

## CIENCIA DE DATOS:

# APRENDE LOS FUNDAMENTOS DE MANERA PRÁCTICA



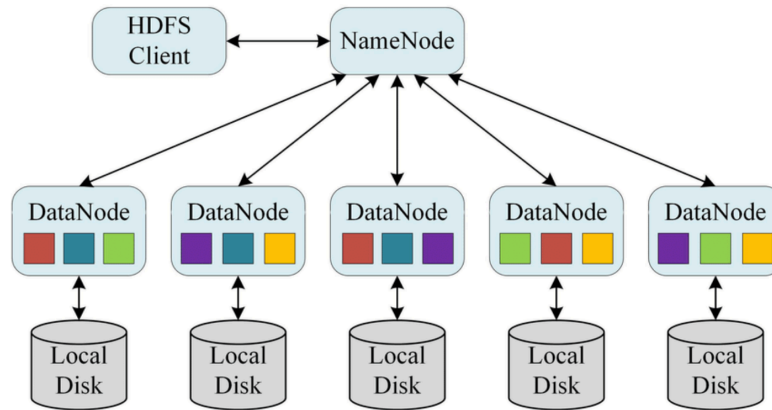
## LABORATORIO 03 SISTEMA OPERATIVO LINUX

**Juan Chipoco**  
mindquasar@gmail.com

# ÍNDICE

OBJETIVO.....	4
SISTEMA OPERATIVO LINUX .....	5
COMANDOS LINUX.....	15
COMANDOS HDFS.....	21

## OBJETIVO



El objetivo de este laboratorio es profundizar en el sistema de archivos Linux y sus comandos. Proporciona conceptos básicos y avanzados de Linux.

Linux es un sistema operativo de código abierto. Es como Windows, Mac, Android, etc.

Unix es también un sistema operativo como Linux. Es un sistema operativo comercial. Consta de tres partes: Kernel, Shell y Programas. La mayoría de los comandos de Unix y Linux son de naturaleza similar.

Nuestro tutorial de Linux incluye todos los temas del sistema operativo Linux, como comandos de Linux, directorios, archivos, páginas de manual, contenidos de archivos, permisos de archivos, shells, editor VI, etc. También se proporcionan preguntas de entrevistas de Linux para ayudarlo a comprender mejor el sistema operativo Linux.

## SISTEMA OPERATIVO LINUX

### *¿Qué es el sistema operativo Linux?*

Linux es un sistema operativo de código abierto como otros sistemas operativos como Microsoft Windows, Apple Mac OS, iOS, Google Android, etc. Un sistema operativo es un software que permite la comunicación entre el hardware y el software de la computadora. Transmite la entrada para que el procesador la procese y lleva la salida al hardware para mostrarla. Esta es la función básica de un sistema operativo. Aunque realiza muchas otras tareas importantes, no hablemos de eso.

Linux nos rodea desde mediados de los 90. Se puede utilizar desde relojes de pulsera hasta superordenadores. Está en todas partes en nuestros teléfonos, computadoras portátiles, PC, automóviles e incluso en los refrigeradores. Es muy famoso entre los desarrolladores y usuarios normales de computadoras.

### *Evolución del sistema operativo Linux*

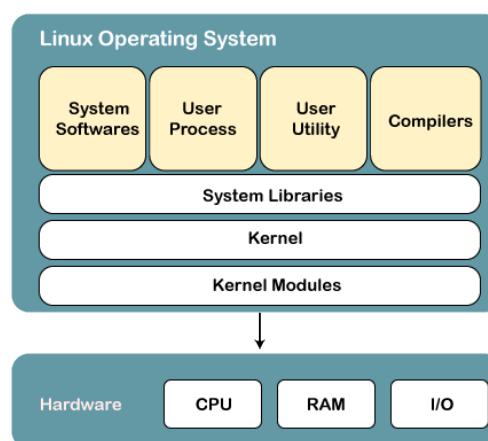
El sistema operativo Linux fue desarrollado por Linus Torvalds en 1991, que surgió como una idea para mejorar el sistema operativo UNIX. Sugirió mejoras pero fue rechazado por los diseñadores de UNIX. Por lo tanto, pensó en lanzar un sistema operativo, diseñado de manera que pudiera ser modificado por sus usuarios.

Hoy en día, Linux es el sistema operativo de más rápido crecimiento. Se utiliza desde teléfonos hasta supercomputadoras por casi todos los principales dispositivos de hardware.

### *Estructura del sistema operativo Linux*

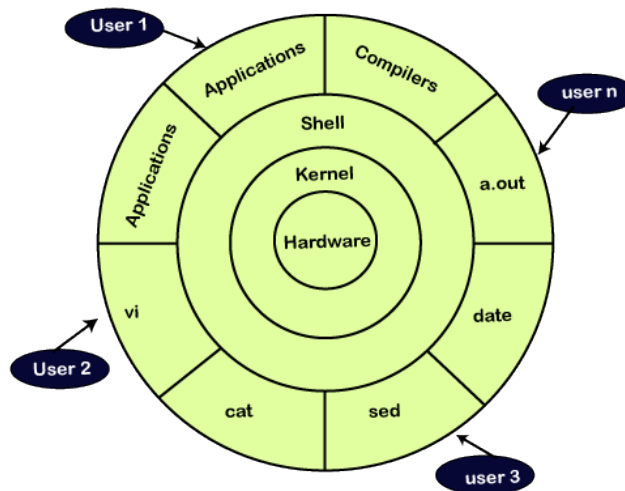
Un sistema operativo es una colección de software, cada uno diseñado para una función específica.

