



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* García Morales Karina

*Asignatura:* Fundamentos de Programación

*Grupo:* 25

*No. de práctica(s):* 2

*Integrante(s):* Enriquez Ramirez Damian

*No. de lista o brigada:* 08

*Semestre:* 2025-2

*Fecha de entrega:* 26/02/2025

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

# Guía práctica de estudio 02: GNU/Linux

## - Objetivos.

El alumno identificará al sistema operativo como una parte esencial de un sistema de cómputo. Explorará un sistema operativo GNU/Linux con el fin de conocer y utilizar sus comandos básicos.

## - Desarrollo de la Practica.

### - Introducción

El Sistema Operativo es el conjunto de programas y datos que administra los recursos tanto de hardware (dispositivos) como de software (programas y datos) de un sistema de cómputo y/o comunicación. Además, funciona como interfaz entre la computadora y el usuario o aplicaciones.

En la actualidad existen diversos sistemas operativos; por ejemplo, para equipos de cómputo están Windows, Linux, Mac OS entre otros. Para el caso de dispositivos móviles se encuentran Android, IOS, Windows Phone, etcétera. Cada uno de ellos tiene diferentes versiones y distribuciones que se ajustan a los diversos equipos de cómputo y comunicación en los que trabajan.

### -Sistema Operativo Linux.

Linux es un sistema operativo semejante a Unix, de código abierto y desarrollado por una comunidad, para computadoras, servidores, mainframes, dispositivos móviles y dispositivos embebidos. Es compatible con casi todas las principales plataformas informáticas, incluyendo x86, ARM y SPARC, por lo que es uno de los sistemas operativos más soportados.

### -Software libre.

Un software libre es aquel que se puede adquirir de manera gratuita, es decir, no se tiene que pagar algún tipo de licencia a alguna casa desarrolladora de software por el uso de éste. Además, que un software sea libre implica también que el software viene acompañado del código fuente, es decir, se pueden realizar cambios en el funcionamiento del sistema si así se desea.

**- Kernel de GNU/Linux:** El kernel Linux es el elemento principal de los sistemas operativos (SO) Linux, y es la interfaz fundamental entre el hardware de una computadora y sus procesos. Los comunica entre sí y gestiona los recursos de la manera más eficiente posible.

Se llama kernel porque se encuentra dentro del sistema operativo y controla todas las funciones principales del hardware, ya sea un teléfono, una computadora portátil, un servidor o cualquier otro tipo de equipo., y es la interfaz fundamental entre el hardware de una computadora y sus procesos.

**- Interfaz de línea de comandos (CLI) o shell de GNU/Linux:** El Shell de GNU/Linux permite introducir órdenes (comandos) y ejecutar programas en el sistema operativo. Todas las órdenes de UNIX/Linux son programas que están almacenados en el sistema de archivos y a los que llamamos comandos, por lo tanto, todo en GNU/Linux se puede controlar mediante comandos.

**- Comandos básicos:** Para trabajar en Linux utilizando comandos, se debe abrir una “terminal” o “consola” que es una ventana donde aparece la “línea de comandos” en la cual se escribirá la orden o comando. La terminal permite un mayor grado de funciones y configuración de lo que queremos hacer con una aplicación o acción en general respecto a un entorno gráfico.

La sintaxis que siguen los comandos es la siguiente: comando [-opciones] [argumentos] Esto es, el nombre del comando, seguido de algunas banderas (opciones) para modificar la ejecución de este y, al final, se puede incluir un argumento (ruta, ubicación, archivo, etcétera) dependiendo del comando. Tanto las opciones como los argumentos son opcionales.

**- Comando ls:** El comando ls permite listar los elementos que existen en alguna ubicación del sistema de archivos de Linux. Por defecto lista los elementos que existen en la ubicación actual

```
estudiante@Bosnia25 ~ % ls
Desktop      Downloads    Movies       Pictures
Documents    Library      Music        Public
```

**-Comando ls . :** realizan exactamente lo mismo que el commando ls.

```
estudiante@Bosnia25 ~ % ls .
Desktop      Downloads    Movies       Pictures
Documents    Library      Music        Public
```

- **ls -l**: El comando ls realiza acciones distintas dependiendo de las banderas que utilice, por ejemplo, si se utiliza la opción l se genera un listado largo de la ubicación actual.

```
estudiante@Bosnia25 ~ % ls -l
total 0
drwx-----+ 3 estudiante staff 96 Feb 19 11:12 Desktop
drwx-----+ 4 estudiante staff 128 Feb 19 12:35 Documents
drwx-----+ 3 estudiante staff 96 Feb 19 11:12 Downloads
drwx-----@ 80 estudiante staff 2560 Feb 19 19:09 Library
drwx----- 3 estudiante staff 96 Feb 19 11:12 Movies
drwx-----+ 3 estudiante staff 96 Feb 19 11:12 Music
drwx-----+ 4 estudiante staff 128 Feb 19 11:12 Pictures
drwxr-xr-x+ 4 estudiante staff 128 Feb 19 11:12 Public
```

