

Actividad | #2 Comandos Básicos en Linux

Ingeniería en Desarrollo de Software



TUTOR: Francisco Ortega Rivera

ALUMNO: Edgar Enrique Cuamea Ochoa

FECHA: 23 de abril del 2024

índice.

Lineamientos básicos.....	Página 3
¿Qué aprenderemos en esta actividad?	Pagina 4
¿Por qué necesitamos aprender comandos en Linux?.....	Pagina 5
Comandos Básicos.....	Pagina 6
Escritura y ejecución de comandos.....	Pagina7
Pwd.....	Pagina 8
Cd.....	Pagina 8
Mkdir.....	Pagina 9
Ls.....	Página 10
Touch.....	Pagina 10
Mv.....	Pagina 10
Cp.....	Pagina 11
Clear.....	Pagina 11
Rmdir.....	Pagina 13
Rm.....	Pagina 13
Uname.....	Pagina 14
Exit.....	Pagina 14
¿Que aprendimos en esta actividad?.....	pagina 15
Referencias.....	Pagina 16

Lineamientos básicos.

En esta segunda actividad veremos algunos comandos básicos que necesitaremos aprender para el correcto uso y manejo de nuestro sistema operativo. Utilizaremos el sistema operativo Ubuntu previamente instalado en una máquina virtual, se crearán algunas carpetas, copiar y renombrar archivos, movernos entre directorios, mover archivos, eliminar directorios, ver en que directorio nos encontramos a través de estos comandos que ejecutaremos en Shell proporcionada por el sistema operativo, etc.

Se explicarán como se escribe el comando y que función ejecuta en la terminal de nuestro sistema operativo con el fin de entender bien el uso correcto de estos comandos lo que nos ayudara a realizar un trabajo de manera limpia y eficiente en el transcurso de esta actividad, así como familiarizarnos más con el sistema de línea de códigos que ejecutaremos en Shell y conocer mas sobre el sistema operativo de Ubuntu, también aprenderemos a navegar a través del sistema y facilitar la comprensión de este por medio de dichos comando básicos.

¿Qué aprenderemos en esta actividad?

En esta actividad se realizará la ejecución de distintos códigos con el fin de aprender a navegar por el sistema y familiarizarnos con Shell, necesitaremos una comprensión de la terminal y escritura correcta de los códigos para un uso óptimo de estos, así como entender mas del sistema operativo para aquellos que solo han usado una terminal grafica como en Windows y nunca han ejecutado código en alguna terminal en sistemas operativos tales como Windows o Mac OS.

Entenderemos la función de distintos comandos de Linux ya que son diferentes al uso de los comandos de CMD de Windows, aunque la función de los comandos puede ser la misma en Linux como en Windows, la escritura de estos comandos puede ser distintos por el hecho de que Windows es un sistema operativo de Microsoft y Ubuntu es un sistema operativo de código libre basado en Linux, lo que hace más fácil la comprensión de Linux por la gran comunidad que hay detrás de este sistema.

¿Por qué necesitamos aprender comandos en Linux?

Es importante conocer el correcto uso de los comandos escritos en Linux por ser el sistema en el que se basan los servidores, al ser estos tan grandes en información, nos será más fácil trabajar con archivos concretos conociendo los directorios en los que se encuentran sin buscar en todo el sistema haciendo nuestro trabajo más rápido y eficiente.

La correcta escritura de esta también es fundamental en la ejecución de código en la terminal, ya que un solo error en la gramática de los comandos puede cambiar el funcionamiento de este o incluso no ejecutarse en Shell provocando un error y por consecuencia volver a escribir el comando completo en la terminal, para evitar estos errores es importante revisar la sintaxis de cada comando, tener los permisos necesarios, ya que sin estos habrán algunos comandos que no se ejecutaran en la terminal.

Nos apoyaremos con el comando “**help**” escrito en la terminal para acceder a una lista de comandos, así como ver sus funciones en Linux.

Comandos Básicos.

