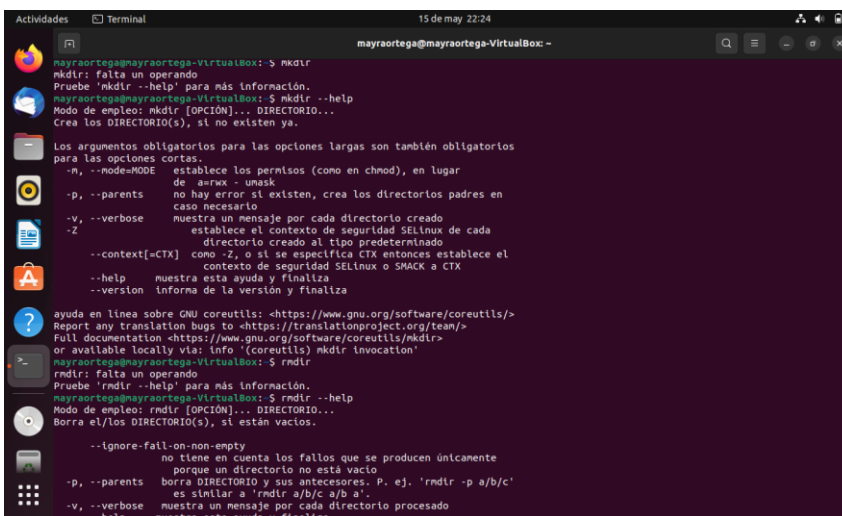
El comando `pwd`; nos da la ruta absoluta para saber en que directorio nos encontramos ejemplo
`/home/username` este comando se denota por `/`.



El comando `ls`; nos ayuda a saber que archivo hay en el directorio en el que se encuentra,
podemos ver archivos ocultos usando `ls -a`.



Cd; sirve para ir a un directorio ejemplo, si estas en la carpeta inicio y deseas ir a la carpeta de descargas puedes escribir, cd descargas.



Mkdir y rmdir; mkdir se usa para crear una carpeta o un directorio ejemplo, si quieres crear un directorio llamado DIY, puedes poner mkdir DIY, rmdir se utiliza para eliminar un directorio y solo se puede usar para un directorio vacío.


```
mayraortega@mayraortega-VirtualBox: ~$ rm
rm: falta un operando
Pruebe 'rm --help' para más información.
mayraortega@mayraortega-VirtualBox: ~$ rm --help
Modo de empleo: rm [OPCIÓN]... [FICHERO]...
Borra (desenlaza) los FICHERO(s).

-f, --force          descarta los ficheros y argumentos que no existan,
                    sin preguntar
-i, --interactive    pregunta antes de cada borrado
-I, --ignore-existing pregunta una vez antes de borrar más de tres ficheros,
                    o cuando se borra recursivamente; menos intrusivo
                    que -i, pero todavía protege contra la mayoría de
                    las equivocaciones
--interactive[=CUÁNDO] pregunta de acuerdo con CUÁNDO: never (nunca),
                    once (una vez, -i), o always (siempre, -I);
                    sin CUÁNDO, pregunta siempre
--one-file-system    cuando se borra una jerarquía recursivamente, salta
                    cualquier directorio que esté en un sistema de
                    ficheros distinto de aquel que corresponde con el
                    argumento de la línea de órdenes
--no-preserve-root   do not treat '/' specially
--preserve-root[=all] do not remove '/' (default);
                    with 'all', reject any command line argument
                    on a separate device from its parent
-r, -R, --recursive remove directories and their contents recursively
-d, --dir            remove empty directories
-v, --verbose        explain what is being done
--help              muestra esta ayuda y finaliza
--version            informa de la versión y finaliza

Por omisión, rm no borra directorios. Utilice la opción --recursive (-r o -R)
para borrar también cada directorio listado, junto con todo su contenido.

Para borrar un fichero cuyo nombre comience con un '.', por ejemplo '.fu',
utilice una de las siguientes órdenes:
rm -- -fu
rm -rfu
```

