

ACADEMÍA JAVA XIDERAL MONTERREY, N.L



SEMANA 1 INTRODUCCIÓN A LINUX Y LÍNEA DE COMANDOS

**Realizado por:
Luis Miguel Sánchez Flores**

INTRODUCCIÓN

Linux es un sistema operativo de código abierto basado en Unix con herramientas de multiprocesamiento. Tiene una amplia selección de características para distinguirlo como un sistema operativo único.

El sistema operativo fue creado por Linus Torvalds en el año 1991, y desde entonces ha ganado popularidad gracias a ser un sistema de alta flexibilidad, escalabilidad y personalizable, convirtiéndolo en una opción popular para una amplia gama de aplicaciones, utilizado desde sistemas embebidos pequeños hasta supercomputadoras.

Además, al ser de código abierto, los usuarios aportan avances relacionados al desarrollo de software, lo que produjo la creación de diferentes “distribuciones” que empaquetan varios programas y utilidades que satisfacen las necesidades de un grupo determinado de usuarios. Una distribución puede ser fácil de usar mediante interfaces y aplicaciones familiares, como Zorin o Ubuntu, o puede ser demasiado avanzada dirigida a los usuarios técnicos expertos, como Arch Linux. Informáticos de distintas naciones fortalecen a los servidores Linux



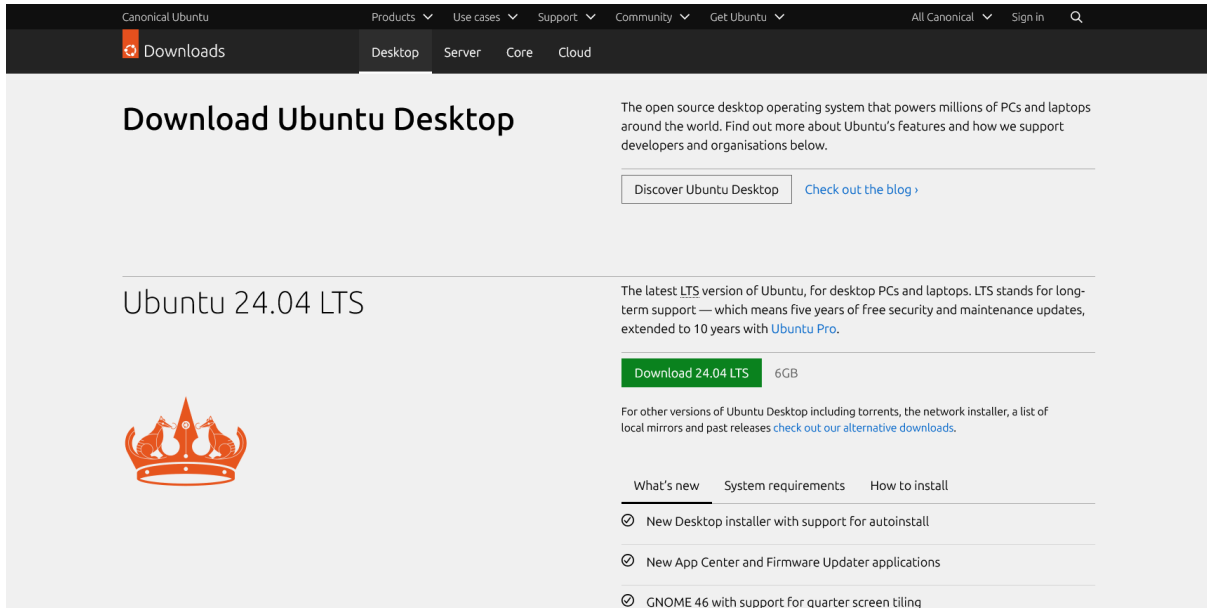
Lo que es común ver utilizar a los usuarios de linux, independientemente de la distribución que utilizan es la **línea de comandos**, ya que es una parte fundamental del sistema operativo que permite el acceso a las funcionalidades del sistema y donde normalmente tiene lugar la personalización y el control.

A través de la línea de comandos, los usuarios pueden añadir, modificar y eliminar contenidos de su unidad, y escribir scripts para automatizar tareas repetitivas / mundanas de una manera rápida y eficaz, por lo que es una opción preferida para los usuarios más experimentados y administradores del sistema operativo.

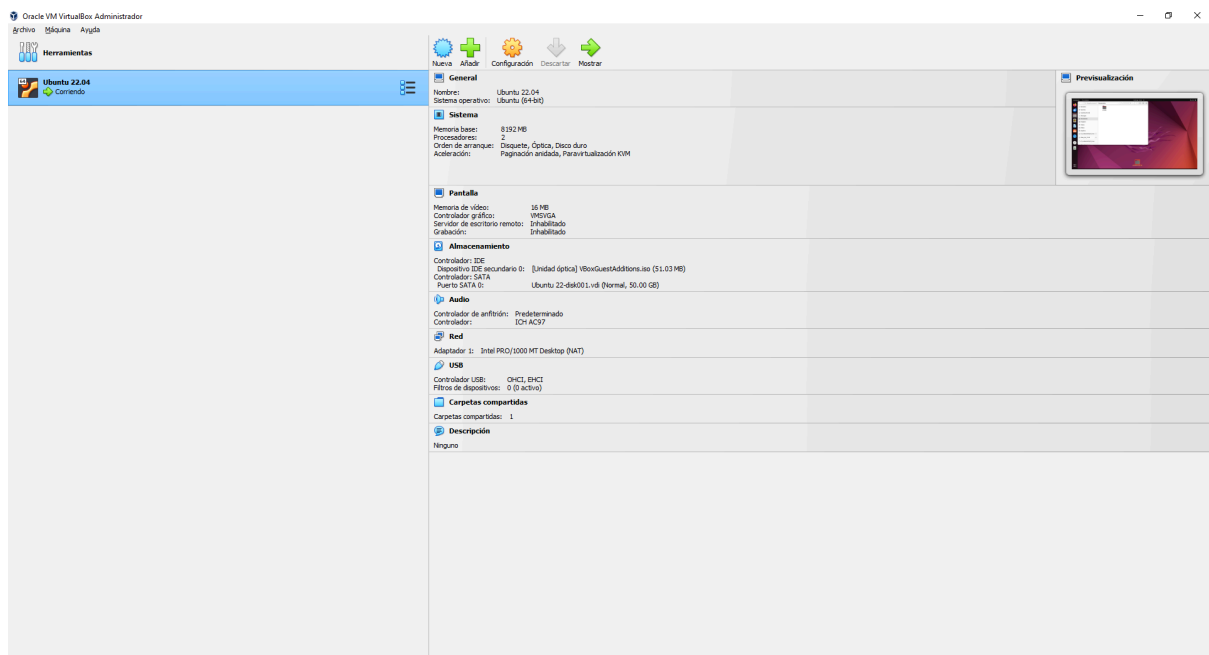
Los usuarios principiantes tienen la suposición que, debido a un sistema altamente personalizable y flexible, puede ser muy complejo utilizar la línea de comandos, y que se requiere de un conocimiento profundo de programación para poder usarlo. No obstante, con la práctica adecuada con los comandos más sencillos y tareas pequeñas hacen de una experiencia agradable para el usuario y empezar a profundizar en el potente ecosistema que es Linux.

