

**Instituto Politécnico Nacional
Escuela Superior de Cómputo
“ESCOM”**



**Unidad de Aprendizaje:
Cómputo Paralelo**

**Práctica No. 1
El entorno de GNU-Linux**

**Integrantes:
Sánchez De Los Ríos Flavio Josué - 6BV1
Tinoco Videgaray Sergio Ernesto - 6BV1
Ibarra González Emilio Francisco- 6BV1**

**Maestro:
Jiménez Benítez José Alfredo**

Fecha de Entrega: 12/03/23

Índice

Introducción	3
Marco Teórico	3
Desarrollo	7
Instalación de GNU\Linux	7
Entorno de GNU\Linux	16
Conclusiones	36
Referencias	36

Introducción

En la presente práctica se llevaron a cabo los pasos necesarios para la instalación del sistema operativo GNU/Linux con la distribución Ubuntu en un entorno virtualizado. Del mismo modo se realizaron ejecuciones de comandos básicos para el aprendizaje y familiarización del manejo, gestión, creación y direccionamiento de archivos en el sistema operativo.

Marco teórico

Máquinas Virtuales

Las **máquinas virtuales** se definen como equipos virtuales o equipos definidos por software dentro de servidores físicos, donde solo existen como código [1]. Las máquinas virtuales se utilizan comúnmente en entornos de desarrollo y pruebas de software, para crear y probar entornos de sistemas operativos y aplicaciones en una variedad de configuraciones. También se utilizan para virtualizar servidores y alojar múltiples sistemas operativos en una sola máquina física, lo que puede reducir los costos y mejorar la eficiencia.

Distribuciones GNU/Linux

Una **distribución de Linux** es una colección de paquetes de software, recopilados y mantenidos por un grupo de personas [2]. Incluye paquetes de software específicos para satisfacer las necesidades de un grupo de usuarios en particular. Las ediciones de la distribución varían según su propósito, siendo comunes las ediciones para uso doméstico, empresarial y para servidores. La mayoría del software utilizado en estas es de código abierto, aunque también pueden incluir aplicaciones y controladores de software privados.. Además, a menudo cuentan con un Sistema de gestión de paquetes para facilitar la instalación y actualización del software incluido.

Los requisitos de instalación del sistema operativo GNU/Linux pueden variar según la distribución específica que se desee instalar. Sin embargo, en general, los requisitos mínimos para instalar y ejecutar GNU/Linux son los siguientes [3]:

- Un procesador compatible con la arquitectura de la distribución elegida (por ejemplo, x86, x86_64, ARM, etc.)
- Al menos 1 GB de RAM (aunque se recomienda al menos 2 GB para un mejor rendimiento)
- Al menos 10 GB de espacio libre en el disco duro
- Una unidad de DVD o una memoria USB para arrancar el sistema operativo
- Una conexión a Internet para descargar actualizaciones y software adicional durante la instalación (opcional, pero recomendado)

Estadísticas del Uso de GNU/Linux en el Mundo

Linux como sistema operativo principal entre la comunidad de desarrollo de software [4] ha estado aumentando constantemente durante los últimos 5 años. De 2018 a 2021 se registró un crecimiento constante con un 23,2 %, 25,6 %, 26,6 %, 25,3 % y, finalmente, en 2022 el uso fue del 40,23 %. El uso de Linux fue mayor que macOS en 2021, pero solo por un pequeño margen. 2022 ahora es un 9% más que macOS.

Sin embargo en el ámbito popular Linux tiene muy poca popularidad. De la encuesta realizada por Statcounter GlobalStats [5] Windows acaba el año con un 75,11% de popularidad, por el 15,6% de macOS y el 2,77% de Linux.

GNOME y KDE

GNOME es un entorno de escritorio diseñado para ser fácil de usar y elegante, con una interfaz de usuario simple e intuitiva [6]. Es conocido por su enfoque minimalista y su diseño limpio. GNOME se basa en el toolkit GTK y utiliza el administrador de ventanas Metacity. También cuenta con una amplia gama de aplicaciones integradas, incluyendo un navegador web, gestor de archivos, reproductor multimedia y mucho más.

Por otro lado, KDE es un entorno de escritorio altamente personalizable y flexible, que se basa en la plataforma Qt [6]. KDE se caracteriza por tener una gran cantidad de características y herramientas avanzadas, que lo hacen ideal para usuarios avanzados y programadores. KDE cuenta con un amplio conjunto de aplicaciones integradas, así como herramientas para personalizar la apariencia del escritorio, como widgets y efectos visuales.

Command Line Interface y Graphical User Interface

Los Entornos **Command Line Interface** (CLI) y los Entornos **Graphical User Interface** (GUI) son dos interfaces diferentes que se utilizan en los sistemas operativos para interactuar con la computadora.

La CLI es una interfaz de usuario que se basa en texto y se utiliza a través de la línea de comandos o terminal [7]. En una CLI, los usuarios escriben comandos para realizar acciones en la computadora, y los resultados se muestran en forma de texto en la pantalla. La CLI es muy potente y eficiente para realizar tareas repetitivas o automatizar procesos, pero puede requerir más conocimiento técnico y habilidad para usarla.

La GUI es una interfaz de usuario que utiliza elementos visuales como iconos, ventanas y menús para interactuar con la computadora [7]. En una GUI, los usuarios pueden hacer clic en los elementos visuales en la pantalla para realizar acciones en la computadora. La GUI es muy fácil de usar y aprender, pero puede requerir más recursos del sistema y ser menos eficiente para realizar tareas repetitivas o automatizadas.

Terminal GNU/Linux

La **Terminal de GNU/Linux** es una aplicación que permite a los usuarios interactuar con el sistema operativo mediante CLI [8]. También se le conoce como terminal, consola o shell. En la Terminal de GNU/Linux, los usuarios pueden escribir comandos para realizar tareas en el sistema operativo, como crear archivos, copiar y pegar archivos, instalar software, administrar permisos de usuario, configurar redes y mucho más. Los comandos se escriben en un intérprete de comandos (shell) que se ejecuta en la terminal.

Usuarios en GNU/Linux

El concepto de **usuario** en Linux permite separar entornos de ejecución para diferentes propósitos. Dos personas pueden trabajar simultáneamente en el mismo sistema, teniendo cada uno un usuario diferente, y un directorio personal diferente. En GNU/Linux, existen tres tipos de usuarios [9]:

1. Usuario Root (Superusuario o Administrador): es el usuario con todos los privilegios en el sistema operativo. El usuario root tiene acceso completo a todos los archivos y directorios del sistema, y puede realizar cualquier tarea en la computadora, incluyendo la instalación y eliminación de software y la modificación de configuraciones críticas del sistema.
2. Usuarios del sistema: son los usuarios normales que utilizan el sistema operativo para realizar tareas cotidianas, como navegar por la web, escribir documentos y enviar correos electrónicos. Los usuarios del sistema no tienen los mismos privilegios que el usuario root, lo que significa que no pueden realizar cambios en el sistema sin permiso. Cada usuario del sistema tiene su propio directorio de inicio, donde se almacenan sus archivos y configuraciones.
3. Usuarios especiales: son usuarios que se utilizan para realizar tareas específicas en el sistema operativo. Los usuarios especiales tienen permisos y privilegios específicos para realizar su tarea, pero no tienen acceso completo al sistema operativo.

Rutas Absolutas y Relativas

Las **rutas absolutas** especifican la ubicación completa de un archivo o directorio desde la raíz del sistema de archivos [10]. Comienzan con una barra diagonal (/) y siguen la jerarquía de directorios hasta llegar al archivo o directorio deseado. Por ejemplo, la ruta absoluta para el directorio de inicio del usuario "john" sería /home/john.

Las **rutas relativas** indican el camino para encontrar un elemento, pero basándonos en el directorio desde el que se ejecuta la orden [10]. Son mucho más cortas que las absolutas, pero para saber si son correctas o no tenemos que saber siempre desde dónde se han utilizado. Comienzan con el nombre del archivo o directorio y pueden incluir puntos (.) o puntos-puntos (..) para representar el directorio actual o el directorio padre, respectivamente. Un ejemplo sería ../john/documents/.

Redireccionamiento

En GNU/Linux, el **redireccionamiento** es una técnica para manipular las entradas y salidas de los comandos [11]. El redireccionamiento se utiliza para enviar la salida de un comando a un archivo en lugar de mostrarla en la pantalla, o para leer la entrada de un comando desde un archivo en lugar del teclado. A continuación se muestran algunos operadores de redireccionamiento:

- ">" : redirecciona la salida de un comando a un archivo. Por ejemplo, el comando "ls > listado.txt" redirecciona la lista de archivos del directorio actual a un archivo llamado "listado.txt".
- ">>" : agrega la salida de un comando al final de un archivo existente, en lugar de sobrescribir el archivo. Por ejemplo, el comando "echo "Hola" >> saludo.txt" agrega la palabra "Hola" al final del archivo "saludo.txt".
- "<" : redirecciona la entrada de un comando desde un archivo. Por ejemplo, el comando "sort < lista.txt" ordena las líneas del archivo "lista.txt" en orden alfabético y muestra el resultado en la pantalla.
- "|" : redirecciona la salida de un comando a la entrada de otro comando. Por ejemplo, el comando "ls | grep ".txt"" muestra solo los archivos con extensión ".txt" en el directorio actual.

Comandos y su Clasificación

Los comandos de GNU/Linux se clasifican de la siguiente manera [12]:

- Comandos de gestión de archivos: Estos comandos se utilizan para manipular archivos y directorios en el sistema de archivos. Algunos ejemplos de comandos en esta categoría son ls, cp, mv, rm, mkdir, rmdir, etc.
- Comandos de gestión de procesos: Estos comandos se utilizan para gestionar procesos en el sistema. Algunos ejemplos de comandos en esta categoría son ps, kill, top.
- Comandos de gestión de usuarios: Estos comandos se utilizan para gestionar usuarios y grupos en el sistema. Algunos ejemplos de comandos en esta categoría son useradd, usermod, userdel, groupadd, groupmod, groupdel.
- Comandos de gestión de redes: Estos comandos se utilizan para configurar y administrar la red en el sistema. Algunos ejemplos de comandos en esta categoría son ifconfig, ping, traceroute, netstat, iptables.
- Comandos de gestión del sistema: Estos comandos se utilizan para configurar y administrar el sistema en general. Algunos ejemplos de comandos en esta categoría son shutdown, reboot, df, du, free, uname.