Rm; elimina archivos y directorios, para eliminar solo el directorio se utiliza rm -r y rm elimina la carpeta y archivos que contiene.

```
mayraortega@mayraortega-VirtualBox: ~$ touch
touch: falta un archivo como argumento
Pruebe 'touch --help' para más información.
mayraortega@mayraortega-VirtualBox: ~$ touch --help
Modo de empleo: touch [OPCIÓN]... FICHERO...
Actualiza la fecha de acceso y modificación de cada FICHERO a la hora actual.

Un argumento FICHERO que no exista se crea vacío, a menos que se
especifique -c o -h.

Una cadena de argumento FICHERO que sea - se maneja especialmente y provoca
que touch cambie la fecha del fichero asociado con la salida estándar.

Los argumentos obligatorios para las opciones largas son también obligatorios
para las opciones cortas.
-a          cambia solamente la fecha de acceso
-c, --no-create no crea ningún fichero
-d, --date=CADENA examina y utiliza CADENA en lugar de la fecha actual
-f          (no tiene efecto)
-h, --no-dereference afecta a cada enlace simbólico en lugar de a los
                    ficheros referidos (solamente es útil en sistemas
                    que pueden cambiar el propietario de un enlace
                    simbólico)
-m          cambia solamente la fecha de modificación
-r, --reference=FICHERO utiliza la fecha de este FICHERO en lugar de la fecha
                    actual
-t FECHA      utiliza [[SS]AA]MMDDhhmm[[.ss]] en lugar de la fecha
                    actual
--time=PALABRA establece la fecha dada por PALABRA:
                    PALABRA es access, atime o use: equivale a -a
                    PALABRA es modify o mtime: equivale a -m
--help      muestra esta ayuda y finaliza
--version   informa de la versión y finaliza

Tenga en cuenta que las opciones -d y -t aceptan formatos de
hora-fecha distintos.

ayuda en línea sobre GNU coreutils: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
```