PRIMEROS PASOS

En este caso, se hace uso de la distribución gratuita de **Ubuntu** que se puede [descargar](#) e instalar ya sea en una computadora física o con el uso de una máquina virtual mediante programas como Virtualbox o VMWare.

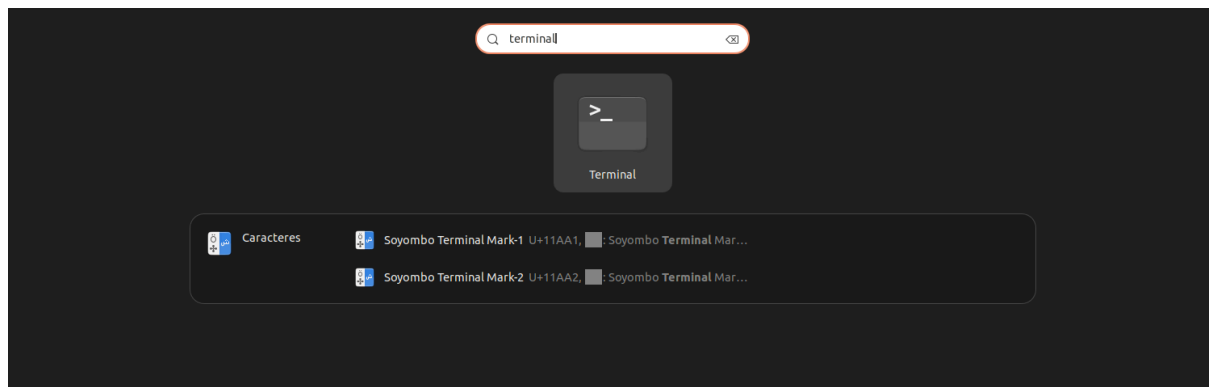
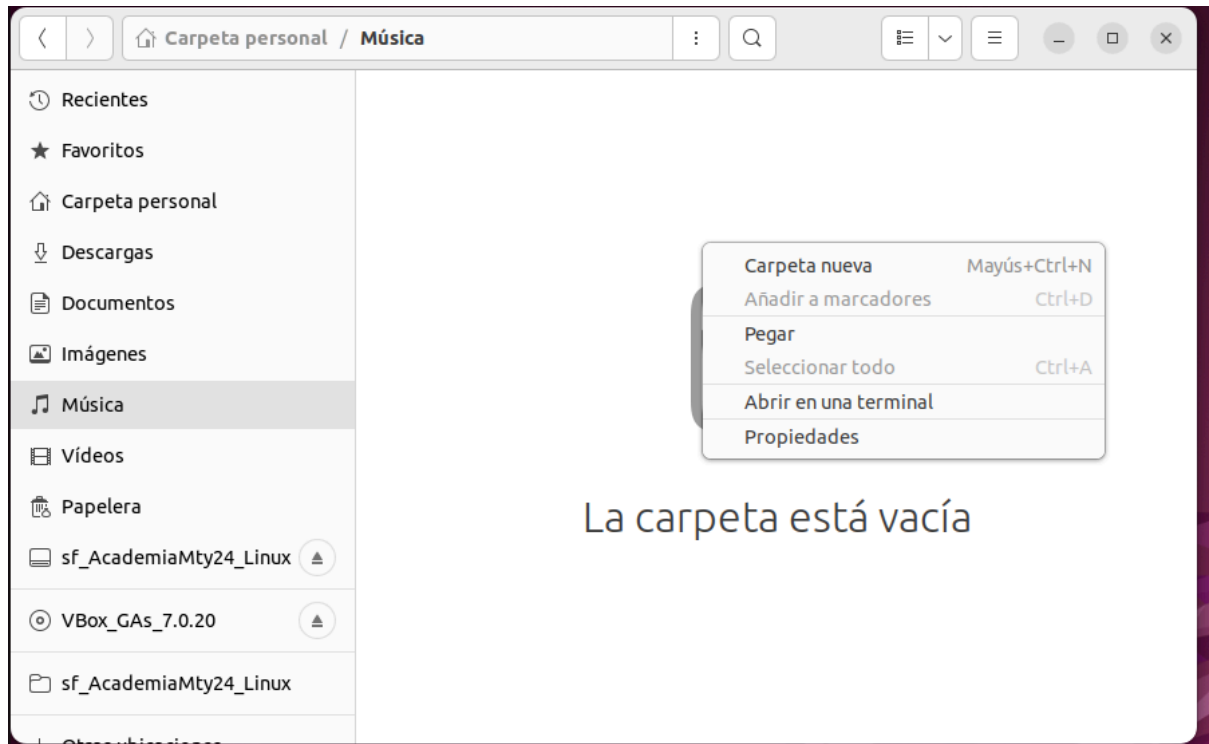


The screenshot shows the Canonical Ubuntu website's 'Download Ubuntu Desktop' page. The header includes navigation links for Products, Use cases, Support, Community, and Get Ubuntu. The main heading is 'Download Ubuntu Desktop', followed by a description of Ubuntu as an open-source desktop operating system. A 'Discover Ubuntu Desktop' button and a link to 'Check out the blog' are present. Below this, the 'Ubuntu 24.04 LTS' section features the Ubuntu logo and a 'Download 24.04 LTS' button indicating a 6GB download. A list of features for the latest LTS version is provided, including a new Desktop installer, App Center, and GNOME 46. Links for 'What's new', 'System requirements', and 'How to install' are also visible.