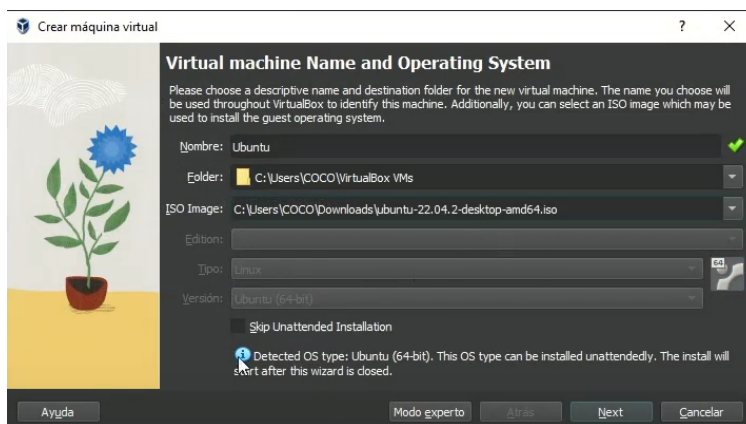
- Comandos de gestión de paquetes: Estos comandos se utilizan para instalar, actualizar y gestionar paquetes de software en el sistema. Algunos ejemplos de comandos en esta categoría son apt, yum, dnf.
- Comandos de gestión de seguridad: Estos comandos se utilizan para gestionar la seguridad del sistema. Algunos ejemplos de comandos en esta categoría son passwd, su, chown, chmod.
- Comandos de edición de texto: Estos comandos se utilizan para editar y manipular archivos de texto en el sistema. Algunos ejemplos de comandos en esta categoría son nano, vim, sed, awk, grep.

Variables de Entorno

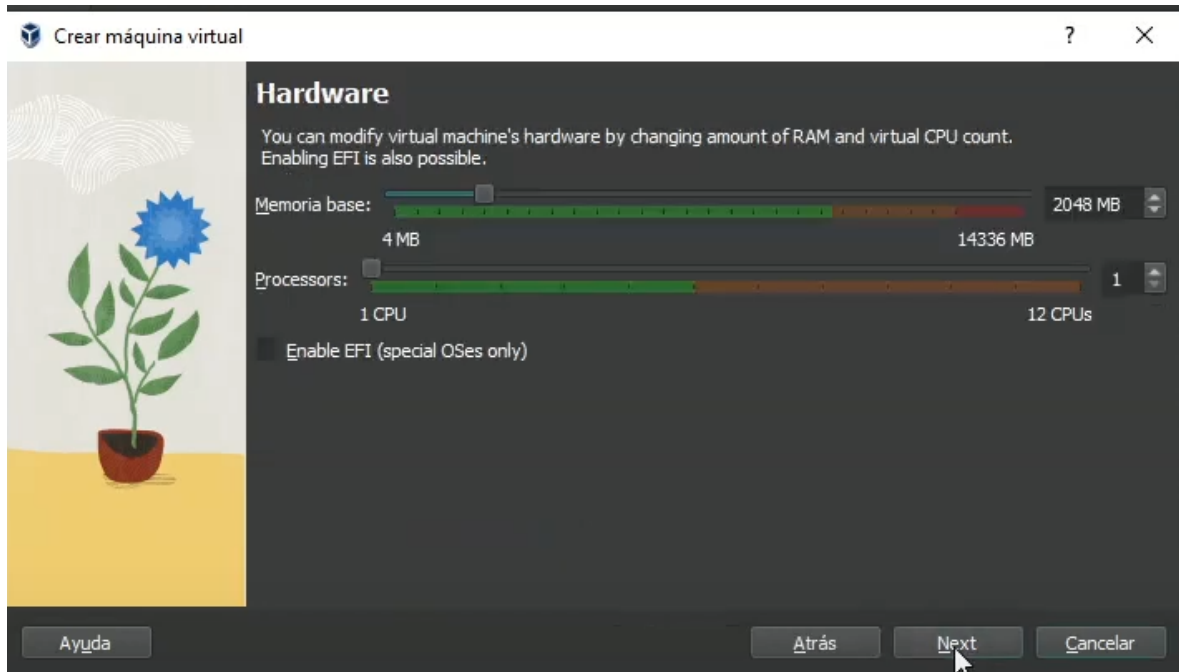
Las **variables de entorno** son valores dinámicos que afectan los programas o procesos que se ejecutan en un servidor [13]. Existen en todos los sistemas operativos y su tipo puede variar. Las variables de entorno se pueden crear, editar, guardar y eliminar. En Linux, las variables de entorno son marcadores de posición para la información almacenada dentro del sistema que pasa datos a los programas iniciados en shells (intérpretes de comando) o sub-shells.

