

# Guía práctica de estudio 06: GNU/Linux

---



---

***Elaborado por:***

Ing. Jorge A. Solano Gálvez  
M.C. Edgar E. García Cano

***Revisado por:***

M.C.

***Autorizado por:***

M.C. Alejandro Velázquez Mena

# Guía práctica de estudio 06: GNU/Linux

## Objetivo:

Explorar un sistema operativo GNU/Linux.

## Actividades:

- Conocer los comandos básicos en GNU/Linux.
- Crear y editar archivos de texto a través del editor vi.

## Introducción

Linux es un sistema operativo tipo Unix de libre distribución para computadoras personales, servidores y estaciones de trabajo.

El sistema está conformado por el núcleo (kernel) y un gran número de programas y bibliotecas. Muchos programas y bibliotecas han sido posibles gracias al proyecto GNU, por lo mismo, se conoce a este sistema operativo como GNU/Linux.

## 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 del mismo.

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.

Linux se distribuye bajo la Licencia Pública General de GNU por lo tanto, el código fuente tiene que estar siempre accesible y cualquier modificación ó trabajo derivado tiene que tener esta licencia.

## Licencia GNU

La Licencia Pública General de GNU o GNU General Public License (GNU GPL) es una licencia creada por la Free Software Foundation en 1989 y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

## Kernel de GNU/Linux

El kernel ó núcleo de linux se puede definir como el corazón del sistema operativo. Es el encargado de que el software y el hardware del equipo se puedan comunicar.

Entre las funciones más importantes del kernel están:

- Administración de la memoria para todos los programas y procesos en ejecución.
- Administración del tiempo de procesador que los programas y procesos en ejecución utilizan.
- Administra el acceso a los periféricos y/o elementos de la computadora de una manera cómoda.



**Figura 1:** Capas que componen al sistema operativo GNU/Linux.

## 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

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 del mismo y, al final, se puede incluir un argumento (ruta, ubicación, archivo, etcétera) dependiendo del comando. Tanto las opciones como los argumentos son opcionales.

### *Ejemplo (comando ls)*

El comando `ls` permite listar los elementos que existen en alguna ubicación del sistema operativo. Por defecto lista los elementos que existen en la ubicación actual:

```
ls
```

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:

```
ls -l
```

Es posible listar los elementos que existen en cualquier ubicación del sistema operativo, para ello hay que ejecutar el comando especificando como argumento la ubicación donde se desean listar los elementos:

```
ls /home
```

Tanto las opciones como los argumentos se pueden combinar para generar una ejecución más específica:

```
ls -l /home
```

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

```
man ls
```

### *Ejemplo (comando touch)*

El comando touch permite crear un archivo de texto, su sintaxis es la siguiente:

```
touch nombre_archivo[.ext]
```

En GNU/Linux no es necesario agregar una extensión al archivo creado, sin embargo, es recomendable hacerlo para poder identificar el tipo de archivo creado.

### *Ejemplo (comando mkdir)*

El comando mkdir permite crear una carpeta, su sintaxis es la siguiente:

```
mkdir nombre_carpeta
```

### *Ejemplo (comando cd)*

El comando cd permite acceder a una carpeta, su sintaxis es la siguiente:

```
cd nombre_carpeta
```

### *Ejemplo (comando pwd)*

El comando pwd permite conocer la ruta ubicación actual, su sintaxis es la siguiente:

```
pwd
```

### *Ejemplo (comando find)*

El comando find permite buscar un elemento dentro del sistema, su sintaxis es la siguiente:

```
find . -name cadena_buscar
```

Al comando find hay que indicar en qué parte del sistema va a iniciar la búsqueda. En el ejemplo anterior la búsqueda se inicia en la posición actual. Además, utilizando la bandera `-name` permite determinar la cadena a buscar.

### *Ejemplo (comando clear)*

El comando clear permite limpiar la consola, su sintaxis es la siguiente:

```
clear
```

### *Ejemplo (comando cp)*

El comando cp permite copiar un archivo, su sintaxis es la siguiente:

```
cp archivo_origen archivo_destino
```

### *Ejemplo (comando mv)*

El comando mv mover un archivo de un lugar a otro, su sintaxis es la siguiente:

```
mv ubicación_actual/archivo ubicación_destino
```

El comando mueve el archivo desde su ubicación actual hacia la ubicación deseada. Este comando también puede ser usado para cambiar el nombre de un archivo.

### *Ejemplo (comando rm)*

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

```
rm nombre_archivo  
rm nombre_carpeta
```

Cuando la carpeta que se desea contiene información se debe utilizar la bandera `-f` para forzar la eliminación. Si la carpeta contiene otras carpetas, se debe utilizar la opción `-r`, para realizar la eliminación recursiva.