Después de configurar el sistema operativo en la máquina, podemos acceder a la **terminal** mediante una de las siguientes maneras:

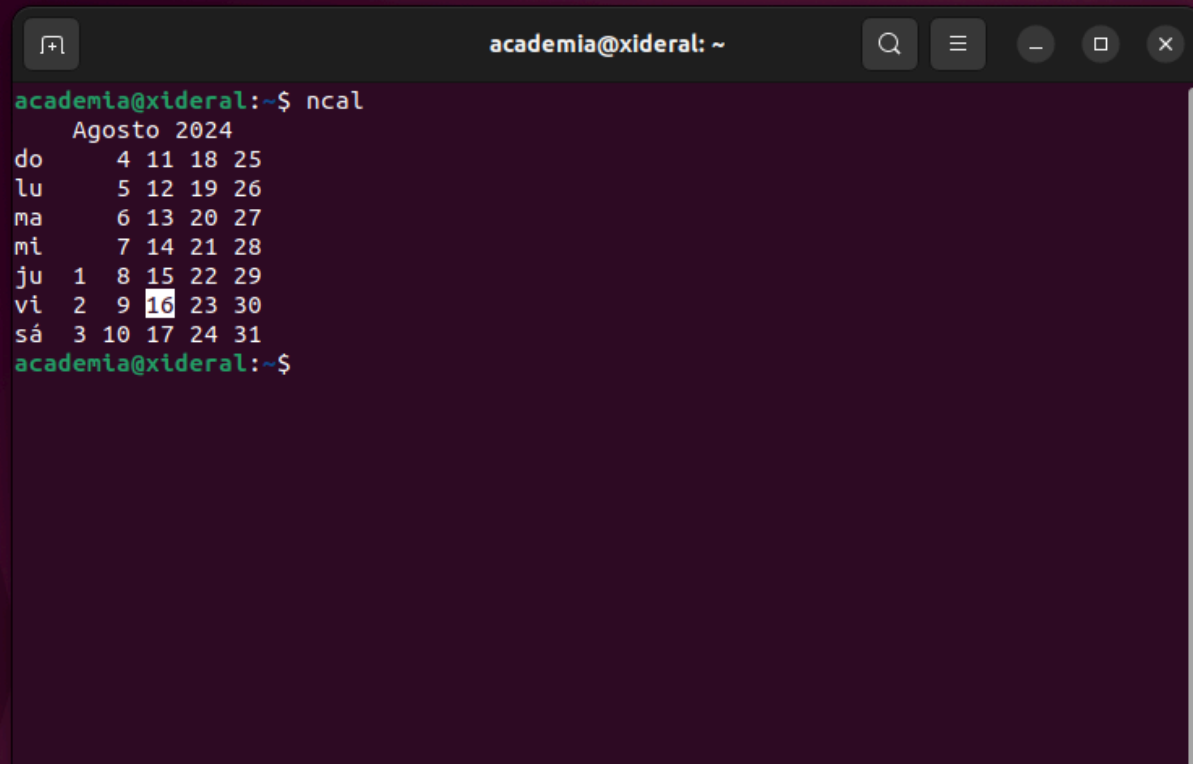
- Con el uso de las teclas de acceso directo: CTRL+ALT+T
- A través de la lista de aplicaciones y seleccionando Terminal
- Dentro del explorador de archivos integrado, dando clic derecho en la ventana y seleccionando la opción de Terminal



El **Terminal** es un software comúnmente utilizado por los usuarios de Linux para interactuar con el **Shell** del sistema operativo (conocido como Bash en Ubuntu). El Shell trabaja como un intermediario entre los comandos del usuario y el **Kernel** en el que interpreta, ejecuta y despliega los resultados del comando hacia el usuario. El Kernel es el componente núcleo del sistema operativo que interactúa entre el hardware y software del ordenador, así como administrar los servicios para componentes esenciales, como el control de dispositivos, administración de memoria, entre otros.

Al abrir el terminal, el shell se prepara para aceptar las peticiones / comandos que provoquen la acción deseada por el usuario. Estos comandos pueden ser rápidos, de varias líneas o pueden tardar algún tiempo en ser procesados por el ordenador.

Empezamos con el comando simple de **ncal**, que despliega el calendario del mes actual:

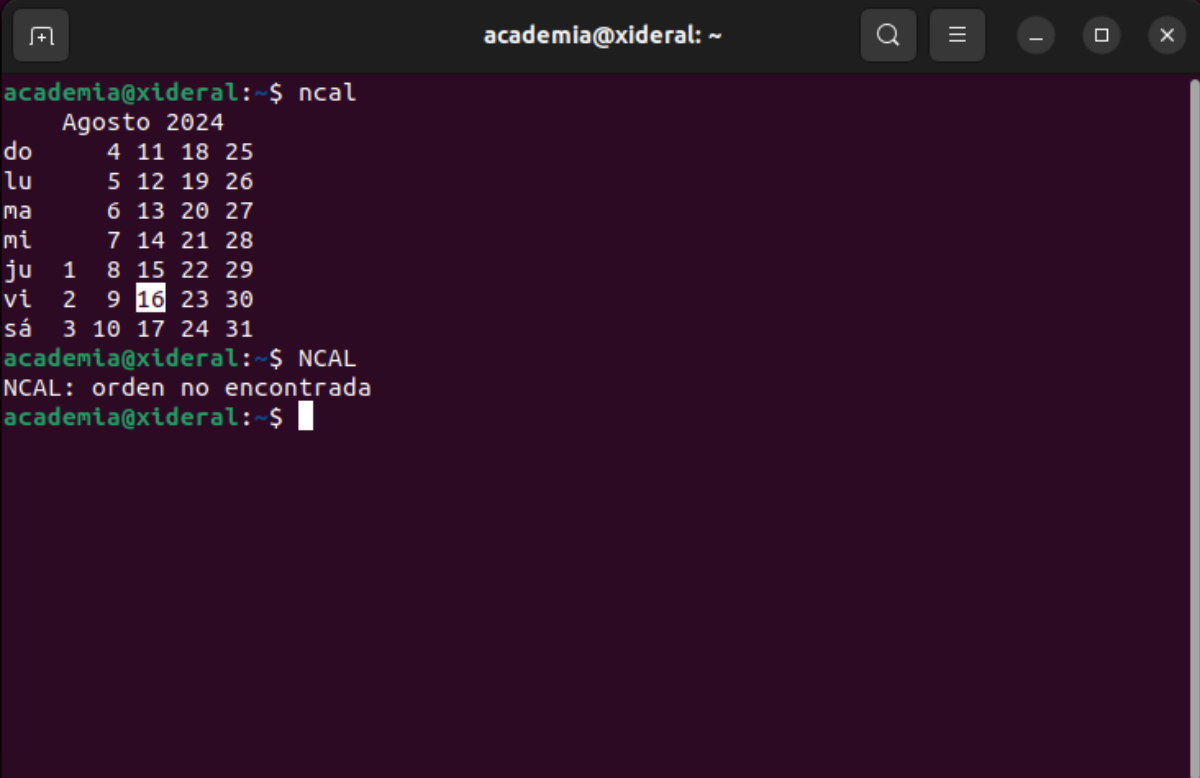


```
academia@xideral: ~  
academia@xideral:~$ ncal  
    Agosto 2024  
do      4 11 18 25  
lu      5 12 19 26  
ma      6 13 20 27  
mi      7 14 21 28  
ju  1  8 15 22 29  
vi  2  9 16 23 30  
sá  3 10 17 24 31  
academia@xideral:~$
```

The screenshot shows a terminal window with a dark background. The title bar at the top reads 'academia@xideral: ~'. The prompt 'academia@xideral:~\$' is followed by the command 'ncal'. The output displays a calendar for August 2024. The days of the week are abbreviated as 'do', 'lu', 'ma', 'mi', 'ju', 'vi', and 'sá'. The dates are arranged in a grid, with the 16th of August highlighted in a white box. The prompt 'academia@xideral:~\$' appears again at the bottom of the output.

