



BASH
THE BOURNE-AGAIN SHELL

Directorios de Linux

¿Qué son los comandos?

Un comando es una instrucción que le damos a nuestra computadora para que haga lo que queramos. En Mac OS y Linux se llama terminal, mientras que en Windows se llama símbolo del sistema. Los comandos siempre distinguen entre mayúsculas y minúsculas.

Los comandos se ejecutan escribiendo en la línea de comando y luego presionando la tecla Intro. Este comando pasa además al shell que lee el comando y lo ejecuta. Shell es un método para que el usuario interactúe con el sistema. El shell predeterminado en Linux se llama bash (Bourne-Again Shell).

Hay dos tipos de comandos de shell:

- **Comandos de shell incorporados:** Son parte de un shell. Cada shell tiene algunos comandos integrados.
- **Comandos externos/Linux:** Cada comando externo es un programa ejecutable separado escrito en C u otros lenguajes de programación.

Comandos de directorio de Linux

Comando de directorio	Descripción
<u>pwd</u>	El comando pwd significa (directorio de trabajo de impresión). Muestra la ubicación de trabajo actual o el directorio del usuario. Muestra toda la ruta de trabajo comenzando con /. Es un comando incorporado.
<u>ls</u>	El comando ls se usa para mostrar la lista de una carpeta. Enumerará todos los archivos en la carpeta dirigida.
<u>cd</u>	El comando cd significa (cambiar de directorio). Se utiliza para cambiar al directorio en el que desea trabajar desde el directorio actual.
<u>mkdir</u>	Con el comando mkdir puede crear su propio directorio.
<u>rmdir</u>	El comando rmdir se usa para eliminar un directorio de su sistema.

Comando pwd de Linux

El comando pwd (imprimir directorio de trabajo) de Linux muestra su ubicación en la que está trabajando actualmente. Dará la ruta completa desde la raíz hasta el directorio.

Sintaxis:

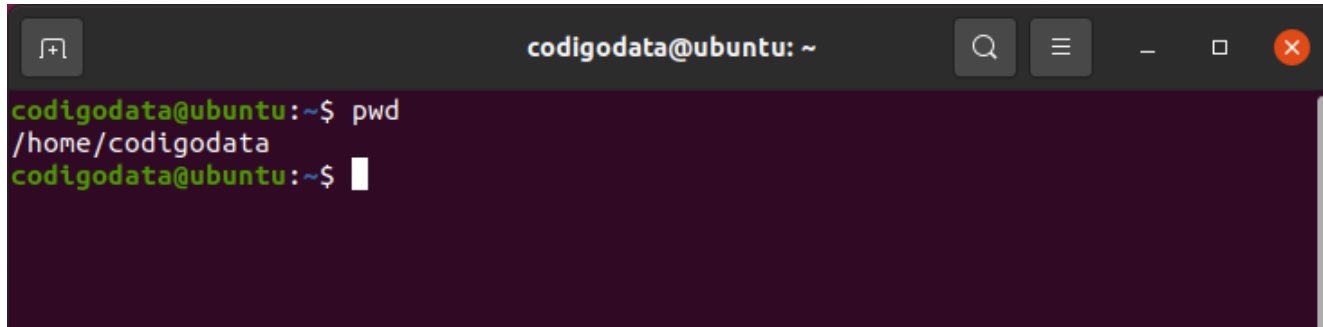
pwd

Ejemplo:

Veamos un ejemplo de comando pwd.

Abra su terminal y escriba pwd, presione la tecla enter. Puede ver la ruta de su directorio. Aquí, mi ruta es /home/sssit y mi ubicación actual es sssit.

Observe aquí que esa ubicación se mostrará desde la raíz o desde el sistema de archivos.



```
codigodata@ubuntu: ~  
codigodata@ubuntu:~$ pwd  
/home/codigodata  
codigodata@ubuntu:~$
```

Directorio de inicio de Linux

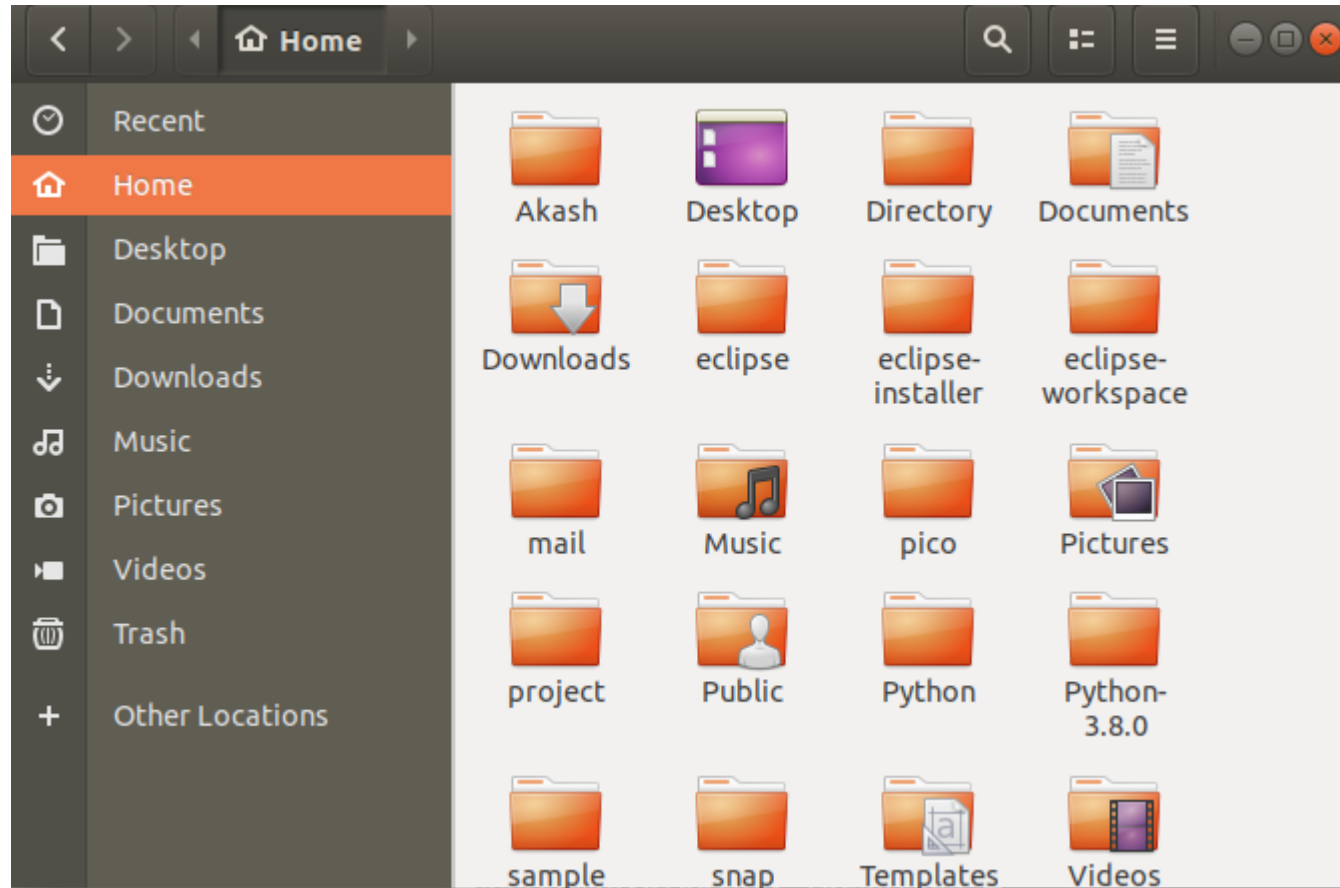
El directorio de inicio de Linux es un directorio para un usuario particular del sistema y consta de archivos individuales. También se le conoce como el **directorío de inicio de sesión** . Este es el primer lugar que ocurre después de iniciar sesión en un sistema Linux. Se crea automáticamente como **"/home"** para cada usuario en el directorio. Es un subdirectorío estándar del directorío raíz. El directorío raíz contiene todos los demás directoríos, subdirectoríos y archivos del sistema. Se denota con una barra diagonal (/).

El directorío de inicio se puede decir como un espacio de trabajo personal para todos los usuarios, excepto para root. Hay un directorío separado para cada usuario. Por ejemplo, dos usuarios 'jtp1' y 'jtp2' tendrán directoríos como "/home/jtp1" y "/home/jtp2". Estos usuarios tendrán todos los derechos en sus archivos de directorío.

El usuario raíz (administrativo) es el único usuario que tiene su directorío de inicio en una ubicación diferente de manera predeterminada. La ruta del usuario root es '/root/', donde tiene control sobre todos los directoríos y archivos.

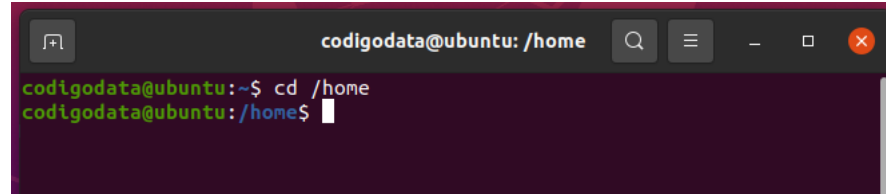
¿Cómo encontrar el directorio de inicio?

Hay varias formas de acceder y volver al directorio de inicio. Algunos comandos son muy útiles para los directorios, como [cd](#) , [pwd](#) , [mkdir](#) , [pwd](#) , [ls](#) y [rmdir](#) . Para acceder al directorio de inicio de forma gráfica, abra la aplicación de **archivos** y haga clic en la opción **Inicio** en el menú del lado izquierdo. Considere la siguiente imagen:



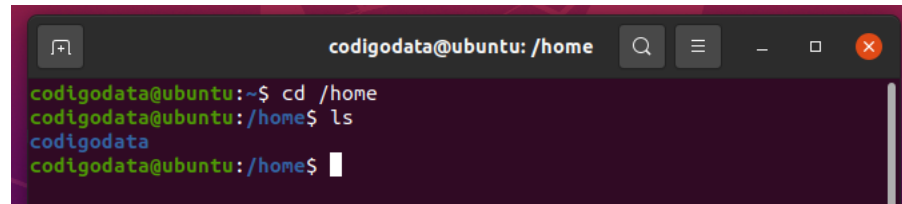
Aquí, podemos explorar nuestro directorio de inicio.

Generalmente, nuestra terminal se abre con el directorio particular del usuario. Para cambiar el directorio al directorio de inicio, ejecute el **comando cd** de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: /home'. The prompt is 'codigodata@ubuntu:~\$'. The user enters 'cd /home'. The prompt changes to 'codigodata@ubuntu:/home\$' with a cursor at the end.

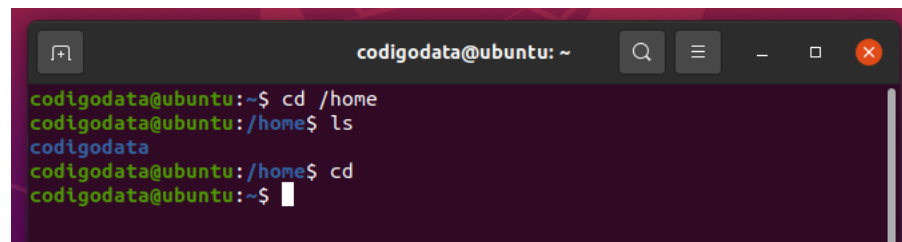
```
codigodata@ubuntu:~$ cd /home
codigodata@ubuntu:/home$
```

El comando anterior cambiará el directorio a casa. Para enumerar el directorio de inicio, ejecute el comando **ls** de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: /home'. The prompt is 'codigodata@ubuntu:/home\$'. The user enters 'ls'. The output is 'codigodata'. The prompt returns to 'codigodata@ubuntu:/home\$' with a cursor at the end.

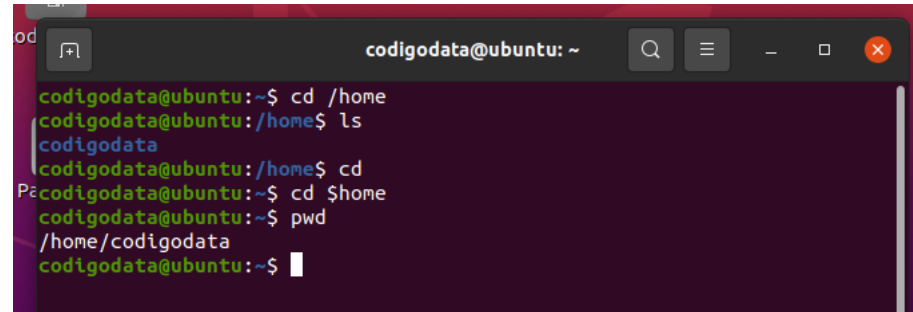
```
codigodata@ubuntu:/home$ ls
codigodata
codigodata@ubuntu:/home$
```

Podemos volver a nuestro directorio de inicio ejecutando el comando **cd** sin ningún argumento. Volverá a nuestro directorio de inicio desde cualquier directorio en el que estemos trabajando. Ejecutarlo de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: ~'. The prompt is 'codigodata@ubuntu:/home\$'. The user enters 'cd'. The prompt changes back to 'codigodata@ubuntu:~\$' with a cursor at the end.

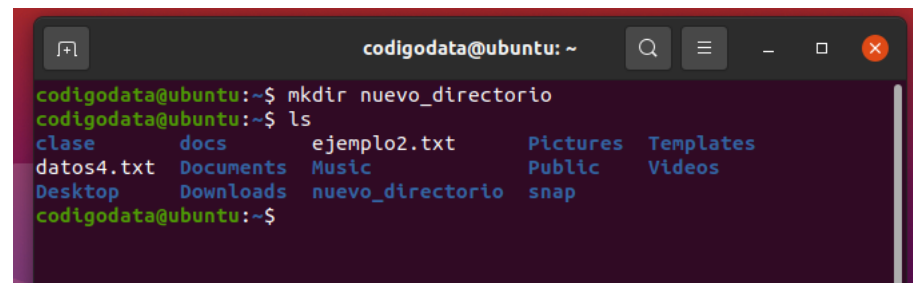
```
codigodata@ubuntu:/home$ cd
codigodata@ubuntu:~$
```

De la salida anterior, estábamos en /Directorio/archivos, al ejecutar solo el comando `cd`, llegamos a nuestro directorio de inicio. También podemos usar los comandos "`cd ~` o `cd $HOME`" para hacer lo mismo. Considere los siguientes comandos:

A terminal window titled 'codigodata@ubuntu: ~' with search, menu, and window control icons. It shows a sequence of commands: 'cd /home', 'ls', 'cd', 'cd \$home', and 'pwd', resulting in the current directory being '/home/codigodata'.

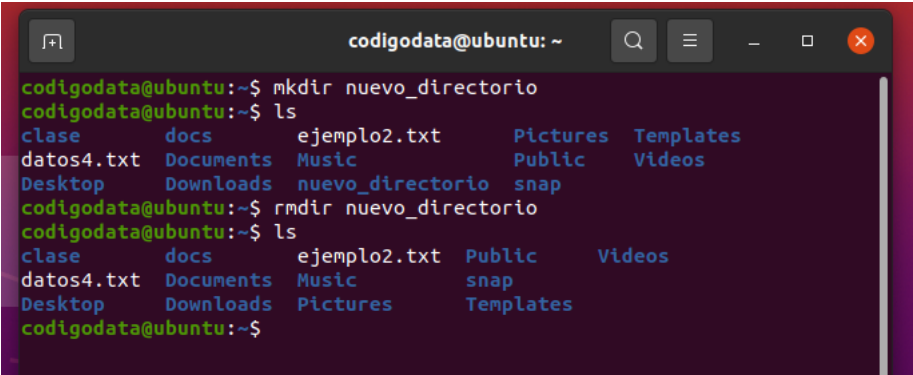
```
codigodata@ubuntu:~$ cd /home
codigodata@ubuntu:/home$ ls
codigodata
codigodata@ubuntu:/home$ cd
codigodata@ubuntu:~$ cd $home
codigodata@ubuntu:~$ pwd
/home/codigodata
codigodata@ubuntu:~$
```

Para crear un directorio bajo un directorio, ejecute el **comando `mkdir`** de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: ~' showing the command 'mkdir nuevo_directorio' being executed. A subsequent 'ls' command shows the new directory 'nuevo_directorio' in the output list.

```
codigodata@ubuntu:~$ mkdir nuevo_directorio
codigodata@ubuntu:~$ ls
clase      docs      ejemplo2.txt  Pictures  Templates
datos4.txt Documents  Music         Public    Videos
Desktop    Downloads nuevo_directorio snap
```


También podemos eliminar un directorio. Para eliminar un directorio, ejecute el comando `rmdir` de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: ~' with standard window controls. The terminal shows a sequence of commands: 'mkdir nuevo_directorio', 'ls' (listing files like clase, docs, ejemplo2.txt, etc.), 'rmdir nuevo_directorio', and another 'ls' (showing the directory removed).

```
codigodata@ubuntu:~$ mkdir nuevo_directorio
codigodata@ubuntu:~$ ls
clase      docs      ejemplo2.txt  Pictures  Templates
datos4.txt Documents Music        Public    Videos
Desktop    Downloads nuevo_directorio snap
codigodata@ubuntu:~$ rmdir nuevo_directorio
codigodata@ubuntu:~$ ls
clase      docs      ejemplo2.txt  Public    Videos
datos4.txt Documents Music        snap
Desktop    Downloads Pictures      Templates
codigodata@ubuntu:~$
```

Diferencia entre directorio raíz y de inicio

Algunas diferencias clave entre el directorio raíz y el de inicio son las siguientes:

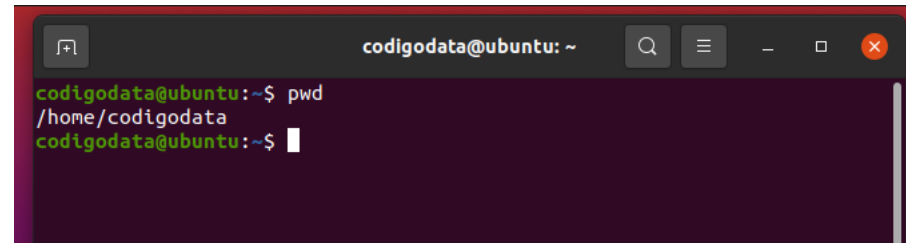
Directorio raíz	Directorio de inicio
El directorio raíz es el nivel superior de la unidad del sistema.	El directorio de inicio es un subdirectorio del directorio raíz.
Se denota con una barra inclinada '/'. El administrador tiene acceso para realizar cualquier cambio en los archivos y la configuración.	Se denota por '~' y tiene la ruta " /usuarios/nombre de usuario ". Ningún usuario que no sea el usuario raíz puede cambiar la configuración de todo el sistema.
El administrador puede crear un usuario.	Cualquier usuario que tenga un directorio de inicio no puede crear un usuario.
En el sistema de archivos de Linux, todo se encuentra en el directorio raíz.	El directorio de inicio contiene los datos de un usuario en particular.

Comando pwd de Linux

El comando pwd (imprimir directorio de trabajo) de Linux muestra su ubicación en la que está trabajando actualmente. Dará la ruta completa desde la raíz hasta el directorio.

Abra su terminal y escriba pwd, presione la tecla enter. Puede ver la ruta de su directorio. Aquí, mi ruta es /home/sssit y mi ubicación actual es sssit.

Observe aquí que esa ubicación se mostrará desde la raíz o desde el sistema de archivos.



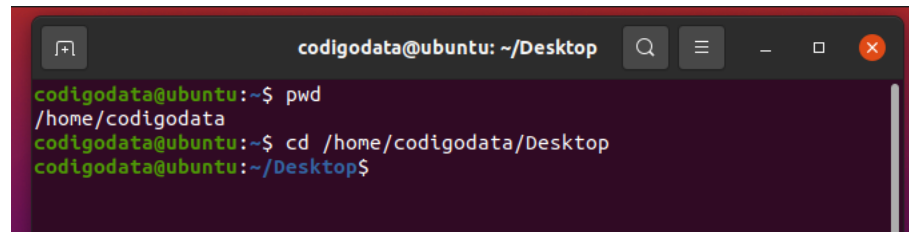
```
codigodata@ubuntu: ~  
codigodata@ubuntu:~$ pwd  
/home/codigodata  
codigodata@ubuntu:~$
```

Comando cd de Linux | Cambiar directorio de Linux

El **comando cd** de Linux se usa para cambiar el directorio de trabajo actual (es decir, en el que está trabajando el usuario actual). El "cd" significa '**cambiar directorio**'. Es uno de los comandos más utilizados en la terminal de Linux.