Desarrollo

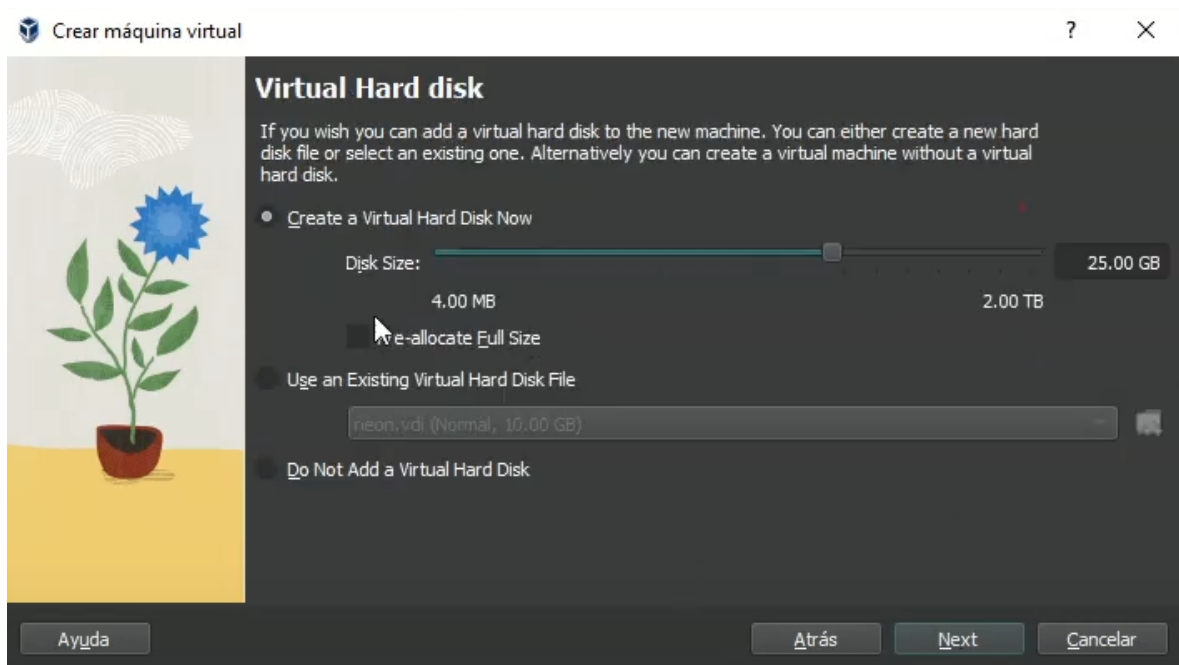
Instalación de GNU-Linux



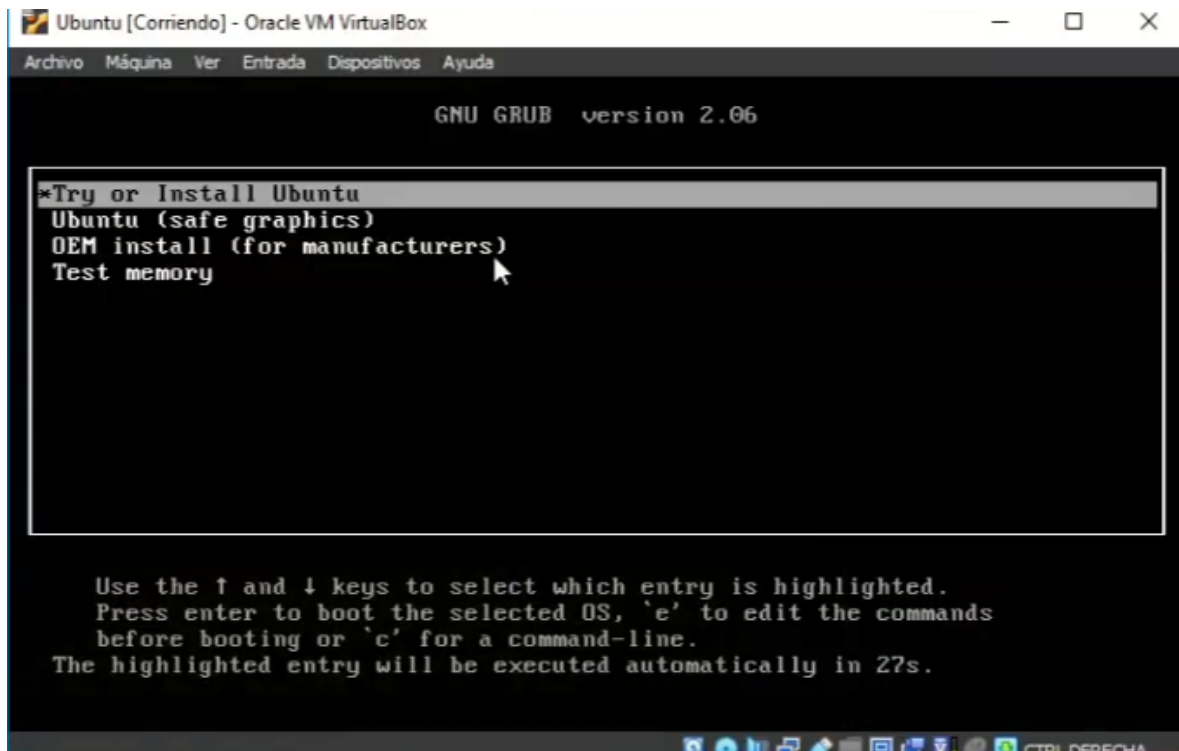
1.0 Importación de la ISO en VB



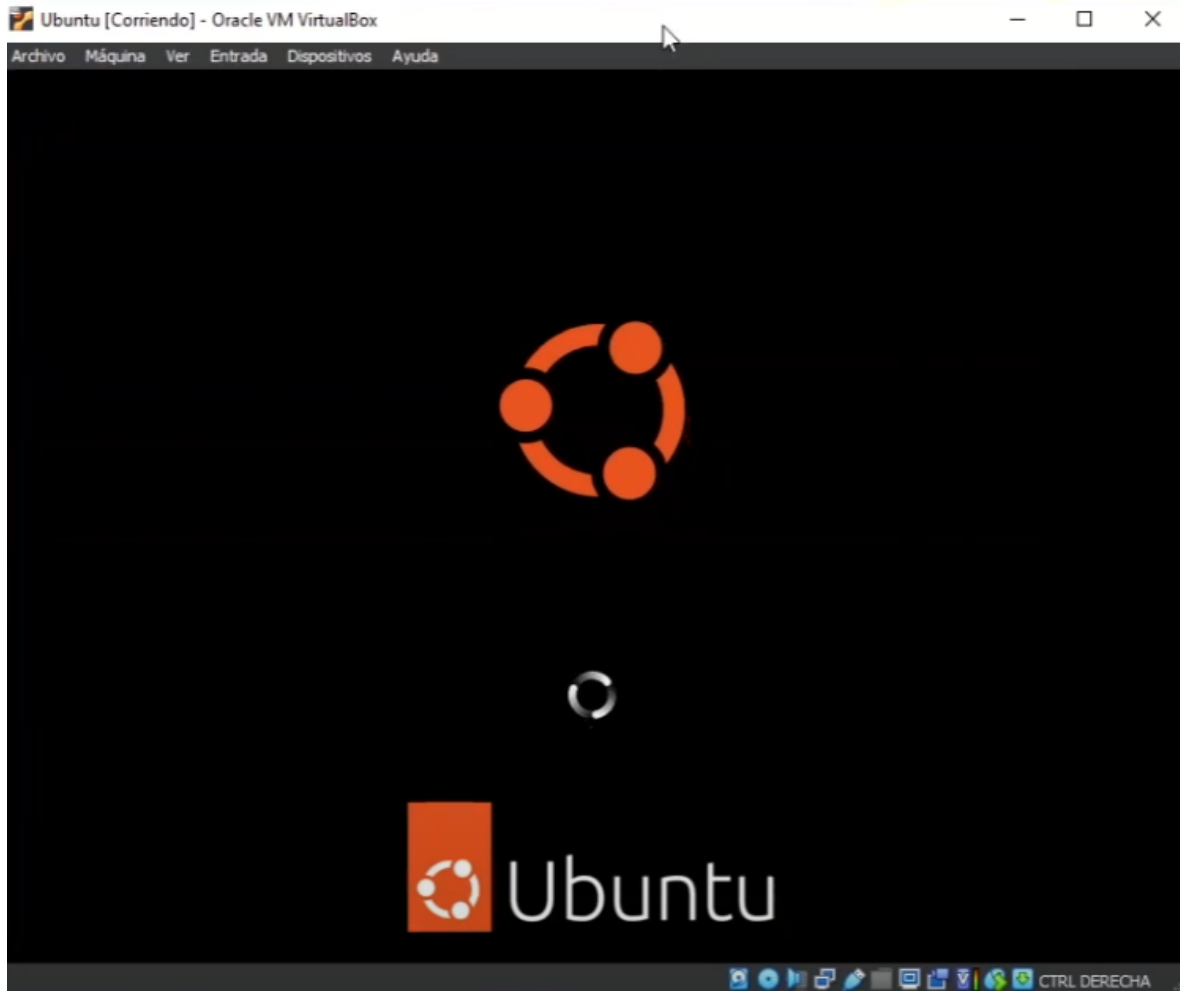
1.1 Particionamiento de memoria



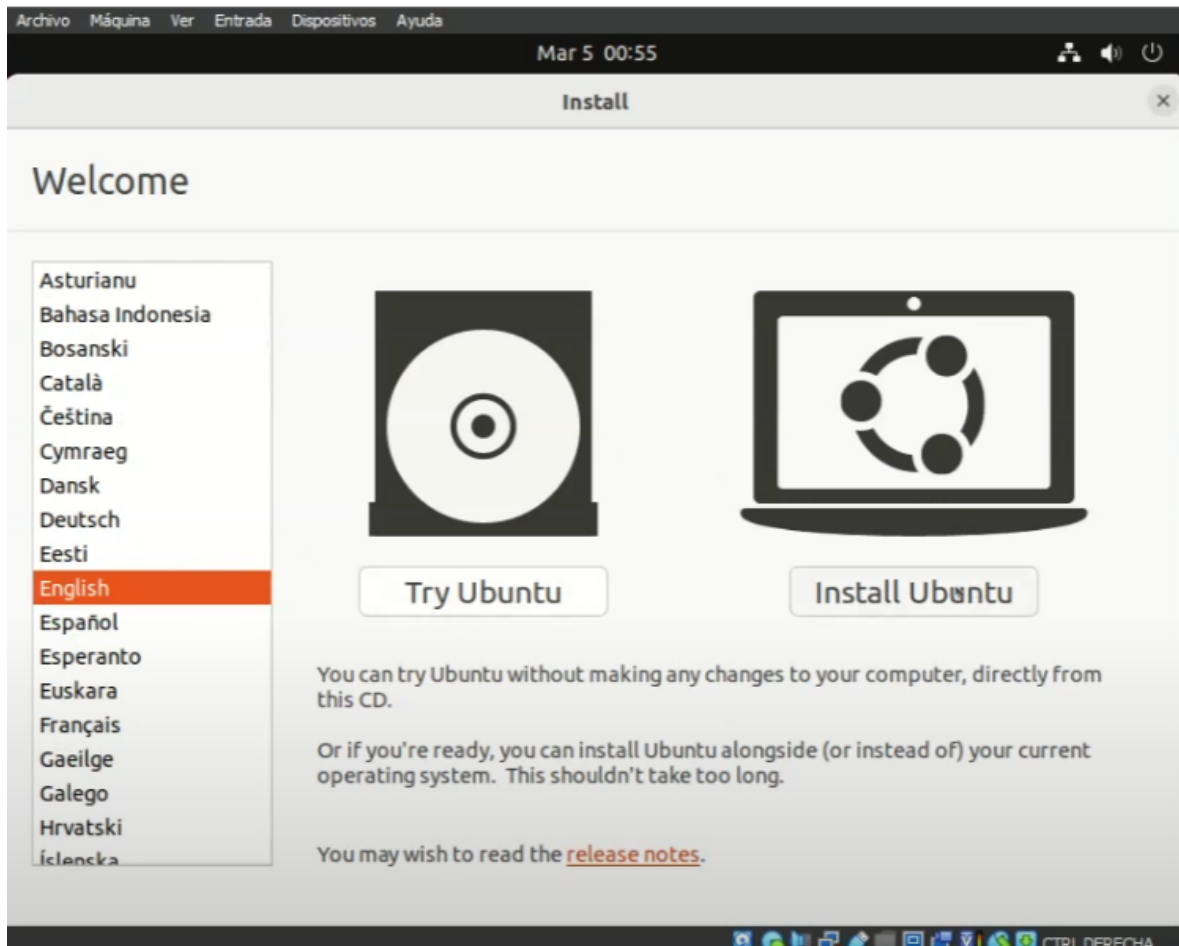
1.2 Creación de un disco duro virtual



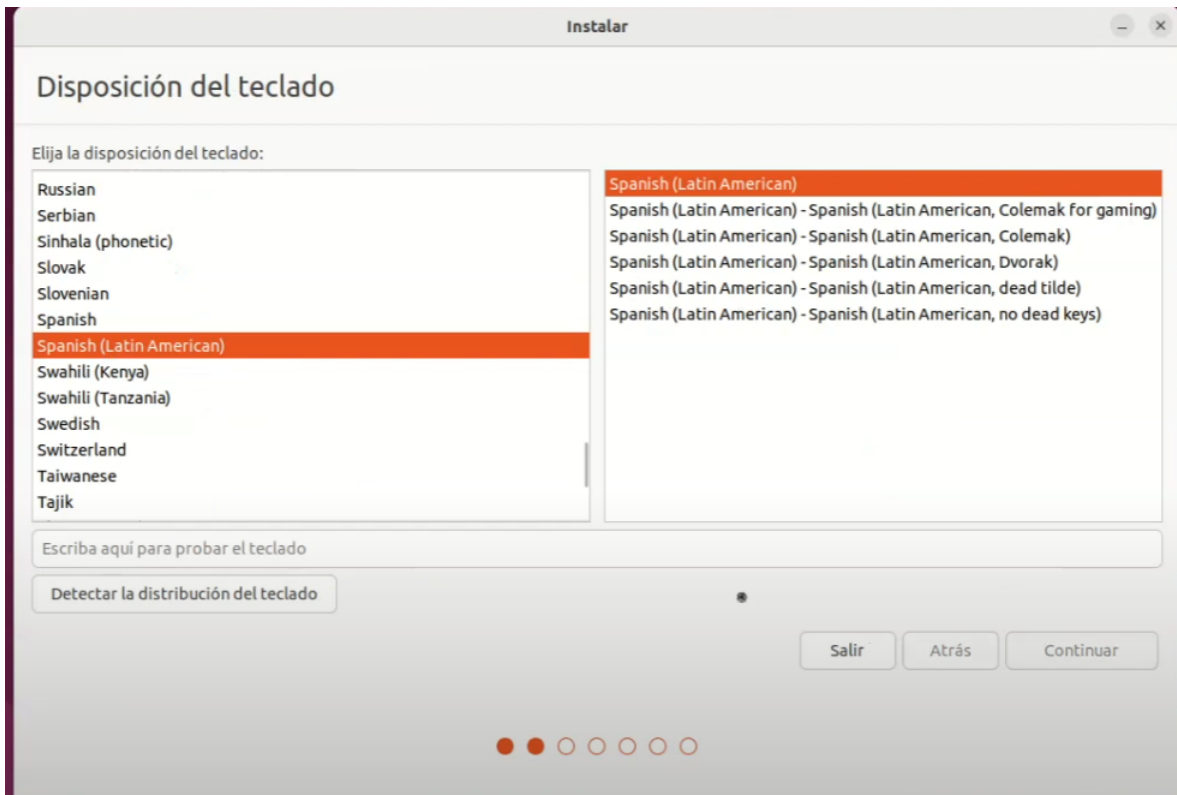
1.3 Boot menu of UBUNTU



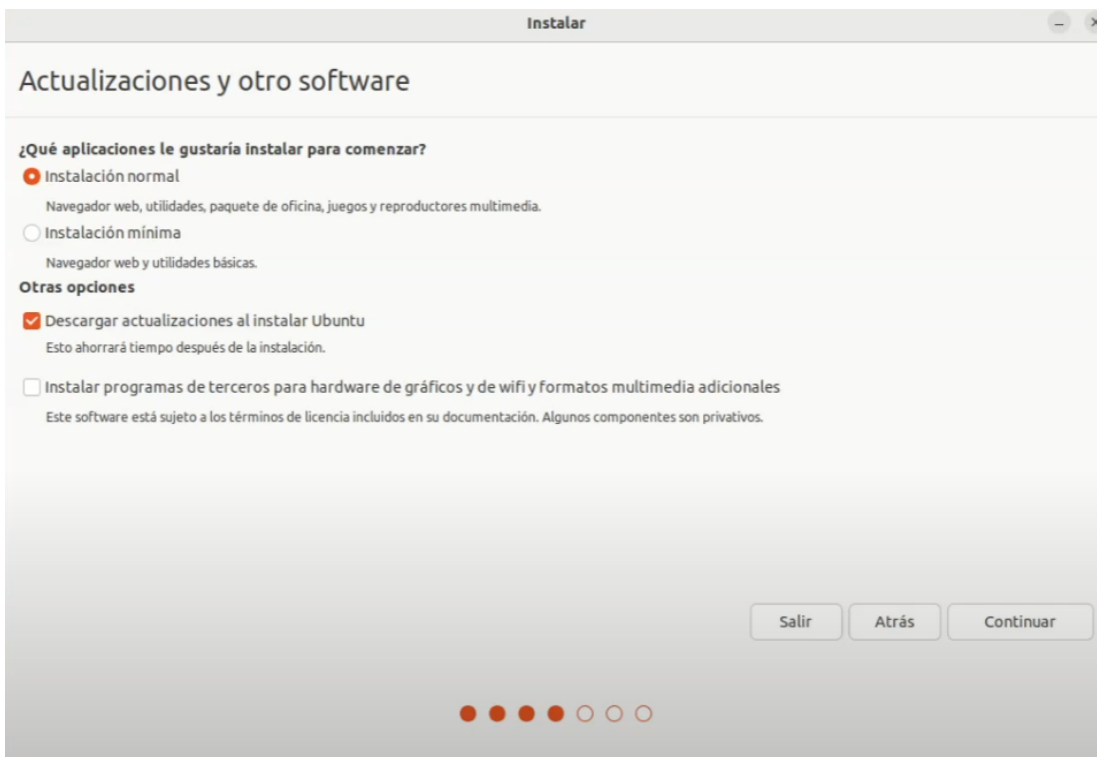
1.4 Inicialización de UBUNTU



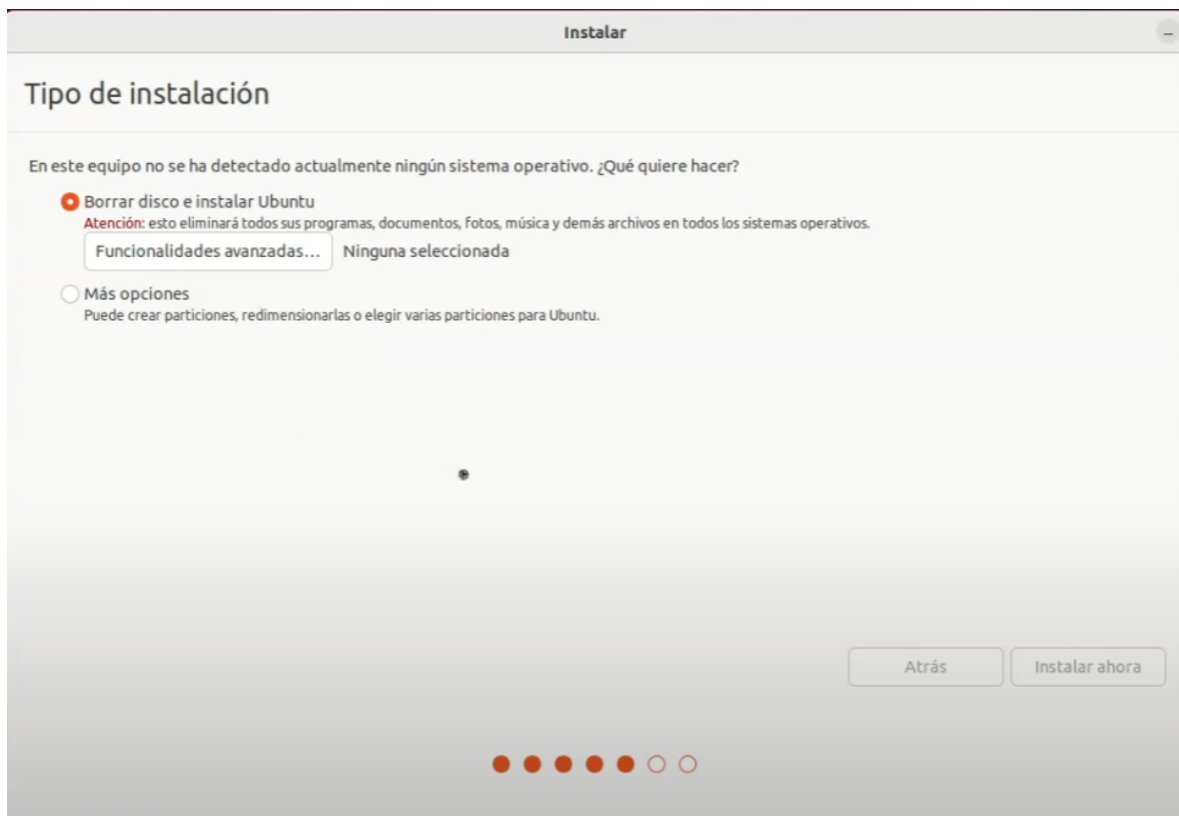
1.5 First UBUNTU SetUp



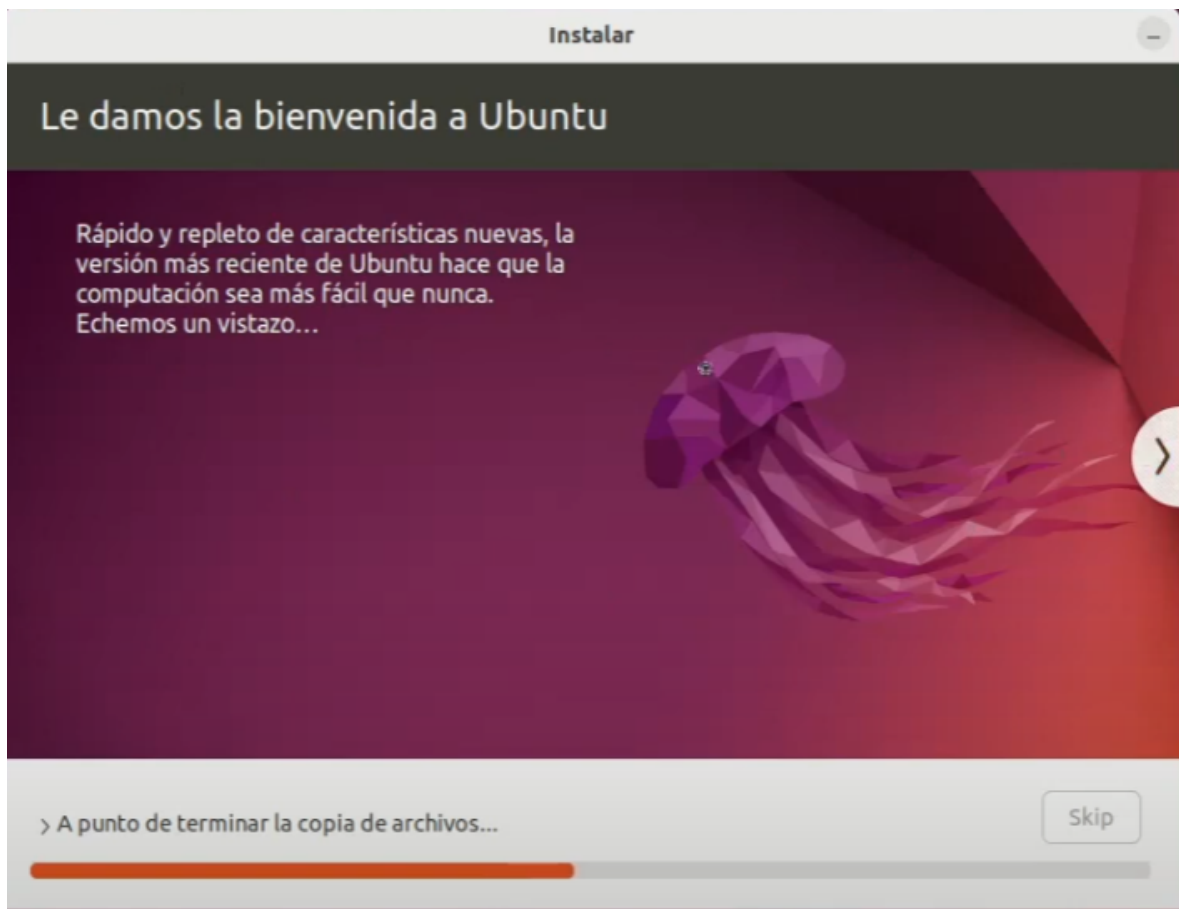
1.6 Selección del Teclado



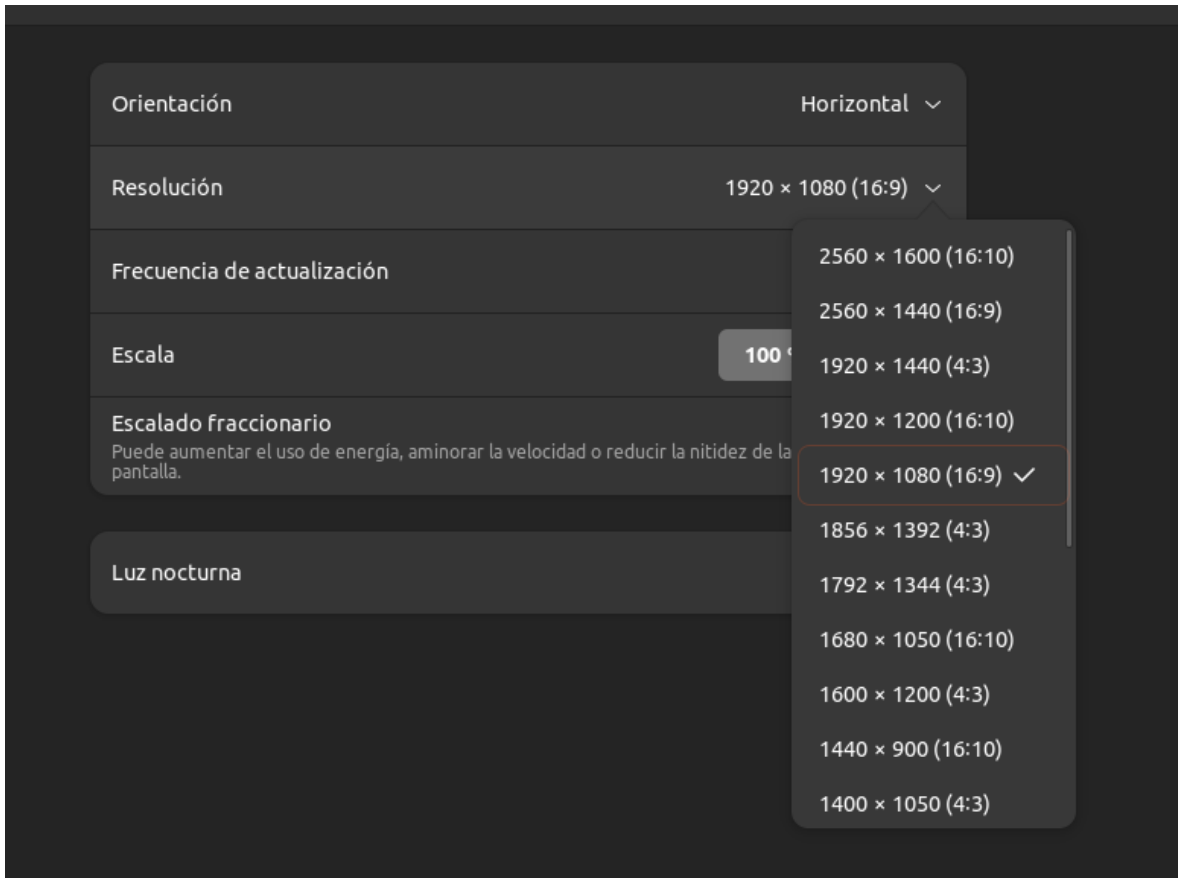
1.7 Preferencias de Actualizaciones



1.8 Seleccionar tipo de Instalación

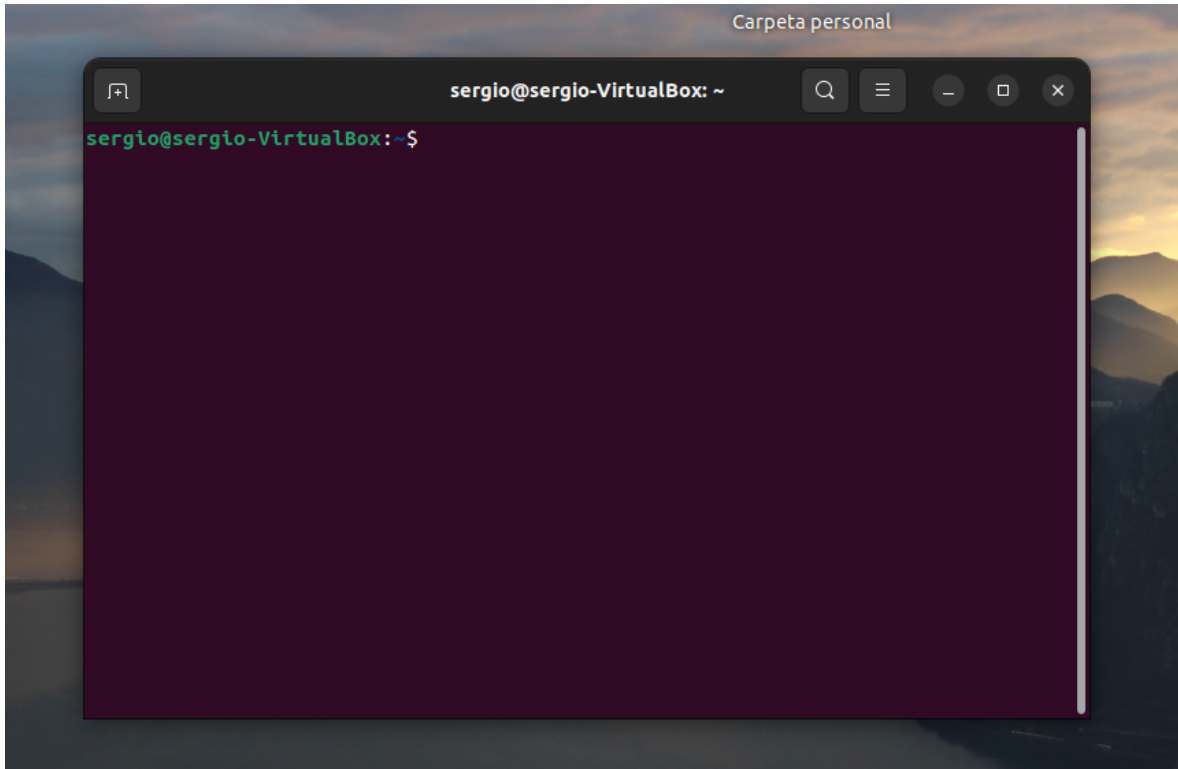


1.9 Instalación de UBUNTU



2.0 Opciones de Pantalla UBUNTU

Entorno de GNU-Linux



1. Cal: Muestra el calendario del mes actual.

```
sergio@sergio-VirtualBox:~$ cal
  Marzo 2023
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 29 30 31
```

2. Date: Muestra la fecha actual en formato dd/mm/yyyy h:min:seg

```
sergio@sergio-VirtualBox:~$ date
dom 12 mar 2023 19:53:03 CST
sergio@sergio-VirtualBox:~$
```


3. **echo:** Imprime en consola el mensaje entre comillas

```
sergio@sergio-VirtualBox:~$ echo "hola mundo"
hola mundo
sergio@sergio-VirtualBox:~$
```

4. **time:** muestra el tiempo consumido por la ejecución de la tubería.

```
sergio@sergio-VirtualBox:~$ time

real    0m0.000s
user    0m0.000s
sys     0m0.000s
```

5. **uname:** Muestra el nombre del sistema, en este caso el kernel que es el de Linux.

```
sergio@sergio-VirtualBox:~$ uname
Linux
sergio@sergio-VirtualBox:~$