Mediante las teclas dirección podemos realizar algunas cosas interesantes dentro de la terminal. Con las teclas de flecha ARRIBA y ABAJO podemos desplazarnos por los comandos anteriores que se aplicaron en la sesión actual, mientras que IZQUIERDA y DERECHA nos permiten movernos por la entrada actual y realizar los cambios necesarios en caso de error, lo que nos puede ahorrar tiempo, sobretodo con comandos extensos. Otra tecla útil es la TABULACIÓN que autocompleta partes del comando, también ayudando significativamente en la ejecución de comandos largos.

Cabe mencionar que los comandos **son sensibles a las mayúsculas y minúsculas**, lo que significa que se **debe escribir los comandos de forma exacta**, de lo contrario se interpretará como un comando diferente. Al intentar ejecutar **ncal** como **NCAL** produce el siguiente resultado erróneo:



```
academia@xideral: ~  
academia@xideral:~$ ncal  
    Agosto 2024  
do      4 11 18 25  
lu      5 12 19 26  
ma      6 13 20 27  
mi      7 14 21 28  
ju  1   8 15 22 29  
vi  2   9 16 23 30  
sá  3  10 17 24 31  
academia@xideral:~$ NCAL  
NCAL: orden no encontrada  
academia@xideral:~$
```

Podemos sacarle más jugo a los comandos con el uso de **argumentos** y **-opciones** que modifican el comportamiento del comando proporcionando información adicional sobre la que el comando puede actuar, y modificar las acciones de la misma, respectivamente.

Por ejemplo, podemos pasar diferentes cadenas de texto como argumentos al comando **echo** para imprimirlos de forma seguida:

A terminal window with a dark purple background. The title bar at the top reads 'academia@xideral: ~'. On the left of the title bar is a window icon, and on the right are search, menu, and window control buttons. The terminal content shows the prompt 'academia@xideral:~\$' followed by the command 'echo Hola Mundo!'. The output 'Hola Mundo!' is displayed on the next line. The prompt 'academia@xideral:~\$' appears again on the third line.

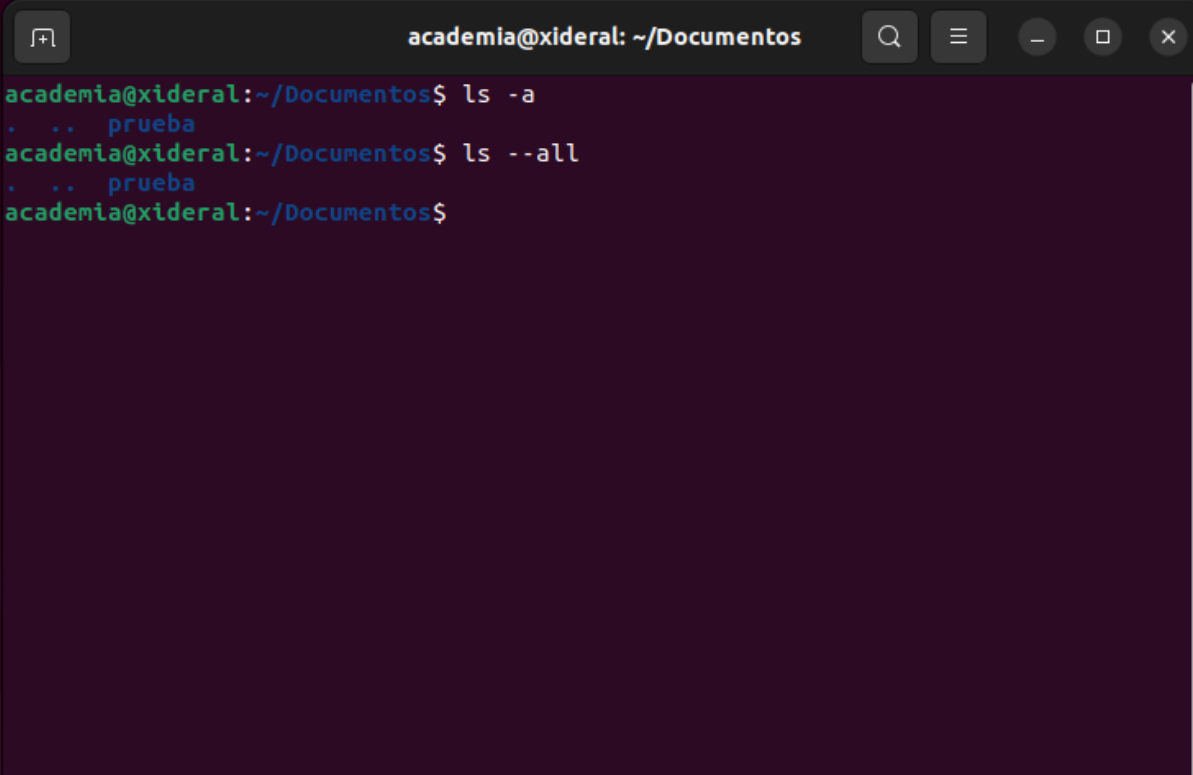
```
academia@xideral:~$ echo Hola Mundo!  
Hola Mundo!  
academia@xideral:~$
```

O podemos proporcionar una opción a **ncal** como **-3** que regresa el mes anterior, el mes actual y el mes siguiente en el calendario:


```
academia@xideral: ~  
academia@xideral:~$ ncal -3  
      Julio 2024      Agosto 2024      Septiembre 2024  
do       7 14 21 28      4 11 18 25      1  8 15 22 29  
lu    1  8 15 22 29      5 12 19 26      2  9 16 23 30  
ma    2  9 16 23 30      6 13 20 27      3 10 17 24  
mi    3 10 17 24 31      7 14 21 28      4 11 18 25  
ju    4 11 18 25      1  8 15 22 29      5 12 19 26  
vi    5 12 19 26      2  9 16 23 30      6 13 20 27  
sá    6 13 20 27      3 10 17 24 31      7 14 21 28  
academia@xideral:~$
```

En el caso de las opciones, normalmente constan de un solo carácter, pero pueden hacer que los comandos parezcan confusos, por lo que algunas de ellas ofrecen variantes de forma larga que logran el mismo resultado, pero hacen que el prompt sea más claro y fácil de entender.