1) Cambiar del directorio actual a un nuevo directorio

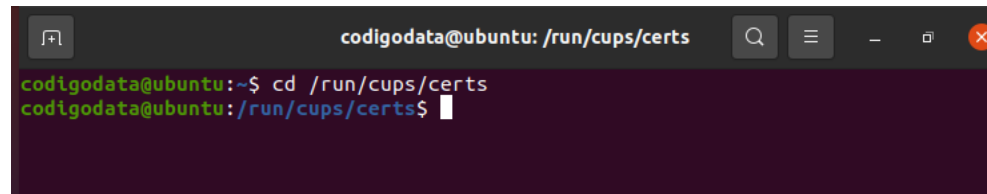
Podemos cambiar nuestro directorio del directorio de trabajo actual a un directorio específico. Para mostrar el directorio de trabajo actual, ejecute el comando de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: ~/Desktop'. The prompt is 'codigodata@ubuntu:~\$'. The user enters 'pwd', and the output is '/home/codigodata'. Then, the user enters 'cd /home/codigodata/Desktop', and the prompt changes to 'codigodata@ubuntu:~/Desktop\$'.

```
codigodata@ubuntu:~$ pwd
/home/codigodata
codigodata@ubuntu:~$ cd /home/codigodata/Desktop
codigodata@ubuntu:~/Desktop$
```

2) Cambiar directorio usando una ruta absoluta

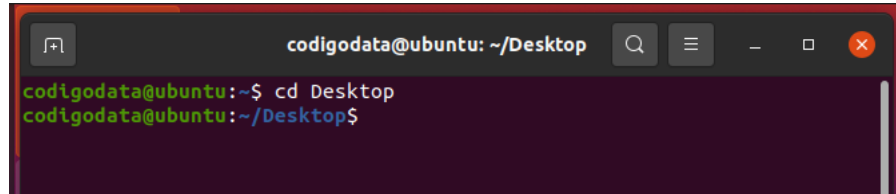
Para cambiar el directorio usando una ruta absoluta, debemos mencionar la ruta completa comenzando desde la raíz. Considere el siguiente ejemplo:

A terminal window titled 'codigodata@ubuntu: /run/cups/certs'. The prompt is 'codigodata@ubuntu:~\$'. The user enters 'cd /run/cups/certs', and the prompt changes to 'codigodata@ubuntu:/run/cups/certs\$' with a cursor.

```
codigodata@ubuntu:~$ cd /run/cups/certs
codigodata@ubuntu:/run/cups/certs$
```

3) Cambiar directorio usando una ruta relativa

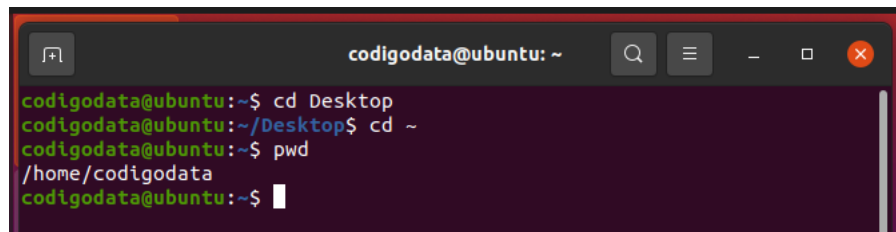
Podemos cambiar nuestro directorio usando una ruta relativa; una ruta relativa es una ubicación relativa al directorio actual. Considere el siguiente ejemplo:

A terminal window titled 'codigodata@ubuntu: ~/Desktop' showing a successful directory change. The user enters 'cd Desktop' and the prompt changes to '~/Desktop\$'.

```
codigodata@ubuntu: ~/Desktop
codigodata@ubuntu:~$ cd Desktop
codigodata@ubuntu:~/Desktop$
```

4) Cambiar al directorio de inicio

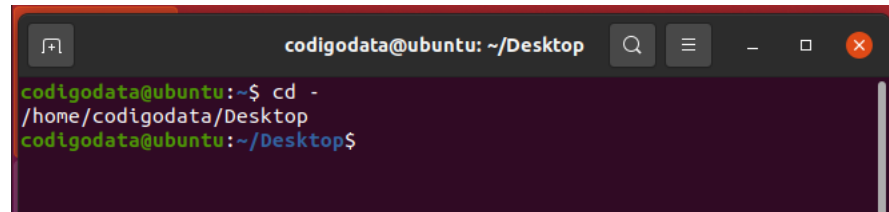
Para cambiar el directorio al directorio de inicio desde el directorio de trabajo actual, ejecute el comando de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: ~' showing a sequence of commands to return to the home directory. The user enters 'cd Desktop', then 'cd ~', and finally 'pwd', which outputs '/home/codigodata'.

```
codigodata@ubuntu: ~
codigodata@ubuntu:~$ cd Desktop
codigodata@ubuntu:~/Desktop$ cd ~
codigodata@ubuntu:~$ pwd
/home/codigodata
codigodata@ubuntu:~$
```

5) Cambiar al directorio anterior

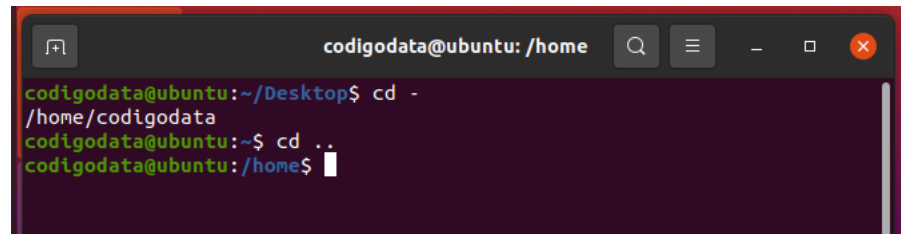
Para cambiar al directorio anterior desde el directorio de trabajo actual, ejecute el comando de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: ~/Desktop' with search, menu, and window control icons. The terminal shows the user running 'cd -' to move from the Desktop directory to the home directory. The prompt changes from '~/Desktop\$' to '~/home\$'.

```
codigodata@ubuntu: ~/Desktop
codigodata@ubuntu:~$ cd -
/home/codigodata/Desktop
codigodata@ubuntu:~/Desktop$
```

6) Cambiar al directorio principal

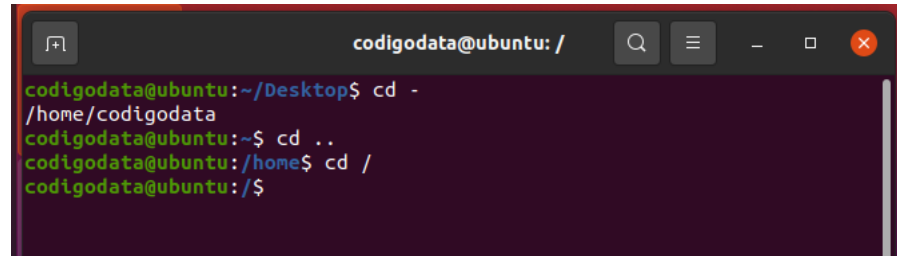
Para cambiar el directorio al directorio principal del directorio de trabajo actual, ejecute el comando de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: /home' with search, menu, and window control icons. The terminal shows the user running 'cd -' to move from the Desktop directory to the home directory, and then running 'cd ..' to move to the parent directory. The prompt changes from '~/Desktop\$' to '~/home\$'.

```
codigodata@ubuntu:~/Desktop$ cd -
/home/codigodata
codigodata@ubuntu:~$ cd ..
codigodata@ubuntu:/home$
```

7) Cambiar al directorio raíz

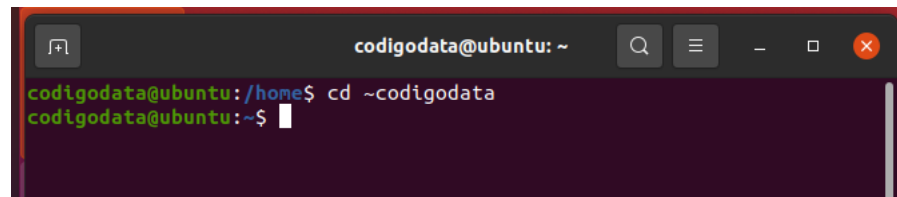
Para navegar el directorio al directorio raíz de todo el sistema desde el directorio de trabajo actual, ejecute el comando de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: /' with search, menu, and window control icons. The terminal shows a sequence of commands: 'cd -' from '~/Desktop', resulting in '/home/codigodata'; 'cd ..' from '~', resulting in '/home'; and 'cd /' from '/home', resulting in '/'.

```
codigodata@ubuntu:~/Desktop$ cd -  
/home/codigodata  
codigodata@ubuntu:~$ cd ..  
codigodata@ubuntu:/home$ cd /  
codigodata@ubuntu:/$
```

8) Cambiar al directorio de inicio de otro usuario

Podemos cambiar el directorio del directorio de trabajo actual al directorio de inicio de un usuario ejecutando el comando de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: ~' with search, menu, and window control icons. The terminal shows the command 'cd ~codigodata' being executed from the '/home' directory, resulting in the prompt '~\$'.

```
codigodata@ubuntu:/home$ cd ~codigodata  
codigodata@ubuntu:~$
```

Archivos Linux

En el sistema Linux, todo es un archivo y si no es un archivo, es un proceso. Un archivo no incluye solo archivos de texto, imágenes y programas compilados, sino que también incluye particiones, controladores de dispositivos de hardware y directorios. Linux considera todo como un archivo.

Los archivos siempre distinguen entre mayúsculas y minúsculas

Tipos de archivos:

1. Archivos regulares (-): Contiene programas, archivos ejecutables y archivos de texto.

2. Directorio de archivos (d): Se muestra en color azul. Contiene una lista de archivos.

3. Archivos especiales

- 1. Archivo de bloque (b)**
- 2. Archivo de dispositivo de caracteres (c)**
- 3. Archivo de canalización con nombre (p)**
- 4. Archivo de enlace simbólico (l)**
- 5. Archivo (s) de socket**

Sistema de archivos Linux

Un sistema de archivos de Linux es una colección estructurada de archivos en una unidad de disco o una partición. Una partición es un segmento de memoria y contiene algunos datos específicos. En nuestra máquina, puede haber varias particiones de la memoria. Generalmente, cada partición contiene un sistema de archivos.

El sistema informático de propósito general necesita almacenar datos sistemáticamente para que podamos acceder fácilmente a los archivos en menos tiempo. Almacena los datos en discos duros (HDD) o algún tipo de almacenamiento equivalente. Puede haber las siguientes razones para mantener el sistema de archivos:

- Principalmente, la computadora guarda datos en el almacenamiento de RAM; puede perder los datos si se apaga. Sin embargo, hay RAM no volátil (Flash RAM y SSD) que está disponible para mantener los datos después de la interrupción de la alimentación.
- Se prefiere el almacenamiento de datos en discos duros en comparación con la RAM estándar, ya que la RAM cuesta más que el espacio en disco. Los costes de los discos duros están bajando progresivamente en comparación con los de la memoria RAM.

El sistema de archivos de [Linux contiene las siguientes secciones:](#)

- El directorio raíz (/)
- Un formato de almacenamiento de datos específico (EXT3, EXT4, BTRFS, XFS, etc.)
- Una partición o volumen lógico que tiene un sistema de archivos particular.

¿Qué es el sistema de archivos de Linux?

El sistema de archivos de Linux es generalmente una capa integrada de un [sistema operativo Linux que se](#) utiliza para manejar la gestión de datos del almacenamiento. Ayuda a organizar el archivo en el almacenamiento del disco. Administra el nombre del archivo, el tamaño del archivo, la fecha de creación y mucha más información sobre un archivo.

Si tenemos un formato de archivo no compatible en nuestro sistema de archivos, podemos descargar un software para solucionarlo.

Estructura del sistema de archivos de Linux

El sistema de archivos de Linux tiene una estructura de archivos jerárquica, ya que contiene un directorio raíz y sus subdirectorios. Se puede acceder a todos los demás directorios desde el directorio raíz. Por lo general, una partición solo tiene un sistema de archivos, pero puede tener más de un sistema de archivos.

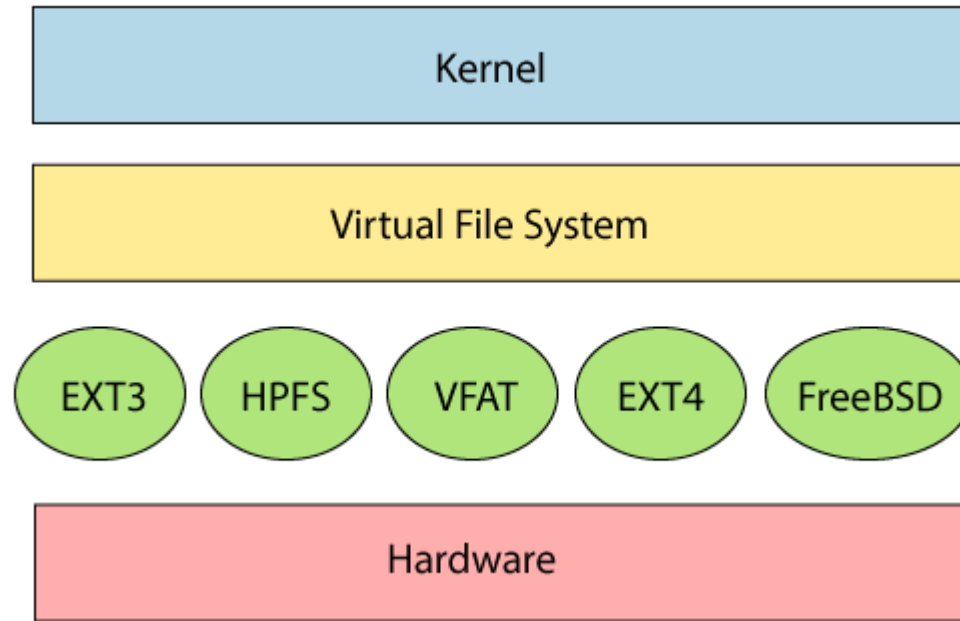
Un sistema de archivos está diseñado de manera que pueda administrar y proporcionar espacio para datos de almacenamiento no volátiles. Todos los sistemas de archivos requerían un espacio de nombres que es una metodología organizativa y de nombres. El espacio de nombres define el proceso de denominación, la longitud del nombre del archivo o un subconjunto de caracteres que se pueden usar para el nombre del archivo. También define la estructura lógica de los archivos en un segmento de memoria, como el uso de directorios para organizar archivos específicos. Una vez que se describe un espacio de nombres, se debe definir una descripción de metadatos para ese archivo en particular.

La estructura de datos debe admitir una estructura de directorio jerárquica; esta estructura se utiliza para describir el espacio en disco disponible y utilizado para un bloque en particular. También tiene otros detalles sobre los archivos, como el tamaño del archivo, la fecha y la hora de creación, la actualización y la última modificación.

Además, almacena información avanzada sobre la sección del disco, como particiones y volúmenes.

Los datos avanzados y las estructuras que representan contienen la información sobre el sistema de archivos almacenado en la unidad; es distinto e independiente de los metadatos del sistema de archivos.

El sistema de archivos de Linux contiene una arquitectura de implementación de software de sistema de archivos de dos partes. Considere la siguiente imagen:



El sistema de archivos requiere una API (interfaz de programación de aplicaciones) para acceder a las llamadas de función para interactuar con los componentes del sistema de archivos, como archivos y directorios. [La API](#)

facilita tareas como la creación, eliminación y copia de archivos. Facilita un algoritmo que define la disposición de los archivos en un sistema de archivos.

Las dos primeras partes del sistema de archivos dado juntas se denominan **sistema de archivos virtual de Linux** . Proporciona un único conjunto de comandos para que el kernel y los desarrolladores accedan al sistema de archivos. Este sistema de archivos virtual requiere el controlador de sistema específico para brindar una interfaz al sistema de archivos.

Características del sistema de archivos de Linux

En Linux, el sistema de archivos crea una estructura de árbol. Todos los archivos están dispuestos como un árbol y sus ramas. El directorio superior llamado directorio **raíz (/)** . Se puede acceder a todos los demás directorios en Linux desde el directorio raíz.

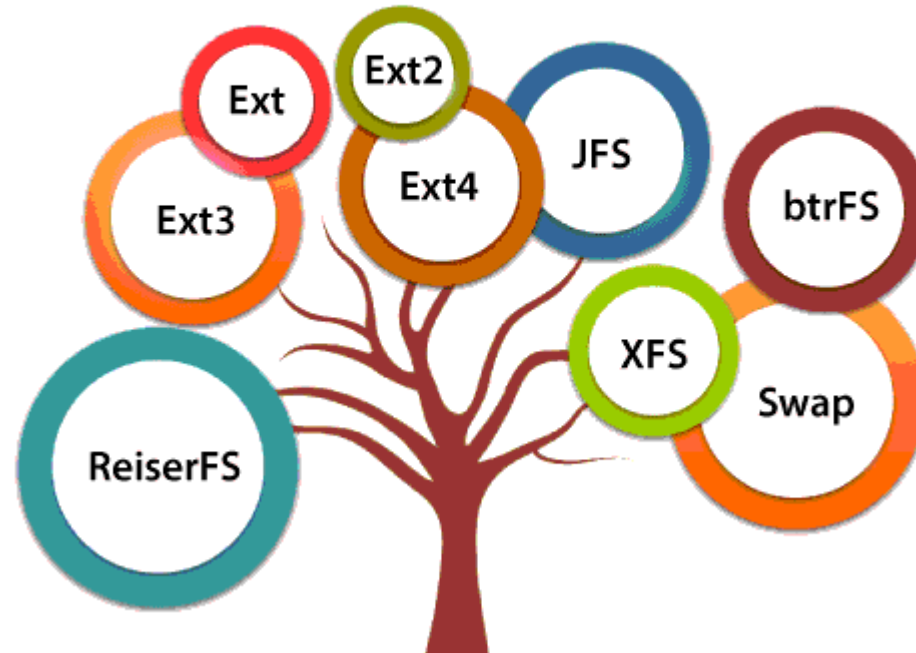
Algunas [características clave del sistema de archivos de Linux](#) son las siguientes:

- Especificación de rutas:** Linux no usa la barra invertida (\) para separar los componentes; utiliza una barra diagonal (/) como alternativa. Por ejemplo, al igual que en Windows, los datos pueden almacenarse en C:\Mis documentos\Trabajo, mientras que, en Linux, se almacenarían en /home/Mi documento/Trabajo.
- Partición, directorios y unidades:** Linux no usa letras de unidad para organizar la unidad como lo hace Windows. En Linux, no podemos saber si nos dirigimos a una partición, un dispositivo de red o un directorio "ordinario" y una unidad.
- Sensibilidad a mayúsculas y minúsculas:** el sistema de archivos de Linux distingue entre mayúsculas y minúsculas. Distingue entre nombres de archivo en minúsculas y mayúsculas. Por ejemplo, hay una diferencia entre test.txt y Test.txt en Linux. Esta regla también se aplica para directorios y comandos de Linux.
- Extensiones de archivo:** en Linux, un archivo puede tener la extensión '.txt', pero no es necesario que un archivo tenga una extensión de archivo. Al trabajar con Shell, los principiantes tienen algunos problemas para diferenciar entre archivos y directorios. Si usamos el administrador de archivos gráfico, simboliza los archivos y carpetas.
- Archivos ocultos:** Linux distingue entre archivos estándar y archivos ocultos, la mayoría de los archivos de configuración están ocultos en el sistema operativo Linux. Por lo general, no necesitamos acceder o leer los archivos ocultos. Los archivos ocultos en Linux están representados por un punto (.) antes del nombre del archivo (p. ej., .ignorar). Para acceder a los archivos, necesitamos cambiar la vista en el administrador de archivos o usar un comando específico en el shell.

Tipos de sistema de archivos de Linux

Cuando instalamos el sistema operativo Linux, Linux ofrece muchos sistemas de archivos como **Ext**, **Ext2**, **Ext3**, **Ext4**, **JFS**, **ReiserFS**, **XFS**, **btrfs** y **swap**

Types of Linux File System



1. Sistema de archivos Ext, Ext2, Ext3 y Ext4

El sistema de archivos Ext significa **Extended File System** . Fue desarrollado principalmente para **MINIX OS** . El sistema de archivos Ext es una versión anterior y ya no se usa debido a algunas limitaciones.

Ext2 es el primer sistema de archivos de Linux que permite administrar dos terabytes de datos. Ext3 se desarrolla a través de Ext2; es una versión mejorada de Ext2 y contiene compatibilidad con versiones anteriores. El principal inconveniente de Ext3 es que no admite servidores porque este sistema de archivos no admite la recuperación de archivos ni la instantánea del disco.