```

6. **w:** Muestra el estado de salida del proceso identificado por un ID.

```
sergio@sergio-VirtualBox:~$ w
19:56:11 up 15 min,  1 user,  load average: 1.04, 0.71, 0.46
USUARIO  TTY      DE              LOGIN@  IDLE   JCPU   PCPU WHAT
sergio   tty2      tty2            19:40   15:59  0.03s  0.03s /usr/libexec/gn
```

7. **who:** Muestra la información del usuario y la fecha.

```
sergio@sergio-VirtualBox:~$ who
sergio   tty2            2023-03-12 19:40 (tty2)
sergio@sergio-VirtualBox:~$
```

8. **apt-get:** Proporciona una interfaz de línea de órdenes para la obtención de paquetes y de información sobre ellos.

```
sergio@sergio-VirtualBox:~$ apt-get
apt 2.5.3 (amd64)
Uso: apt-get [opciones] orden
      apt-get [opciones] install|remove paq1 [paq2 ...]
      apt-get [opciones] source paq1 [paq2 ...]
```

9. clear: Limpia la consola pero no el historial.

```
sergio@sergio-VirtualBox:~$
```

10. df: Muestra la cantidad de espacio en disco disponible en el sistema con cada bloque de archivos.

```
sergio@sergio-VirtualBox:~$ df
```

S.ficheros	bloques de 1K	Usados	Disponibles	Uso%	Montado en
tmpfs	202296	1600	200696	1%	/run
/dev/sda3	25106692	12111488	11694520	51%	/
tmpfs	1011468	0	1011468	0%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
/dev/sda2	524252	6216	518036	2%	/boot/efi
tmpfs	202292	116	202176	1%	/run/user/1000

11. du: Muestra el número de bloques usado por archivos.

```

sergio@sergio-VirtualBox:~$ du
4      ./Descargas
4      ./Escritorio
4      ./Música
4      ./Imágenes
8      ./snap/gedit/664/.config/gedit
4      ./snap/gedit/664/.config/dconf
4      ./snap/gedit/664/.config/ibus