Un ejemplo es el comando **ls** que despliega los archivos y directorios en la ubicación actual. La opción **-a** ofrece la variante **--all** que producen el mismo resultado de desplegar todos los archivos:



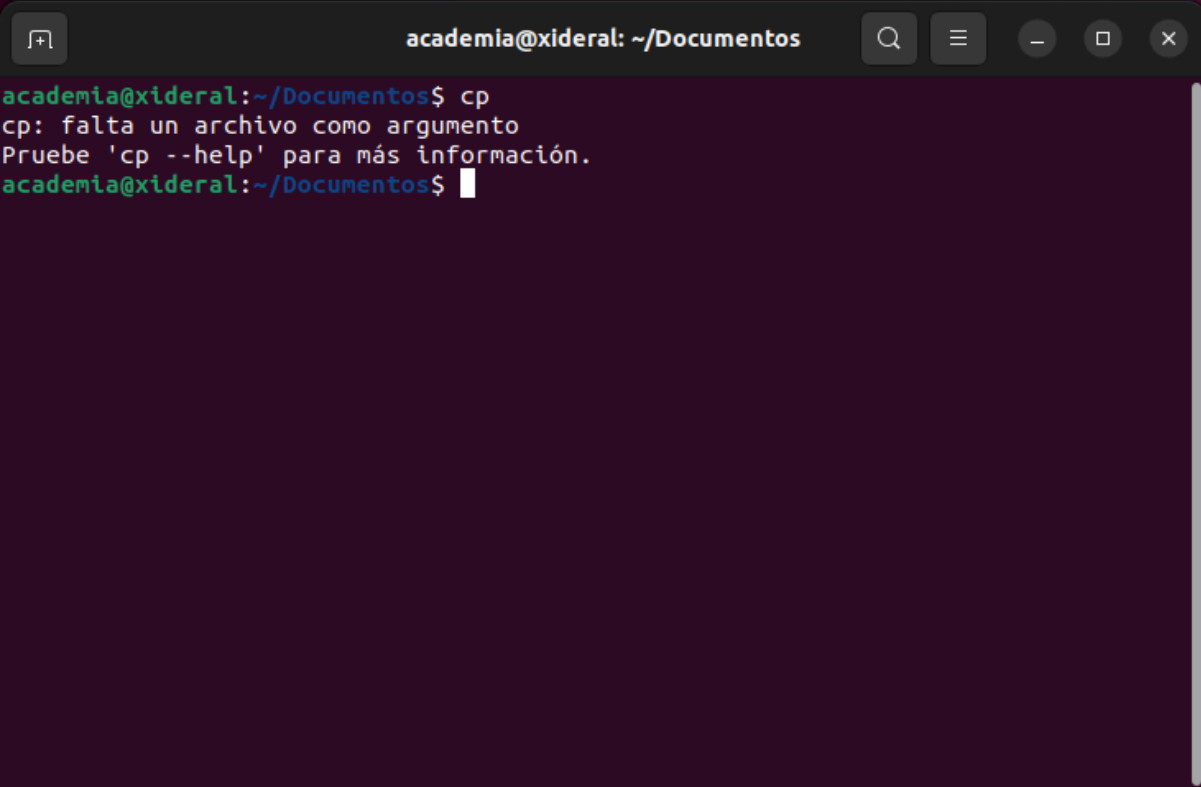
```
academia@xideral: ~/Documentos
academia@xideral:~/Documentos$ ls -a
.  ..  prueba
academia@xideral:~/Documentos$ ls --all
.  ..  prueba
academia@xideral:~/Documentos$
```

Algunas opciones pueden requerir valores adicionales llamados **parámetros** para funcionar. Al igual que con los propios comandos, las opciones son sensibles a las mayúsculas y minúsculas.

Regresando al comando **ncal**, ofrece las opciones -A y -B que regresa los meses siguientes y los meses anteriores hasta la fecha, respectivamente. Ambas opciones requieren un número para especificar cuantos meses se deben desplegar. En este caso, un valor de 3 para ambas opciones regresaron tres meses:

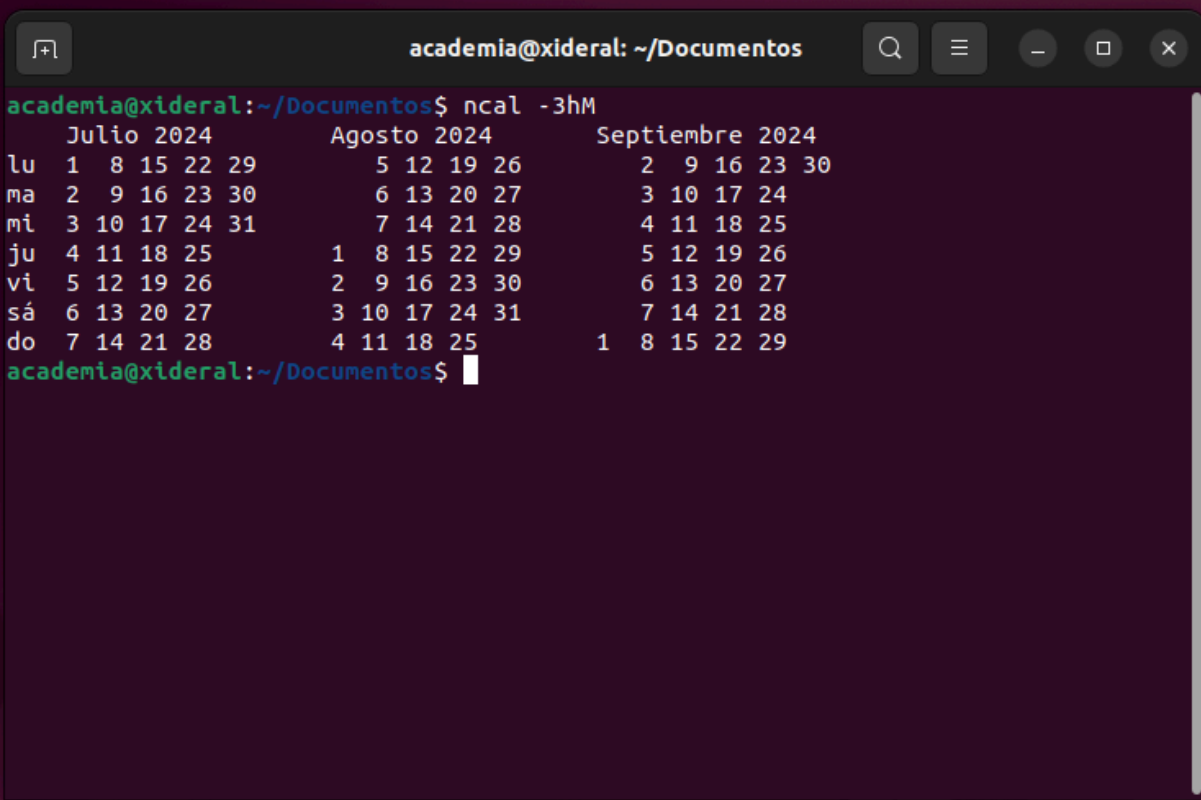
```
academia@xideral: ~/Documentos
academia@xideral:~/Documentos$ ncal -A3
    Agosto 2024      Septiembre 2024      Octubre 2024      Noviembre 2024
do      4 11 18 25      1  8 15 22 29      6 13 20 27      3 10 17 24
lu      5 12 19 26      2  9 16 23 30      7 14 21 28      4 11 18 25
ma      6 13 20 27      3 10 17 24      1  8 15 22 29      5 12 19 26
mi      7 14 21 28      4 11 18 25      2  9 16 23 30      6 13 20 27
ju  1  8 15 22 29      5 12 19 26      3 10 17 24 31      7 14 21 28
vi  2  9 16 23 30      6 13 20 27      4 11 18 25      1  8 15 22 29
sá  3 10 17 24 31      7 14 21 28      5 12 19 26      2  9 16 23 30
academia@xideral:~/Documentos$ ncal -B3
                                2024
    Mayo      Junio      Julio      Agosto
do      5 12 19 26      2  9 16 23 30      7 14 21 28      4 11 18 25
lu      6 13 20 27      3 10 17 24      1  8 15 22 29      5 12 19 26
ma      7 14 21 28      4 11 18 25      2  9 16 23 30      6 13 20 27
mi  1  8 15 22 29      5 12 19 26      3 10 17 24 31      7 14 21 28
ju  2  9 16 23 30      6 13 20 27      4 11 18 25      1  8 15 22 29
vi  3 10 17 24 31      7 14 21 28      5 12 19 26      2  9 16 23 30
sá  4 11 18 25      1  8 15 22 29      6 13 20 27      3 10 17 24 31
academia@xideral:~/Documentos$
```

Los argumentos son simplemente cadenas alfanuméricas que pueden utilizarse como entrada o salida del comando. Sin embargo, estos comandos podrían validar si el argumento sigue el formato requerido para que se ejecute correctamente, de lo contrario podría devolver un error. Los comandos pueden pedir al usuario que pase números, nombres de archivos, rutas, etc., para ejecutarse correctamente. Por ejemplo, ejecutando el comando **cp** sin argumentos marca un error:

A terminal window with a dark purple background. The title bar at the top reads 'academia@xideral: ~/Documentos'. The terminal shows a command prompt 'academia@xideral:~/Documentos\$' followed by the command 'cp'. The next line shows an error message: 'cp: falta un archivo como argumento' and a suggestion: 'Pruebe 'cp --help' para más información.' The prompt returns to 'academia@xideral:~/Documentos\$' with a cursor. The window has standard Linux window controls (minimize, maximize, close) and a search icon on the right side of the title bar.

```
academia@xideral:~/Documentos$ cp
cp: falta un archivo como argumento
Pruebe 'cp --help' para más información.
academia@xideral:~/Documentos$
```

Podemos combinar varios argumentos y opciones para indicar al shell las acciones deseadas y los datos que el ordenador debe procesar para conseguir el resultado deseado:



```
academia@xideral: ~/Documentos
academia@xideral:~/Documentos$ ncal -3hM
      Julio 2024      Agosto 2024      Septiembre 2024
lu  1  8 15 22 29      5 12 19 26      2  9 16 23 30
ma  2  9 16 23 30      6 13 20 27      3 10 17 24
mi  3 10 17 24 31      7 14 21 28      4 11 18 25
ju  4 11 18 25      1  8 15 22 29      5 12 19 26
vi  5 12 19 26      2  9 16 23 30      6 13 20 27
sá  6 13 20 27      3 10 17 24 31      7 14 21 28
do  7 14 21 28      4 11 18 25      1  8 15 22 29
academia@xideral:~/Documentos$
```

OBTENIENDO AYUDA

Con una amplia gama de comandos, cada uno con su propio conjunto diverso de opciones y argumentos, puede ser complicado recordar que es lo que hace que. Afortunadamente, el shell de Bash ofrece formas incorporadas para obtener referencias de lo que hacen los comandos, así como las opciones y argumentos disponibles.

Una de ellas es a través de un **manual** incorporado, que se puede invocar con el comando **man** junto con el nombre del comando que se quiera conocer.

```
academia@xideral: ~/Documents$ man echo
```

El usuario es presentado una página de documentación sobre el comando, en el que se incluye una breve explicación sobre lo que hace, una sinopsis de su sintaxis y una descripción detallando las funcionalidades para cada una de las opciones y argumentos disponibles:

```
ECHO(1) User Commands ECHO(1)
NAME
    echo - display a line of text
SYNOPSIS
    echo [SHORT-OPTION]... [STRING]...
    echo LONG-OPTION
DESCRIPTION
    Echo the STRING(s) to standard output.
    -n    do not output the trailing newline
    -e    enable interpretation of backslash escapes
    -E    disable interpretation of backslash escapes (default)
    --help display this help and exit
    --version
           output version information and exit
Manual page echo(1) line 1 (press h for help or q to quit)
```