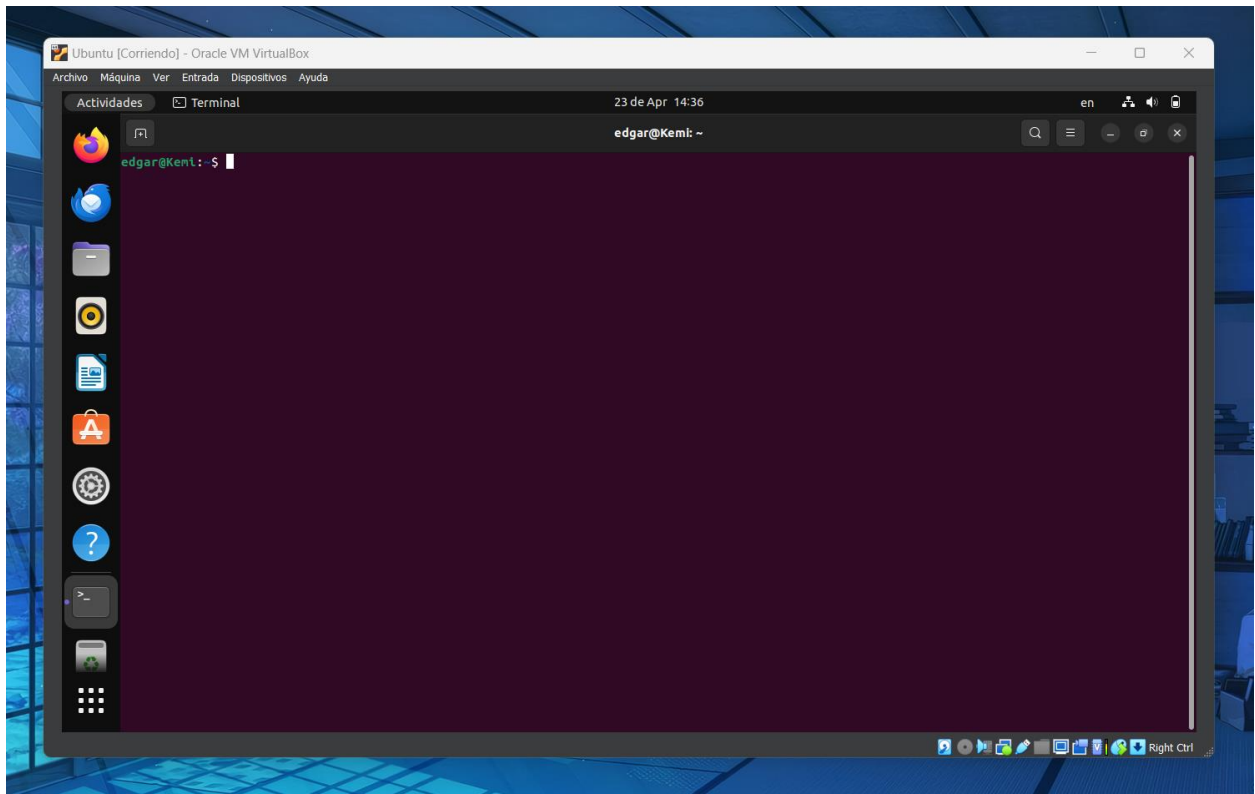
Algunos comandos básicos de Linux que veras serán los siguientes.

- **Pdw.** Muestra el directorio en el que estamos.
- **Ls.** sirve para saber que archivos hay en el directorio en el que se encuentra.
- **Cd.** Sirve para movernos entre directorios.
- **Mkdir.** Crea un nuevo directorio o carpeta.
- **Rmdir.** Elimina un directorio vacío del sistema.
- **Touch.** Crea un archivo,
- **Cp.** Copia archivos y directorios de un lugar a otro.
- **Mv.** Mueve archivos de lugar.
- **Exit.** Cierra la Shell actual.
- **Uname.** Muestra el kernel que se está utilizando.
- **Clear.** Se utiliza para eliminar todo el texto escrito de la terminal.
- **Rm.** Elimina archivos.

Esto con el fin de que aprendamos el uso correcto de estos comandos y saber diferenciar sus funciones en la terminal.

Escritura y ejecución de comandos.

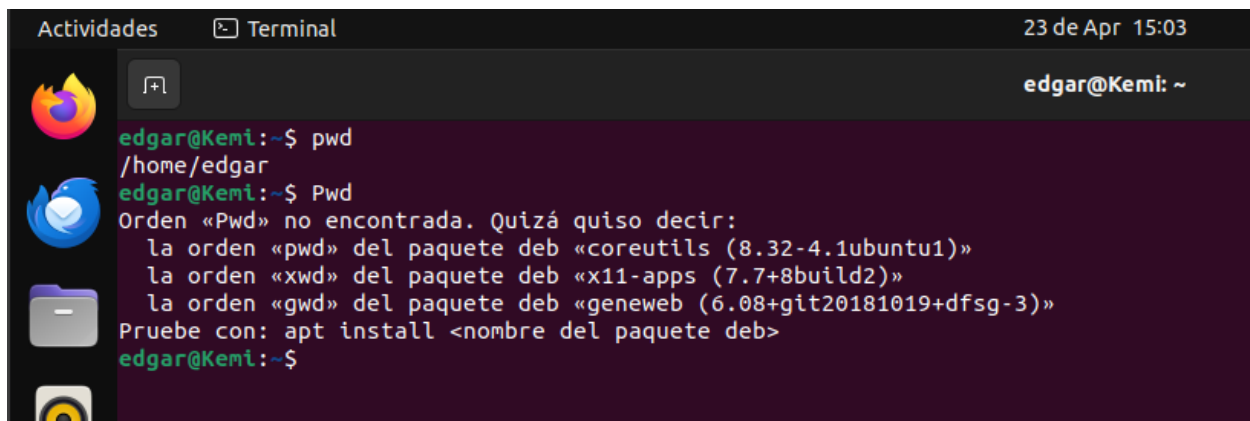
En este primer paso en la escritura y ejecución de comandos en Shell será necesario encender nuestra máquina virtual con nuestro sistema operativo Ubuntu y abrir la terminal como se muestra en la siguiente imagen.