8      ./snap/gedit/664/.config/fontconfig
4      ./snap/gedit/664/.config/gtk-3.0
4      ./snap/gedit/664/.config/gtk-2.0
52     ./snap/gedit/664/.config
68     ./snap/gedit/664/.local/share/glib-2.0/schemas
72     ./snap/gedit/664/.local/share/glib-2.0
4      ./snap/gedit/664/.local/share/icons
84     ./snap/gedit/664/.local/share
88     ./snap/gedit/664/.local
148    ./snap/gedit/664
72     ./snap/gedit/common/.cache/immodules
36     ./snap/gedit/common/.cache/fontconfig
28     ./snap/gedit/common/.cache/gio-modules
144    ./snap/gedit/common/.cache
148    ./snap/gedit/common

```

12. pstree: Muestra la ejecución de procesos a manera de árbol.

```

sergio@sergio-VirtualBox:~$ pstree
systemd--ModemManager--2*[{ModemManager}]
        --NetworkManager--2*[{NetworkManager}]
        --accounts-daemon--2*[{accounts-daemon}]
        --avahi-daemon--avahi-daemon
        --colord--2*[{colord}]
        --cron
        --cups-browsed--2*[{cups-browsed}]
        --cupsd
        --dbus-daemon
        --fwupd--4*[{fwupd}]
        --gdm3--gdm-session-wor--gdm-wayland-ses--gnome-session-b--2*[{gnom+
                |                |                |
                |                |                +--2*[{gdm-wayland-ses}]
                |                +--2*[{gdm-session-wor}]
                +--2*[{gdm3}]
        --gpg-agent
        --2*[kerneloops]
        --packagekitd--2*[{packagekitd}]
        --polkitd--2*[{polkitd}]
        --power-profiles--2*[{power-profiles-}]
        --rsyslogd--3*[{rsyslogd}]
        --rtkit-daemon--2*[{rtkit-daemon}]
        --snapd--8*[{snapd}]
        --switcheroo-cont--2*[{switcheroo-cont}]