El sistema operativo Linux tiene los siguientes componentes:



## 1) Kernel

El kernel de Linux es la parte central del sistema operativo. Establece la comunicación entre los dispositivos y el software. Además, gestiona los recursos del sistema. Tiene cuatro responsabilidades:



- Administración de dispositivos: un sistema tiene muchos dispositivos conectados, como CPU, un dispositivo de memoria, tarjetas de sonido, tarjetas gráficas, etc. Un kernel almacena todos los datos relacionados con todos los dispositivos en el controlador del dispositivo (sin este kernel no podrá para controlar los dispositivos). Por lo tanto, kernel sabe qué puede hacer un dispositivo y cómo manipularlo para obtener el mejor rendimiento. También gestiona la comunicación entre todos los dispositivos. El núcleo tiene ciertas reglas que deben seguir todos los dispositivos.
- Gestión de la memoria: Otra función que tiene que gestionar el kernel es la gestión de la memoria. El kernel realiza un seguimiento de la memoria utilizada y no utilizada y se asegura de que los procesos no manipulen los datos entre sí utilizando direcciones de memoria virtual.
- Gestión de procesos: en el proceso, el kernel de gestión asigna suficiente tiempo y da prioridades a los procesos antes de manejar la CPU a otros procesos. También se ocupa de la información de seguridad y propiedad.
- Manejo de llamadas al sistema: el manejo de llamadas al sistema significa que un programador puede escribir una consulta o pedirle al núcleo que realice una tarea.

## 2) Bibliotecas del sistema

Las bibliotecas del sistema son programas especiales que ayudan a acceder a las funciones del kernel. Un núcleo tiene que activarse para realizar una tarea, y esta activación la realizan las aplicaciones. Pero las aplicaciones deben saber cómo realizar una llamada al sistema porque cada kernel tiene un conjunto diferente de llamadas al sistema. Los programadores han desarrollado una biblioteca estándar de procedimientos para comunicarse con el núcleo. Cada sistema operativo admite estos estándares y luego estos se transfieren a las llamadas del sistema para ese sistema operativo. Ejemplo Standard POSIX.

La biblioteca del sistema más conocida para Linux es Glibc (biblioteca GNU C).

## 3) Herramientas del sistema

El sistema operativo Linux tiene un conjunto de herramientas de utilidad, que generalmente son comandos simples. Es un software que el proyecto GNU ha escrito y publicado bajo su licencia de código abierto para que el software esté disponible gratuitamente para todos.

Con la ayuda de los comandos, puede acceder a sus archivos, editar y manipular datos en sus directorios o archivos, cambiar la ubicación de los archivos o cualquier cosa.

## 4) Herramientas de desarrollo

Con los tres componentes anteriores, su sistema operativo se está ejecutando y funcionando. Pero para actualizar su sistema, tiene herramientas y bibliotecas adicionales. Estas herramientas y bibliotecas adicionales están escritas por los programadores y se denominan cadena de herramientas. Una cadena de herramientas es una herramienta de desarrollo vital utilizada por los desarrolladores para producir una aplicación funcional.

## 5) Herramientas de usuario final

Estas herramientas finales hacen que un sistema sea único para un usuario. Las herramientas finales no son necesarias para el sistema operativo, pero son necesarias para un usuario.

Algunos ejemplos de herramientas finales son las herramientas de diseño gráfico, las suites ofimáticas, los navegadores, los reproductores multimedia, etc. Por ejemplo GIMP, Libre Office, Firefox, Opera, Chrome.

## ¿Por qué usar Linux?

Esta es una de las preguntas más frecuentes sobre los sistemas Linux. ¿Por qué usamos un sistema operativo diferente y un poco complejo, si tenemos un sistema operativo simple como Windows? Así que hay varias características de Linux sistemas que lo hacen completamente diferente y uno de los sistemas operativos más utilizados. Linux puede ser un sistema operativo perfecto si desea deshacerse de virus, malware, ralentizaciones,

bloqueos, reparaciones costosas y muchos más. Además, proporciona varias ventajas sobre otros sistemas operativos, y no tenemos que pagar por ello.



### *Sistema operativo gratuito y de código abierto*

La mayoría de los sistemas operativos vienen en un formato compilado, lo que significa que el código fuente principal se ha ejecutado a través de un programa llamado compilador que traduce el código fuente a un lenguaje conocido por la computadora. Modificar este código compilado es un trabajo duro.

Por otro lado, el código abierto es completamente diferente. El código fuente se incluye con la versión compilada y permite la modificación por cualquier persona que tenga algún conocimiento. Nos da la libertad de ejecutar el programa, la libertad de cambiar el código de acuerdo con nuestro uso, la libertad de redistribuir sus copias y la libertad de distribuir copias, que son modificadas por nosotros.