Veremos que por defecto viene escrito algo parecido a esto: `edgar@Kemi:~$`. Aparecerá nuestro nombre de usuario que proporcionamos en la instalación de nuestro dispositivo, en mi caso mi primer nombre, seguido de la letra “@” y después el nombre de nuestra máquina, en mi caso yo lo llame Kemi, seguido de esto encontraremos dos símbolos más, el primer símbolo representara en que directorio nos encontramos y el segundo símbolo representa la solicitud de nuestro usuario donde a continuación podremos escribir el siguiente comando, esto aparecerá al principio de cualquier comando escrito para que podamos saber que usuario esta usando la terminal en caso de que se tengan distintos usuarios en nuestro sistema operativo y diferenciar

cuando se este usando el modo de administrador o super usuario, y poder diferenciar del comando escrito y la ejecución de dicho comando.

Empecemos viendo en que directorio nos encontramos escribiendo el siguiente comando en nuestra terminal. Todos los comandos los escribiremos en minúsculas para así respetar la sintaxis para así obtener el resultado correcto del comando.



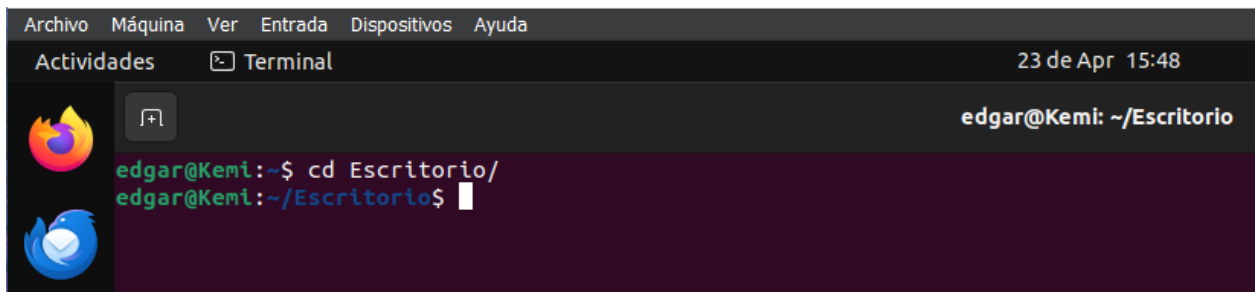
```
edgar@Kemi:~$ pwd
/home/edgar
edgar@Kemi:~$ Pwd
Orden «Pwd» no encontrada. Quizá quiso decir:
  la orden «pwd» del paquete deb «coreutils (8.32-4.1ubuntu1)»
  la orden «xwd» del paquete deb «x11-apps (7.7+8build2)»
  la orden «gwd» del paquete deb «geneweb (6.08+git20181019+dfsg-3)»
Pruebe con: apt install <nombre del paquete deb>
edgar@Kemi:~$
```

Pwd. Este comando nos servirá para ver en que directorio nos encontramos, en este caso me encuentro en el directorio “home” seguido de mi nombre de usuario.

Es importante respetar la escritura de este ya que, si no equivocamos en ella, no nos mostrara el resultado esperado, en este caso la primera vez lo escribí correctamente y la segunda vez puse la primera letra en mayúscula con el fin de que veas el error que causa no respetar la sintaxis, en este segundo caso se nos da algunas recomendaciones de algunos comandos sugeridos y de que paquete se encuentra y la recomendación de instalar algún otro paquete.

Cd. Empecemos creando una carpeta en un directorio que queramos, en este caso yo lo creare en el escritorio, para ello usaremos el comando “cd” para movernos al escritorio.

Escribiremos el comando en el terminal seguido de la ruta al cual queremos movernos, ya sea Descargas, Música, Documentos, Escritorio, Etc.