```

trap -1 para procesar señales de entrada.

```

sergio@sergio-VirtualBox:~$ trap -1
bash: trap: -1: opción no válida
trap: modo de empleo: trap [-lp] [[arg] id_señal ...]
sergio@sergio-VirtualBox:~$

```

13. pwd: Muestra la ruta del directorio actual.

```
sergio@sergio-VirtualBox:~$ pwd
/home/sergio
sergio@sergio-VirtualBox:~$
```

14. su:

```
sergio@sergio-VirtualBox:~$ su
Contraseña:
```

15. sudo: Ejecuta comandos con privilegios de superusuario.

```
sergio@sergio-VirtualBox:~$ sudo
usage: sudo -h | -K | -k | -V
usage: sudo -v [-ABknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-ABknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
[command]
usage: sudo [-ABbEHknPS] [-r role] [-t type] [-C num] [-D directory] [-g group]
[-h host] [-p prompt] [-R directory] [-T timeout] [-u user]
[VAR=value] [-i|-s] [<command>]
usage: sudo -e [-ABknS] [-r role] [-t type] [-C num] [-D directory] [-g group]
[-h host] [-p prompt] [-R directory] [-T timeout] [-u user] file .
```

16. apt: Nos proporciona una interfaz para la gestión de paquetes de Linux.

```

sergio@sergio-VirtualBox:~$ apt
apt 2.5.3 (amd64)
Uso: apt [opciones] orden

apt es un gestor de paquetes de línea de órdenes y proporciona órdenes
para la búsqueda y gestión así como la solicitud de información sobre
los paquetes. Proporciona la misma funcionalidad que las herramientas
APT especializadas, como apt-get y apt-cache, pero permite de forma predetermina
da
opciones más apropiadas para un uso interactivo.

Órdenes más utilizadas:
list - lista los paquetes según los nombres
search - busca en las descripciones de los paquetes
show - muestra detalles del paquete
install - instala paquetes
reinstall - reinstalar paquetes
remove - elimina paquetes
autoremove - Elimina automáticamente todos los paquetes sin utilizar
update - actualiza la lista de paquetes disponibles
upgrade - actualiza el sistema instalando/actualizando paquetes
full-upgrade - actualiza el sistema eliminando/instalando/actualizando paquete

```

17. exit: Cierra la consola.

```

sergio@sergio-VirtualBox:~$ exit
exit

```

18. ps: Proporciona información acerca de un proceso con el fin de mostrar su estado.

```

sergio@sergio-VirtualBox:~$ ps
  PID TTY          TIME CMD
 2873 pts/0        00:00:00 bash
 28179 pts/0        00:00:00 ps

```

19. ps -fea: Nos permite gestionar un proceso que se encuentra en ejecución, por lo que es necesario activar el modo super usuario.

```

sergio@sergio-VirtualBox:~$ ps -fea
UID      PID     PPID  C  STIME TTY          TIME CMD
root         1         0  0  19:40 ?        00:00:04 /sbin/init splash
root         2         0  0  19:40 ?        00:00:00 [kthreadd]
root         3         2  0  19:40 ?        00:00:00 [rcu_gp]
root         4         2  0  19:40 ?        00:00:00 [rcu_par_gp]
root         5         2  0  19:40 ?        00:00:00 [slub_flushwq]
root         6         2  0  19:40 ?        00:00:00 [netns]
root         8         2  0  19:40 ?        00:00:00 [kworker/0:0H-events_highpri]
root        10         2  0  19:40 ?        00:00:00 [mm_percpu_wq]
root        11         2  0  19:40 ?        00:00:00 [rcu_tasks_kthread]
root        12         2  0  19:40 ?        00:00:00 [rcu_tasks_rude_kthread]
root        13         2  0  19:40 ?        00:00:00 [rcu_tasks_trace_kthread]
root        14         2  0  19:40 ?        00:00:01 [ksoftirqd/0]
root        15         2  0  19:40 ?        00:00:01 [rcu_preempt]
root        16         2  0  19:40 ?        00:00:00 [migration/0]
root        17         2  0  19:40 ?        00:00:00 [idle_inject/0]
root        19         2  0  19:40 ?        00:00:00 [cpuhp/0]
root        20         2  0  19:40 ?        00:00:00 [kdevtmpfs]
root        21         2  0  19:40 ?        00:00:00 [inet_frag_wq]
root        22         2  0  19:40 ?        00:00:00 [kauditd]
root        23         2  0  19:40 ?        00:00:00 [khungtaskd]

```

20. man: Nos proporciona la documentación del sistema.

```

MAN(1)                               Utilidades de paginador del manual                               MAN(1)

NOMBRE
    man - interfaz de los manuales de referencia del sistema

SINOPSIS
    man [opciones de man] [[sección] página ...] ...
    man -k [opciones de apropos] regexp ...
    man -K [opciones de man] [sección] term ...
    man -f [whatis opciones] página ...
    man -l [opciones de man] archivo ...
    man -w|-W [opciones de man] página ...

DESCRIPCIÓN
    man es el paginador de manuales del sistema. Cada argumento de página
    dado a man normalmente es el nombre de un programa, utilidad o función.
    La página de manual asociada con cada uno de estos argumentos es, pues,
    encontrada y mostrada. Si se proporciona una sección, man mirará solo
    en esa sección del manual. La acción predeterminada es buscar en todas
    las secciones disponibles siguiendo un orden predefinido (véase
    DEFAULTS), y mostrar solo la primera página encontrada, incluso si la
    página existe en varias secciones.

```

21. cat: Es un comando muy versátil que nos permite hacer un sinfín de acciones como concatenar archivos de texto, concatenar comandos, redireccionar mensajes a la salida estándar etc. En este caso se utilizó el comando “cat” para concatenar dos archivos de texto.

```

sergio@sergio-VirtualBox:~/Documentos/Pract8$ cat sistemas.txt operativos.txt
> avanzados.txt
sergio@sergio-VirtualBox:~/Documentos/Pract8$

```

```
sistemas.txt x operativos.txt avanzados.txt
Nombre: Sergio Tinoco
Grupo: 6BV1
Materia: Computo Paralelo
```

```
sistemas.txt operativos.txt x avanzados.txt
Esta es la practica 1
Esto es parecido a MS-DOS
```

```
sistemas.txt operativos.txt avanzados.txt
Nombre: Sergio Tinoco
Grupo: 6BV1
Materia: Computo Paralelo
Esta es la practica 1
Esto es parecido a MS-DOS
```

22. **cd:** Al igual que el CMD de Windows, este comando nos permite navegar entre los directorios del sistema.

```
sergio@sergio-VirtualBox:~/Documentos/Pract8$ cd ubuntu/
sergio@sergio-VirtualBox:~/Documentos/Pract8/ubuntu$
```

23. **ls:** Nos proporciona la lista de archivos, carpetas, ejecutables y archivos comprimidos de la carpeta actual.

```
sergio@sergio-VirtualBox:~/Documentos/Pract8$ ls
avanzados.txt  kubuntu  lubuntu  operativos.txt  sistemas.txt  ubuntu  xubuntu
sergio@sergio-VirtualBox:~/Documentos/Pract8$
```

24. **ls -la:** Nos proporciona una vista de todos los archivos y el número de elementos dentro de cada carpeta a partir de la ruta actual, así como la última hora de modificación.

```
sergio@sergio-VirtualBox:~/Documentos/Pract8$ ls -la
total 36
drwxrwxr-x 6 sergio sergio 4096 mar 12 20:08 .
drwxr-xr-x 3 sergio sergio 4096 mar 12 00:09 ..
-rw-rw-r-- 1 sergio sergio 108 mar 12 20:08 avanzados.txt
drwxrwxr-x 3 sergio sergio 4096 mar 12 00:11 kubuntu
drwxrwxr-x 3 sergio sergio 4096 mar 12 00:12 lubuntu
-rw-rw-r-- 1 sergio sergio 48 mar 12 00:25 operativos.txt
-rw-rw-r-- 1 sergio sergio 60 mar 12 00:23 sistemas.txt
drwxrwxr-x 3 sergio sergio 4096 mar 12 00:10 ubuntu
drwxrwxr-x 3 sergio sergio 4096 mar 12 00:12 xubuntu
sergio@sergio-VirtualBox:~/Documentos/Pract8$
```

25. **rm:** Elimina un archivo o carpeta vacío.


```
rm: no se puede borrar 'xubuntu/': ES un directorio
sergio@sergio-VirtualBox:~/Documentos/Pract8$ rm sistemas.txt
sergio@sergio-VirtualBox:~/Documentos/Pract8$
```

```
sergio@sergio-VirtualBox:~/Documentos/Pract8$ ls
avanzados.txt  kubuntu  lubuntu  operativos.txt  ubuntu  xubuntu
sergio@sergio-VirtualBox:~/Documentos/Pract8$
```

26. mv: Cambia de carpeta el archivo indicado, ya sea usando rutas absolutas o relativas.

```
sergio@sergio-VirtualBox:~/Documentos/Pract8$ mv operativos.txt ..
sergio@sergio-VirtualBox:~/Documentos/Pract8$ ls
avanzados.txt  kubuntu  lubuntu  ubuntu  xubuntu
sergio@sergio-VirtualBox:~/Documentos/Pract8$
```

27. more: Proporciona una vista del contenido de un archivo por pantalla hasta que el usuario le indique que cambie de vista.

```
sergio@sergio-VirtualBox:~/Documentos/Pract8$ more avanzados.txt
Nombre: Sergio Tinoco
Grupo: 6BV1
Materia: Computo Paralelo
Esta es la practica 1
Esto es parecido a MS-DOS
```

28. less: Similar a more, muestra la información de un archivo en una pantalla a la vez hasta que el usuario le indique cambiar de pantalla agregando la posibilidad de poder desplazarse por el fichero del archivo.

```
Nombre: Sergio Tinoco
Grupo: 6BV1
Materia: Computo Paralelo
Esta es la practica 1
Esto es parecido a MS-DOS
avanzados.txt (END)
```