El sistema de archivos Ext4 es el sistema de archivos más rápido entre todos los sistemas de archivos Ext. Es una opción muy compatible para los discos SSD (unidad de estado sólido), y es el sistema de archivos predeterminado en la distribución de Linux.

2. Sistema de archivos JFS

JFS son las siglas de **Journaled File System** y lo ha desarrollado **IBM para AIX Unix** . Es una alternativa al sistema de archivos Ext. También se puede usar en lugar de Ext4, donde se necesita estabilidad con pocos recursos. Es un sistema de archivos útil cuando la potencia [de la CPU](#) es limitada.

3. Sistema de archivos ReiserFS

ReiserFS es una alternativa al sistema de archivos Ext3. Tiene un rendimiento mejorado y funciones avanzadas. Anteriormente, ReiserFS se usaba como el sistema de archivos predeterminado en SUSE Linux, pero luego cambió algunas políticas, por lo que SUSE volvió a Ext3. Este sistema de archivos admite dinámicamente la extensión del archivo, pero tiene algunos inconvenientes en el rendimiento.

4. Sistema de archivos XFS

El sistema de archivos XFS se consideró como JFS de alta velocidad, desarrollado para el procesamiento de E/S en paralelo. La NASA todavía usa este sistema de archivos con su servidor de almacenamiento alto (servidor de 300+ Terabyte).

5. Sistema de archivos Btrfs

Btrfs significa el **sistema de archivos de árbol B**. Se utiliza para tolerancia a fallas, sistema de reparación, administración divertida, configuración de almacenamiento extensiva y más. No es un buen traje para el sistema de producción.

6. Sistema de intercambio de archivos

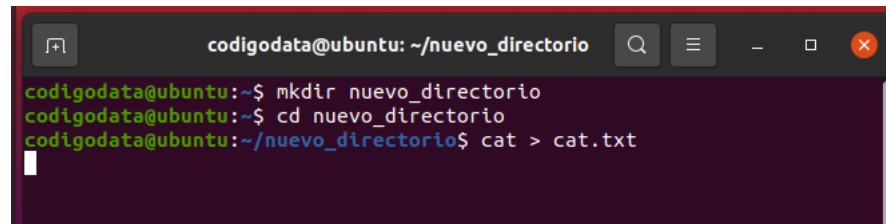
El sistema de archivos de intercambio se utiliza para la paginación de memoria en el sistema operativo Linux durante la hibernación del sistema. Un sistema que nunca entra en estado de hibernación debe tener un espacio de intercambio igual al tamaño de su [RAM](#)

Crear archivo de Linux

El sistema de archivos de Linux considera todo como un archivo en Linux; ya sean imágenes de archivos de texto, particiones, programas compilados, directorios o dispositivos de hardware. Si no es un archivo, entonces debe ser un proceso. Para gestionar los datos, forma una estructura de árbol.

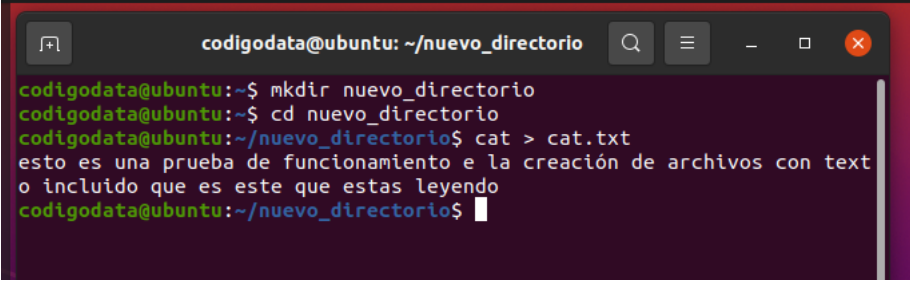
1. Usando el comando cat

Creamos un directorio y ejecutamos el comando cat para crear un archivo:



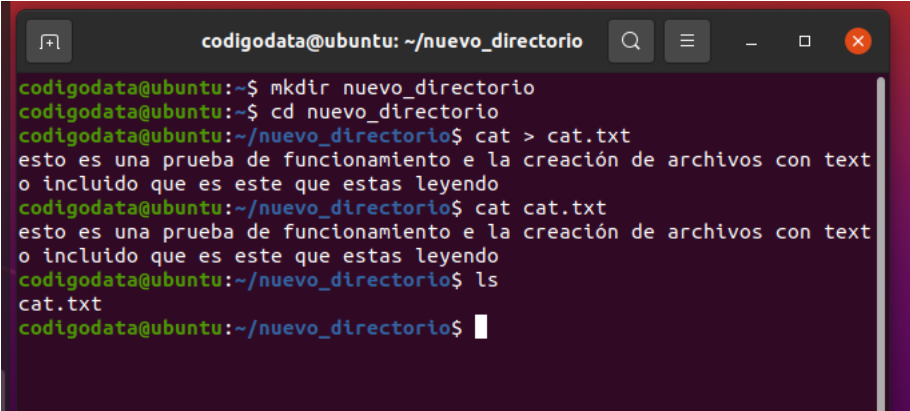
```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~$ mkdir nuevo_directorio
codigodata@ubuntu:~$ cd nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ cat > cat.txt
```


El comando anterior creará un archivo de texto y entrará en el modo editor. Ahora, ingrese el texto deseado y presione la tecla **CTRL + D** para guardar y salir del archivo y volverá a la línea de comando.



```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~$ mkdir nuevo_directorio
codigodata@ubuntu:~$ cd nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ cat > cat.txt
esto es una prueba de funcionamiento e la creación de archivos con text
o incluido que es este que estas leyendo
codigodata@ubuntu:~/nuevo_directorio$
```

Para mostrar el contenido del archivo, ejecute el comando cat de la siguiente manera:

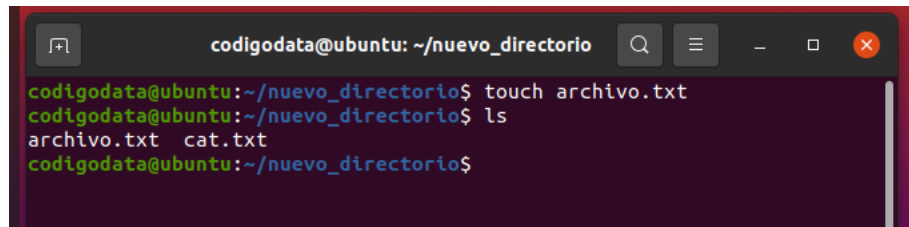


```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~$ mkdir nuevo_directorio
codigodata@ubuntu:~$ cd nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ cat > cat.txt
esto es una prueba de funcionamiento e la creación de archivos con text
o incluido que es este que estas leyendo
codigodata@ubuntu:~/nuevo_directorio$ cat cat.txt
esto es una prueba de funcionamiento e la creación de archivos con text
o incluido que es este que estas leyendo
codigodata@ubuntu:~/nuevo_directorio$ ls
cat.txt
codigodata@ubuntu:~/nuevo_directorio$
```

2. Usando el comando touch

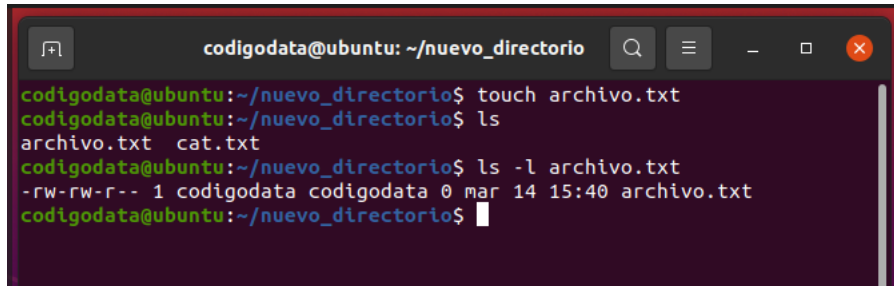
El comando **táctil** también es uno de los comandos populares en Linux. Se utiliza para **crear un nuevo archivo, actualizar la marca de tiempo en archivos y directorios existentes** . También puede crear archivos vacíos en Linux.

El **comando touch** es la forma más sencilla de crear un nuevo archivo desde la línea de comandos. Podemos crear varios archivos ejecutando este comando a la vez. Para crear un archivo, ejecute el comando táctil seguido del nombre del archivo como se indica a continuación:

A screenshot of a Linux terminal window. The window title is 'codigodata@ubuntu: ~/nuevo_directorio'. The terminal shows the following commands and output: 'touch archivo.txt' is executed, followed by 'ls' which outputs 'archivo.txt cat.txt'.

```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ touch archivo.txt
codigodata@ubuntu:~/nuevo_directorio$ ls
archivo.txt  cat.txt
codigodata@ubuntu:~/nuevo_directorio$
```

Para listar la información del archivo creado, ejecute el siguiente comando:

A terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio'. The user enters 'touch archivo.txt', then 'ls' showing 'archivo.txt cat.txt', and finally 'ls -l archivo.txt' showing file permissions and details.

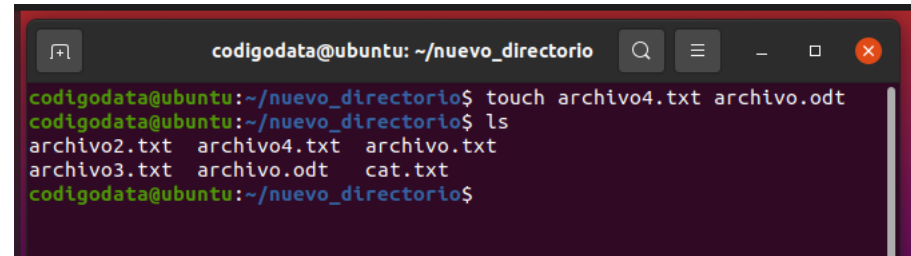
```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ touch archivo.txt
codigodata@ubuntu:~/nuevo_directorio$ ls
archivo.txt  cat.txt
codigodata@ubuntu:~/nuevo_directorio$ ls -l archivo.txt
-rw-rw-r-- 1 codigodata codigodata 0 mar 14 15:40 archivo.txt
codigodata@ubuntu:~/nuevo_directorio$
```

Para crear varios archivos a la vez, especifique los archivos y sus extensiones después del comando touch junto con un solo espacio. Ejecute el siguiente comando para crear tres archivos a la vez:

A terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio'. The user enters 'touch archivo.txt', then 'ls' showing 'archivo.txt cat.txt', then 'ls -l archivo.txt' showing file details, and finally 'touch archivo2.txt archivo2.txt a rchivo3.txt' (note the typo 'a rchivo3.txt' in the original image). The final 'ls' command shows 'archivo2.txt archivo3.txt archivo.txt cat.txt'.

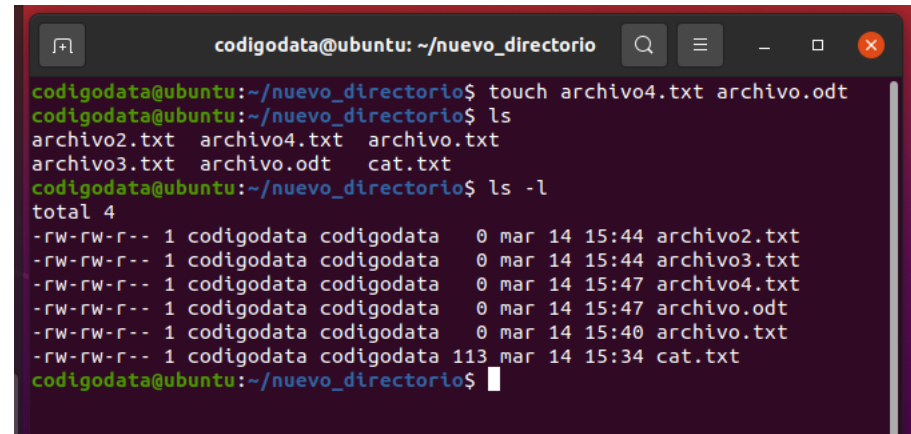
```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ touch archivo.txt
codigodata@ubuntu:~/nuevo_directorio$ ls
archivo.txt  cat.txt
codigodata@ubuntu:~/nuevo_directorio$ ls -l archivo.txt
-rw-rw-r-- 1 codigodata codigodata 0 mar 14 15:40 archivo.txt
codigodata@ubuntu:~/nuevo_directorio$ touch archivo2.txt archivo2.txt a
rchivo3.txt
codigodata@ubuntu:~/nuevo_directorio$ ls
archivo2.txt  archivo3.txt  archivo.txt  cat.txt
codigodata@ubuntu:~/nuevo_directorio$
```

Para crear dos tipos diferentes de archivos, ejecute el comando de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio'. The user enters 'touch archivo4.txt archivo.odt' and then 'ls'. The output of 'ls' shows 'archivo2.txt', 'archivo4.txt', 'archivo.txt', 'archivo3.txt', 'archivo.odt', and 'cat.txt'.

```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ touch archivo4.txt archivo.odt
codigodata@ubuntu:~/nuevo_directorio$ ls
archivo2.txt  archivo4.txt  archivo.txt
archivo3.txt  archivo.odt   cat.txt
codigodata@ubuntu:~/nuevo_directorio$
```

Para mostrar el archivo y su marca de tiempo, ejecute el comando `ls` de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio'. The user enters 'touch archivo4.txt archivo.odt', then 'ls', and finally 'ls -l'. The output of 'ls -l' shows detailed file information including permissions, owner, group, size, date, and filename.

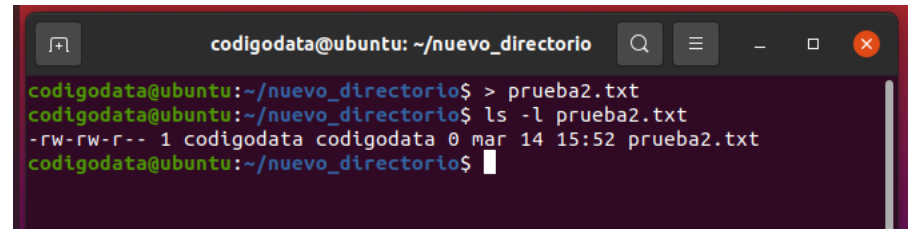
```
codigodata@ubuntu:~/nuevo_directorio$ touch archivo4.txt archivo.odt
codigodata@ubuntu:~/nuevo_directorio$ ls
archivo2.txt  archivo4.txt  archivo.txt
archivo3.txt  archivo.odt   cat.txt
codigodata@ubuntu:~/nuevo_directorio$ ls -l
total 4
-rw-rw-r-- 1 codigodata codigodata 0 mar 14 15:44 archivo2.txt
-rw-rw-r-- 1 codigodata codigodata 0 mar 14 15:44 archivo3.txt
-rw-rw-r-- 1 codigodata codigodata 0 mar 14 15:47 archivo4.txt
-rw-rw-r-- 1 codigodata codigodata 0 mar 14 15:47 archivo.odt
-rw-rw-r-- 1 codigodata codigodata 0 mar 14 15:40 archivo.txt
-rw-rw-r-- 1 codigodata codigodata 113 mar 14 15:34 cat.txt
codigodata@ubuntu:~/nuevo_directorio$
```

Si pasamos el nombre de un archivo existente, cambiará la marca de tiempo de ese archivo.

3. Usar el símbolo de redirección (>)

También podemos crear un archivo usando el símbolo de redirección (>) en la línea de comando. Para crear un archivo, solo tenemos que escribir un símbolo de redirección (>) seguido del nombre del archivo. Este símbolo se usa principalmente para redirigir la salida. Hay dos formas de redirigir la salida. Si usamos el **operador** > , sobrescribirá el archivo existente y el operador >> agregará la salida.

Para crear un archivo con operador de redirección (>), ejecute el comando de la siguiente manera:

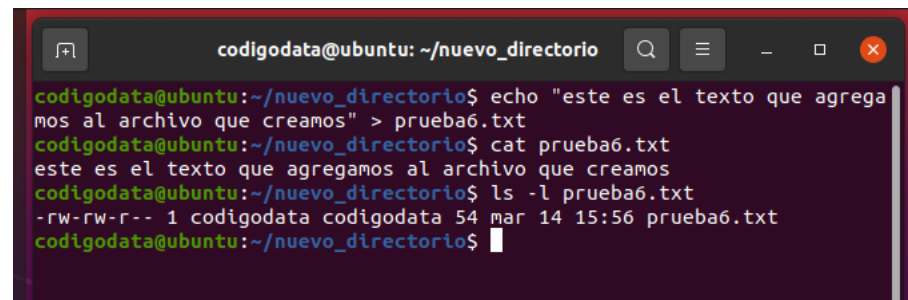
A terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio' showing the following commands and output:

```
codigodata@ubuntu:~/nuevo_directorio$ > prueba2.txt
codigodata@ubuntu:~/nuevo_directorio$ ls -l prueba2.txt
-rw-rw-r-- 1 codigodata codigodata 0 mar 14 15:52 prueba2.txt
codigodata@ubuntu:~/nuevo_directorio$
```

4. Usando el comando de eco

El comando **echo** se usa para crear un archivo, pero debemos especificar el contenido del archivo en la línea de comando.

Para crear el archivo con el comando echo, ejecute el comando de la siguiente manera:

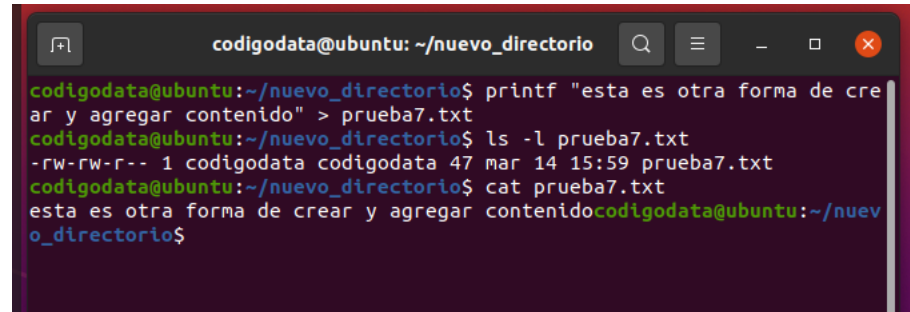
A terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio' showing the following commands and output:

```
codigodata@ubuntu:~/nuevo_directorio$ echo "este es el texto que agregamos al archivo que creamos" > prueba6.txt
codigodata@ubuntu:~/nuevo_directorio$ cat prueba6.txt
este es el texto que agregamos al archivo que creamos
codigodata@ubuntu:~/nuevo_directorio$ ls -l prueba6.txt
-rw-rw-r-- 1 codigodata codigodata 54 mar 14 15:56 prueba6.txt
codigodata@ubuntu:~/nuevo_directorio$
```

5. Usando el comando printf

También podemos crear un archivo usando el comando **printf** . Para esto necesitamos especificar el contenido del archivo en la línea de comando.

Para crear un archivo con el comando printf, ejecute el comando de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio' showing a series of commands and their outputs. The commands are: 'printf "esta es otra forma de crear y agregar contenido" > prueba7.txt', 'ls -l prueba7.txt', and 'cat prueba7.txt'. The output of 'ls' shows a file 'prueba7.txt' with permissions '-rw-rw-r--', size '47', and timestamp 'mar 14 15:59'. The output of 'cat' shows the text 'esta es otra forma de crear y agregar contenido'.

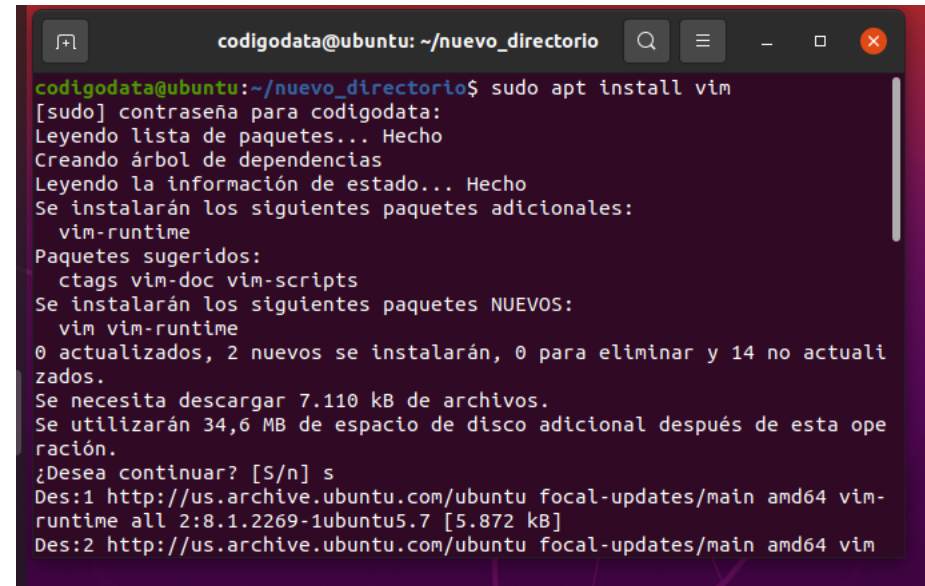
```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ printf "esta es otra forma de crear y agregar contenido" > prueba7.txt
codigodata@ubuntu:~/nuevo_directorio$ ls -l prueba7.txt
-rw-rw-r-- 1 codigodata codigodata 47 mar 14 15:59 prueba7.txt
codigodata@ubuntu:~/nuevo_directorio$ cat prueba7.txt
esta es otra forma de crear y agregar contenido
codigodata@ubuntu:~/nuevo_directorio$
```

6. Usando el editor de texto

También podemos crear un archivo usando los diferentes editores de texto como **vim**, **nano**, **vi** y más.