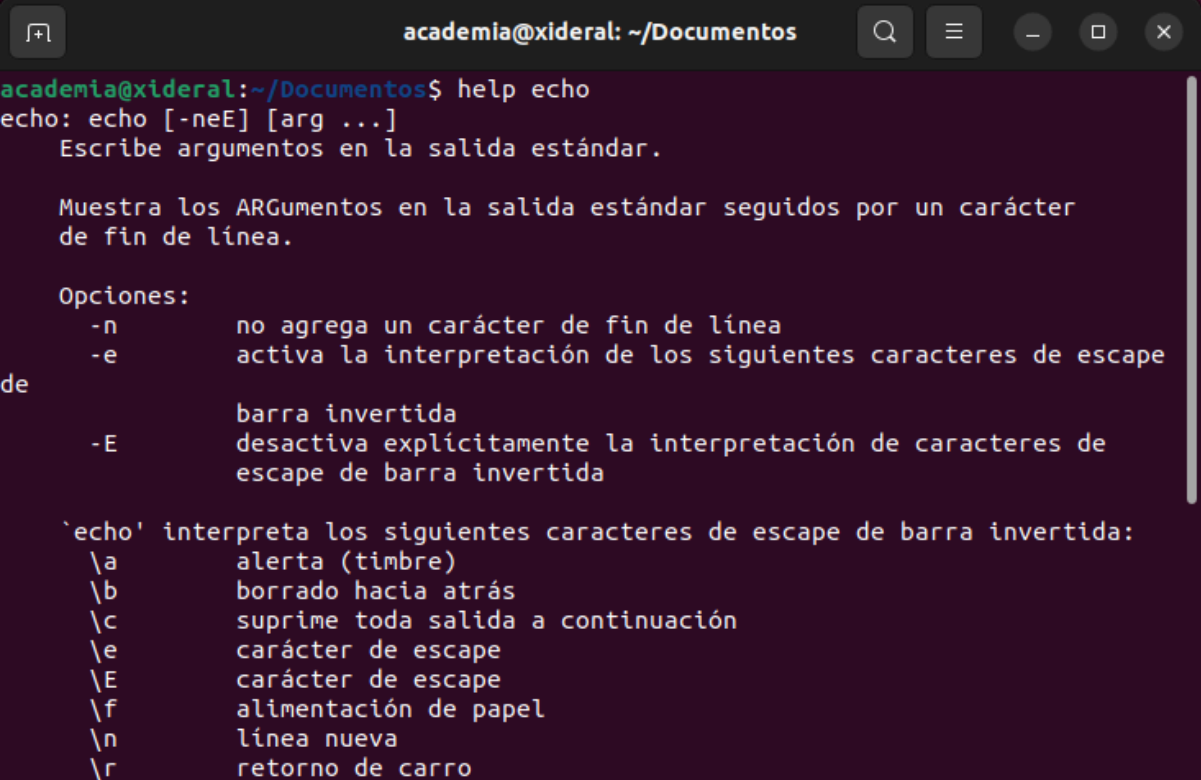
La sinopsis del documento hace conocer los valores requeridos y opcionales del comando, que se presentan sin corchetes y con corchetes, respectivamente. Aquellas opciones que no utilizan algún parámetro se agrupan como uno solo, mientras que las opciones que sí requieren un parámetro son separadas:

```
ncal [-3bhjJpwySM] [-A number] [-B number] [-W number] [-s country_code]
    [[month] year]
```

Algunos de los valores pueden tener puntos suspensivos (...) que ayudan a indicar que más de dos valores pueden ser aceptadas como parte de la opción o argumento.

```
echo [SHORT-OPTION]... [STRING]...
```

Los comandos incorporados tienen sus páginas de documentación incluidas, pero los comandos personalizados o las utilidades descargadas normalmente no tienen una página dedicada. En este caso, se recomienda hacer uso del comando **help** para obtener más información.



```
academia@xideral: ~/Documentos
academia@xideral:~/Documentos$ help echo
echo: echo [-neE] [arg ...]
    Escribe argumentos en la salida estándar.

    Muestra los ARGUMENTOS en la salida estándar seguidos por un carácter
    de fin de línea.

    Opciones:
      -n      no agrega un carácter de fin de línea
      -e      activa la interpretación de los siguientes caracteres de escape
de
      barra invertida
      -E      desactiva explícitamente la interpretación de caracteres de
escape de barra invertida

'echo' interpreta los siguientes caracteres de escape de barra invertida:
  \a      alerta (timbre)
  \b      borrado hacia atrás
  \c      suprime toda salida a continuación
  \e      carácter de escape
  \E      carácter de escape
  \f      alimentación de papel
  \n      línea nueva
  \r      retorno de carro
```

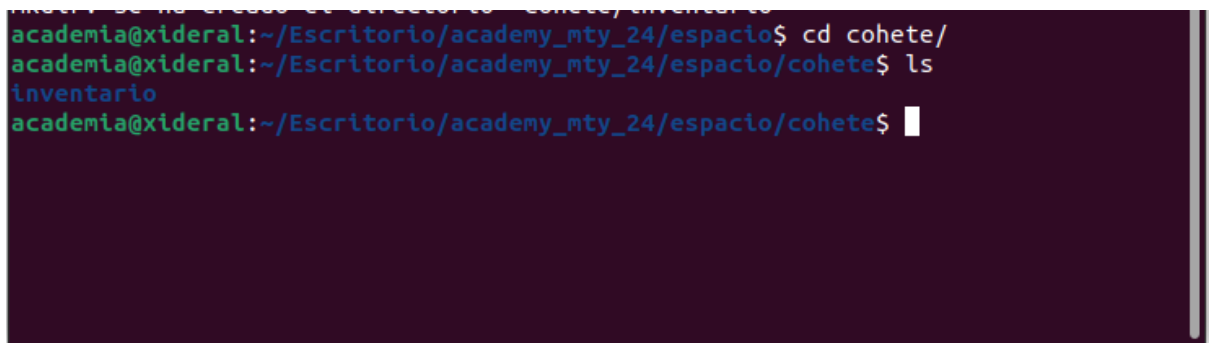
NAVEGACIÓN Y CREACIÓN DE ARCHIVOS / DIRECTORIOS

Aprender a navegar a través de los directorios, así como la creación y modificación de archivos a través de la línea de comandos es una habilidad

esencial para los usuarios tanto principiantes como expertos del sistema, ya que puede haber ocasiones en los que una interfaz gráfica no se encuentre disponible en el momento, así como el acceso a archivos relevantes y ocultos para la administración adecuada del sistema. Aprender a navegar a través del sistema mediante la línea de comandos permite a los usuarios aprovechar el potencial de los sistemas Linux y trabajar con mayor eficacia.