En resumen, Linux es un sistema operativo que es "para la gente, por la gente".

Y podemos sumergirnos en Linux sin pagar ningún costo. Podemos instalarlo en Múltiples máquinas sin pagar ningún costo.

### *Es seguro*

Linux admite varias opciones de seguridad que lo salvarán de virus, malware, ralentizaciones y bloqueos. Además, mantendrá sus datos protegidos. Su característica de seguridad es la razón principal por la que es la opción más favorable para los desarrolladores. No es completamente seguro, pero es menos vulnerable que otros. Cada aplicación debe ser autorizada por el usuario administrador. El virus no puede ejecutarse hasta que el administrador proporcione la contraseña de acceso. Los sistemas Linux no requieren ningún programa antivirus.

### *Elección favorable de los desarrolladores*

Linux es adecuado para los desarrolladores, ya que admite casi todos los lenguajes de programación más utilizados, como C/C++

- , Java
- , Python
- , rubí on rails

, y más. Además, facilita con una amplia gama de aplicaciones útiles para el desarrollo (IDE).

Los desarrolladores encuentran que la terminal de Linux es mucho mejor que la línea de comandos de Windows, por lo que prefieren la terminal a la línea de comandos de Windows. El administrador de paquetes en el sistema Linux ayuda a los programadores a comprender cómo se hacen las cosas. Secuencias de comandos bash es también una característica funcional para los programadores. Además, el soporte SSH ayuda a administrar los servidores rápidamente.

### *Un sistema operativo flexible*

Linux es un sistema operativo flexible, ya que puede usarse para aplicaciones de escritorio, sistemas integrados y aplicaciones de servidor. Se puede utilizar desde relojes de pulsera hasta superordenadores. Esta en todas partes en nuestros teléfonos, computadoras portátiles, PC, automóviles e incluso en los refrigeradores. Además, admite varias opciones de personalización.

### *Distribuciones de Linux*

Muchas agencias modificaron el sistema operativo Linux y hacen sus distribuciones de Linux. Hay muchas distribuciones de Linux disponibles en el mercado. Proporciona un sabor diferente del sistema operativo Linux a los usuarios. Podemos elegir cualquier distribución según nuestras necesidades. Algunas distribuciones populares son Ubuntu, Fedora, Debian, Linux Mint, Arch Linux y muchas más.

Para los principiantes, Ubuntu y Linux Mint se consideran útiles y, para los desarrolladores expertos, Debian y Fedora serían una buena opción. Para obtener una lista de distribuciones, visite [Distribuciones de Linux](#).

### *¿Cómo funciona Linux?*

Linux es similar a UNIX

Sistema operativo, pero es compatible con una variedad de dispositivos de hardware, desde teléfonos hasta supercomputadoras. Cada sistema operativo basado en Linux tiene el kernel de Linux y un conjunto de paquetes de software para administrar los recursos de hardware.

Además, el sistema operativo Linux incluye algunas herramientas básicas de GNU para proporcionar una forma de administrar los recursos del kernel, instalar software, configurar la configuración de seguridad y el rendimiento, y mucho más. Todas estas herramientas se empaquetan juntas para crear un sistema operativo funcional.

### *¿Cómo usar Linux?*

Podemos utilizar Linux tanto a través de una interfaz de usuario interactiva como desde la terminal (Command Line Interface). Las diferentes distribuciones tienen una interfaz de usuario ligeramente diferente, pero casi todos los comandos tendrán el mismo comportamiento para todas las distribuciones. Para ejecutar Linux desde la terminal,



presione las teclas "CTRL+ALT+T". Y, para explorar su funcionalidad, presione el botón de la aplicación que se encuentra en la esquina inferior izquierda de su escritorio.

### *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.

### **Linux Directory Commands**

Directory Command	Description
<b>pwd</b>	The pwd command stands for (print working directory). It displays the current working location or directory of the user. It displays the whole working path starting with /. It is a built-in command.
<b>ls</b>	The ls command is used to show the list of a folder. It will list out all the files in the directed folder.
<b>cd</b>	The cd command stands for (change directory). It is used to change to the directory you want to work from the present directory.
<b>mkdir</b>	With mkdir command you can create your own directory.
<b>rmdir</b>	The rmdir command is used to remove a directory from your system.

### *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. Entendámoslo a través de un ejemplo.

Tipos de archivos:

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

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

Archivos especiales

Archivo de bloque (b)

Archivo de dispositivo de caracteres (c)

Archivo de canalización con nombre (p)

Archivo de enlace simbólico (l)

Archivo (s) de socket

## Linux File Commands

Command	Description
file	Determines file type.
touch	Used to create a file.
rm	To remove a file.
cp	To copy a file.
mv	To rename or to move a file.
rename	To rename file.

## Linux man

El "man" 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.

### *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:

Commands	Function
<code>head</code>	It displays the beginning of a file.
<code>tail</code>	It displays the last last part of a file.
<code>cat</code>	This command is versatile and multi worker.
<code>tac</code>	Opposite of cat.
<code>more</code>	Command line diaplays contents in pager form that is either in more format.
<code>less</code>	Command line diaplays contents in pager form that is either in less format.