Touch; crea un archivo nuevo de cualquier tipo desde un archivo txt vacío hasta un archivo zip, ejemplo touch new.txt

```
Actividades Terminal 15 de may 22:27
mayraortega@mayraortega-VirtualBox: ~
¿Qué página de manual quiere?
Por ejemplo, pruebe 'man man'.
mayraortega@mayraortega-VirtualBox: ~$ man cd
Ninguna entrada del manual para cd
mayraortega@mayraortega-VirtualBox: ~$ man --help
Uso: man [OPCIÓN...] [SECCIÓN] PÁGINA...

-C, --config-file=FICHERO  emplee este fichero de configuración de usuario
-d, --debug                emite mensajes de depuración
-D, --default              restablece todas las opciones a sus valores
                             predeterminados
--warnings[=AVISOS]       activa avisos desde groff

Modos principales de operación:
-f, --whatis               equivalente a whatis
-k, --apropos              equivalente a apropos
-K, --global-apropos       busca texto en todas las páginas
-l, --local-file            interpreta argumento(s) PÁGINA como nombre(s) de
                             fichero(s) local(es)
-W, --where, --path, --location  escribe localización física de página(s) man
-M, --where-cat, --location-cat  escribe localización física de fichero(s) cat

-C, --catman               empleado por catman para reformatear páginas cat
                             desactualizadas
-R, --recode=CODIFICACIÓN  página origen de salida codificada en
                             CODIFICACIÓN

Encontrando páginas del manual:
-L, --local=LOCALIZACIÓN  define la localización para esta búsqueda de man
                             particular
-M, --system=SISTEMA       emplea páginas del manual desde otros sistemas
-o bien: cp [OPCIÓN]... t DIRECTORIO ORIGEN...
Copia ORIGEN a DESTINO, o varios ORIGEN(es) a DIRECTORIO.

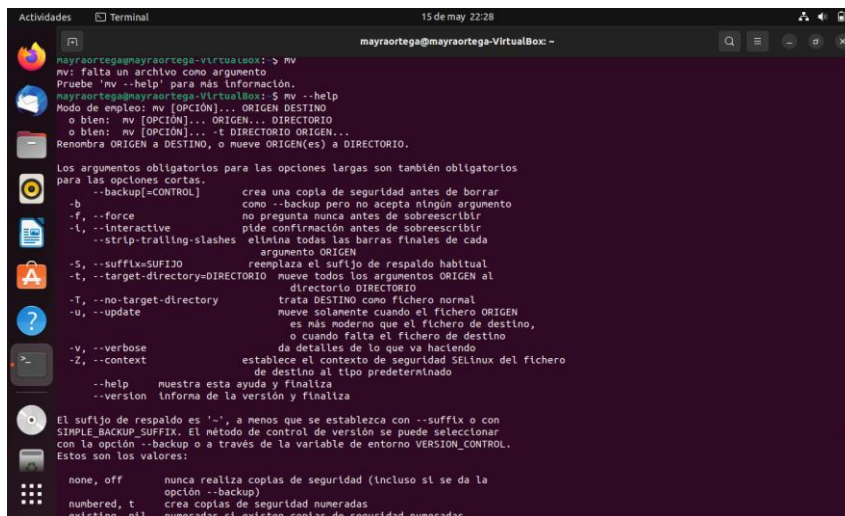
Los argumentos obligatorios para las opciones largas son también obligatorios
para las opciones cortas.
-a, --archive              lo mismo que -dR --preserve=all
--attributes-only          no copia los datos del fichero, solamente los
                             atributos
--backup[=CONTROL]         crea una copia de seguridad de cada fichero de
                             destino que exista
-b                          como --backup pero no acepta ningún argumento
--copy-contents            copia el contenido de los ficheros especiales
                             cuando opera recursivamente
-d                          lo mismo que --no-dereference --preserve=link
-f, --force                si un fichero de destino no se puede abrir, lo
                             borra y lo intenta de nuevo (no se tiene en
                             cuenta si se utiliza también la opción -n)
-i, --interactive          pide confirmación antes de sobrescribir
-H                          sigue los enlaces simbólicos de la línea
                             de órdenes
-L, --link                 crea enlaces duros de los ficheros en vez de copiarlos
-L, --dereference           siempre sigue los enlaces simbólicos en ORIGEN
-n, --no-clobber            no sobrescribe un fichero que exista
                             (tiene prioridad sobre una opción -i anterior)
--no-dereference           nunca sigue los enlaces simbólicos en ORIGEN
-p                          igual que --preserve=mode,ownership,timestamps
--preserve[=LISTA_ATTR]    conserva si puede los atributos especificados,
                             (por omisión: mode,ownership,timestamps)
                             atributos adicionales: context, links, xattr,
                             all
--no-preserve=LISTA_ATTR  no conserva los atributos especificados
--parents                 añade el directorio de origen a DIRECTORIO
-R, --recursive            copia directorios recursivamente
--preserve=LISTA_ATTR     conserva si puede los atributos especificados,
                             (por omisión: mode,ownership,timestamps)
                             atributos adicionales: context, links, xattr,
                             all
```