A continuación, se presenta una serie de comandos esenciales para la navegación por directorios, así como la creación y modificación de los archivos:

- **cd DIRECTORIO:** Permite navegar desde el directorio actual hasta el directorio introducido. El comando sin argumentos te regresa al directorio “home”, mientras que el argumento con doble puntos “..” te regresa al directorio anterior



```
academia@xideral:~/Escritorio/academy_mty_24/espacio$ cd cohete/  
academia@xideral:~/Escritorio/academy_mty_24/espacio/cohete$ ls  
inventario  
academia@xideral:~/Escritorio/academy_mty_24/espacio/cohete$
```

- **mkdir DIRECTORIO:** Genera un nuevo directorio en la ubicación actual. Al introducir múltiples argumentos se pueden generar varios directorios a la vez. También se puede especificar la opción -p para generar un subdirectorio dentro de un directorio padre, que también lo genera en dado caso que no exista en primer lugar


```
academia@xideral: ~/Escritorio/academy_mty_24
academia@xideral:~/Escritorio/academy_mty_24$ mkdir espacio
academia@xideral:~/Escritorio/academy_mty_24$ ls
espacio
academia@xideral:~/Escritorio/academy_mty_24$
```

```
academia@xideral:~/Escritorio/academy_mty_24/espacio$ mkdir -pv cohete/inventario
mkdir: se ha creado el directorio 'cohete'
mkdir: se ha creado el directorio 'cohete/inventario'
academia@xideral:~/Escritorio/academy_mty_24/espacio$ cd cohete/
academia@xideral:~/Escritorio/academy_mty_24/espacio/cohete$ ls
inventario
```

- **touch NOMBRE:** Es utilizada para la creación de archivos vacíos dentro del directorio actual. Pasando varios argumentos permite la creación de varios archivos a la vez:

```
academia@xideral: ~/Escritorio/academy_mty_24/espacio/gal...  
academia@xideral:~/Escritorio/academy_mty_24/espacio/galaxia/planetas$ touch tie  
rra.txt marte.txt jupiter.txt neptuno.txt  
academia@xideral:~/Escritorio/academy_mty_24/espacio/galaxia/planetas$ ls  
jupiter.txt marte.txt neptuno.txt tierra.txt  
academia@xideral:~/Escritorio/academy_mty_24/espacio/galaxia/planetas$
```

- **file ARCHIVO:** Despliega información sobre un archivo en particular, como el tipo de formato:

```
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/imagenes$ file tie  
rra.png  
tierra.png: empty  
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/imagenes$
```

- **cp ORIGEN DESTINO:** Copia el archivo ORIGEN hasta la ubicación DESTINO.

```
academia@xideral: ~/Escritorio/academy_mty_24/espacio/rec...  
academia@xideral:~/Escritorio/academy_mty_24/espacio$ cp galaxia/planetas/tierra  
.txt recursos/imagenes/  
  
academia@xideral:~/Escritorio/academy_mty_24/espacio$ cd recursos/imagenes/ && l  
s  
tierra.png tierra.txt  
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/imagenes$
```

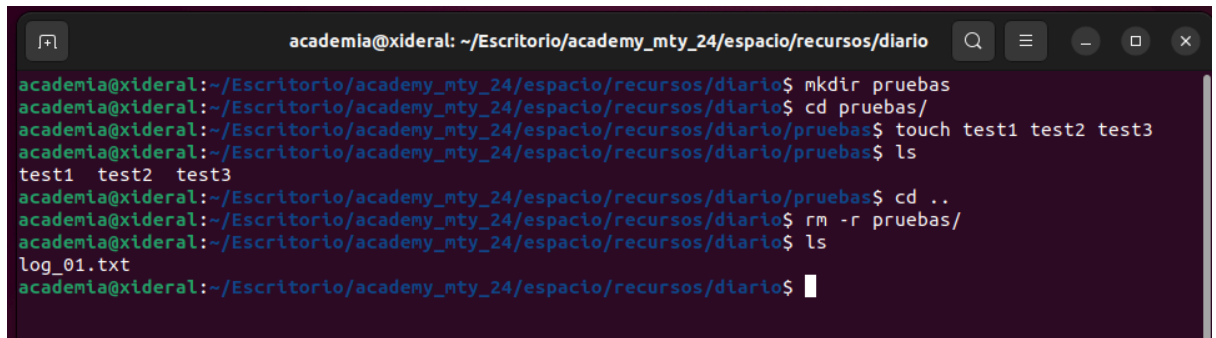
- **mv ORIGEN DESTINO:** Permite mover el archivo ORIGEN hasta el directorio DESTINO. También puede ser utilizado para renombrar archivos, pasando como argumento el nombre original como ORIGEN, y el nuevo nombre como DESTINO:

```
academia@xideral: ~/Escritorio/academy_mty_24/espacio/rec...
academia@xideral:~/Escritorio/academy_mty_24/espacio$ mv recursos/diario/log_01.txt cohete/inventario/
academia@xideral:~/Escritorio/academy_mty_24/espacio$ cd cohete/inventario/ && ls
comida.txt dispositivos.txt gas.txt log_01.txt medicinas.txt
academia@xideral:~/Escritorio/academy_mty_24/espacio/cohete/inventario$ cd ../../
/
academia@xideral:~/Escritorio/academy_mty_24/espacio$ cd recursos/diario/ && ls
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario$
```

```
academia@xideral: ~/Escritorio/academy_mty_24/espacio/co...
academia@xideral:~/Escritorio/academy_mty_24/espacio$ cd cohete/
academia@xideral:~/Escritorio/academy_mty_24/espacio/cohete$ cd inventario/
academia@xideral:~/Escritorio/academy_mty_24/espacio/cohete/inventario$ ls
comida.txt dispositivos.txt gas.txt log_01.txt medicinas.txt
academia@xideral:~/Escritorio/academy_mty_24/espacio/cohete/inventario$ mv gas.txt consumo_gas.txt
academia@xideral:~/Escritorio/academy_mty_24/espacio/cohete/inventario$ ls
comida.txt consumo_gas.txt dispositivos.txt log_01.txt medicinas.txt
academia@xideral:~/Escritorio/academy_mty_24/espacio/cohete/inventario$
```

- **rm ARCHIVO:** Comando para eliminar el ARCHIVO. Si se introduce la opción -r, se puede eliminar de forma recursiva los archivos de un directorio, así como el propio directorio:

```
academia@xideral: ~/Escritorio/academy_mty_24/espacio/recursos/diario
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario$ rm log_01.txt
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario$ ls
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario$ ls -a
.
..
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario$
```



```
academia@xideral: ~/Escritorio/academy_mty_24/espacio/recursos/diario
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario$ mkdir pruebas
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario$ cd pruebas/
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario/pruebas$ touch test1 test2 test3
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario/pruebas$ ls
test1  test2  test3
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario/pruebas$ cd ..
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario$ rm -r pruebas/
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario$ ls
log_01.txt
academia@xideral:~/Escritorio/academy_mty_24/espacio/recursos/diario$
```