29. mkdir: Similar al CMD permite la creación de nuevos directorios ya sea con rutas absolutas o relativas.

```
sergio@sergio-VirtualBox:~/Documentos/Pract8$ mkdir nueva
sergio@sergio-VirtualBox:~/Documentos/Pract8$ ls
avanzados.txt  kubuntu  lubuntu  nueva  ubuntu  xubuntu
sergio@sergio-VirtualBox:~/Documentos/Pract8$
```


30. pico: Proporciona un editor de texto para Unix.



```
GNU nano 6.4                               Nuevo búfer

[ Bienvenido a nano.  Para obtener ayuda básica, pulse Ctrl+G. ]
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich.^_ Reemplazar  ^U Pegar      ^J Justificar ^/ Ir a línea
```

31. nano: Similar a pico proporciona un editor de texto para Unix, con la principal diferencia de que nano es más simple y por ende mejor de entender.

```

                                Texto de ayuda principal de nano

El editor nano está diseñado para emular la funcionalidad y sencillez de
uso del editor de texto UW Pico. El editor cuenta con cuatro secciones
principales. La línea superior muestra la versión del programa, el nombre
del fichero que se está editando, y si ha sido modificado o no. La
ventana principal del editor muestra lo que está siendo editado. La línea
de estado es la tercera empezando por abajo y muestra mensajes
importantes. Las dos últimas líneas muestran las combinaciones de teclas
usadas más a menudo en el editor.

Las combinaciones de teclas se escriben como sigue: las secuencias con la
tecla Control se indican con '^' y se pueden introducir tanto pulsando la
tecla Ctrl como pulsando dos veces Esc. Las secuencias con la tecla Meta
se indican con 'M-' y se pueden introducir con las teclas Alt, Cmd o Esc,
dependiendo de su configuración de teclado. Además, si pulsa dos veces
Esc y escribe después un código decimal de tres dígitos entre 000 y 255,
introducirá el carácter de valor correspondiente. Dispone de las
siguientes pulsaciones en la ventana principal del editor. Las
pulsaciones alternativas se muestran entre paréntesis:

^L Actualizar ^W Buscar      M-Q Anterior ^P Línea ant. ^Y Pág. ant. M-\ Pri. línea
^X Cerrar     ^O Buscar atr M-W Siguiente ^N Línea sig. ^V Pág. sig. M-/ Últ. línea

```

32. vi: Proporciona un editor de texto más funcional.

```

sergio@sergio-VirtualBox: ~/Documentos/Pract8
VIM - Vi IMproved

version 9.0.242
by Bram Moolenaar et al.
Modified by team+vim@tracker.debian.org
Vim is open source and freely distributable

Become a registered Vim user!
type  :help register<Enter>   for information

type  :q<Enter>               to exit
type  :help<Enter> or <F1>    for on-line help
type  :help version9<Enter>   for version info

Running in Vi compatible mode
type  :set nocp<Enter>        for Vim defaults
type  :help cp-default<Enter> for info on this

```

Rutas absolutas:

Las rutas absolutas se definen desde la carpeta raíz (en este caso la carpeta home) hasta la carpeta o directorio donde se va a trabajar. Podría decirse que está definida de forma lineal.

Ejemplos:

El usuario se desplaza a la carpeta lubuntu.

```
sergio@sergio-VirtualBox:~$ cd /home/sergio/Documentos/Pract8/lubuntu/  
sergio@sergio-VirtualBox:~/Documentos/Pract8/lubuntu$
```

El usuario se desplaza a la carpeta descargas.

```
sergio@sergio-VirtualBox:~/Documentos/Pract8/lubuntu$ cd /home/sergio/Descargas/  
sergio@sergio-VirtualBox:~/Descargas$
```

El usuario se desplaza a la carpeta Tareas del escritorio.

```
sergio@sergio-VirtualBox:~/Descargas$ cd /home/sergio/Escritorio/Tareas/  
sergio@sergio-VirtualBox:~/Escritorio/Tareas$
```

Rutas relativas:

Las rutas relativas se definen a partir de la carpeta actual en el bash, es decir la carpeta en la que se encuentra el usuario, estas van a cambiar dependiendo la carpeta en la que el usuario se encuentre, ya que la carpeta de interés puede estar localizada fuera de la misma o bien dentro de alguna carpeta del directorio actual.

Ejemplos:

El usuario se desplaza a la carpeta ubuntu desde la carpeta Tareas.

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas$ cd ../../Documentos/Pract8/ubuntu/  
sergio@sergio-VirtualBox:~/Documentos/Pract8/ubuntu$
```

El usuario se desplaza a la carpeta versiones de Xubuntu desde la carpeta ubuntu.

```
sergio@sergio-VirtualBox:~/Documentos/Pract8/ubuntu$ cd ../xubuntu/versiones/16/  
sergio@sergio-VirtualBox:~/Documentos/Pract8/xubuntu/versiones/16$
```

El usuario se desplaza a la carpeta lubuntu desde la carpeta versiones de Xubuntu.

```
sergio@sergio-VirtualBox:~/Documentos/Pract8/xubuntu/versiones/16$ cd ../../../lubuntu/  
sergio@sergio-VirtualBox:~/Documentos/Pract8/lubuntu$
```

Como se pudo observar las rutas relativas suelen ser más simples que las absolutas, sin embargo es importante conocer el árbol de directorios en el que nos desplazamos y de igual forma saber aplicar el comando `cd ../` para desplazarse una o más carpetas hacia afuera.

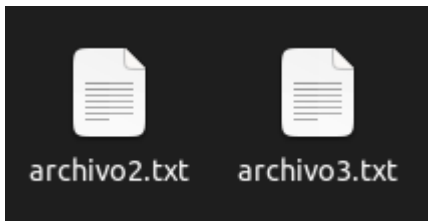
Borrado de archivos.

El borrado de archivos puede aplicarse ya sea por medio de rutas absolutas o relativas.

Ejemplos:

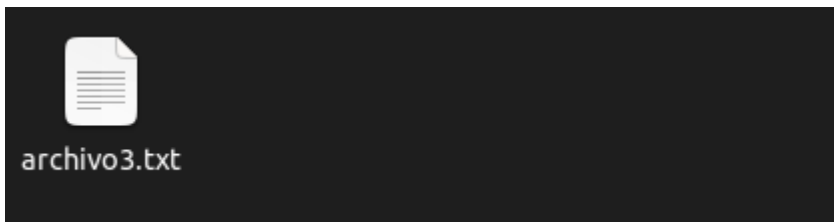
El usuario elimina el archivo1 utilizando el operador ? como comodín de un carácter.

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas/c1$ rm archiv?1.txt
sergio@sergio-VirtualBox:~/Escritorio/Tareas/c1$
```



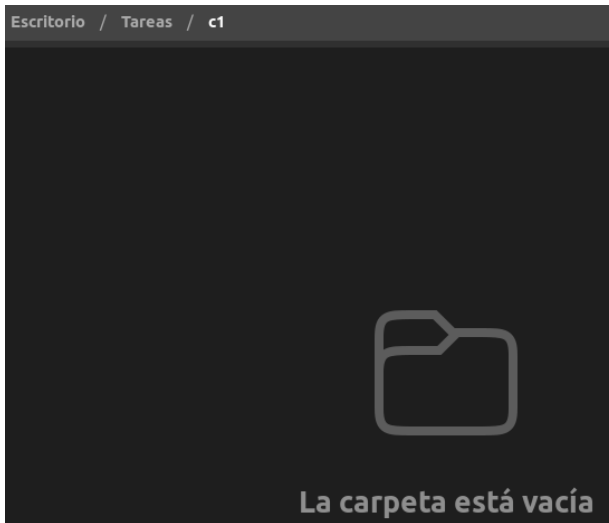
El usuario elimina el archivo 2 haciendo uso de rutas relativas y añadiendo el operador * en la extensión del archivo.

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas/c1$ cd ..
sergio@sergio-VirtualBox:~/Escritorio/Tareas$ rm ./c1/archivo2.*
```



El usuario elimina un archivo con extensión .txt haciendo uso de rutas absolutas.

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas$ rm /home/sergio/Escritorio/Tareas/c1/*.txt
sergio@sergio-VirtualBox:~/Escritorio/Tareas$
```



Redireccionamiento.

El redireccionamiento nos permite mandar a la salida estándar 2 o más elementos como archivos de texto o salidas de algún proceso.

Ejemplos:

El usuario manda a un archivo la salida del texto “Hola Mundo”

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas$ echo "Hola Mundo" > archivo.txt
sergio@sergio-VirtualBox:~/Escritorio/Tareas$
```

```
Hola Mundo
archivo.txt (END)
```

El usuario agrega el texto “Esta es la práctica 1” al archivo previo.

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas$ echo "Esta es la practica 1" >> archivo.txt
sergio@sergio-VirtualBox:~/Escritorio/Tareas$ less archivo.txt
sergio@sergio-VirtualBox:~/Escritorio/Tareas$
```

```
Hola Mundo
Esta es la practica 1
archivo.txt (END)
```

Instalación de Software.

La instalación de programas o paquetes se puede realizar ya sea por medio de comandos o por medio de la interfaz del sistema operativo.