•Usando el editor de texto Vim

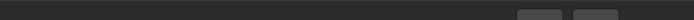
Podemos crear un archivo usando el editor de **texto Vim** . Si no tiene el editor vim instalado en su máquina, ejecute el siguiente comando:

A terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio' showing the command 'sudo apt install vim' and its output. The output includes the password prompt, package list reading, dependency tree creation, state information reading, additional packages to be installed (vim-runtime), suggested packages (ctags, vim-doc, vim-scripts), new packages to be installed (vim, vim-runtime), disk space requirements, and the confirmation to proceed with installation. The terminal shows the progress of the installation with progress bars.

```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ sudo apt install vim
[sudo] contraseña para codigodata:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
vim-runtime
Paquetes sugeridos:
ctags vim-doc vim-scripts
Se instalarán los siguientes paquetes NUEVOS:
vim vim-runtime
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 14 no actualizados.
Se necesita descargar 7.110 kB de archivos.
Se utilizarán 34,6 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 vim-runtime all 2:8.1.2269-1ubuntu5.7 [5.872 kB]
Des:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 vim
```

El editor de texto Vim se instaló correctamente.

Para crear un archivo usando el editor de texto vim, ejecute el siguiente comando:



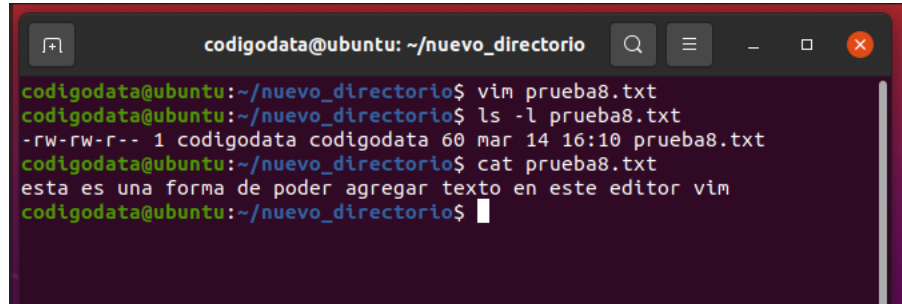
```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ vim prueba8.txt
codigodata@ubuntu:~/nuevo_directorio$
```

El comando anterior abrirá el editor de texto, presione la tecla **i** para ir al modo de inserción del editor.

Ingrese el contenido del archivo, presione la **tecla Esc** precedida por **:wq** para guardar y salir del archivo. El editor de texto tiene el siguiente aspecto:

A screenshot of a terminal window titled "codigodata@ubuntu: ~/nuevo_directorio". The terminal shows a Vim editor interface with a dark background. The first line contains the text "esta es una forma de poder agregar texto en este editor vim" followed by a cursor. Below this, there are several tilde (~) symbols representing empty lines. At the bottom left, it says "-- INSERTAR --". At the bottom right, it shows "1,60" and "Todo". The terminal has standard window controls at the top.

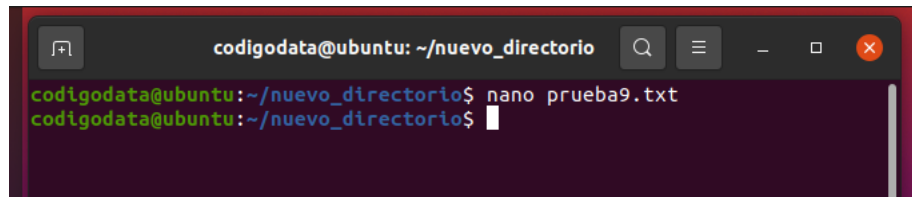
Para mostrar la información del archivo, ejecute el comando **ls** y para ver el contenido del archivo, ejecute el comando **cat** de la siguiente manera:

A terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio' with search, menu, and window control icons. The terminal shows a sequence of commands: 'vim prueba8.txt', 'ls -l prueba8.txt', and 'cat prueba8.txt'. The output of 'ls' shows file permissions, owner, group, size, date, and filename. The output of 'cat' shows the text content of the file.

```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ vim prueba8.txt
codigodata@ubuntu:~/nuevo_directorio$ ls -l prueba8.txt
-rw-rw-r-- 1 codigodata codigodata 60 mar 14 16:10 prueba8.txt
codigodata@ubuntu:~/nuevo_directorio$ cat prueba8.txt
esta es una forma de poder agregar texto en este editor vim
codigodata@ubuntu:~/nuevo_directorio$
```

7. Usando el editor Nano

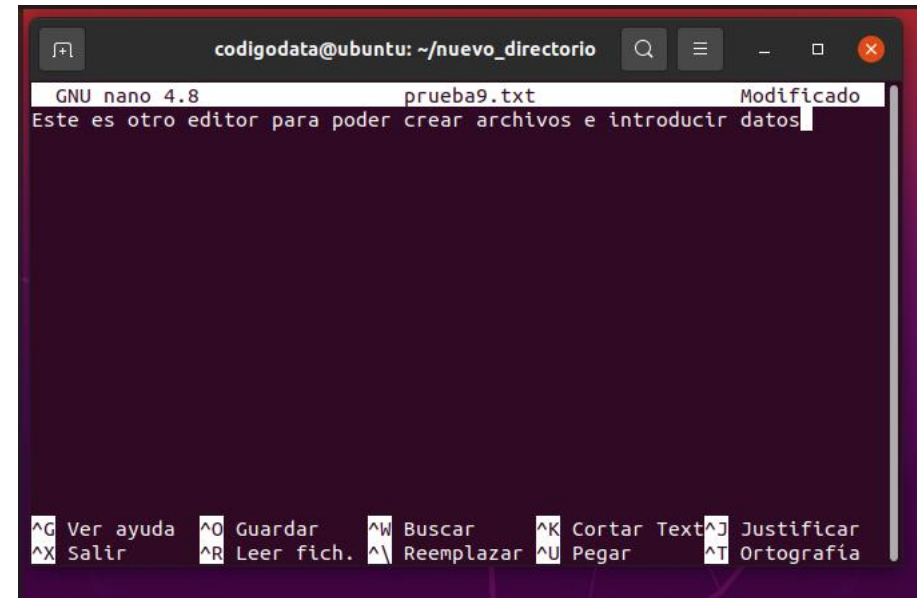
Podemos crear un archivo usando el editor de texto **nano** . Para crear un archivo, ejecute el siguiente comando:

A terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio' with search, menu, and window control icons. The terminal shows the command 'nano prueba9.txt' being entered at the prompt.

```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ nano prueba9.txt
codigodata@ubuntu:~/nuevo_directorio$
```



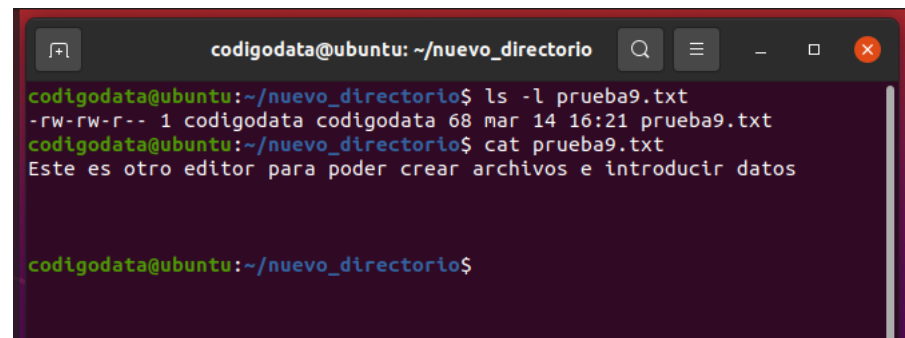

The screenshot shows the nano text editor interface. The title bar indicates the user is 'codigodata' on an 'ubuntu' machine, in the directory '~/nuevo_directorio'. The editor is editing a file named 'prueba9.txt'. The main editing area is empty. At the bottom, there is a menu bar with various shortcuts: ^G Ver ayuda, ^O Guardar, ^W Buscar, ^K Cortar Text, ^J Justificar, ^X Salir, ^R Leer fich., ^\ Reemplazar, ^U Pegar, and ^T Ortografía. A small box above the menu bar says '[Archivo nuevo]'.



The screenshot shows the nano text editor interface. The title bar indicates the user is 'codigodata' on an 'ubuntu' machine, in the directory '~/nuevo_directorio'. The editor is editing a file named 'prueba9.txt'. The main editing area contains the text 'Este es otro editor para poder crear archivos e introducir datos'. The menu bar at the bottom is the same as in the previous screenshot.

Para mostrar la información del archivo, ejecute el comando **ls**.

Para ver el contenido del archivo, ejecute el comando **cat** de la siguiente manera:




```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ ls -l prueba9.txt
-rw-rw-r-- 1 codigodata codigodata 68 mar 14 16:21 prueba9.txt
codigodata@ubuntu:~/nuevo_directorio$ cat prueba9.txt
Este es otro editor para poder crear archivos e introducir datos

codigodata@ubuntu:~/nuevo_directorio$
```

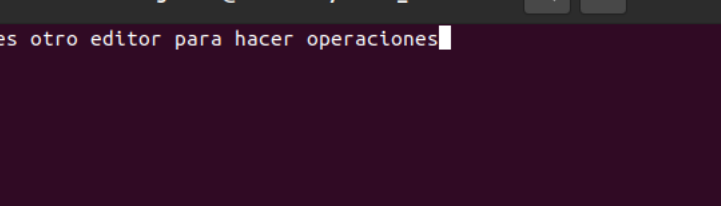
Usando el editor Vi

Para crear un archivo con el editor Vi, ejecute el siguiente comando:



```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ vi prueba10.txt
codigodata@ubuntu:~/nuevo_directorio$
```

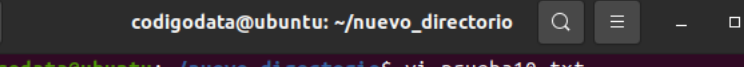
El comando anterior abrirá el editor Vi. Presione la tecla i para el modo de inserción e ingrese el contenido del archivo. Presione la tecla Esc y :wq para guardar y salir del archivo del editor.



codigodata@ubuntu: ~/nuevo_directorio

Este es otro editor para hacer operaciones

-- INSERTAR -- 1,43 Todo

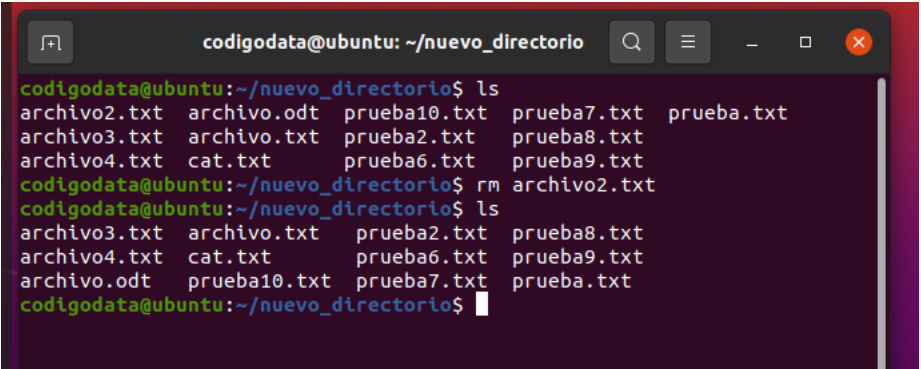


The screenshot shows a terminal window with a dark background. The title bar at the top reads "codigodata@ubuntu: ~/nuevo_directorio". The terminal contains the following commands and output:

```
codigodata@ubuntu:~/nuevo_directorio$ vi prueba10.txt
codigodata@ubuntu:~/nuevo_directorio$ ls -l prueba10.txt
-rw-rw-r-- 1 codigodata codigodata 29  mar 14 16:30 prueba10.txt
codigodata@ubuntu:~/nuevo_directorio$ cat prueba10.txt
Este es otro editor de texto
codigodata@ubuntu:~/nuevo_directorio$
```

linux rm | Eliminar archivo de Linux

El 'rm' significa eliminar. Este comando se utiliza para eliminar un archivo. La línea de comando no tiene una papelera de reciclaje o papelera a diferencia de otras GUI para recuperar los archivos. Por lo tanto, tenga mucho cuidado al usar este comando. Una vez que haya eliminado un archivo, se eliminará de forma permanente.

A screenshot of a terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio'. The terminal shows the following commands and output:
1. `ls` command output: `archivo2.txt archivo.odt prueba10.txt prueba7.txt prueba.txt`
2. `rm archivo2.txt` command executed.
3. `ls` command output: `archivo3.txt archivo.txt prueba2.txt prueba8.txt`
4. `archivo4.txt cat.txt prueba6.txt prueba9.txt`
5. `archivo.odt prueba10.txt prueba7.txt prueba.txt`
The cursor is at the end of the last line.

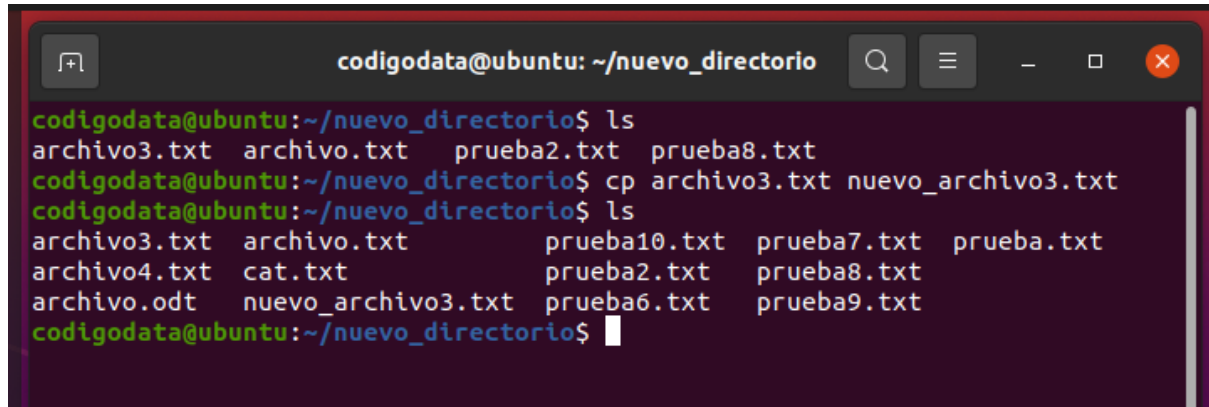
```
codigodata@ubuntu: ~/nuevo_directorio
codigodata@ubuntu:~/nuevo_directorio$ ls
archivo2.txt  archivo.odt  prueba10.txt  prueba7.txt  prueba.txt
archivo3.txt  archivo.txt  prueba2.txt   prueba8.txt
archivo4.txt  cat.txt      prueba6.txt   prueba9.txt
codigodata@ubuntu:~/nuevo_directorio$ rm archivo2.txt
codigodata@ubuntu:~/nuevo_directorio$ ls
archivo3.txt  archivo.txt  prueba2.txt   prueba8.txt
archivo4.txt  cat.txt      prueba6.txt   prueba9.txt
archivo.odt   prueba10.txt prueba7.txt    prueba.txt
codigodata@ubuntu:~/nuevo_directorio$
```

Opciones de rm

Opción	Descripción
<code>rm *extensión</code>	Se utiliza para eliminar archivos que tienen la misma extensión.
<code>rm -r o R</code>	Para eliminar un directorio recursivamente.
<code>rm-i</code>	Eliminar un archivo de forma interactiva.
<code>rm-rf</code>	Eliminar un directorio a la fuerza.

cp linux | Copiar archivo de Linux

'cp' significa copia. El comando 'cp' se usa para copiar un archivo o un directorio. Para copiar un archivo en el mismo directorio, la sintaxis será,

A terminal window titled 'codigodata@ubuntu: ~/nuevo_directorio' with standard window controls. The terminal shows the following commands and output:

```
codigodata@ubuntu:~/nuevo_directorio$ ls
archivo3.txt  archivo.txt  prueba2.txt  prueba8.txt
codigodata@ubuntu:~/nuevo_directorio$ cp archivo3.txt nuevo_archivo3.txt
codigodata@ubuntu:~/nuevo_directorio$ ls
archivo3.txt  archivo.txt      prueba10.txt  prueba7.txt  prueba.txt
archivo4.txt  cat.txt          prueba2.txt   prueba8.txt
archivo.odt   nuevo_archivo3.txt  prueba6.txt   prueba9.txt
codigodata@ubuntu:~/nuevo_directorio$
```

En la instantánea anterior, creamos una copia de 'archivo3.txt' y la llamamos 'nuevo_archivo3.txt' . Si en el caso, (en nuestro caso es 'nuevo_archivo3.txt') ya existe, simplemente sobrescribirá el archivo anterior.

Opciones de cp

Opción	Función
<u>cp-r</u>	Para copiar un directorio junto con sus subdirectorios.
<u>cp archivo 1 archivo 2 nombre de directorio</u>	Para copiar varios archivos o directorios en un directorio.
<u>cp -backup</u>	Para hacer una copia de seguridad del archivo existente antes de sobrescribirlo.
<u>cp -i</u>	Pide confirmación.
<u>cp-l</u>	Para crear un archivo de enlace duro.
<u>cp-p</u>	Conserva el atributo de un archivo.
<u>cp-u-v</u>	Para asegurarse de que el archivo de origen sea más nuevo que el archivo de destino.

Linux mv | Mover archivo de Linux

El comando mv de Linux se usa para mover un archivo o directorio existente de una ubicación a otra. También se utiliza para cambiar el nombre de un archivo o directorio. Si desea cambiar el nombre de un solo directorio o archivo, será mejor usar la opción '**mv**'.

Cómo cambiar el nombre de un archivo

Al cambiar el nombre de un archivo, el número de inodo de ambos archivos seguirá siendo el mismo.

A terminal window titled 'codigodata@ubuntu: ~/Downloads' showing a sequence of commands and their outputs. The user first runs 'ls -i -l' to list files with their inodes. Then, they run 'mv archivo.txt documento.txt' to rename the file. Finally, they run 'ls -i -l' again to confirm the change. The inodes for both files remain the same (660458).

```
codigodata@ubuntu:~/Downloads$ ls -i -l
total 8
660385 -rw-rw-r-- 1 codigodata codigodata 30 mar 14 17:17 archivo_tranferir.txt
660458 -rw-rw-r-- 1 codigodata codigodata 37 mar 14 16:56 archivo.txt
codigodata@ubuntu:~/Downloads$ mv archivo.txt documento.txt
codigodata@ubuntu:~/Downloads$ ls -i -l
total 8
660385 -rw-rw-r-- 1 codigodata codigodata 30 mar 14 17:17 archivo_tranferir.txt
660458 -rw-rw-r-- 1 codigodata codigodata 37 mar 14 16:56 documento.txt
codigodata@ubuntu:~/Downloads$
```

En el ejemplo anterior, hemos cambiado el nombre del archivo '**docc**' a '**document**'. Pero el número de inodo de ambos archivos sigue siendo el mismo.