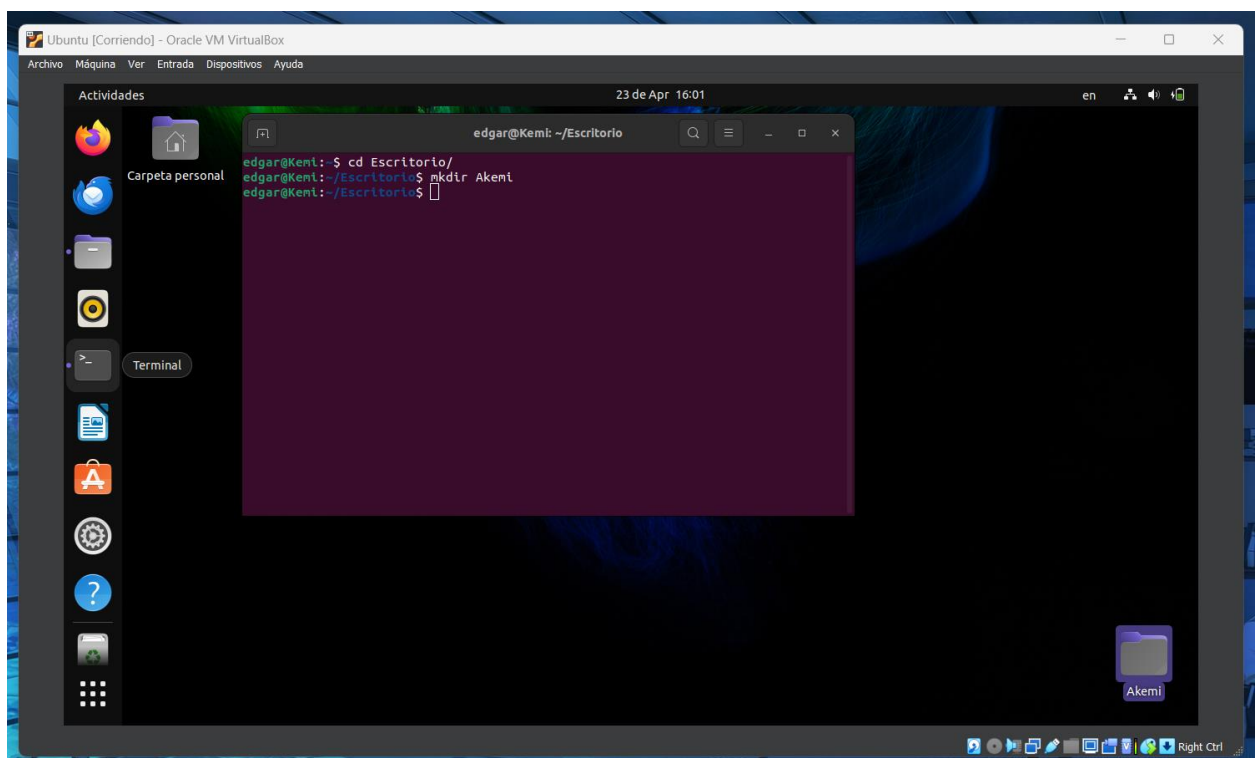


```
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal  23 de Apr 15:48
edgar@Kemi: ~/Escritorio

edgar@Kemi:~$ cd Escritorio/
edgar@Kemi:~/Escritorio$
```

Una vez ejecutado el comando nos encontraremos en el directorio deseado y escribiremos el siguiente comando para crear una carpeta en el directorio deseado.

Mkdir. Este comando funciona para crear una carpeta dentro del directorio donde nos encontramos, para eso escribiremos el comando seguido del nombre que le daremos a la carpeta y presionamos “enter” para efectuar el comando.



En este caso se creo la carpeta deseada con el nombre que queramos en la ruta especifica que escribimos, en mi caso se creo la carpeta “Akemi” en la esquina inferior derecha.

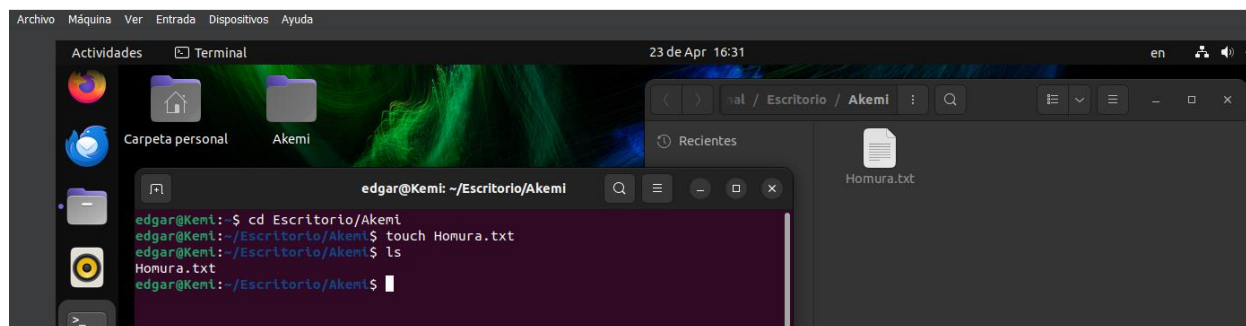
Ls. Para verificar que si se haya creado la carpeta sin minimizar la terminal podemos usar el siguiente comando “ls”, este comando nos servira para ver todos los archivos que se encuentren en el directorio escrito a travez de la terminal, en este caso en el escritorio.



```
edgar@Kemi:~$ cd Escritorio/
edgar@Kemi:~/Escritorio$ ls
Akemi
edgar@Kemi:~/Escritorio$
```