### Estándar de jerarquía del sistema de archivos de Linux (FHS)

El estándar de jerarquía del sistema de archivos describe la estructura de directorios y su contenido en Unix y un sistema operativo similar a Unix. Explica dónde se deben ubicar los archivos y directorios y qué debe contener.

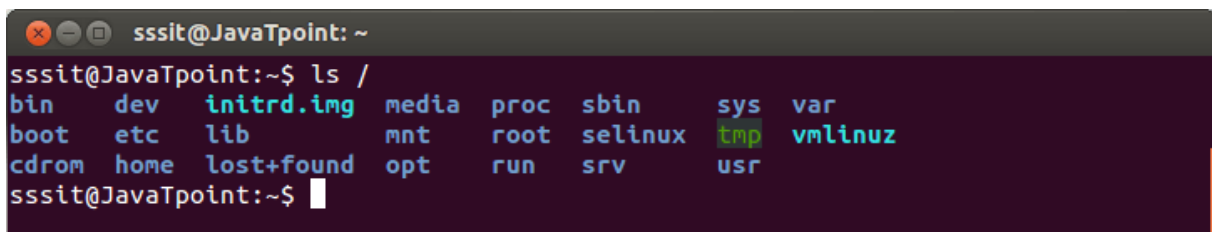
Su versión actual es 3.0 lanzada el 3 de junio de 2015 y es mantenida por Free Standards Group. En Unix como sistema operativo todo se considera como un archivo.

Solo las distribuciones de Linux siguen el FHS y eso también parcialmente. Porque cada distribución tiene su propia política, por lo que puede notar algunas diferencias en la estructura del árbol de directorios de las diferentes distribuciones.

Si desea obtener información sobre el FHS de su sistema, ingrese el comando `man hier`. Mostrará la estructura de directorios de su sistema.

#### *El directorio raíz*

Todos los directorios del sistema Linux se encuentran bajo el directorio raíz, que está representado por una barra inclinada (/). Todo en su sistema se puede encontrar en este directorio raíz, incluso si están almacenados en diferentes dispositivos virtuales o físicos.



```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ ls /  
bin    dev    initrd.img  media  proc  sbin    sys  var  
boot  etc    lib         mnt    root  selinux tmp  vmlinuz  
cdrom  home  lost+found  opt    run   srv     usr  
sssit@JavaTpoint:~$
```

Mire la instantánea anterior, le hemos mostrado el directorio raíz de nuestro sistema, es decir; Ubuntu con la ayuda del comando `"ls/"`. Aquí, hemos escrito (/) para representar el directorio raíz.

## Directorios de Linux

Hemos clasificado los directorios según el tipo de archivo como se indica a continuación:

Directory type	Types of files stored
Binary directories	Contains binary or compiled source code files, eg, /bin, /sbin, etc.
Configuration directories	Contains configuration files of the system, eg, /etc, /boot.
Data directories	Stores data files, eg, /home, /root, etc.
Memory directories	Stores device files which doesn't take up actual hard disk space, eg, /dev, /proc, /sys.
Usr (Unix System Resources)	Contains sharable, read only data, eg, /usr/bin, /usr/lib, etc.
var (variable directory)	Contains larger size data, eg, /var/log, /var/cache, etc.
Non-standard directories	Directories which do not come under standard FHS, eg, lost+found, /run, etc.

## Introducción de Linux a los usuarios

Aca te mostraremos cómo identificar la cuenta de usuario de un sistema con comandos como `quién`, `quién soy`, etc.

Si más de una persona usa un solo sistema, entonces todos pueden tener su propia cuenta de usuario. Aquí, será útil conocer los detalles de la cuenta de usuario.

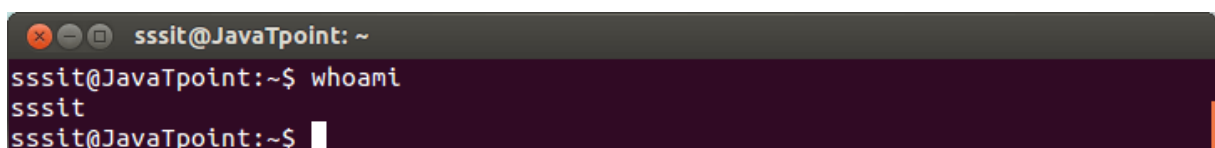
También explica cómo crear una segunda cuenta de usuario y ejecutar un programa con la ayuda de los comandos `su` y `sudo`.

`quién soy`

Le informa sobre el nombre de usuario del sistema.

### Sintaxis:

1. `quién soy`



```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ whoami  
sssit  
sssit@JavaTpoint:~$
```

Mire la instantánea anterior, '`sssit`' es el nombre de usuario de nuestro sistema.

`quién`

El comando `who` brinda información sobre los usuarios que iniciaron sesión en el sistema.