Cómo cambiar el nombre de un directorio

Los directorios se pueden renombrar de la misma manera que los archivos. En este caso, también el número de inodo seguirá siendo el mismo.

```
codigodata@ubuntu: ~/Downloads
codigodata@ubuntu:~/Downloads$ ls -i -l
total 8
660385 -rw-rw-r-- 1 codigodata codigodata 30 mar 14 17:17 archivo_tranf
erir.txt
660458 -rw-rw-r-- 1 codigodata codigodata 37 mar 14 16:56 documento.txt
codigodata@ubuntu:~/Downloads$
codigodata@ubuntu:~/Downloads$ mkdir dir
codigodata@ubuntu:~/Downloads$ ls
archivo_tranferir.txt  dir  documento.txt
codigodata@ubuntu:~/Downloads$ mv dir tpm
codigodata@ubuntu:~/Downloads$ ls
archivo_tranferir.txt  documento.txt  tpm
codigodata@ubuntu:~/Downloads$
```

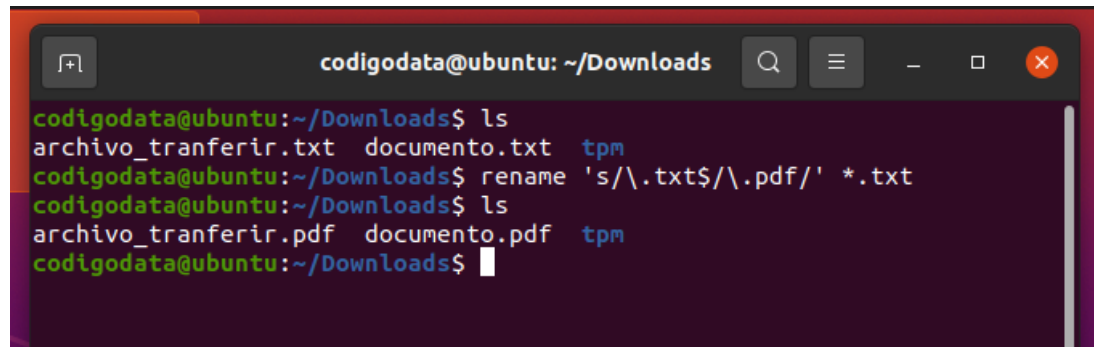
Opción mv

Opción	Función
mv -i	Pide permiso para sobrescribir.
mv *	Mover varios archivos a un directorio específico.
mv --suffix	Se utiliza para realizar una copia de seguridad antes de sobrescribir.
mv -u	Solo mueva aquellos archivos que no existen.

Linux Cambiar nombre de archivo y directorio

Para cambiar el nombre de un archivo, también hay otros comandos como '**mv**'. Pero el comando '**cambiar nombre**' es ligeramente más avanzado que otros. Este comando rara vez se usará y funciona de manera diferente en diferentes distribuciones de Linux. Trabajaremos en ejemplos de Debian/Ubuntu.

En el siguiente ejemplo, hemos convertido todos los archivos que terminan en '**.txt**' en archivos que terminan en '**.pdf**'.



```
codigodata@ubuntu: ~/Downloads
codigodata@ubuntu:~/Downloads$ ls
archivo_tranferir.txt  documento.txt  tpm
codigodata@ubuntu:~/Downloads$ rename 's/\.txt$/\.pdf/' *.txt
codigodata@ubuntu:~/Downloads$ ls
archivo_tranferir.pdf  documento.pdf  tpm
codigodata@ubuntu:~/Downloads$
```


opción de cambio de nombre:

rename tiene algunos argumentos opcionales, pero una expresión perl obligatoria que viene con cada opción y la guía sobre cómo trabajar.

Opción	Función
<u>rename -n</u>	Verifique los cambios antes de ejecutar el comando.
<u>rename -v</u>	Imprima la salida.
<u>rename (a-z)(A-Z)/ (A-Z)(a-z)</u>	Convertir a mayúsculas/minúsculas.
<u>rename -f</u>	Sobrescribir con fuerza los archivos existentes.

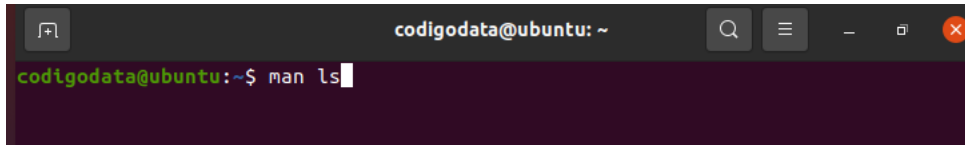
Comando man de Linux

Es un término corto para la página del manual. En sistemas operativos similares a Unix, como Linux, man es una interfaz para ver el manual de referencia del sistema.

Un usuario puede solicitar mostrar una página de manual simplemente escribiendo man seguido de un espacio y luego un argumento. Aquí su argumento puede ser un comando, una utilidad o una función. Se muestra una página de manual asociada con cada uno de estos argumentos.

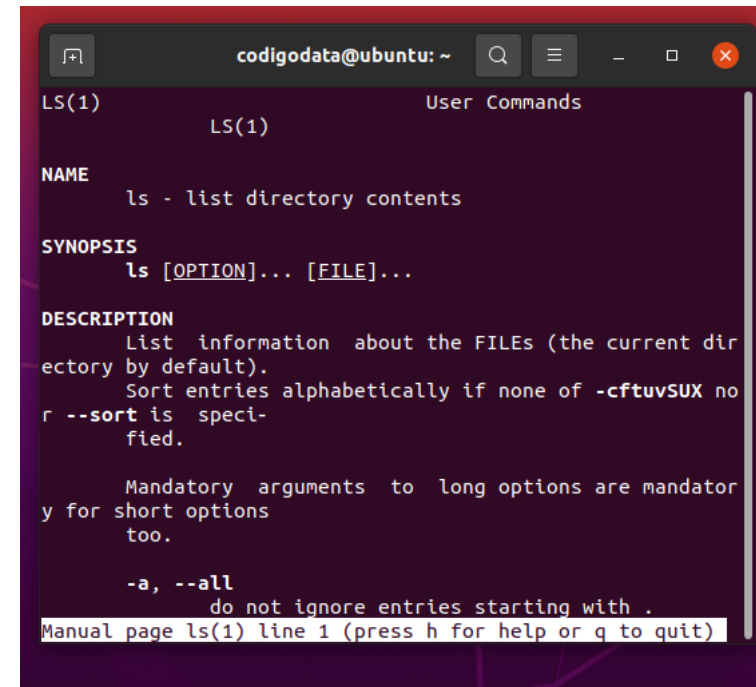
Si proporciona un número de sección en el comando, se le indicará al hombre que busque ese número de sección del manual y se mostrará la página de esa sección. Y si no, de forma predeterminada mostrará la primera página y tendrá que pasar por todas las secciones de una manera predefinida.

Leeremos sobre el número de sección en este tutorial.



```
codigodata@ubuntu: ~  
codigodata@ubuntu:~$ man ls
```

Este comando mostrará toda la información sobre el comando **'ls'** como se muestra en la captura de pantalla.



```
codigodata@ubuntu: ~  
LS(1) User Commands  
LS(1)  
  
NAME  
    ls - list directory contents  
  
SYNOPSIS  
    ls [OPTION]... [FILE]...  
  
DESCRIPTION  
    List information about the FILES (the current directory by default).  
    Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.  
  
    Mandatory arguments to long options are mandatory for short options too.  
  
    -a, --all  
        do not ignore entries starting with .  
Manual page ls(1) line 1 (press h for help or q to quit)
```

Cómo se muestra la salida en Command Shell

man muestra su salida a través del buscapersonas. Un buscapersonas es un programa que muestra su salida una pantalla completa a la vez, lo que significa que el texto completo no aparece a la vez y no hay opción para desplazarse hacia abajo en la página.

Dos puntos en la parte inferior muestran el final de la página en pantalla. Para ir a la siguiente página puede usar la 'barra espaciadora' o 'f' y para ir a la página anterior puede usar 'b'.

Para salir de la página en pantalla, use 'q' y será dirigido al programa shell. Y para ayuda presione 'h'.

Secciones en la página man:

La página man está dividida en diferentes secciones. Cada sección está dividida en función de un tema en particular.

Las páginas man tienen un número escrito entre paréntesis después del comando. Estos números representan el número de sección. En la imagen de arriba, puedes ver LS(1) en la parte superior, lo que muestra que es de la sección 1.

Como dije anteriormente, si menciona un número de sección en particular, su página de manual se dirigirá a esa sección. O bien, debe pasar por todas las secciones de una manera predefinida para ir a la sección deseada.

Secciones:

- 1.Executable programs and shell commands
- 2.System calls
- 3.Library calls
- 4.Special files
- 5.File formats and conventions
- 6.Games
- 7.Miscellaneous
- 8.System administration commands
- 9.Kernel routines
- n. Tcl/Tk (a programming language)

man Opciones

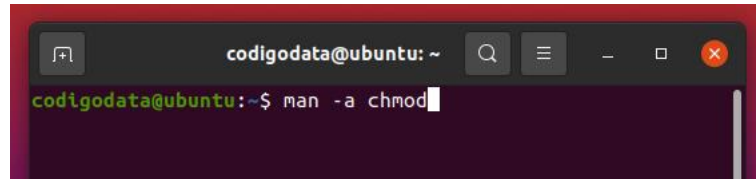
Comandos	Función
<u>man -aw</u>	Muestra todas las secciones disponibles de un comando.
<u>man -a</u>	Para ver todas las páginas man de un comando.
<u>sman -k (apropos)</u>	Muestra una lista de resultados en la página del manual que contiene una coincidencia de palabra clave.
<u>-f, whatis</u>	Muestra la descripción de la página del manual si está disponible.
<u>whereis</u>	Se utiliza para determinar la ubicación de una página de manual

Linux man -a

Este comando nos ayuda a mostrar todas las secciones de la página de manual de un tema en particular. Las secciones vendrán en orden ascendente y el número de sección más bajo aparecerá primero.

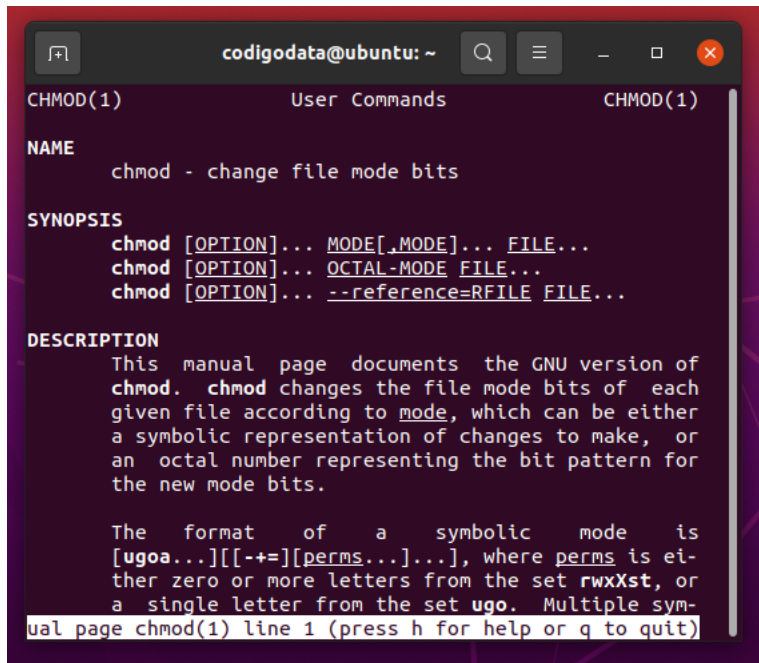
Para ir a la siguiente sección, presione 'q' y luego ingrese.

En la imagen de abajo, hemos dado el comando '**man -a chmod**'. Sabemos que chmod tiene dos secciones (1) y (2).



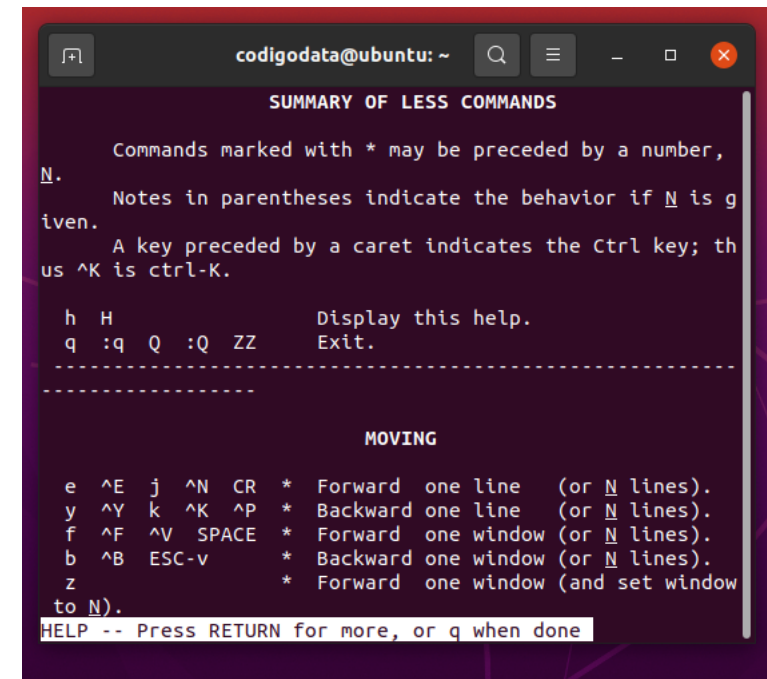
```
codigodata@ubuntu: ~  
codigodata@ubuntu:~$ man -a chmod
```

La página del manual de la sección 1 de chmod aparecerá primero como se muestra a continuación.



```
codigodata@ubuntu: ~  
CHMOD(1) User Commands CHMOD(1)  
  
NAME  
chmod - change file mode bits  
  
SYNOPSIS  
chmod [OPTION]... MODE[.MODE]... FILE...  
chmod [OPTION]... OCTAL-MODE FILE...  
chmod [OPTION]... --reference=RFILE FILE...  
  
DESCRIPTION  
This manual page documents the GNU version of  
chmod. chmod changes the file mode bits of each  
given file according to mode, which can be either  
a symbolic representation of changes to make, or  
an octal number representing the bit pattern for  
the new mode bits.  
  
The format of a symbolic mode is  
[ugoa...][[-+=][perms...]...], where perms is ei-  
ther zero or more letters from the set rwXst, or  
a single letter from the set ugo. Multiple sym-  
ual page chmod(1) line 1 (press h for help or q to quit)
```

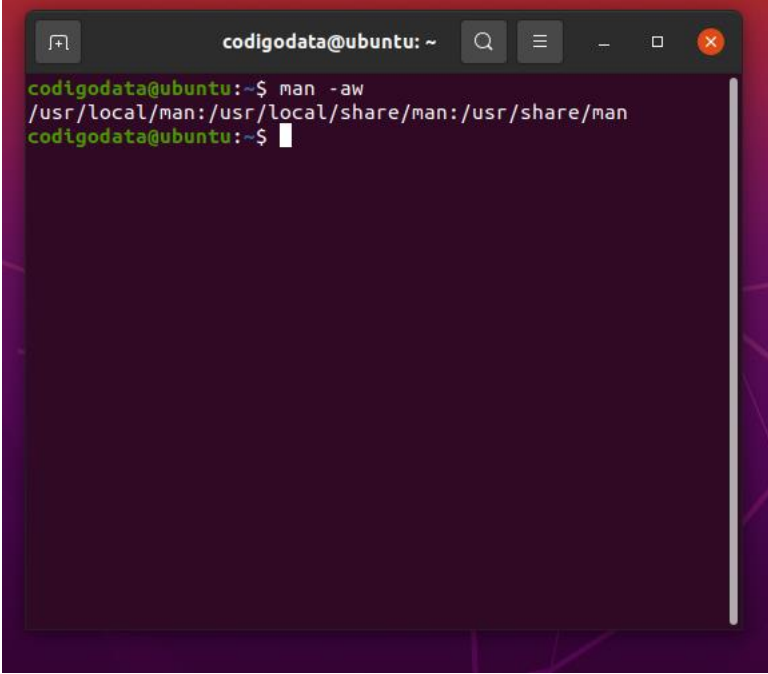
Para ir a la siguiente sección, presione q seguido de la tecla Intro y se nos dirigirá a la página de abajo de la sección 2.



```
codigodata@ubuntu: ~  
SUMMARY OF LESS COMMANDS  
  
Commands marked with * may be preceded by a number,  
N.  
Notes in parentheses indicate the behavior if N is g-  
iven.  
A key preceded by a caret indicates the Ctrl key; th-  
us ^K is ctrl-K.  
  
h H Display this help.  
q :q Q :Q ZZ Exit.  
-----  
  
MOVING  
  
e ^E j ^N CR * Forward one line (or N lines).  
y ^Y k ^K ^P * Backward one line (or N lines).  
f ^F ^V SPACE * Forward one window (or N lines).  
b ^B ESC-v * Backward one window (or N lines).  
z * Forward one window (and set window  
to N).  
HELP -- Press RETURN for more, or q when done
```

Linux man -aw

Este comando se usa para ver todas las secciones de un tema en particular en una página de manual.

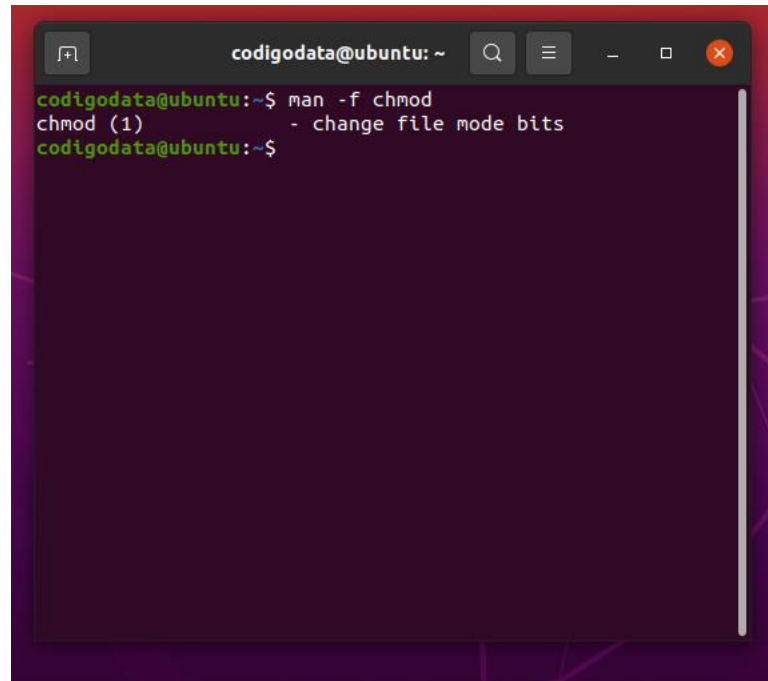
A terminal window with a dark background and a light-colored title bar. The title bar contains the text 'codigodata@ubuntu: ~' and standard window control icons. The terminal shows the command 'man -aw' being executed, followed by the output: '/usr/local/man:/usr/local/share/man:/usr/share/man'. The prompt 'codigodata@ubuntu:~\$' is visible at the end of the line.

```
codigodata@ubuntu: ~  
codigodata@ubuntu:~$ man -aw  
/usr/local/man:/usr/local/share/man:/usr/share/man  
codigodata@ubuntu:~$
```

Hay dos secciones para el comando 'chmod' como puede ver arriba. Entonces puede elegir cualquiera de las secciones (1) que son programas ejecutables o (2) que son llamadas al sistema.

Linux man -f (qué es)

Este comando funciona como el comando 'qué es', que estudiaremos en secciones posteriores. Busca el nombre de la página del manual para un comando dado y muestra el resultado con una breve descripción sobre la página del manual en particular.

A terminal window titled 'codigodata@ubuntu: ~' with standard window controls. The prompt is 'codigodata@ubuntu:~\$'. The command 'man -f chmod' has been entered. The output shows 'chmod (1)' followed by a description: '- change file mode bits'. The prompt 'codigodata@ubuntu:~\$' is visible again at the bottom.

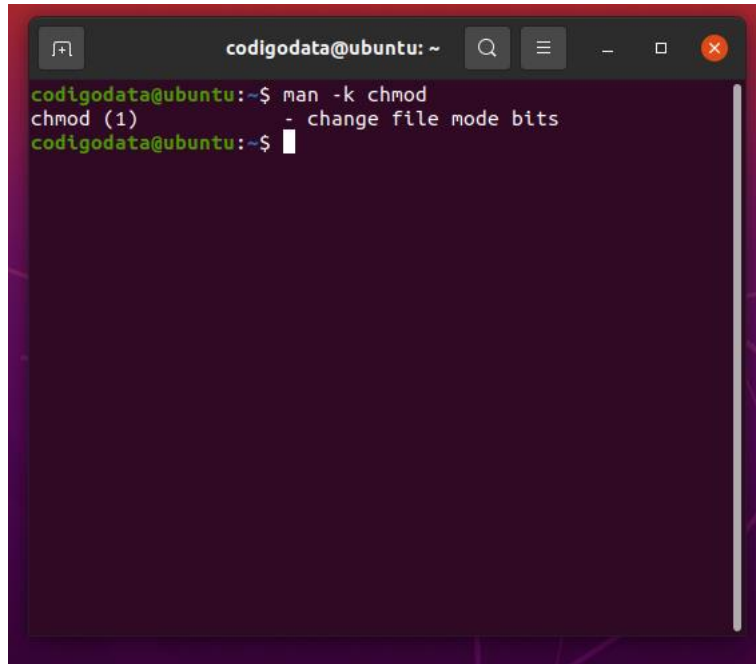
```
codigodata@ubuntu:~$ man -f chmod
chmod (1)
- change file mode bits
codigodata@ubuntu:~$
```

El ejemplo anterior muestra todas las páginas del manual junto con una breve descripción de esa página.