Man &--help; man se utiliza para saber mas de un comando y como usarlo ejemplo, man cd muestra las paginas manual del comando cd y help muestra de que manera se puede usar el comando.

```
Actividades Terminal 15 de may 22:28
mayraortega@mayraortega-VirtualBox: ~
cp: falta un archivo como argumento
Pruebe 'cp --help' para más información.
mayraortega@mayraortega-VirtualBox: ~$ cp --help
Modo de empleo: cp [OPCIÓN]... ORIGEN DESTINO
o bien: cp [OPCIÓN]... t DIRECTORIO ORIGEN...
Copia ORIGEN a DESTINO, o varios ORIGEN(es) a DIRECTORIO.

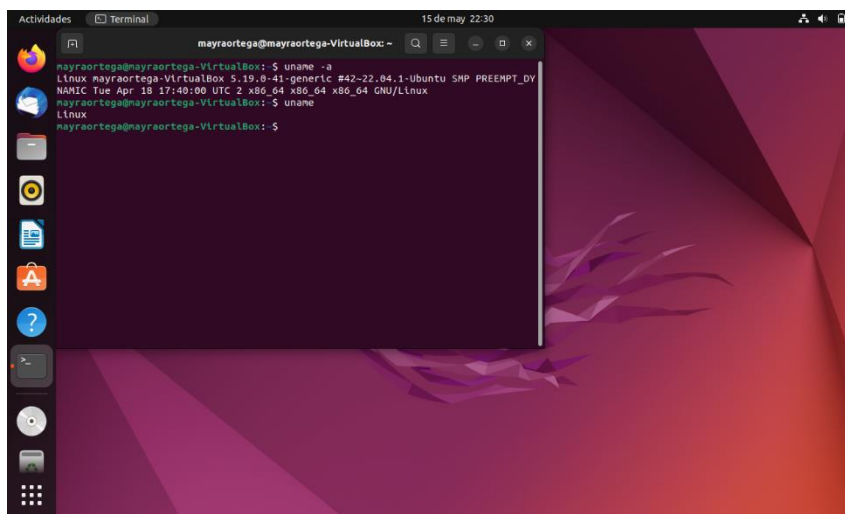
Los argumentos obligatorios para las opciones largas son también obligatorios
para las opciones cortas.
-a, --archive              lo mismo que -dR --preserve=all
--attributes-only          no copia los datos del fichero, solamente los
                             atributos
--backup[=CONTROL]         crea una copia de seguridad de cada fichero de
                             destino que exista
-b                          como --backup pero no acepta ningún argumento
--copy-contents            copia el contenido de los ficheros especiales
                             cuando opera recursivamente
-d                          lo mismo que --no-dereference --preserve=link
-f, --force                si un fichero de destino no se puede abrir, lo
                             borra y lo intenta de nuevo (no se tiene en
                             cuenta si se utiliza también la opción -n)
-i, --interactive          pide confirmación antes de sobrescribir
-H                          sigue los enlaces simbólicos de la línea
                             de órdenes
-L, --link                 crea enlaces duros de los ficheros en vez de copiarlos
-L, --dereference           siempre sigue los enlaces simbólicos en ORIGEN
-n, --no-clobber            no sobrescribe un fichero que exista
                             (tiene prioridad sobre una opción -i anterior)
--no-dereference           nunca sigue los enlaces simbólicos en ORIGEN
-p                          igual que --preserve=mode,ownership,timestamps
--preserve[=LISTA_ATTR]    conserva si puede los atributos especificados,
                             (por omisión: mode,ownership,timestamps)
                             atributos adicionales: context, links, xattr,
                             all
--no-preserve=LISTA_ATTR  no conserva los atributos especificados
--parents                 añade el directorio de origen a DIRECTORIO
-R, --recursive            copia directorios recursivamente
--preserve=LISTA_ATTR     conserva si puede los atributos especificados,
                             (por omisión: mode,ownership,timestamps)
                             atributos adicionales: context, links, xattr,
                             all
```

Cp; copia los archivos través de la línea de comando, toma dos argumentos ubicados del archivo que se va a copiar y el segundo la dirección a donde se va a copiar.



```
mayraortega@mayraortega-VirtualBox: ~  
$ mv  
mv: falta un archivo como argumento  
Pruebe 'mv --help' para más información.  
mayraortega@mayraortega-VirtualBox: ~$ mv --help  
Modo de empleo: mv [OPCIÓN]... ORIGEN DESTINO  
o bien: mv [OPCIÓN]... ORIGEN... DIRECTORIO  
o bien: mv [OPCIÓN]... -t DIRECTORIO ORIGEN...  
Renombra ORIGEN a DESTINO, o mueve ORIGEN(es) a DIRECTORIO.  
  
Los argumentos obligatorios para las opciones largas son también obligatorios  
para las opciones cortas.  
-b, --backup[=CONTROL] crea una copia de seguridad antes de borrar  
como --backup pero no acepta ningún argumento  
-f, --force no pregunta nunca antes de sobrescribir  
-i, --interactive pide confirmación antes de sobrescribir  
--strip-trailing-slashes elimina todas las barras finales de cada  
argumento ORIGEN  
-s, --suffix=SUFFIX reemplaza el sufijo de respaldo habitual  
-t, --target-directory=DIRECTORIO mueve todos los argumentos ORIGEN al  
directorio DIRECTORIO  
-T, --no-target-directory trata DESTINO como fichero normal  
-u, --update mueve solamente cuando el fichero ORIGEN  
es más moderno que el fichero de destino,  
o cuando falta el fichero de destino  
-v, --verbose da detalles de lo que va haciendo  
-Z, --context establece el contexto de seguridad SELinux del fichero  
de destino al tipo predeterminado  
--help muestra esta ayuda y finaliza  
--version informa de la versión y finaliza  
  
El sufijo de respaldo es '-', a menos que se establezca con --suffix o con  
SIMPLE_BACKUP_SUFFIX. El método de control de versión se puede seleccionar  
con la opción --backup o a través de la variable de entorno VERSION_CONTROL.  
Estos son los valores:  
  
none, off nunca realiza copias de seguridad (incluso si se da la  
opción --backup)  
numbered, t crea copias de seguridad numeradas  
existing, nil numeradas si existen copias de seguridad numeradas.
```