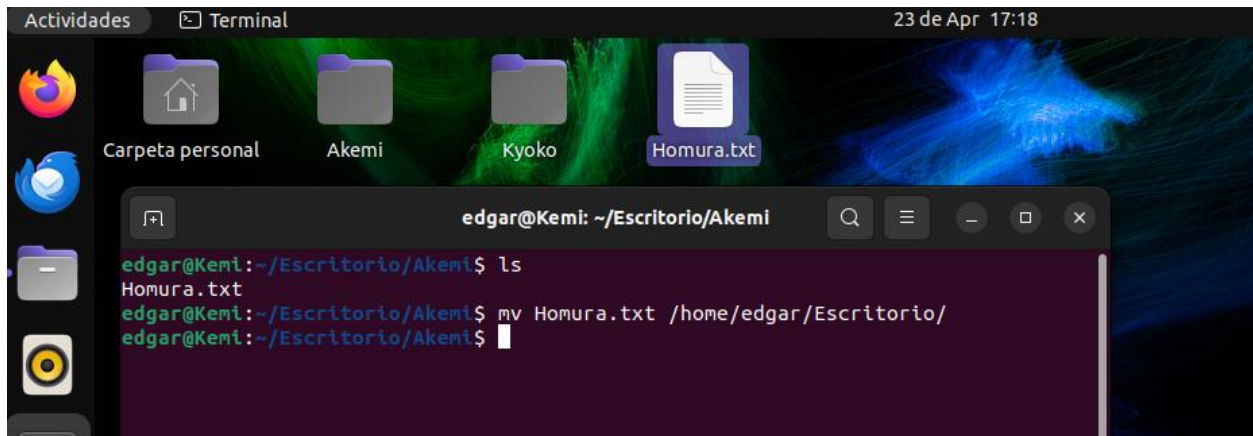
A continuación, crearemos un archivo de texto en la carpeta que acabamos de crear, para ello utilizaremos el comando.

Touch. Nos moveremos a la carpeta creada con el comando “cd” a la carpeta “Akemi” y ejecutaremos el comando “touch” seguido del nombre del archivo y del fichero “.txt” para especificar que será un archivo de texto y no otro tipo de archivo.



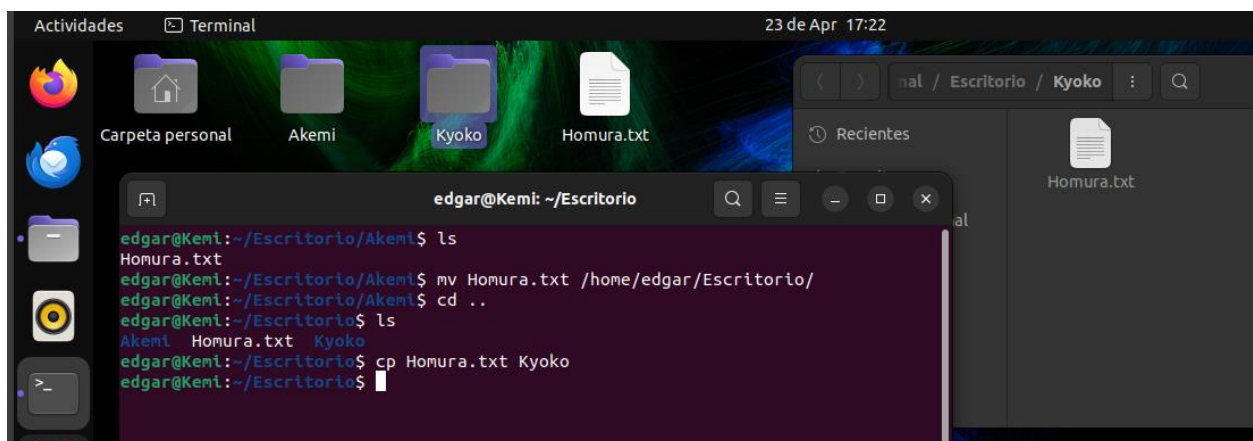
En este caso yo lo llame “Homura” y podemos comprobar con el comando “ls” en la terminal que efectivamente se creo este archivo de texto vacío en el directorio especifico.

Mv. A continuación, moveremos el archivo de la carpeta “Akemi” y la pegaremos en el escritorio con el comando “mv” que nos permite mover archivos a otro directorio.



En esta imagen se muestra como mover un archivo en el escritorio, para ello tendremos que escribir la ruta absoluta donde queremos mover nuestro archivo de texto.