Linux man -k (a propósito)

Este comando mostrará todo tipo de resultados relacionados con su búsqueda de palabras clave en la página de manual, tal como lo hace apropos (que estudiaremos en secciones posteriores). Puede ser útil en el caso de que no sepa el nombre exacto de un comando.

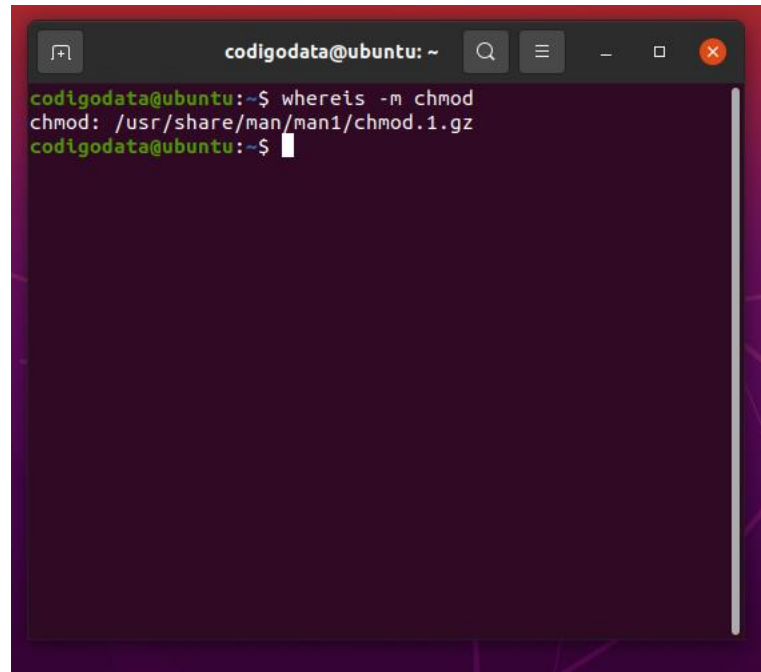
El resultado se mostrará en un formato con título, número de sección y su descripción como se muestra a continuación. Apropos no distingue entre mayúsculas y minúsculas, lo que lo hace más cómodo de usar.

A terminal window with a dark background and a red title bar. The title bar contains the text 'codigodata@ubuntu: ~' and standard window control icons. The terminal shows the command 'man -k chmod' being executed. The output is 'chmod (1)' followed by a description '- change file mode bits'. The prompt 'codigodata@ubuntu:~\$' is visible at the end of the line.

```
codigodata@ubuntu: ~  
codigodata@ubuntu:~$ man -k chmod  
chmod (1)  
- change file mode bits  
codigodata@ubuntu:~$
```

Linux whereis -m

Este comando se utiliza para encontrar la ubicación de una página de manual.

A terminal window with a dark background and light green text. The window title is 'codigodata@ubuntu: ~'. The command 'whereis -m chmod' has been entered and executed. The output shows the location of the manual page for 'chmod' as '/usr/share/man/man1/chmod.1.gz'.

```
codigodata@ubuntu: ~  
codigodata@ubuntu:~$ whereis -m chmod  
chmod: /usr/share/man/man1/chmod.1.gz  
codigodata@ubuntu:~$
```

Comando de contenido de archivos de Linux

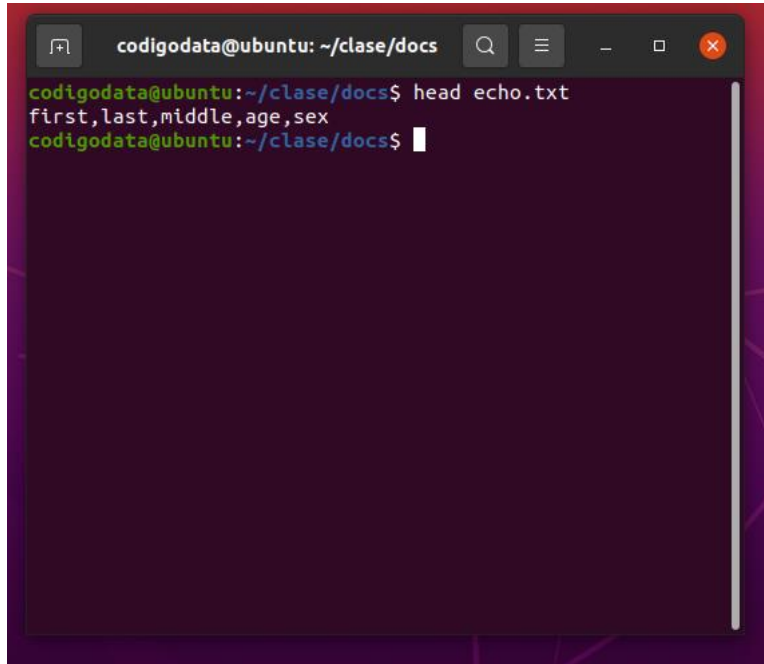
Hay muchos comandos que ayudan a ver el contenido de un archivo. Ahora veremos algunos de los comandos como head, tac, cat, less & more y strings.

Discutiremos sobre los siguientes contenidos de archivo que se dan en la tabla:

Comandos	Función
<u>head</u>	Muestra el comienzo de un archivo.
<u>tail</u>	Muestra la última última parte de un archivo.
<u>cat</u>	Este comando es polivalente y multi trabajador.
<u>tac</u>	Contrario de gato.
<u>more</u>	La línea de comandos muestra el contenido en forma de buscapersonas que está en más formato.
<u>less</u>	La línea de comandos muestra el contenido en forma de buscapersonas que tiene menos formato.

Comando principal de Linux

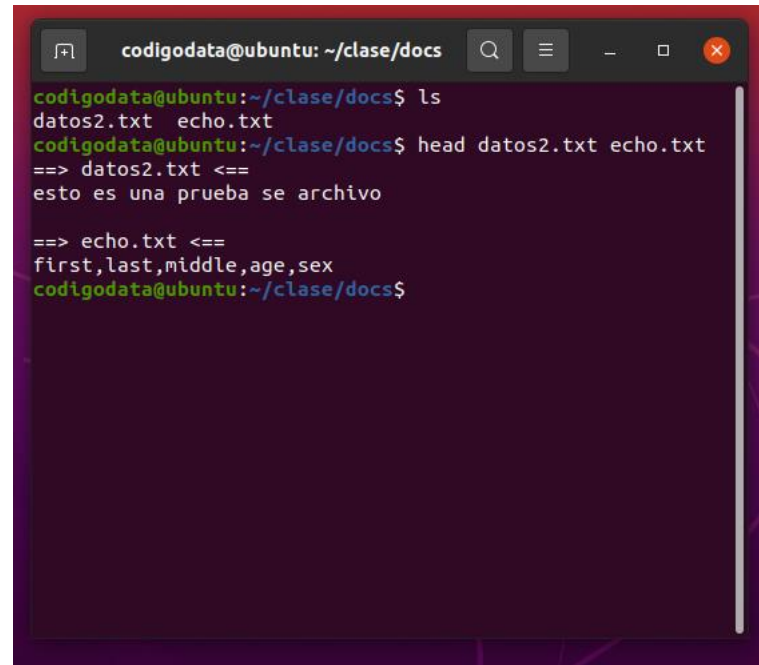
El comando 'head' muestra el contenido inicial de un archivo. De forma predeterminada, muestra las 10 líneas iniciales de cualquier archivo.



```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ head echo.txt
first,last,middle,age,sex
codigodata@ubuntu:~/clase/docs$
```

Comando principal para múltiples archivos

Si escribimos dos nombres de archivo, mostrará las primeras diez líneas (en este caso, el archivo tiene solo cinco líneas) de cada archivo separados por un encabezado.

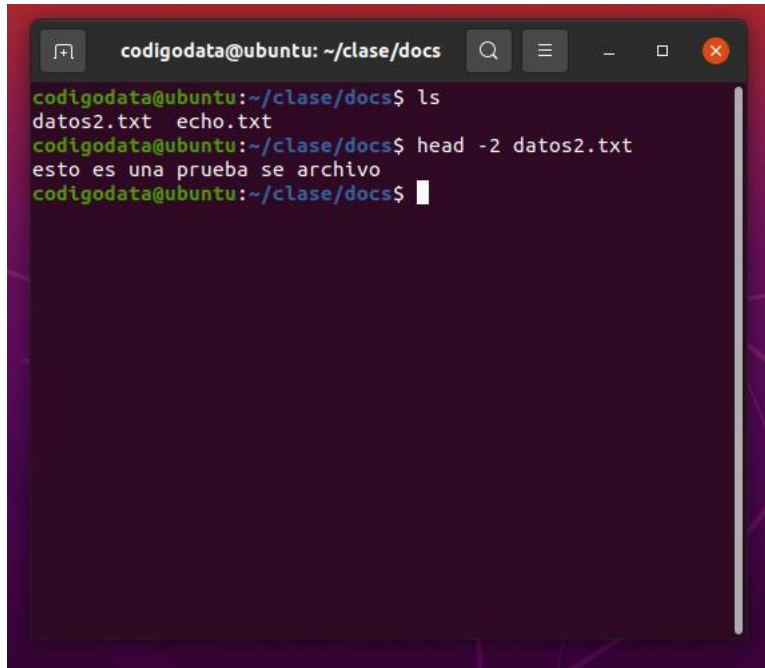


```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ ls
datos2.txt  echo.txt
codigodata@ubuntu:~/clase/docs$ head datos2.txt echo.txt
==> datos2.txt <==
esto es una prueba se archivo

==> echo.txt <==
first,last,middle,age,sex
codigodata@ubuntu:~/clase/docs$
```

Cabeza de Linux -n

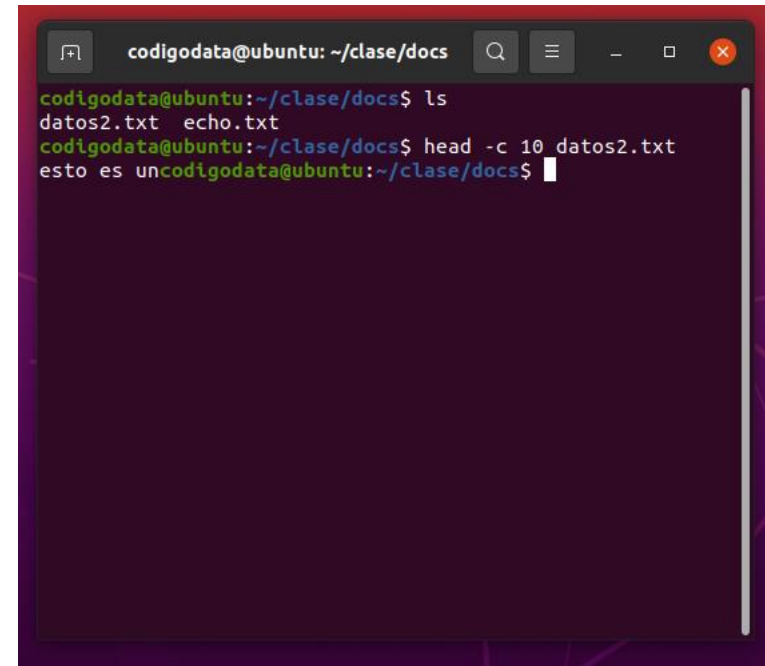
La opción 'head -n' muestra un número específico de líneas.

A terminal window titled 'codigodata@ubuntu: ~/clase/docs' with search, menu, and window control icons. It shows the execution of 'ls' listing 'datos2.txt' and 'echo.txt', followed by 'head -2 datos2.txt' which outputs the first two lines of the file: 'esto es una prueba' and 'se archivo'.

```
codigodata@ubuntu:~/clase/docs$ ls
datos2.txt  echo.txt
codigodata@ubuntu:~/clase/docs$ head -2 datos2.txt
esto es una prueba
se archivo
codigodata@ubuntu:~/clase/docs$
```

Cabeza de Linux -c

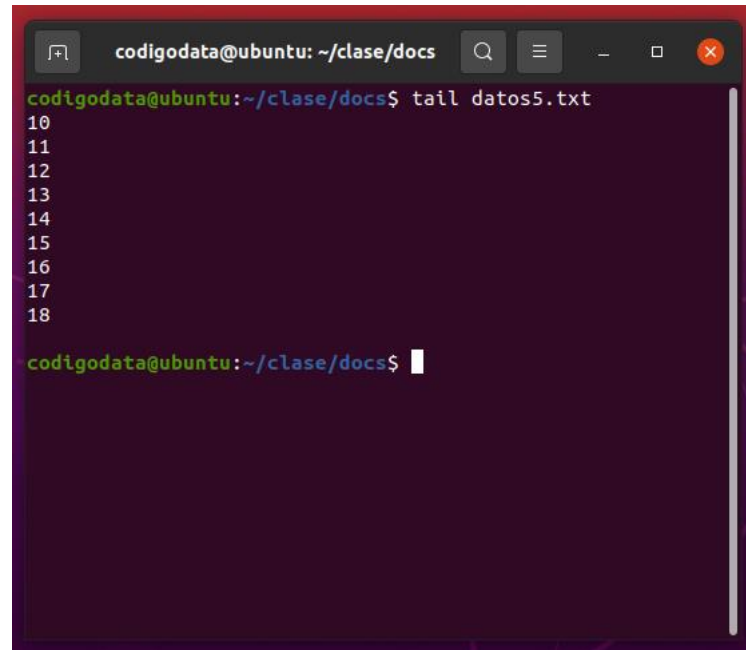
El comando 'head -c' cuenta el número de bytes de un archivo.

A terminal window titled 'codigodata@ubuntu: ~/clase/docs' with search, menu, and window control icons. It shows the execution of 'ls' listing 'datos2.txt' and 'echo.txt', followed by 'head -c 10 datos2.txt' which outputs the first 10 bytes of the file: 'esto es un'.

```
codigodata@ubuntu:~/clase/docs$ ls
datos2.txt  echo.txt
codigodata@ubuntu:~/clase/docs$ head -c 10 datos2.txt
esto es un
codigodata@ubuntu:~/clase/docs$
```

Comando de cola de Linux

El comando tail de Linux se usa para mostrar las últimas diez líneas de uno o más archivos. Su objetivo principal es leer el mensaje de error. Por defecto, muestra las últimas diez líneas de un archivo. Además, se utiliza para monitorear los cambios de archivos en tiempo real. Es un comando complementario del comando [principal](#) .

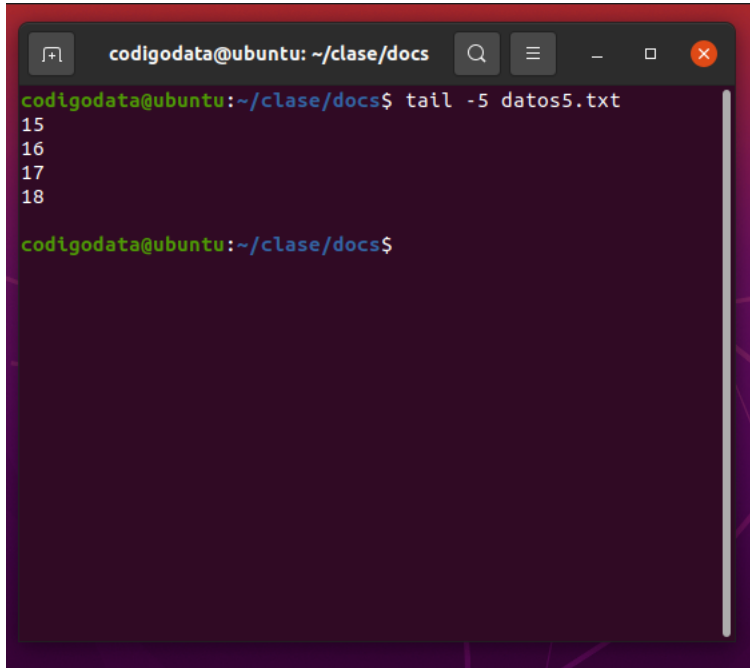
A terminal window titled 'codigodata@ubuntu: ~/clase/docs' with standard window controls. The prompt is 'codigodata@ubuntu:~/clase/docs\$'. The command 'tail datos5.txt' has been executed, resulting in a list of line numbers from 10 to 18. The prompt is now 'codigodata@ubuntu:~/clase/docs\$' with a cursor.

```
codigodata@ubuntu:~/clase/docs$ tail datos5.txt
10
11
12
13
14
15
16
17
18

codigodata@ubuntu:~/clase/docs$
```

Mostrar el número específico de líneas

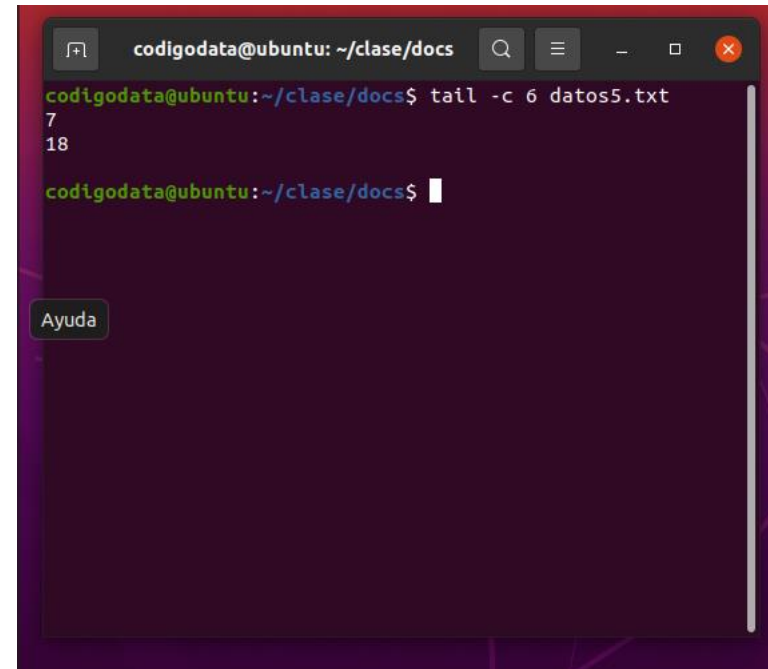
La opción '-n' muestra el número especificado de líneas. Para especificar el número de líneas, ejecute el comando de la siguiente manera:



```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ tail -5 datos5.txt
15
16
17
18
codigodata@ubuntu:~/clase/docs$
```

Mostrar el número especificado de bytes

La opción '-c' muestra el número especificado de bytes desde el último. Para mostrar el número especificado de bytes, ejecute el comando de la siguiente manera:

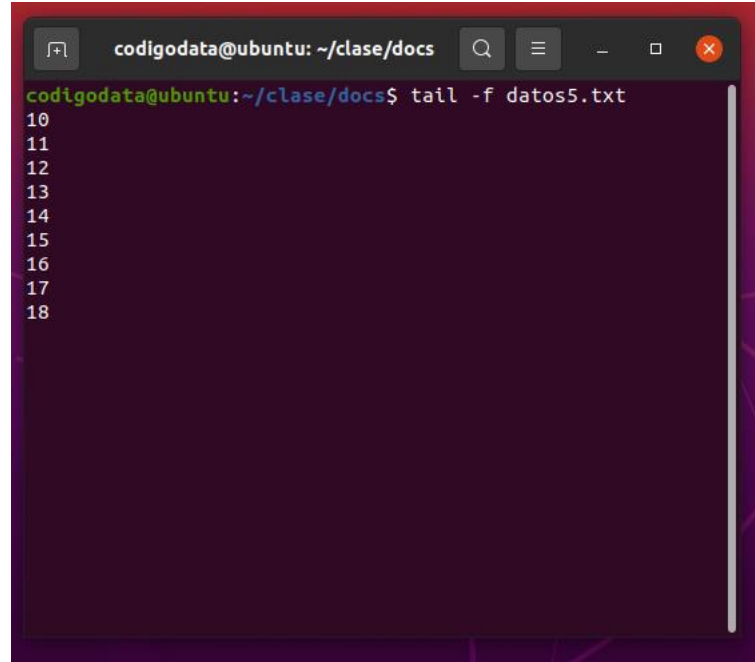


```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ tail -c 6 datos5.txt
7
18
codigodata@ubuntu:~/clase/docs$
```

Ayuda

Seguimiento de un archivo para los cambios

Para rastrear un archivo en busca de cambios, se usa la opción '-f'. Aquí, '-f' representa lo siguiente. Es útil para monitorear archivos de registro. Ejecute el siguiente comando:

A terminal window with a dark background and light-colored text. The window title bar shows 'codigodata@ubuntu: ~/clase/docs'. The prompt is 'codigodata@ubuntu:~/clase/docs\$'. The command 'tail -f datos5.txt' has been entered. Below the command, the numbers 10, 11, 12, 13, 14, 15, 16, 17, and 18 are listed vertically, indicating line numbers in the file being monitored.