mv. es utilizado para mover archivos a través de la línea de comandos, también cambia el nombre de un archivo ejemplo text a new podemos mv text new.



```
mayraortega@mayraortega-VirtualBox: ~  
$ uname -a  
Linux mayraortega-VirtualBox 5.19.0-41-generic #42-22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Tue Apr 18 17:40:00 UTC 2 x86_64 x86_64 x86_64 GNU/Linux  
mayraortega@mayraortega-VirtualBox: ~$ uname  
Linux  
mayraortega@mayraortega-VirtualBox: ~$
```

Uname; muestra la información del sistema que ejecuta su distribución de Linux, uname -a imprime la mayor parte de información en el sistema.

5.1 CONCLUSION

Para VirtualBox su principal ejecución es a treves de los comandos básicos de Shell que nos ayuda a probar otros programas y rutas de acceso más factibles para nuestras tareas además nos ayudan ahorrar tiempo de trabajo además son más eficaces para su administración con esta herramienta junto con una serie de comandos se pueden instalar Software dentro de tu máquina virtual, así mismo para los usuarios y miembros de algunas organizaciones les ayuda ejecutar tareas de una maneras más fácil en el menor tiempo posible. Actualmente la gran parte de las empresas usan este medio de desarrollo ya que ofrece soluciones a cualquier tipo de situaciones de diversos entornos.

Además, este Software no tiene ningún costo ni anualidad por lo que es una buena herramienta para reducir los costos operativos y acortar el tiempo necesario para implementar aplicaciones de forma segura en entornos locales y en la nube.

En otras palabras, una máquina virtual le permite crear otra computadora en una computadora que utilizará sus recursos existentes, esta trabajará de forma aislada.

La virtualización y la máquina virtual en particular, amplían las capacidades de la infraestructura de la tecnología. Este medio es de gran importancia ya que su utilidad es para los desarrolladores en general de Software que buscan mejoras hacia un sistema operativo.

6.1 REFERENCIAS

<https://www.redhat.com/es/topics/linux>

Sitio web proveedor líder a nivel mundial de soluciones empresariales de código abierto, incluyendo Linux, cloud, contenedores y Kubernetes. Ofrece soluciones reforzadas, las cuales permiten que las empresas trabajen en distintas plataformas y entornos con facilidad, desde el centro de datos principal hasta el extremo de la red.

<https://www3.gobiernodecanarias.org/medusa/ecoescuela/seguridad/ciudadania-y-seguridad-tic/principios-legales/software-libre/ubuntu-linux/>

<https://www.virtualbox.org/>

Sistema operativo, VirtualBox se está desarrollando activamente con lanzamientos frecuentes y tiene una lista cada vez mayor de funciones, sistemas operativos invitados compatibles y plataformas en las que se ejecuta.

<https://ubuntu.com/>

distribución Linux basada en Debian GNU/Linux, que incluye principalmente software libre y de código abierto.