### Sintaxis:

1. `quién`

```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ who  
sssit    tty7          2016-06-30 09:24  
sssit    pts/0        2016-06-30 12:47 (:0)  
sssit@JavaTpoint:~$
```

quién soy

Este comando muestra la información sobre el usuario actual únicamente.

**Sintaxis:**

1. quién soy

```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ who am i  
sssit    pts/0        2016-06-30 12:47 (:0)  
sssit@JavaTpoint:~$
```

Mire la instantánea anterior, en nuestro sistema, el usuario que ha iniciado sesión actualmente es **sssit**.

w

Este comando informa sobre los usuarios que han iniciado sesión y qué están haciendo.

**Sintaxis:**

1. w

```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ w  
16:02:31 up 6:37, 2 users, load average: 0.05, 0.12, 0.20  
USER      TTY      FROM          LOGIN@      IDLE        JCPU       PCPU WHAT  
sssit     tty7          :0            09:24       6:37m      9:56       0.23s gnome-session -  
sssit     pts/0        :0            12:47       0.00s      0.17s      0.00s w  
sssit@JavaTpoint:~$
```

identificación

Este comando informa sobre su identificación de usuario, identificación de grupo principal y una lista de grupos que le pertenecen.

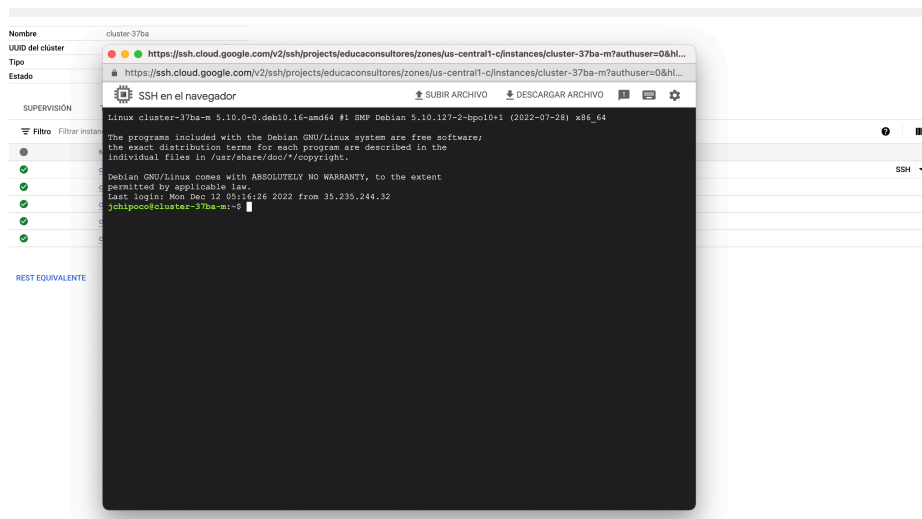
**Sintaxis:**

1. identificación

```
sssit@JavaTpoint: ~  
sssit@JavaTpoint:~$ id  
uid=1000(sssit) gid=1000(sssit) groups=1000(sssit),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),109(lpadmin),124(sambashare)  
sssit@JavaTpoint:~$
```

## COMANDOS LINUX

Primeramente loguearnos a nuestro servidor usando SSH:



### 1. comando pwd

Usa el comando **pwd** para encontrar la ruta del directorio (carpeta) de trabajo actual en el que te encuentras. El comando devolverá una ruta absoluta (completa), que es básicamente una ruta de todos los directorios que comienzan con una barra diagonal (/) Un ejemplo de una ruta absoluta es **/home/nombredeusuario**.

### 2. comando cd

Para navegar por los archivos y directorios de Linux, usa el comando **cd**. Te pedirá la ruta completa o el nombre del directorio, dependiendo del directorio de trabajo actual en el que te encuentres.

Supongamos que estás en **/home/nombredeusuario/Documentos** y deseas ir a **Fotos**, un subdirectorio de **Documentos**. Para hacerlo, simplemente escribe el siguiente comando: **cd Fotos**.

Otro escenario es si deseas ir a un directorio completamente nuevo, por ejemplo, **/home/nombredeusuario/Peliculas**. En este caso, debes escribir **cd** seguido de la ruta absoluta del directorio: **cd /home/ nombredeusuario/Peliculas**.

Hay algunos atajos para ayudarte a navegar rápidamente:

- **cd ..** (con dos puntos) para ir un directorio hacia arriba
- **cd** para ir directamente a la carpeta de inicio
- **cd-** (con un guión) para ir al directorio anterior

Como nota al margen, el shell de Linux distingue entre mayúsculas y minúsculas. Por lo tanto, debes escribir el nombre del directorio de forma exacta.

### 3. comando ls

El comando **ls** se usa para ver el contenido de un directorio. Por defecto, este comando mostrará el contenido de tu directorio de trabajo actual.

Si deseas ver el contenido de otros directorios, escribe **ls** y luego la ruta del directorio. Por ejemplo, ingresa **ls/home/nombredeusuario/Documentos** para ver el contenido de **Documentos**.

Hay variaciones que puedes usar con el comando **ls**:

- **ls -R** también listará todos los archivos en los subdirectorios
- **ls -a** mostrará los archivos ocultos
- **ls -al** listará los archivos y directorios con información detallada como los permisos, el tamaño, el propietario, etc.