- **ls /**: Es posible listar los elementos que existen en cualquier ubicación del sistema de archivos, para ello hay que ejecutar el comando especificando como argumento la ubicación donde se desean listar los elementos. Si queremos ver los archivos que se encuentran en la raíz.

```
estudiante@Bosnia25 ~ % ls /
Applications  System      Volumes     cores        etc           opt          sbin         usr
Library       Users       bin          dev          home         private      tmp          var
```

- **ls /home**: Para ver los usuarios del equipo local, revisamos el directorio home que parte de la raíz (/).

```
estudiante@Bosnia25 ~ % ls /home
```

- **man ls**: GNU/Linux proporciona el comando man, el cual permite visualizar la descripción de cualquier comando, así como la manera en la que se puede utilizar.

```
estudiante@Bosnia25 ~ % man ls
Unknown locale, assuming C
```

- **ls /Volumenes**: Esto es, el argumento se inicia con / indicando que es el directorio raíz, seguido de Volumenes que es el nombre del directorio. Cuando especificamos la ubicación de un archivo partiendo de la raíz, se dice que estamos indicando la “ruta absoluta” del archivo.

```
estudiante@Bosnia25 ~ % ls /Volumes
Macintosh HD
```

Existe otra forma de especificar la ubicación de un archivo, esto es empleando la “ruta relativa”. Si bien el punto (.) es para indicar la ubicación actual, el doble punto (..) se utiliza para referirse al directorio “padre”. De esta forma si deseamos listar los archivos que dependen de mi directorio padre se escribe el siguiente comando: **ls .. o ls ../**

Con los primeros dos puntos se hace referencia al directorio home, con los siguientes dos puntos se refiere al directorio raíz, y finalmente se escribe el nombre del Directorio: **ls ../../Directorio**

- **Comando touch:** El comando touch permite crear un archivo de texto, su sintaxis es la siguiente: **touch nombre archivo[.txt]**.

```
estudiante@Bosnia25 ~ % touch damian.txt
estudiante@Bosnia25 ~ % touch enriquez.txt ramirez.txt
```

- **Comando mkdir:** El comando mkdir permite crear una carpeta a, su sintaxis es la siguiente: **mkdir nombre carpeta.**

```
estudiante@Bosnia25 ~ % mkdir tareas
```

- **Comando cd:** El comando cd permite ubicarse en una carpeta, su sintaxis es la siguiente: **cd nombre carpeta.**

```
estudiante@Bosnia25 ~ % cd tareas/
estudiante@Bosnia25 tareas % cd ..
```

- **comando pwd:** El comando pwd permite conocer la ubicación actual(ruta), su sintaxis es la siguiente: **pwd.**

```
estudiante@Bosnia25 ~ % pwd
/Users/estudiante
```

- **Comando find:** El comando find permite buscar un elemento dentro del sistema de archivos, su sintaxis es la siguiente:

**find . -name cadena\_buscar**

- **Comando clear:**

El comando clear permite limpiar la consola o terminal, su sintaxis es la siguiente: **clear.**

- **Comando cp:** permite copiar un archivo, su sintaxis es la siguiente: **cp archivo\_origen archivo\_destino.**

```
estudiante@Bosnia25 ~ % cp directorio1/telecomunicaciones.txt tareas/telecomunicaciones2.txt
```

- **Comando mv:**

El comando mv mueve un archivo de un lugar a otro, en el sistema de archivos; su sintaxis es la siguiente:

## Mv ubicación origen/archivo ubicación destino

```
[estudiante@Bosnia25 ~ % mv tareas/damian.txt enriquez.txt ramirez.txt borrar/
```

- **Comando rm:** El comando rm permite eliminar un archivo o un directorio, su sintaxis es la siguiente:

**rm nombre\_archivo**

**rm nombre\_carpeta**

```
estudiante@Bosnia25 ~ % rm -r directorio1
estudiante@Bosnia25 ~ % rm -r directorio2 directorio3
estudiante@Bosnia25 ~ % rm -r borrar/
estudiante@Bosnia25 ~ % rm -r tareas/
```

### -Tarea

1.- Crea un directorio que se llame *LAB2025-2\_Nombre* y entra a el.

Ejemplo: *LAB2025-2 KGM*

```
[root@localhost ~]# mkdir LAB2025-2_DER
[root@localhost ~]# cd LAB2025-2_DER
```

2.- Crea los directorios necesarios de las Materias que estas cursando(Mínimo 3) llamado *Materia*.

```
[root@localhost LAB2025-2_DER]# mkdir Algebra
[root@localhost LAB2025-2_DER]# mkdir Calculo
[root@localhost LAB2025-2_DER]# mkdir Programacion
```

3.- Dentro de cada directorio crea un archivo llamado *Nombre\_PENDIENTES\_Materia* (recuerda que puedes utilizar los comandos vistos en la práctica)

```
[root@localhost LAB2025-2_DER]# cd Algebra
[root@localhost Algebra]# mkdir DER_PENDIENTE_Programacion
[root@localhost Algebra]# CD ..
sh: CD: command not found
[root@localhost Algebra]# cd..
sh: cd.: command not found
[root@localhost Algebra]# cd ..
[root@localhost LAB2025-2_DER]# cd Calculo
[root@localhost Calculo]# mkdir DER_PENDIENTE_Calculo
[root@localhost Calculo]# cd ..
[root@localhost LAB2025-2_DER]# Programacion
sh: Programacion: command not found
[root@localhost LAB2025-2_DER]# cd Programacion
[root@localhost Programacion]# mkdir DER_PENDIENTE_Programacion
```