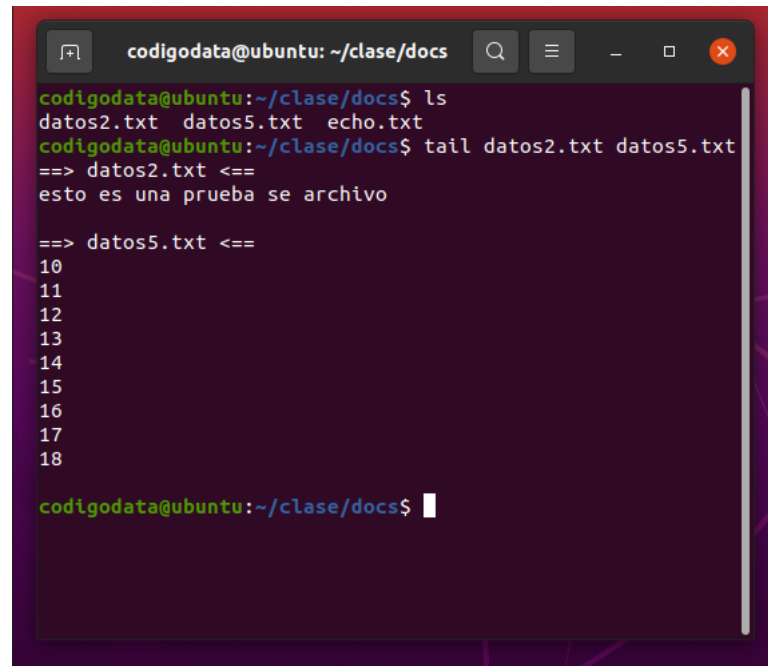
```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ tail -f datos5.txt
10
11
12
13
14
15
16
17
18
```

El comando anterior monitoreará el archivo 'num.txt'. Para salir del monitoreo, presione las teclas "CTRL+C". Considere la siguiente salida:

Mostrar varios archivos

Podemos mostrar varios archivos desde el último a la vez ejecutando el comando tail. Para mostrar varios archivos, proporcione los nombres de los archivos como entrada. Mostrará las últimas diez líneas de los archivos especificados.

Por ejemplo, tenemos otro archivo 'alfabeto.txt' que contiene todos los caracteres del alfabeto en una nueva línea. Para mostrar ambos archivos 'num.txt' y 'alphabet.txt', ejecute el comando de la siguiente manera:

A terminal window with a dark background and light text. The window title is 'codigodata@ubuntu: ~/clase/docs'. The user enters 'ls' and lists 'datos2.txt', 'datos5.txt', and 'echo.txt'. Then, they enter 'tail datos2.txt datos5.txt'. The output shows the last line of 'datos2.txt' and the last ten lines of 'datos5.txt'.

```
codigodata@ubuntu: ~/clase/docs$ ls
datos2.txt  datos5.txt  echo.txt
codigodata@ubuntu: ~/clase/docs$ tail datos2.txt datos5.txt
==> datos2.txt <==
esto es una prueba se archivo

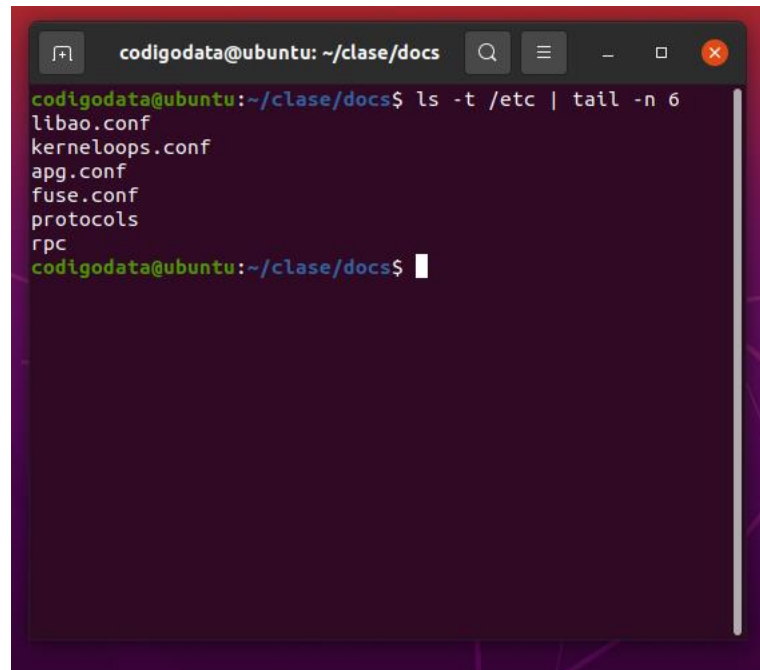
==> datos5.txt <==
10
11
12
13
14
15
16
17
18

codigodata@ubuntu: ~/clase/docs$
```

mostrará las últimas diez líneas de los archivos especificados.

Cola con otros comandos

El comando tail se puede usar con otros comandos. Se puede canalizar a otros comandos para filtrar la salida. Considere el siguiente comando:

A terminal window with a dark background and light-colored text. The window title is 'codigodata@ubuntu: ~/clase/docs'. The prompt is 'codigodata@ubuntu:~/clase/docs\$'. The command entered is 'ls -t /etc | tail -n 6'. The output shows six files: 'libao.conf', 'kerneloops.conf', 'apg.conf', 'fuse.conf', 'protocols', and 'rpc'. The prompt is now 'codigodata@ubuntu:~/clase/docs\$' with a cursor.

```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ ls -t /etc | tail -n 6
libao.conf
kerneloops.conf
apg.conf
fuse.conf
protocols
rpc
codigodata@ubuntu:~/clase/docs$
```

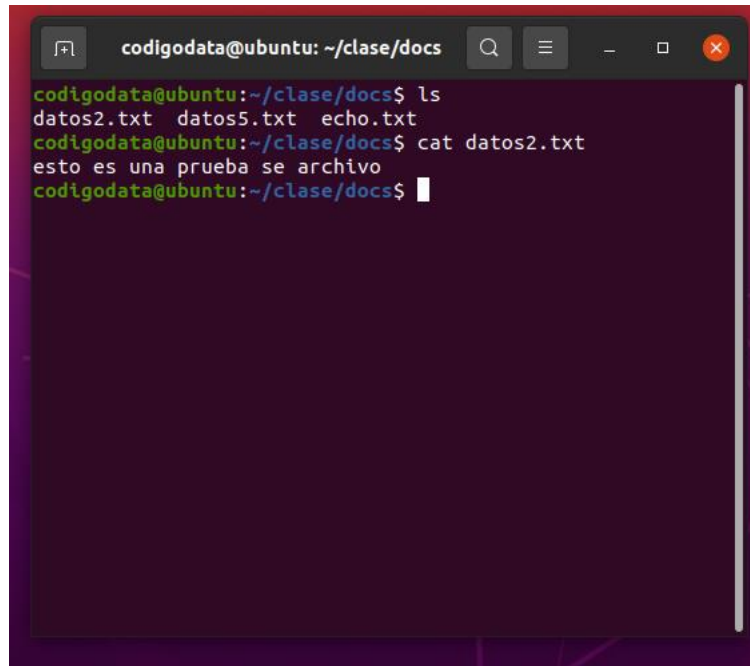
hemos canalizado el comando tail con el comando ls. Solo mostrará seis archivos o carpetas modificados hace más tiempo. Considere la siguiente salida:

Comando gato de Linux

El comando 'gato' es la herramienta más universal y poderosa. Se considera que es uno de los comandos más utilizados. Se puede utilizar para mostrar el contenido de un archivo, copiar contenido de un archivo a otro, concatenar el contenido de varios archivos, mostrar el número de línea, mostrar \$ al final de la línea, etc.

Comando cat de Linux: para mostrar el contenido del archivo

El comando 'cat' se puede utilizar para mostrar el contenido de un archivo.

A terminal window titled 'codigodata@ubuntu: ~/clase/docs' with standard window controls. The terminal shows a sequence of commands and their outputs: first, 'ls' lists files 'datos2.txt', 'datos5.txt', and 'echo.txt'; then, 'cat datos2.txt' displays the content 'esto es una prueba se archivo'. The prompt returns after each command.

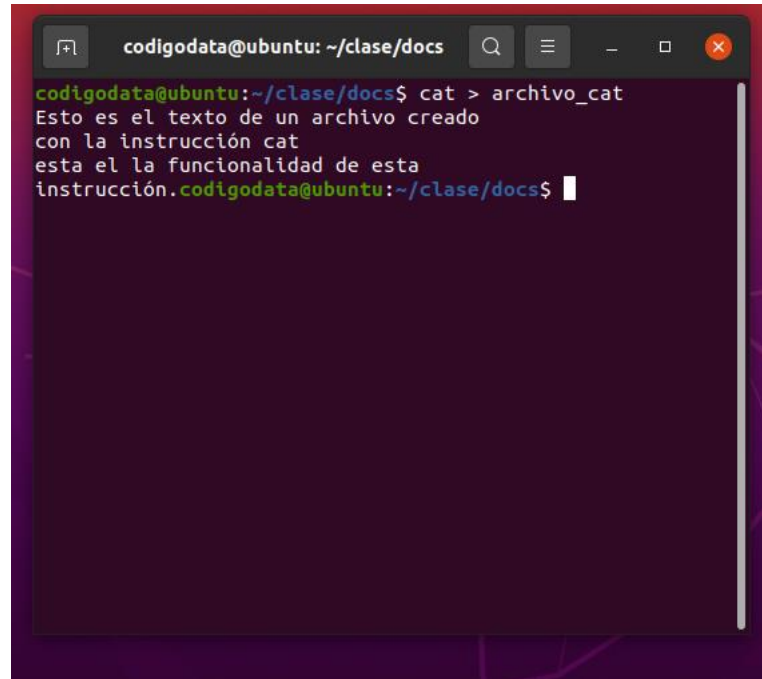
```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ ls
datos2.txt  datos5.txt  echo.txt
codigodata@ubuntu:~/clase/docs$ cat datos2.txt
esto es una prueba se archivo
codigodata@ubuntu:~/clase/docs$
```

Uso del comando cat de Linux

Opción	Función
<u>cat > [nombre de archivo]</u>	Para crear un archivo.
<u>cat [archivo antiguo] > [archivo nuevo]</u>	Para copiar contenido de un archivo anterior a uno nuevo.
<u>cat [archivo1 archivo2 y así sucesivamente] > [nuevo nombre de archivo]</u>	Para concatenar contenidos de múltiples archivos en uno.
<u>cat -n/cat -b [nombre de archivo]</u>	Para mostrar los números de línea.
<u>cat -e [nombre de archivo]</u>	Para mostrar el carácter \$ al final de cada línea.
<u>cat [nombre de archivo] <<EOF</u>	Se utiliza como marcador de final de página.

Comando cat de Linux (para crear un archivo)

El comando 'cat' se puede usar para crear un nuevo archivo con un signo mayor que (>).

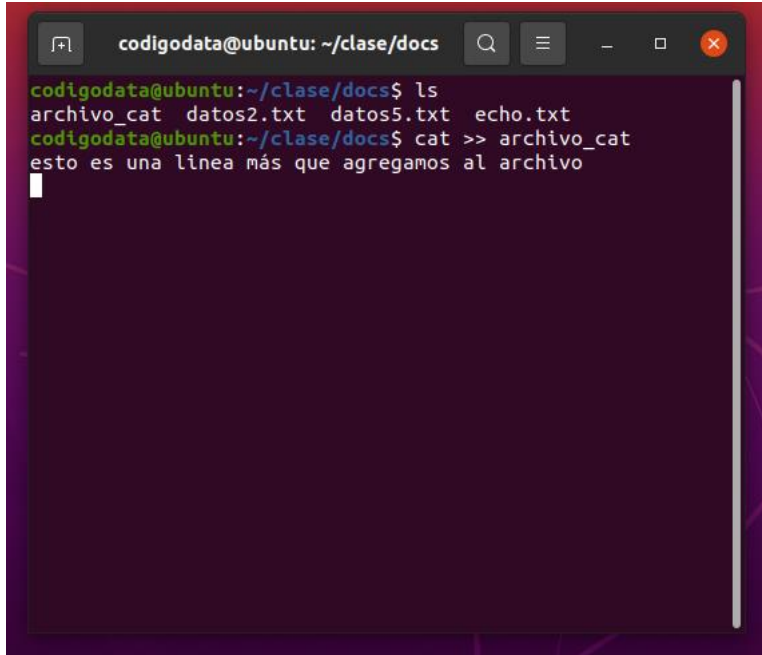
A terminal window titled 'codigodata@ubuntu: ~/clase/docs' with search, menu, and window control icons. The prompt is 'codigodata@ubuntu:~/clase/docs\$'. The command 'cat > archivo_cat' has been entered. The output shows four lines of text: 'Esto es el texto de un archivo creado', 'con la instrucción cat', 'esta el la funcionalidad de esta', and 'instrucción.'. The prompt is now 'codigodata@ubuntu:~/clase/docs\$' with a cursor at the end.

```
codigodata@ubuntu:~/clase/docs$ cat > archivo_cat
Esto es el texto de un archivo creado
con la instrucción cat
esta el la funcionalidad de esta
instrucción.codigodata@ubuntu:~/clase/docs$
```

Presione 'enter' después de cada línea y será dirigido a la siguiente línea. Para guardar su archivo, vaya a la siguiente línea, presione '**ctrl+d**' y su archivo se guardará.

Para agregar el contenido de un archivo

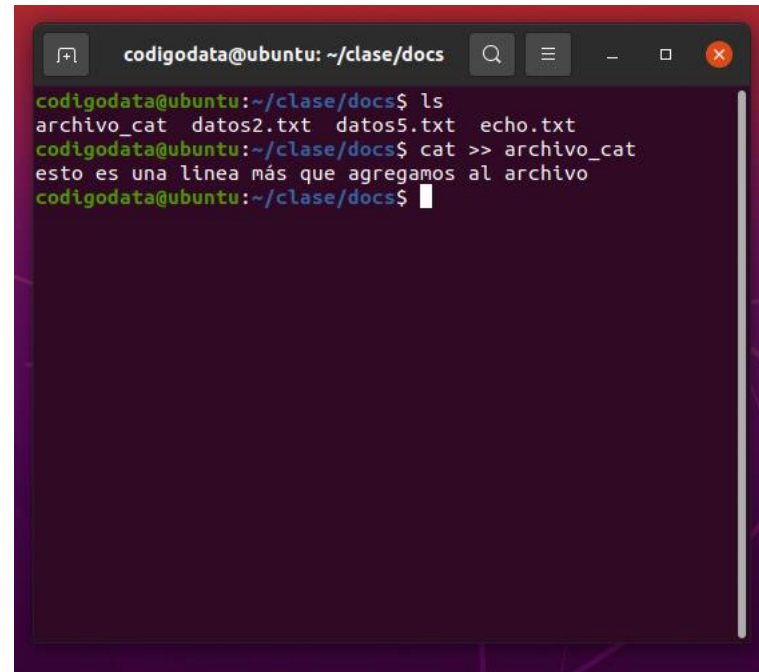
El comando 'gato' con el signo doble mayor que (>>) agrega (agrega algo en el último de un archivo) algo en su archivo ya existente.



```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ ls
archivo_cat  datos2.txt  datos5.txt  echo.txt
codigodata@ubuntu:~/clase/docs$ cat >> archivo_cat
esto es una línea más que agregamos al archivo

```

Mire la instantánea anterior, se agrega una nueva línea al final en el archivo 'archivo_cat'. Después de pasar el comando "**cat >> archivo_cat**", escriba las líneas que desee agregar. Para guardar el archivo presione '**ctrl + d**'.

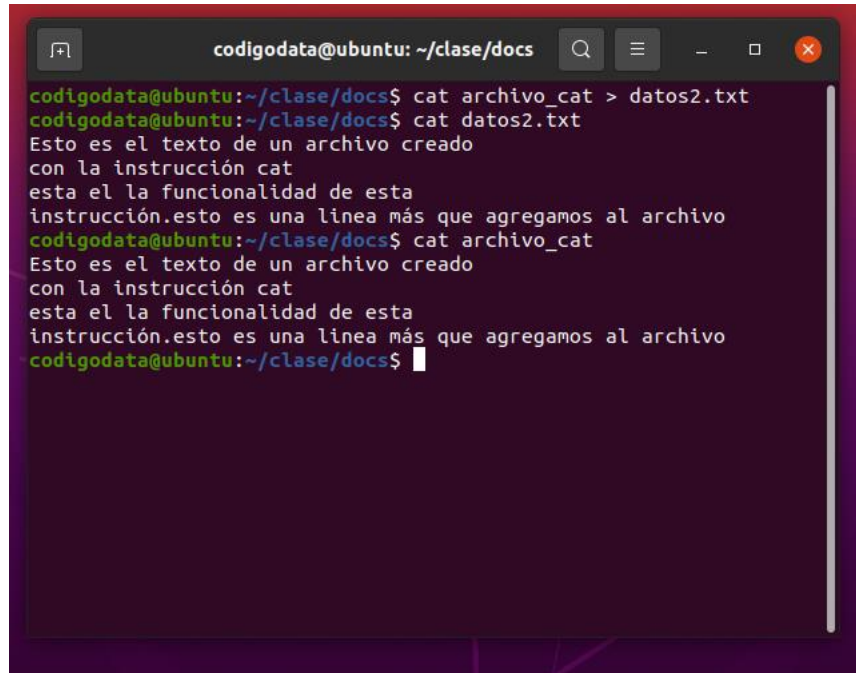


```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ ls
archivo_cat  datos2.txt  datos5.txt  echo.txt
codigodata@ubuntu:~/clase/docs$ cat >> archivo_cat
esto es una línea más que agregamos al archivo
codigodata@ubuntu:~/clase/docs$

```

Comando cat de Linux (para copiar archivo)

El comando 'cat' se puede usar para copiar el contenido de un archivo en otro archivo.

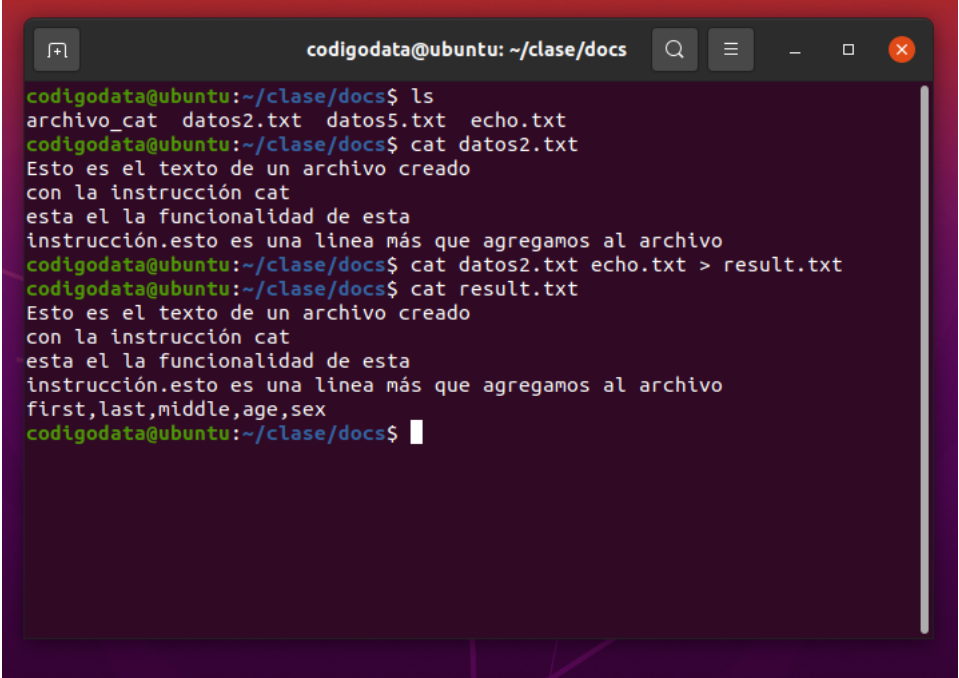
A terminal window with a dark background and light green text. The window title is 'codigodata@ubuntu: ~/clase/docs'. The terminal shows the following commands and output:

```
codigodata@ubuntu:~/clase/docs$ cat archivo_cat > datos2.txt
codigodata@ubuntu:~/clase/docs$ cat datos2.txt
Esto es el texto de un archivo creado
con la instrucción cat
esta el la funcionalidad de esta
instrucción.esto es una línea más que agregamos al archivo
codigodata@ubuntu:~/clase/docs$ cat archivo_cat
Esto es el texto de un archivo creado
con la instrucción cat
esta el la funcionalidad de esta
instrucción.esto es una línea más que agregamos al archivo
codigodata@ubuntu:~/clase/docs$
```

En la instantánea anterior, hemos copiado el contenido del archivo '**archivo_cat**' en el archivo '**datos2.txt**' con el comando "**cat archivo_cat > datos2.txt**".