#### 4. comando cat

**cat** (abreviatura de concatenate, en inglés) es uno de los comandos más utilizados en Linux. Se utiliza para listar el contenido de un archivo en la salida estándar (sdout). Para ejecutar este comando, escribe **cat** seguido del nombre del archivo y su extensión. Por ejemplo: **cat archivo.txt**.

Aquí hay otras formas de usar el comando **cat**:

- **cat > nombredearchivo** crea un nuevo archivo.
- **cat nombredearchivo1 nombredearchivo2>nombredearchivo3** une dos archivos (1 y 2) y almacena la salida de ellos en un nuevo archivo (3)
- convertir un archivo a mayúsculas o minúsculas, **cat nombredearchivo | tr a-z A-Z> salida.txt**

#### 5. comando cp

Usa el comando **cp** para copiar archivos del directorio actual a un directorio diferente. Por ejemplo, el comando **cp escenario.jpg /home/nombredeusuario/Imagenes** crearía una copia de **escenario.jpg** (desde tu directorio actual) en el directorio de **Imagenes**.

#### 6. comando mv

El uso principal del comando **mv** es mover archivos, aunque también se puede usar para cambiar el nombre de los archivos.

Los argumentos en **mv** son similares al comando **cp**. Debes escribir **mv**, el nombre del archivo y el directorio destino. Por ejemplo: **mv archivo.txt /home/nombredeusuario/Documentos**.

Para cambiar el nombre de los archivos, el comando de Linux es **mv nombreviejo.ext nombrenuevo.ext**

#### 7. comando mkdir

Usa el comando **mkdir** para crear un nuevo directorio: si escribes **mkdir Musica**, creará un directorio llamado **Musica**.

También hay comandos adicionales de **mkdir**:

- Para generar un nuevo directorio dentro de otro directorio, usa este comando básico de Linux **mkdir Musica/Nuevoarchivo**
- Usa la opción **p** (padres) para crear un directorio entre dos directorios existentes. Por ejemplo, **mkdir -p Musica/2020/Nuevoarchivo** creará el nuevo archivo «2020».

#### 8. comando rmdir

Si necesitas eliminar un directorio, usa el comando **rmdir**. Sin embargo, **rmdir** solo te permite eliminar directorios vacíos.

## 9. comando **rm**

El comando **rm** se usa para eliminar directorios y el contenido dentro de ellos. Si solo deseas eliminar el directorio, como alternativa a **rmdir**, usa **rm -r**.

**Nota:** Ten mucho cuidado con este comando y verifica en qué directorio te encuentras. Este comando elimina todo y no se puede deshacer.

## 10. comando **touch**

El comando **touch** te permite crear un nuevo archivo en blanco a través de la línea de comando de Linux. Como ejemplo, ingresa **touch /home/nombredeusuario/Documentos/Web.html** para crear un archivo HTML titulado **Web** en el directorio **Documentos**.

## 11. comando **locate**

Puedes usar este comando para **localizar** un archivo, al igual que el comando de búsqueda en Windows. Además, el uso del argumento **-i** junto con este comando hará que no distinga entre mayúsculas y minúsculas, por lo que puedes buscar un archivo incluso si no recuerdas su nombre exacto.

Para buscar un archivo que contenga dos o más palabras, usa un asterisco (\*). Por ejemplo, el comando **locate -i escuela\*nota** buscará cualquier archivo que contenga la palabra «escuela» y «nota», ya sea en mayúsculas o minúsculas.

## 12. comando **find**

Similar al comando **locate**, usando **find** también buscas archivos y directorios. La diferencia es que usas el comando **find** para ubicar archivos dentro de un directorio dado.

Como ejemplo, el comando **find /home/ -name notas.txt** buscará un archivo llamado **notas.txt** dentro del directorio de inicio y sus subdirectorios.

Otras variaciones al usar **find** son:

- Para buscar archivos en el directorio actual, **find . -name notas.txt**
- Para buscar directorios, **/ -type d -name notes.txt**

## 13. comando **grep**

Otro comando básico de Linux que sin duda es útil para el uso diario es **grep**. Te permite buscar a través de todo el texto en un archivo dado.

Para ilustrar, **grep azul notepad.txt** buscará la palabra azul en el archivo del bloc de notas. Las líneas que contienen la palabra buscada se mostrarán.

## 14. comando **sudo**

Abreviatura de «**SuperUser Do**» (SuperUsuario hace), este comando te permite realizar tareas que requieren permisos administrativos o raíz. Sin embargo, no es aconsejable usar este comando para el uso diario, ya que podría ser fácil que ocurra un error si haces algo mal.

## 15. comando **df**

Usa el comando **df** para obtener un informe sobre el uso del espacio en disco del sistema, que se muestra en porcentaje y KB. Si deseas ver el informe en megabytes, escribe **df -m**.



## 16. comando du

Si deseas verificar cuánto espacio ocupa un archivo o un directorio, el comando **du** (Uso del disco, en inglés) es la respuesta. Sin embargo, el resumen de uso del disco mostrará números de bloque de disco en lugar del formato de tamaño habitual. Si deseas verlo en bytes, kilobytes y megabytes, agrega el argumento **-h** a la línea de comando.