4.- Muestra el contenido de cada uno de tus directorios creados dentro de *LAB2025\_2\_Nombre*.

```
Algebra:
DER_PENDIENTE_Algebra

Calculo:
DER_PENDIENTE_Calculo

Programacion:
DER_PENDIENTE_Programacion
[root@localhost LAB2025-2_DER]#
```

5.- Salir del directorio *LAB2025-2\_Nombre*.

```
[root@localhost LAB2025-2_DER]# cd ..
```

6.- Crea un directorio que se llame *Nombre\_COPIA*.

```
[root@localhost ~]# mkdir DER_COPIA
```

7.- Copia los archivos que creaste, al directorio *Nombre\_COPIA*.

```
[root@localhost ~]# cp -r Algebra/ DER_COPIA/
[root@localhost ~]# cp -r Calculo/ DER_COPIA/
[root@localhost ~]# cp -r Programacion/ DER_COPIA/
```

8.- Muestra el contenido de tu directorio *Nombre\_COPIA*.

```
[root@localhost DER_COPIA]# ls -l / Algebra Calculo Programacion

Algebra:
DER_PENDIENTE_Algebra

Calculo:
DER_PENDIENTE_Calculo

Programacion:
DER_PENDIENTE_Programacion
[root@localhost DER_COPIA]# cd ..
```

9.- Mueve el directorio *Nombre\_COPIA* al directorio *LAB2025-2\_Nombre*.

```
[root@localhost ~]# cp -r DER_COPIA/ LAB2025-2_DER/
[root@localhost ~]# cd LAB2025-2_DER
[root@localhost LAB2025-2_DER]# ls
Algebra Calculo DER_COPIA Programacion
```

10.- Muestra los permisos de los archivos contenidos en el directorio *LAB2025-2\_Nombre* y en *Nombre\_COPIA*, explica cada uno de los permisos (Investigar comando `chown` y `chmod`).

-**Comando Chmod:** El comando `chmod` (*change mode*) sirve para gestionar los permisos de los archivos o directorios del sistema.

-**Comando Chown:** El comando `chown` (*change owner*) sirve para indicar quién es el dueño y el grupo de un archivo.

11.- Indica el directorio en el que te encuentras y con que comando lo muestras.

```
[root@localhost LAB2025-2_DER]# pwd
/root/LAB2025-2_DER
```

12.- Teclea el comando *cal* y escribe lo que muestra.

```
[root@localhost LAB2025-2_DER]# cal
February 2025
Su Mo Tu We Th Fr Sa
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28
```

-Al teclear cal la terminal nos arroja un calendario del mes en el que estamos.

13.- Teclea el comando *date* y escribe la salida.

```
[root@localhost LAB2025-2_DER]# date
Wed Feb 26 08:25:44 AM UTC 2025
```

-Al teclear date este comando nos arroja la fecha (dia de la semana, mes, dia, hora y año).

14.- Describe para que empleas el comando man

El comando man nos permite ver la forma en la que podemos utilizar cualquier comando.

15.- Investiga para que se utiliza el comando cat

El **comando cat** en Linux permite concatenar y mostrar el contenido de archivos. Deriva de “concatenar” y se utiliza para visualizar, unir y crear archivos.

Conclusion.

En conclusion usar linux es una herramienta util, Segura y facil es ideal para desarrollar es uno de los sistemas operativos mas utilizados por los desarrolladores, se manejan varios comandos pero para que estos sean funcionales es importante poner atencion a la hora de escribirlos ya que si nos equivocamos este no tendra la function que esperamos.



## **- Bibliografía.**

Laboratorio Salas A y B. (s. f.).

<http://lcp02.fi-b.unam.mx/>

B, G., & B, G. (2023b, agosto 10). *Comando cat de Linux: para qué sirve y ejemplos de uso.* Tutoriales Hostinger.

<https://www.hostinger.com/mx/tutoriales/comando-cat-linux>

Arsys. (s. f.). *Uso del comando Chmod y comando Chown para la gestión de servidores.* <https://www.arsys.es/blog/comandos-chmod-chown>

JSLinux. (s. f.). <https://bellard.org/jslinux/vm.html?cpu=riscv64&url=fedora33-riscv.cfg&mem=256>

AIX 7.2. (s. f.). <https://www.ibm.com/docs/es/aix/7.2?topic=files-copying-cp-command>

¿Qué es el kernel Linux? y ¿Cuáles son sus funciones? (s. f.).

<https://www.redhat.com/es/topics/linux/what-is-the-linux-kernel>

Bigelow, S. J. (2021, 9 abril). *Sistema operativo Linux.* ComputerWeekly.es.

<https://www.computerweekly.com/es/definicion/Sistema-operativo-Linux>