Comando cat de Linux (para concatenar archivos)

El comando 'cat' se puede utilizar para concatenar el contenido de varios archivos en un único archivo nuevo.

A terminal window titled 'codigodata@ubuntu: ~/clase/docs' with search, menu, and window control icons. The terminal shows the following commands and output:

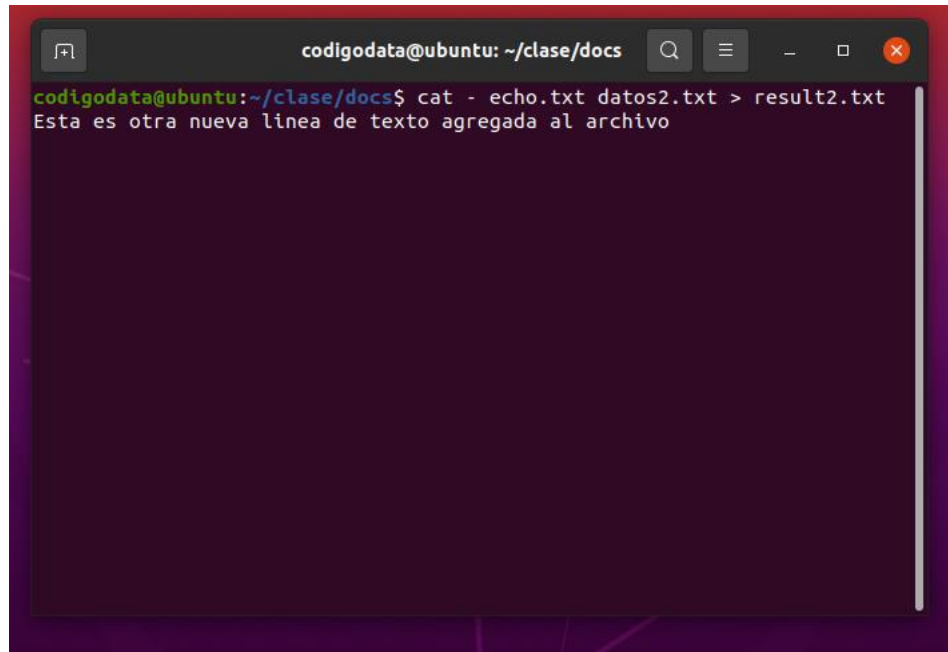
```
codigodata@ubuntu:~/clase/docs$ ls
archivo_cat  datos2.txt  datos5.txt  echo.txt
codigodata@ubuntu:~/clase/docs$ cat datos2.txt
Esto es el texto de un archivo creado
con la instrucción cat
esta el la funcionalidad de esta
instrucción.esto es una línea más que agregamos al archivo
codigodata@ubuntu:~/clase/docs$ cat datos2.txt echo.txt > result.txt
codigodata@ubuntu:~/clase/docs$ cat result.txt
Esto es el texto de un archivo creado
con la instrucción cat
esta el la funcionalidad de esta
instrucción.esto es una línea más que agregamos al archivo
first,last,middle,age,sex
codigodata@ubuntu:~/clase/docs$
```

hemos combinado tres archivos “**datos2.txt**, **echo.txt** en un solo archivo “**result.txt**” con el comando “**cat datos2.txt echo.txt > result.txt**” .

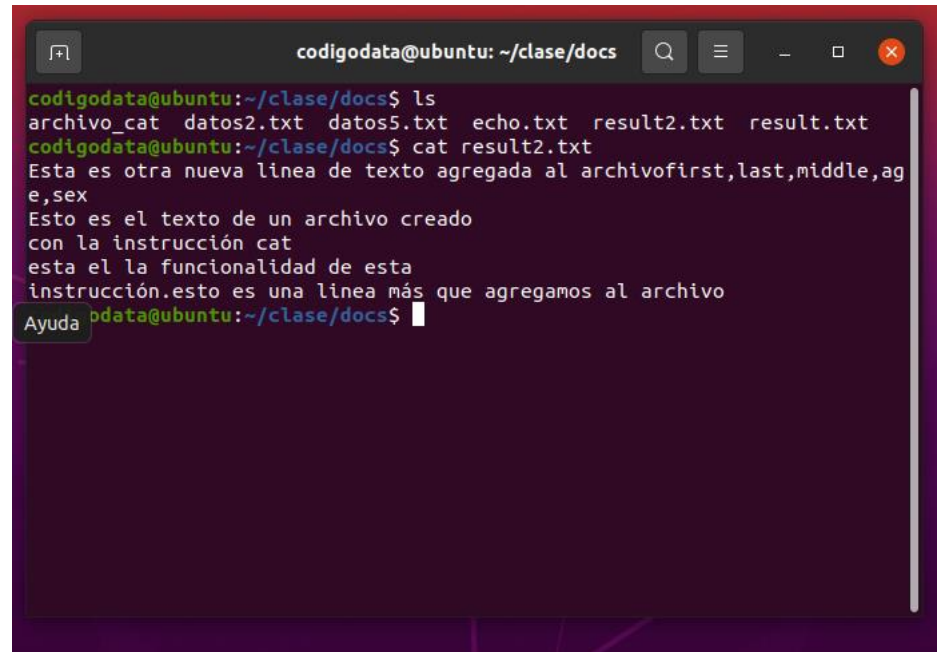
Observe el contenido de tres archivos separados y luego el contenido de un nuevo archivo concatenado que es “**result.txt**” .

Para insertar una nueva línea

Se insertará una nueva línea al concatenar varios archivos usando un guión (-).



```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ cat - echo.txt datos2.txt > result2.txt
Esta es otra nueva línea de texto agregada al archivo
```

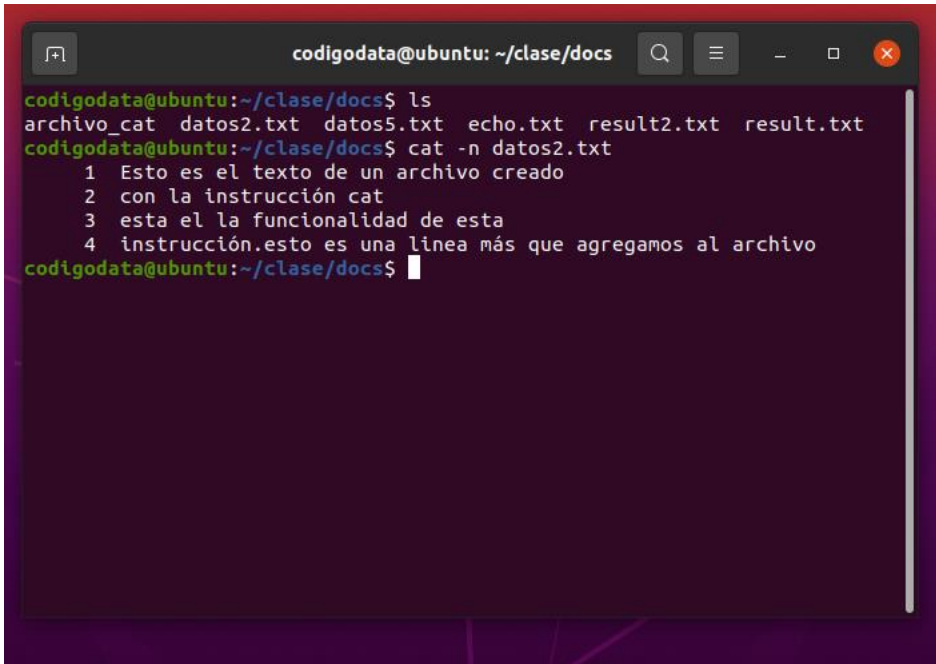


```
codigodata@ubuntu:~/clase/docs$ ls
archivo_cat datos2.txt datos5.txt echo.txt result2.txt result.txt
codigodata@ubuntu:~/clase/docs$ cat result2.txt
Esta es otra nueva línea de texto agregada al archivofirst,last,middle,ag
e,sex
Esto es el texto de un archivo creado
con la instrucción cat
esta el la funcionalidad de esta
instrucción.esto es una línea más que agregamos al archivo
Ayuda odigodata@ubuntu:~/clase/docs$
```

En la instantánea anterior, hemos insertado una nueva línea al principio al concatenar para salir después de la escritura pulsar las teclas ctrl + d

Comando Linux cat -n (para mostrar los números de línea)

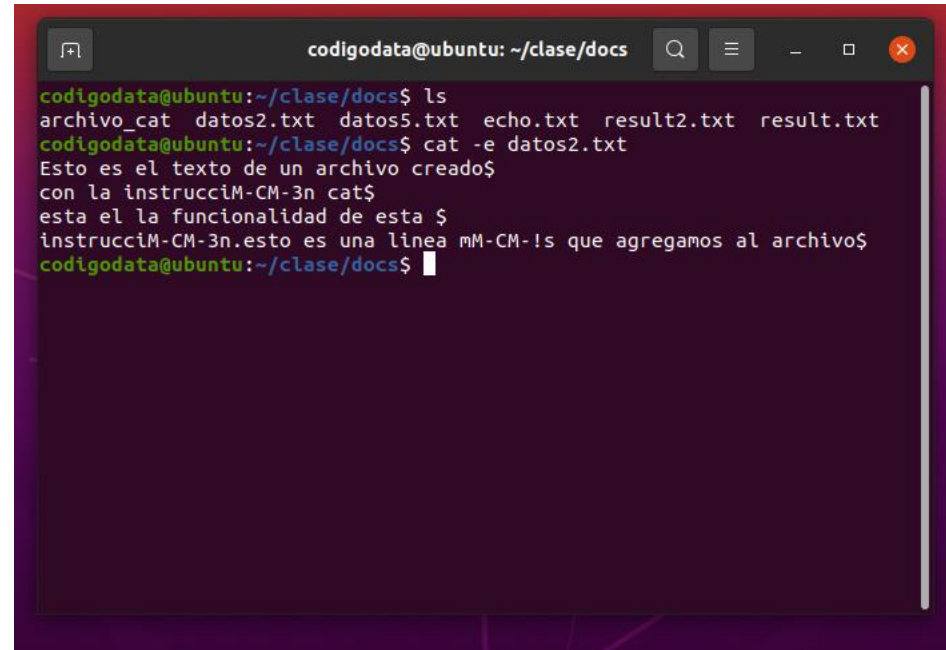
La opción 'cat -n' muestra números de línea delante de cada línea en un archivo.

A terminal window titled 'codigodata@ubuntu: ~/clase/docs'. The user runs 'ls' showing files: 'archivo_cat', 'datos2.txt', 'datos5.txt', 'echo.txt', 'result2.txt', and 'result.txt'. Then they run 'cat -n datos2.txt', which displays the contents of 'datos2.txt' with line numbers 1 through 4. The prompt returns to the shell.

```
codigodata@ubuntu:~/clase/docs$ ls
archivo_cat  datos2.txt  datos5.txt  echo.txt    result2.txt result.txt
codigodata@ubuntu:~/clase/docs$ cat -n datos2.txt
 1 Esto es el texto de un archivo creado
 2 con la instrucción cat
 3 esta el la funcionalidad de esta
 4 instrucción.esto es una línea más que agregamos al archivo
codigodata@ubuntu:~/clase/docs$
```

Comando Linux cat -e (para mostrar \$)

La opción 'cat -e' muestra un signo '\$' al final de cada línea.

A terminal window titled 'codigodata@ubuntu: ~/clase/docs'. The user runs 'ls' showing the same files as the first screenshot. Then they run 'cat -e datos2.txt', which displays the contents of 'datos2.txt' with a dollar sign at the end of each line. The prompt returns to the shell.

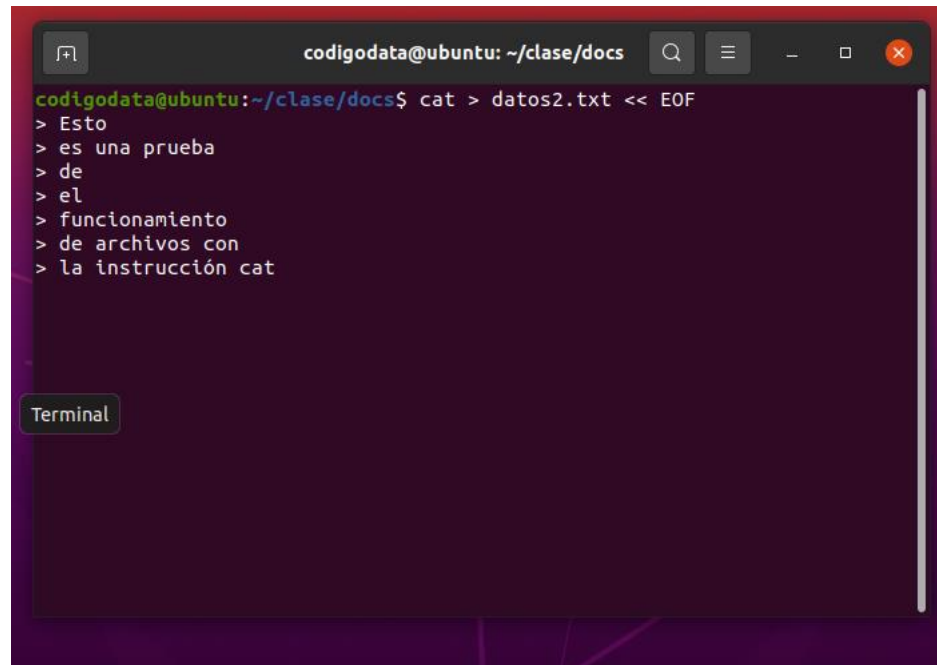
```
codigodata@ubuntu:~/clase/docs$ ls
archivo_cat  datos2.txt  datos5.txt  echo.txt    result2.txt result.txt
codigodata@ubuntu:~/clase/docs$ cat -e datos2.txt
Esto es el texto de un archivo creado$
con la instrucción cat$
esta el la funcionalidad de esta $
instrucción.esto es una línea más que agregamos al archivo$
codigodata@ubuntu:~/clase/docs$
```

Comando cat de Linux (como marcador final)

La opción 'cat << EOF ' muestra un marcador final al final de un archivo. Se llama **aquí directiva** y el contenido del archivo se guardará en el marcador final dado.

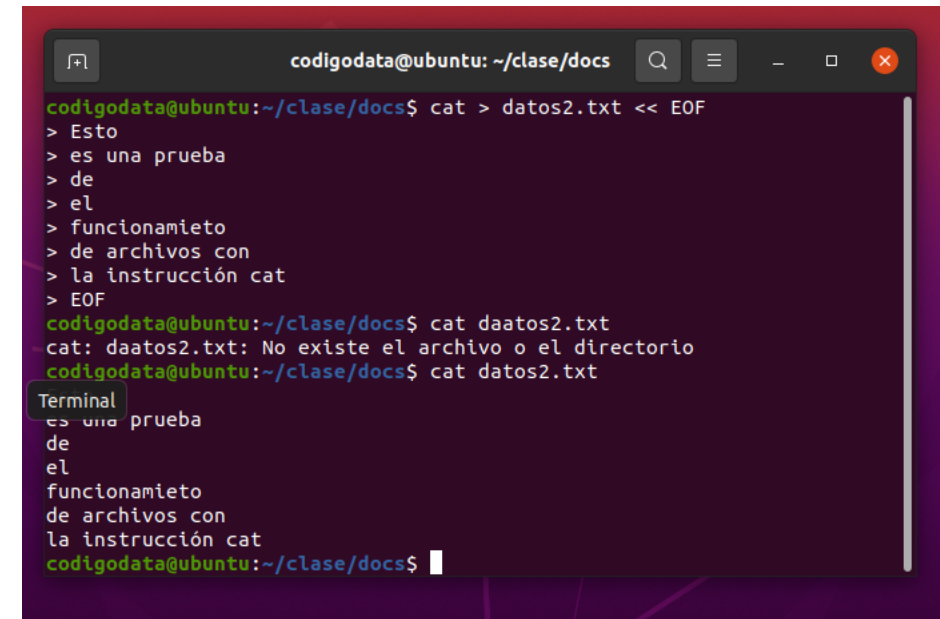
El archivo también se puede guardar con la ayuda de las teclas 'ctrl + d' . Funciona como el marcador final.

Nota: cualquier palabra que no sea 'EOF' se puede utilizar para el marcador final.



```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ cat > datos2.txt << EOF
> Esto
> es una prueba
> de
> el
> funcionamiento
> de archivos con
> la instrucción cat
```

Terminal

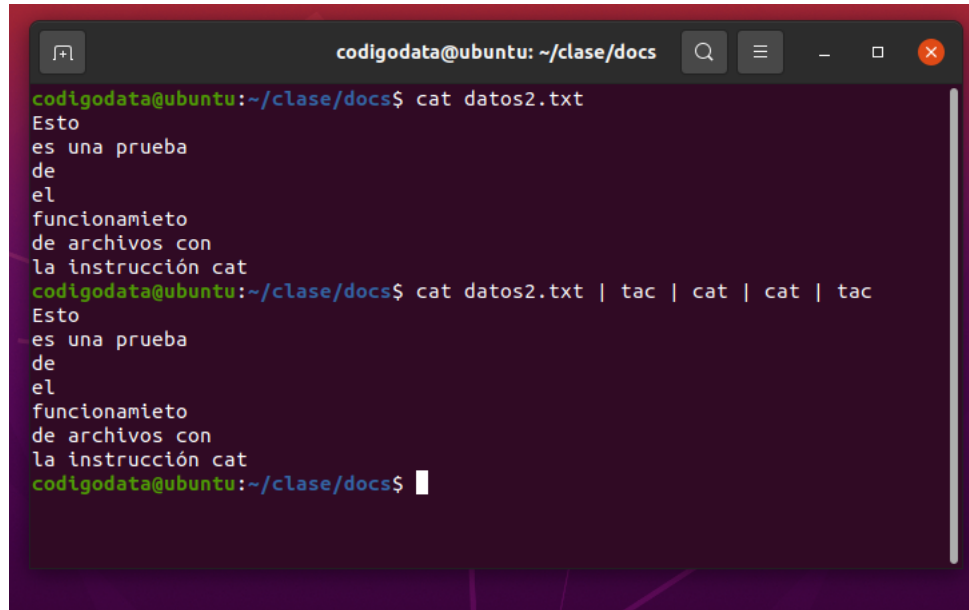


```
codigodata@ubuntu: ~/clase/docs
codigodata@ubuntu:~/clase/docs$ cat > datos2.txt << EOF
> Esto
> es una prueba
> de
> el
> funcionamiento
> de archivos con
> la instrucción cat
> EOF
codigodata@ubuntu:~/clase/docs$ cat daatos2.txt
cat: daatos2.txt: No existe el archivo o el directorio
codigodata@ubuntu:~/clase/docs$ cat datos2.txt
es una prueba
de
el
funcionamiento
de archivos con
la instrucción cat
codigodata@ubuntu:~/clase/docs$
```

Terminal

Filtros cat de Linux

Cuando el comando cat se usa dentro de las tuberías, no hace nada excepto mover stdin a stout.

A terminal window titled 'codigodata@ubuntu: ~/clase/docs' with search, menu, and window control icons. It shows two commands being executed. The first command is 'cat datos2.txt', which outputs five lines of text. The second command is 'cat datos2.txt | tac | cat | cat | tac', which takes the output of the first command, reverses the line order with 'tac', and then passes it through two 'cat' commands before reversing it back with a final 'tac'. The output of the second command is identical to the first.

```
codigodata@ubuntu:~/clase/docs$ cat datos2.txt
Esto
es una prueba
de
el
funcionamiento
de archivos con
la instrucción cat
codigodata@ubuntu:~/clase/docs$ cat datos2.txt | tac | cat | cat | tac
Esto
es una prueba
de
el
funcionamiento
de archivos con
la instrucción cat
codigodata@ubuntu:~/clase/docs$
```

<https://www.javatpoint.com/linux-tac>