## 17. comando head

El comando **head** se usa para ver las primeras líneas de cualquier archivo de texto. De manera predeterminada, mostrará las primeras diez líneas, pero puedes cambiar este número a tu gusto. Por ejemplo, si solo deseas mostrar las primeras cinco líneas, escribe **head -n 5 nombredearchivo.ext**.

## 18. comando tail

Este tiene una función similar al comando **head**, pero en lugar de mostrar las primeras líneas, el comando **tail** mostrará las últimas diez líneas de un archivo de texto. Por ejemplo, **tail -n nombredearchivo.ext**.

## 19. comando diff

Para abreviar diferencia, el comando **diff** compara el contenido de dos archivos línea por línea. Después de analizar los archivos, genera las líneas que no coinciden. Los programadores a menudo usan este comando cuando necesitan hacer modificaciones al programa en lugar de reescribir todo el código fuente.

La forma más simple de usar este comando es **diff archivo1.ext archivo2.ext**

## 20. comando tar

El comando **tar** es el comando más utilizado para guardar múltiples archivos en un **tarball**, un formato de archivo de Linux común que es similar al formato zip, con compresión opcional. Este comando es bastante complejo con una larga lista de funciones, como agregar nuevos archivos a un archivo existente, enumerar el contenido de un archivo, extraer el contenido de un archivo y muchos más.

## 21. comando chmod

**chmod** es otro comando de Linux, utilizado para cambiar los permisos de lectura, escritura y ejecución de archivos y directorios. Como este comando es bastante complicado, puedes leer el [tutorial completo](#) (en inglés) para ejecutarlo correctamente.

## 22. comando chown

En Linux, todos los archivos son propiedad de un usuario específico. El comando **chown** te permite cambiar o transferir la propiedad de un archivo al nombre de usuario especificado. Por ejemplo, **chown usuariolinux2 archivo.ext** hará que **usuariolinux2** sea el propietario del **archivo.ext**.

## 23. comando jobs

El comando **jobs** mostrará todos los trabajos actuales junto con sus estados. Un trabajo es básicamente un proceso iniciado por el shell.

## 24. comando kill

Si tienes un programa que no responde, puedes cerrarlo manualmente utilizando el [comando kill](#). Enviará una cierta señal al programa que se está ejecutando mal y le indica a la aplicación que finalice.

Hay un total de sesenta y cuatro señales que puedes usar, pero las personas generalmente solo usan dos señales:

- **SIGTERM (15)**: solicita que un programa deje de ejecutarse y te da algo de tiempo para guardar todo tu progreso. Si no especificas la señal al ingresar el comando kill, se utilizará esta señal.
- **SIGKILL (9)**: obliga a los programas a detenerse inmediatamente. El progreso no guardado se perderá.

Además de conocer las señales, también debes conocer el número de identificación del proceso (PID) del programa que deseas detener (**kill**). Si no conoces el PID, simplemente ejecute el comando **ps ux**.

Después de saber qué señal deseas usar y el PID del programa, ingresa la siguiente sintaxis:  
**kill [opción de señal] PID.**

## 25. comando ping

Usa el comando **ping** para verificar tu estado de conectividad a un servidor. Por ejemplo, simplemente ingresando **ping google.com**, el comando verificará si puedes conectarte a Google y también medirá el tiempo de respuesta.

## 26. comando wget

La línea de comandos de Linux es muy útil: incluso puedes descargar archivos de Internet con la ayuda del comando **wget**. Para hacerlo, simplemente escribe **wget** seguido del enlace de descarga.

## 27. comando uname

El comando **uname**, abreviatura de Nombre de Unix, imprimirá información detallada sobre tu sistema Linux, como el nombre de la máquina, el sistema operativo, el núcleo, etc.

## 28. comando top

Como un terminal equivalente al Administrador de tareas en Windows, el comando **top** mostrará una lista de los procesos en ejecución y la cantidad de CPU que utiliza cada proceso. Es muy útil monitorear el uso de los recursos del sistema, especialmente para saber qué proceso debe terminarse porque consume demasiados recursos.

## 29. comando history

Cuando hayas estado utilizando Linux durante un cierto período de tiempo, notarás rápidamente que puedes ejecutar cientos de comandos todos los días. Como tal, ejecutar el comando **history** es particularmente útil si deseas revisar los comandos que ingresaste anteriormente.

## 30. comando man

¿Confundido sobre la función de ciertos comandos de Linux? No te preocupes, puedes aprender fácilmente cómo usarlos directamente desde el shell de Linux mediante el comando **man**. Por ejemplo, al ingresar **man tail** se mostrarán las instrucciones manuales del comando tail.

### 31. comando echo

Este comando se usa para mover algunos datos a un archivo. Por ejemplo, si deseas agregar el texto «Hola, mi nombre es John» en un archivo llamado nombre.txt, debes escribir **echo Hola, mi nombre es John >> nombre.txt**

### 32. comando zip, unzip

Usa el comando **zip** para comprimir tus archivos en un archivo zip y use el comando **unzip** para extraer los archivos comprimidos de un archivo zip.