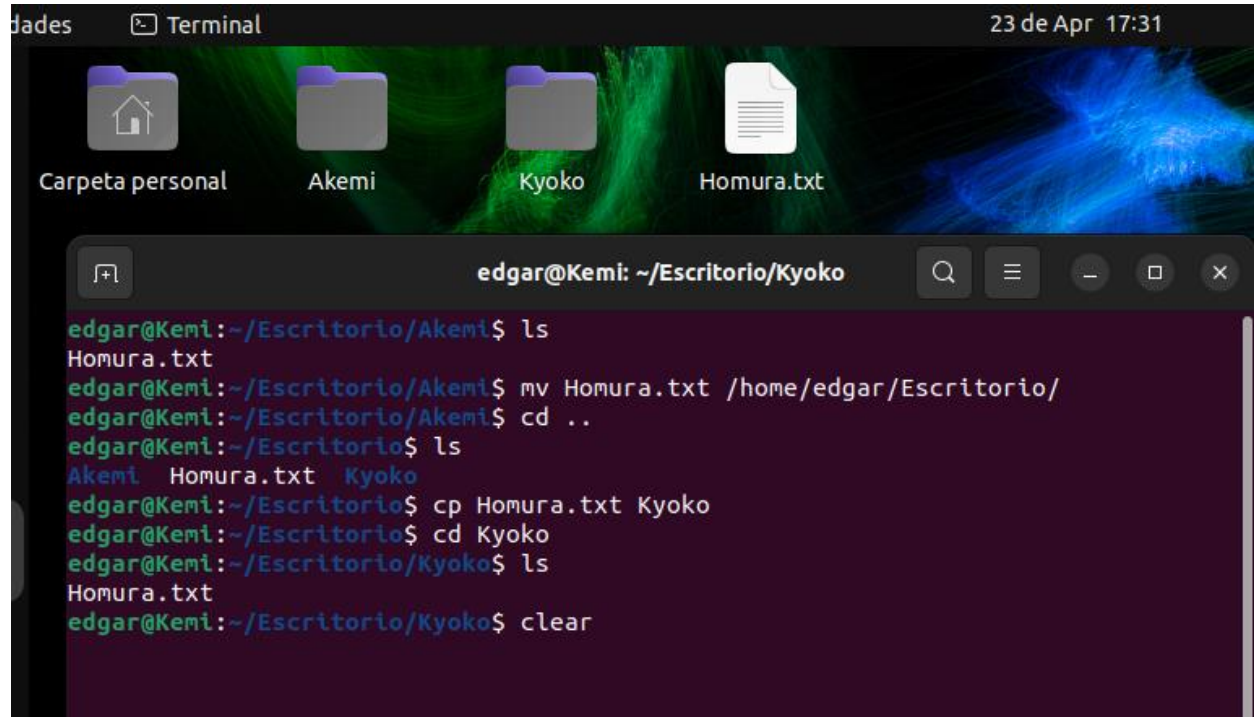
Cp. Con este comando aprenderemos a copiar archivos en este caso cree una carpeta llamada “Kyoko” donde copiaremos nuestro archivo de texto y eliminar la carpeta “Akemi” mas adelante, para copiar el archivo de texto escribiremos el siguiente comando.



Como se muestra en la imagen, uso el comando “cd ..” para regresar a un directorio anterior, en este caso el escritorio y uso el comando “ls” para ver los archivos que tengo en escritorio para así copiar el texto con el comando “cp” como vemos en los archivos, este se copio correctamente, ahora procederemos a limpiar el texto de la terminal.

Clear. Este comando nos servirá para borrar el contenido escrito en shell y empezar con una terminal limpia sin perder el directorio en el que nos encontramos, para eso me moveré a la

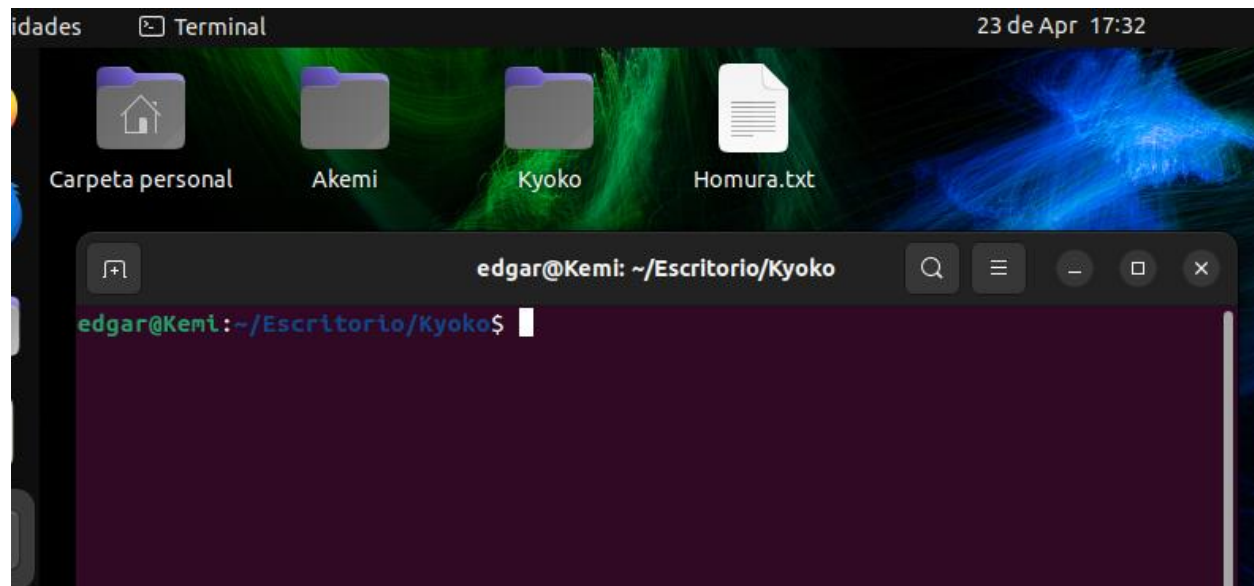
carpeta “Kyoko” y procedere a efectuar el comando como en las siguientes imágenes.