## Editor Visual Interface de GNU/Linux

El editor vi (visual interface) es un editor de texto de pantalla completa que maneja en memoria el texto entero de un archivo.

Es el editor clásico de UNIX; está en todas las versiones. Puede usarse en cualquier tipo de terminal con un mínimo de teclas.

El editor vi posee tres modos operativos:

- Modo comando: En éste modo se ejecuta el editor cada vez que abre un archivo. Permite introducir comandos.
- Modo texto o inserción: Este modo permite la capturar caracteres dentro del documento.
- Modo última línea o reemplazo: Este modo permite escribir comandos en la última línea (al final de la pantalla).

### Modo de comando

i	inserta texto a la izquierda del cursor
a	agrega texto a la derecha del cursor
x	borra el caracter bajo el cursor
o	añadir una línea en blanco
J	borrar el final de línea (une dos líneas)
u	deshacer la última edición
dd	borra una línea
w	avanza una palabra
b	va al inicio de la palabra actual
e	va al fin de la palabra actual
h o flecha izquierda	mueve el cursor un caracter a la izquierda
j o flecha abajo	mueve el cursor una línea hacia abajo
k o flecha arriba	mueve el cursor una línea hacia arriba
l o flecha derecha	mueve el cursor un caracter a la derecha
ESC	vuelve a modo comando

## Modo última línea o reemplazo

:q	sale del editor
:w	guarda el archivo
: w nombreArchivo	guarda el archivo como “nombreArchivo”
:w! nombreArchivo	forza el guardado del archivo
:wq	guarda y cierra el archivo
:x	guarda y cierra el archivo
:q!	cierra el archivo sin guardar cambios
:/cadena	busca cadena hacia adelante
:?cadena	busca la cadena hacia atrás
:set number	agrega número a las líneas
:set nonumber	quita el número de las líneas
:set list	muestra caracteres ocultos
:set nolist	deshabilita set list
:s/lo que busco/lo que reemplazo/c	reemplazar y confirmar
:s/lo que busco/lo que reemplazo/g	reemplazar una línea
:%s/lo que busco/lo que reemplazo/g	reemplazar todo documento
:1,\$s/lo que busco/lo que reemplazo/g	reemplazar todo documento

El editor vi permite crear cualquier tipo de archivo de texto plano: sql, txt, c, java, c++, php, html, jsp, sh, etc.

Además, permite ejecutar comandos propios de linux dentro del modo última línea. La sintaxis para ejecutar un comando de Linux dentro del editor es la siguiente:

*Ejemplo (vi)*

vi permite crear y editar archivos de texto, su sintaxis es la siguiente:

```
vi nombre_archivo[.ext]
```

Por ejemplo, se puede crear un archivo de fortran de la siguiente manera\_:

```
vi holaMundo.f
```



La instrucción anterior abre el editor vi con el archivo holaMundo.f.



Por defecto, vi se abre en modo comando, para empezar a ingresar datos hay que cambiar a modo texto, para ello se puede presionar la tecla i para insertar caracteres:

En este modo se puede teclear el programa:

```

program holaMundo
write(*,*) '¡Hola mundo!'
stop
end
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
5,11      All

```

Para guardar los cambios se debe regresar al modo comando con la tecla ESC y en modo de última línea teclear :wq para guardar y cerrar el archivo.

```

program holaMundo
write(*,*) '¡Hola mundo!'
stop
end
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq

```

Los pasos anteriores se deben realizar para generar el código fuente de un programa. A esta etapa se le conoce como la etapa de edición.

Una vez creado el código fuente se puede entrar a la etapa de compilación. Para compilar un programa en FORTRAN se utiliza el comando gfortran, su sintaxis es la siguiente:

```
gfortran nombre_archivo.f
```

El comando gfortran intenta compilar el programa y, si la compilación es exitosa, se genera el archivo ejecutable con el nombre a.out. Es recomendable generar un archivo ejecutable con el mismo nombre que el código fuente, para ello se puede utilizar la bandera -o. El archivo de salida puede tener extensión .out, .exe o nombrarse sin extensión:

```
gfortran nombre_archivo.f -o nombre_archivo.out  
gfortran nombre_archivo.f -o nombre_archivo.exe  
gfortran nombre_archivo.f -o nombre_archivo
```

Si la etapa de compilación es exitosa, se puede pasar a la etapa de ejecución del programa ejecutable de la siguiente manera:

```
./nombre_archivo.out  
./nombre_archivo.exe  
./nombre_archivo
```

## Bibliografía

- Óscar Vicente Huguet Soriano, Sonia Doménech Gómez. Introducción a Linux. [Figura 1]. Consulta: Junio de 2015. Disponible en: [http://mural.uv.es/oshuso/81\\_introduccion\\_a\\_linux.html](http://mural.uv.es/oshuso/81_introduccion_a_linux.html)