### 33. comando hostname

Si deseas conocer el nombre de tu host/red, simplemente escribe **hostname**. Agregar un **-I** al final mostrará la dirección IP de tu red.

## COMANDOS HDFS

```
hdfs
hdfs dfs
echo Hello World HDFS >> test.txt
ls -alt
cat test.txt
clear
```

```
hdfs dfs -mkdir /prueba
hdfs dfs -ls /
hdfs dfs -put test.txt /prueba
```

```
hdfs dfs -cat /prueba/test.txt
hdfs dfs -mkdir /prueba2
hdfs dfs -cp /prueba/test.txt /prueba2/
```

```
hdfs dfs -rm /prueba2/test.txt
hdfs fsck /prueba/VirtualBox-7.0.4-154605-OSX.dmg -files -blocks -locations
```

### *Listar Archivos en HDFS*

hdfs dfs -ls /	Lista todos los ficheros y directorios para el path /
hdfs dfs -ls -h /	Lista los ficheros con su tamaño en formato legible
hdfs dfs -ls -R /	Lista todos los ficheros y directorios recursivamente (con subdirectorios)
hdfs dfs -ls /file*	Lista todos los ficheros que cumplen el patrón (ficheros que comienzan con 'file')

### *Leer y Escribir Archivos*

hdfs dfs -text /app.log	Imprime el fichero en modo texto por la terminal
hdfs dfs -cat /app.log	Muestra el contenido del fichero en la salida estándar
hdfs dfs -appendToFile /home/file1 /file2	Añade el contenido del fichero local 'file1' al fichero en hdfs 'file2'

### *Cargar y Descargar Archivos*

hdfs dfs -put /home/file1 /hadoop	Copia el fichero 'file1' del sistema de ficheros local a hdfs
hdfs dfs -put -f /home/file1 /hadoop	Copia el fichero 'file1' del sistema de ficheros local a hdfs y lo sobrescribe en el caso de que ya exista
hdfs dfs -put -l /home/file1 /hadoop	Copia el fichero 'file1' del sistema de ficheros local a hdfs. Fuerza replicación 1 y permite al DataNode persistir los datos de forma perezosa.
hdfs dfs -put -p /home/file1 /hadoop	Copia el fichero 'file1' del sistema de ficheros local a hdfs. Mantiene los tiempos de acceso, de modificación y propietario original
hdfs dfs -get /file1 /home/	Copia el fichero 'file1' de hdfs al sistema de ficheros local
hdfs dfs -moveFromLocal /home/file1 /hadoop	Copia el fichero 'file1' del sistema de ficheros local a hdfs y luego lo borra del sist. ficheros local

### *Gestión de Archivos*

hdfs dfs -cp /hadoop/file1 /hadoop1	Copia el fichero al directorio destino en hdfs
-------------------------------------	--

<code>hdfs dfs -cp -p /hadoop/file1 /hadoop1</code>	Copia el fichero al directorio destino en hdfs conservando tiempos de acceso y de modificación, propietario y modo
<code>hdfs dfs -rm /hadoop/file1</code>	Elimina el fichero 'file1' de hdfs y lo envía a la papelera
<code>hdfs dfs -rm -r /hadoop</code>	
<code>hdfs dfs -rm -R /hadoop</code>	
<code>hdfs dfs -rmr /hadoop</code>	Elimina el directorio y su contenido en hdfs
<code>hdfs dfs -rm -skipTrash /file1</code>	Elimina el fichero sin dejarlo en la papelera
<code>hdfs dfs -mkdir /hadoop2</code>	Crea un directorio en hdfs
<code>hdfs dfs -touchz /hadoop3</code>	Crea un fichero en hdfs con tamaño 0

### ***Gestión de Permisos***

<code>hdfs dfs -checksum /hadoop/file1</code>	Muestra la información checksum del fichero
<code>hdfs dfs -chmod 775 /hadoop/file1</code>	Cambia los permisos del fichero en hdfs
<code>hdfs dfs -chmod -R 755 /hadoop</code>	Cambia los permisos de los ficheros recursivamente
<code>hdfs dfs -chown hadoop:hadoop /file1</code>	Cambia el propietario y el grupo del fichero
<code>hdfs dfs -chown -R hadoop:hadoop /file1</code>	Cambia el propietario y el grupo recursivamente
<code>hdfs dfs -chgrp hadoop /file1</code>	Cambia el grupo del fichero

### ***Comandos de Administración HDFS***

<code>hdfs dfs -df /hadoop</code>	Muestra la capacidad y el espacio libre y usado del sistema de ficheros
<code>hdfs dfs -df -h /hadoop</code>	Muestra la capacidad y el espacio libre y usado del sistema de ficheros en formato legible
<code>hadoop version</code>	Muestra la versión de hadoop
<code>hdfs fsck /</code>	Comprueba el estado de salud del sistema de ficheros
<code>hdfs dfsadmin -safemode leave</code>	Deshabilita el modo seguro del NameNode
<code>hdfs namenode -format</code>	Formatea el NameNode