The screenshot shows a Linux desktop with a dark theme. At the top, there's a panel with icons for 'Carpeta personal', 'Akemi', 'Kyoko', and 'Homura.txt'. Below this, a terminal window is open with the title 'edgar@Kemi: ~/Escritorio/Kyoko'. The terminal shows the following commands and their outputs:

```
edgar@Kemi:~/Escritorio/Akemi$ ls
Homura.txt
edgar@Kemi:~/Escritorio/Akemi$ mv Homura.txt /home/edgar/Escritorio/
edgar@Kemi:~/Escritorio/Akemi$ cd ..
edgar@Kemi:~/Escritorio$ ls
Akemi Homura.txt Kyoko
edgar@Kemi:~/Escritorio$ cp Homura.txt Kyoko
edgar@Kemi:~/Escritorio$ cd Kyoko
edgar@Kemi:~/Escritorio/Kyoko$ ls
Homura.txt
edgar@Kemi:~/Escritorio/Kyoko$ clear
```

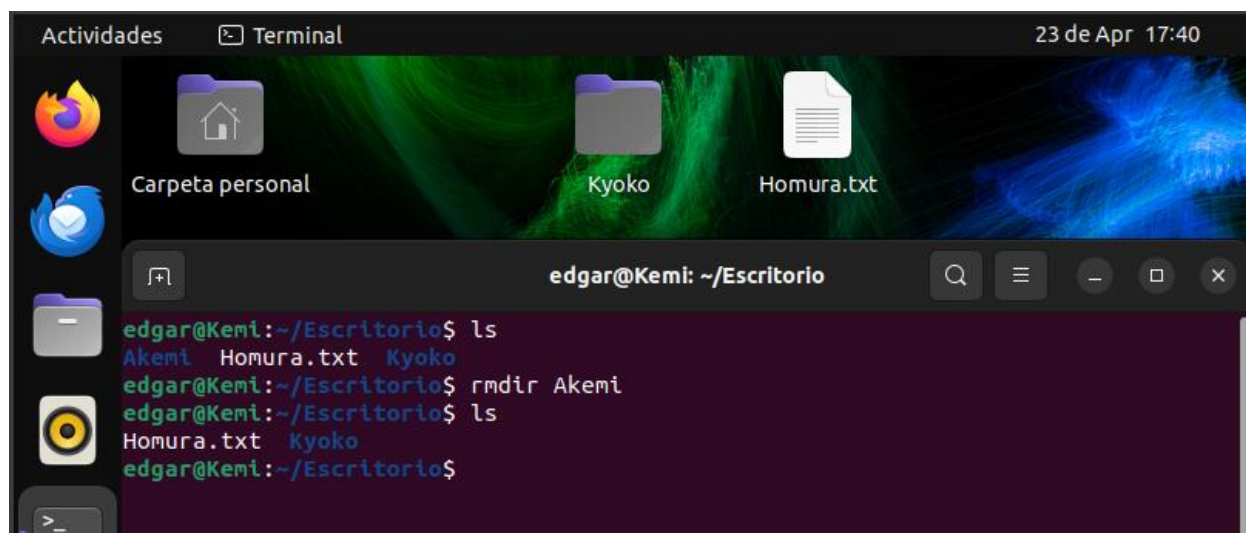
Una vez ejecutado el comando se limpiara la terminal.



This screenshot shows the same desktop environment as the previous one. The terminal window is still open, but the content has been cleared. The prompt now shows 'edgar@Kemi:~/Escritorio/Kyoko\$' followed by a cursor. The desktop icons and the terminal title bar remain the same.

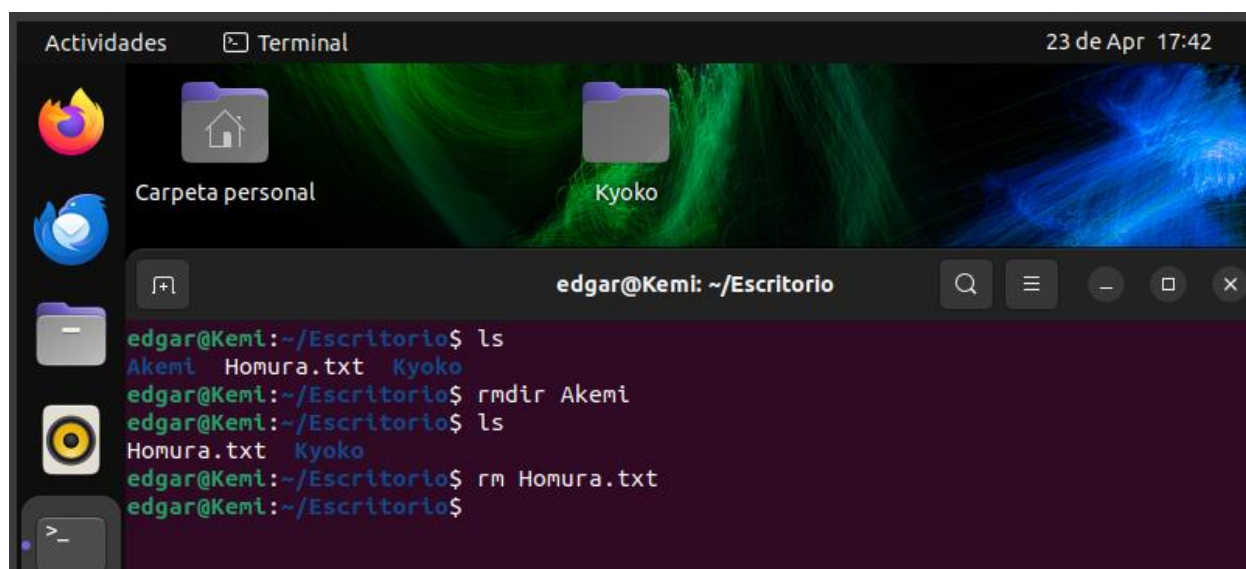
En este punto sabremos que no hay ningun archivo en la carpeta llamada “Akemi”asi que la eliminaremos con el siguiente comando.

Rmdir. Este comando nos servirá para eliminar directorios vacíos, como en la siguiente imagen.



Si queremos eliminar un archivo como el archivo “Homura” tendremos que utilizar un comando parecido.

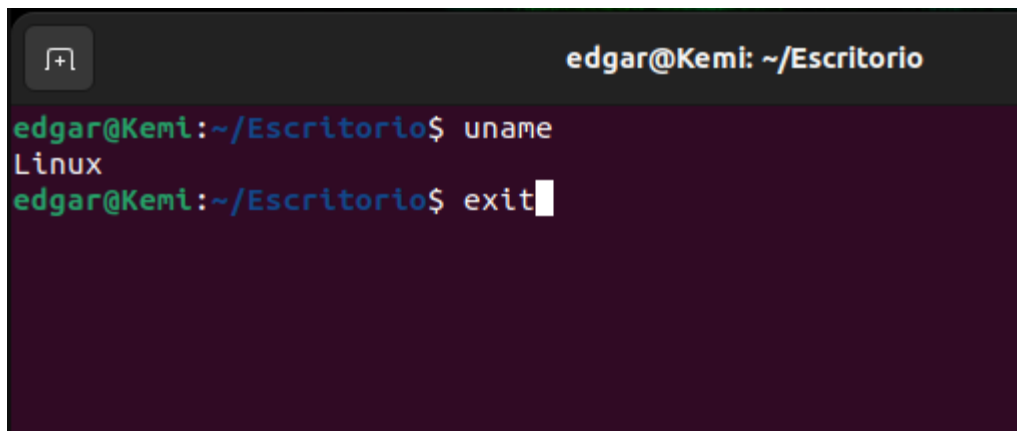
Rm. Este comando nos servirá para eliminar archivos, ya que “rmdir” solo elimina directorios vacíos.



Como pudimos ver en la imagen, estos comandos sirven para borrar contenido, solo que uno es para archivos y otro para directorios, todos estos comandos se realizan en el kernel de Linux

que podemos comprobar con el siguiente comando.

Uname. Este comando nos muestra en que kernel estamos trabajando así como se muestra en la siguiente imagen, al finalizar los comandos o cuando ya no queramos tener la terminal abierta podemos escribir este comando útil que nos servirá al final de nuestro trabajo

A screenshot of a terminal window with a dark background. The title bar at the top shows a window icon and the text 'edgar@Kemi: ~/Escritorio'. The terminal content shows the prompt 'edgar@Kemi:~/Escritorio\$' followed by the command 'uname', which returns the output 'Linux'. Below that, the prompt is again 'edgar@Kemi:~/Escritorio\$' followed by the command 'exit', with a white cursor at the end of the line.

```
edgar@Kemi:~/Escritorio$ uname
Linux
edgar@Kemi:~/Escritorio$ exit
```

Exit. Este comando como su nombre lo indica, nos sirve para cerrar la terminal que usemos, al presionar “enter” este automáticamente se cerrara si mostrar ningún mensaje.

¿Que aprendimos en esta actividad?

En conclusión en esta actividad vimos las maneras en las que podemos desde crear archivos, moverlos, copiarlos y eliminarlos hasta como movernos entre las carpetas y directorios en Ubuntu con la finalidad de poder relacionarnos con el sistema operativo, al principio es complicado entender para que sirve cada comando pero con el tiempo de uso te vas acostumbrando al significado de este y como complementarlos para que sea más fácil la navegación por el sistema, al ser comandos básicos es lo que necesitamos aprender al principio es la sintaxis de estos y a ver que tengamos un control parcial y total sobre lo que hagamos en esta actividad así como tener en cuenta como es que funciona Shell y la línea de comandos para en un futuro utilizarlo para actividades que realizaremos más adelante en esta materia de ingeniería en desarrollo en software por como su nombre indica tendremos que aprender el uso de los comandos y de Shell.

Referencias.

A, D., & A, D. (2024, March 14). *60 Comandos esenciales y populares de Linux*. Tutoriales

Hostinger. <https://www.hostinger.mx/tutoriales/linux-comandos>

Comandos Básicos de Shell (Terminal) | Facultad de Ingeniería. (n.d.).

<https://www.fing.edu.uy/es/sysadmin/salas-estudiantiles-linux/comandos->

[b%C3%A1sicos-de-shell-terminal](https://www.fing.edu.uy/es/sysadmin/salas-estudiantiles-linux/comandos-b%C3%A1sicos-de-shell-terminal)

Corbalán, A. S. (2022, April 8). Los 50 mejores Comandos Linux del Shell Bash que debes

conocer - Antonio Sánchez Corbalán. *Antonio Sánchez Corbalán*.

<https://sanchezcorbalan.es/mejores-comandos-linux-bash/>