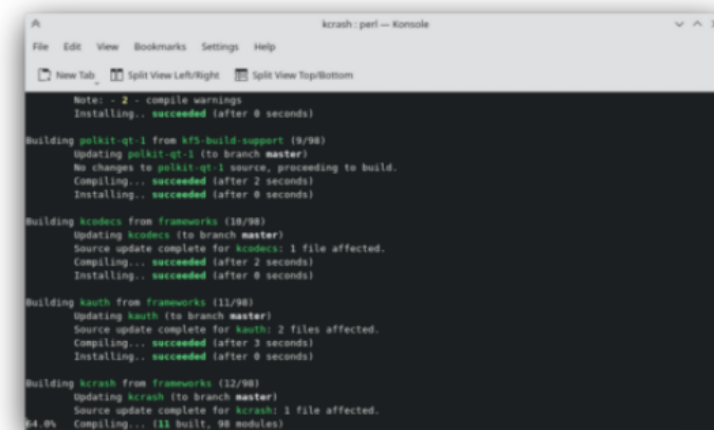
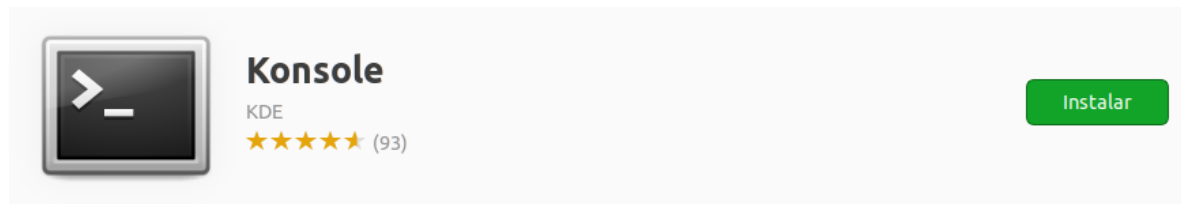
Ejemplos:

Por medio de comandos se utiliza el comando `sudo apt install` junto con el nombre del paquete o programa a instalar.

```
sergio@sergio-VirtualBox:~/Escritorio/areas$ sudo apt install gimp
[sudo] contraseña para sergio:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
gimp-data graphviz i965-va-driver intel-media-va-driver libaac0 libamd2 libann0 libaom3 libavcodec59 libavformat59 libavutil57
libbabi-0.1-0 libbdplus0 libblas3 libbluray2 libcamd2 libccolamd2 libcdt5 libcggraph6 libcholmod3 libchromaprint1 libcjson1
libcodecs2-1.0 libcolamd2 libdavid6 libde265-0 libegl-mesa0 libgbm1 libgegl-0.4-0 libgegl-common libgfortran5 libgimp2.0
libgl1-mesa-dri libglapi-mesa libglx-mesa0 libgme0 libgsm1 libgts-0.7-5 libgts-bin libgvc6 libgvpr2 libheif1 libigdgmm12
libinmath-3-1-29 liblab-gamut1 liblapack3 libmbedcrypto7 libmetis5 libmfx1 libmng2 libmypaint-1.5-1 libmypaint-common libnorm1
libopenexr-3-1-30 libopenmpt0 libpathplan4 libpgm-5.3-0 libquadmath0 librabbitmq4 libraw20 librist4 libshine3 libsnappy1v5
libsodium23 libsoxr0 libsrtp1.5-gnutls libssh-gcrypt-4 libsuitesparseconfig5 libsvtav1enc1 libswresample4 libswscale6 libudfread0
libunifpack5 libva-drm2 libva-x11-2 libva2 libvdpau1 libx264-164 libx265-199 libxatracker2 libxvidcore4 libzmq5 libzvb1-common
```

Por medio de la interfaz.

Haciendo uso de la tienda de aplicaciones de Ubuntu se instala un programa, en este caso “Konsole”

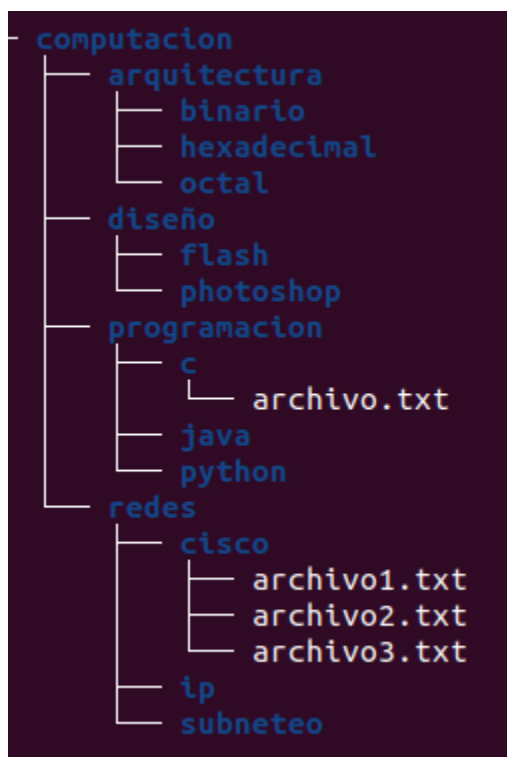


Mover y copiar archivos.

Para modificar la estructura del árbol de directorios se puede hacer uso de comandos como mv (mover) o cp (copiar), ya sean archivos o carpetas. Esto por medio de rutas absolutas o relativas.

Ejemplos:

Para los ejemplos se tiene el siguiente árbol de directorios.



Por medio de rutas absolutas el usuario copia el archivo de texto a la carpeta C de programación.

```
sergio@sergio-VirtualBox: ~/Escritorio/Tareas/practica1/computacion$ touch archivo.txt
sergio@sergio-VirtualBox: ~/Escritorio/Tareas/practica1/computacion$ cp /home/sergio/Escritorio/Tareas/practica1/computacion/archivo.txt /home/sergio/Escritorio/Tareas/practica1/computacion/programacion/c/
```

Haciendo uso de rutas absolutas, el usuario copia el archivo de texto y lo renombra 3 veces dentro de la carpeta cisco.

```
computacion$ cp archivo.txt redes/cisco/archivo2.txt
computacion$ cp archivo.txt redes/cisco/archivo3.txt
computacion$ cp archivo.txt redes/cisco/archivo1.txt
```

```
redes/cisco/archivo1.txt
sergio@sergio-VirtualBox: ~/Escritorio/Tareas/practica1/computacion$ ls redes/cisco/
archivo1.txt  archivo2.txt  archivo3.txt
```

Por medio de rutas relativas y el comodín *, se copiará el contenido de la carpeta cisco a la carpeta actual (computación).

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion$ cp -r redes/cisco/*.* .
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion$ ls
archivo1.txt  archivo2.txt  archivo3.txt  archivo.txt  arquitectura  diseño  programacion  redes
```

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion$ cp -r redes/cisco/*.* .
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion$ ls
archivo1.txt  archivo2.txt  archivo3.txt  arquitectura  diseño  programacion  redes
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion$
```

Por medio de rutas absolutas y el parámetro -r se va a copiar la carpeta cisco junto con el contenido que hay dentro.

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion$ cp -r redes/cisco .
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion$ ls
archivo1.txt  archivo2.txt  archivo3.txt  arquitectura  cisco  diseño  programacion  redes
```

Utilizando el comando mv, se va a mover la carpeta cisco a redes cambiando el nombre a “cisconuevo”.

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion$ mv cisco redes/cisconuevo
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion$ cd redes/
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion/redes$ ls
cisco  cisconuevo  ip  subneteo
```

Haciendo uso de rutas relativas y el comando mv se va a mover el contenido de la carpeta cisco original a la carpeta de la práctica.

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion/redes$ mv cisco/*.*txt ../../
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1/computacion/redes$ cd ../../
sergio@sergio-VirtualBox:~/Escritorio/Tareas/practica1$ ls
archivo1.txt  archivo2.txt  archivo3.txt  computacion
```

Agregar un usuario.

Para agregar un usuario se puede hacer uso del comando adduser o useradd, en este caso se utilizó el comando useradd con el cual se proporcionó un nombre de usuario y contraseña así como los privilegios con el que el nuevo usuario cuenta que en este caso es sudo (super usuario).


```

sergio@sergio-VirtualBox:~$ sudo useradd -m serch
[sudo] contraseña para sergio:
sergio@sergio-VirtualBox:~$ sudo passwd serch
sudo: passwd: orden no encontrada
sergio@sergio-VirtualBox:~$ sudo passwd serch
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 8 caracteres
Vuelva a escribir la nueva contraseña:
Las contraseñas no coinciden.
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 8 caracteres
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
sergio@sergio-VirtualBox:~$ sudo chsh -s /bin/bash serch
sergio@sergio-VirtualBox:~$ sudo usermod -a -G sudo serch
sergio@sergio-VirtualBox:~$ cat /etc/passwd

```

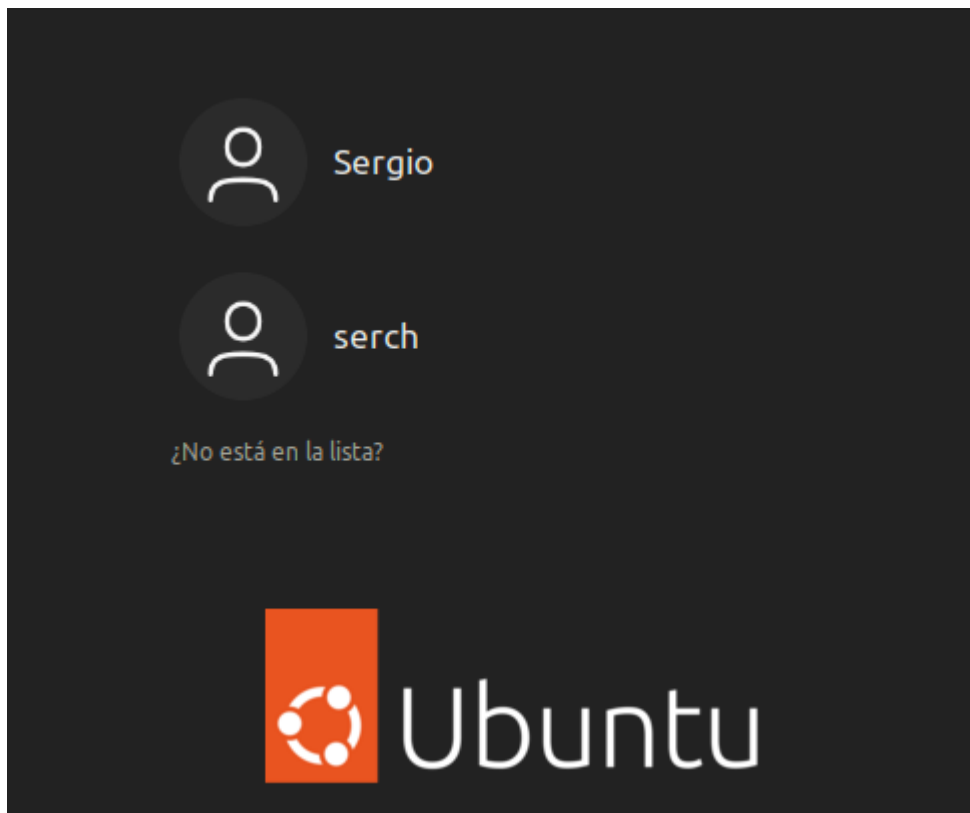
Bash de los 2 usuarios actuales.

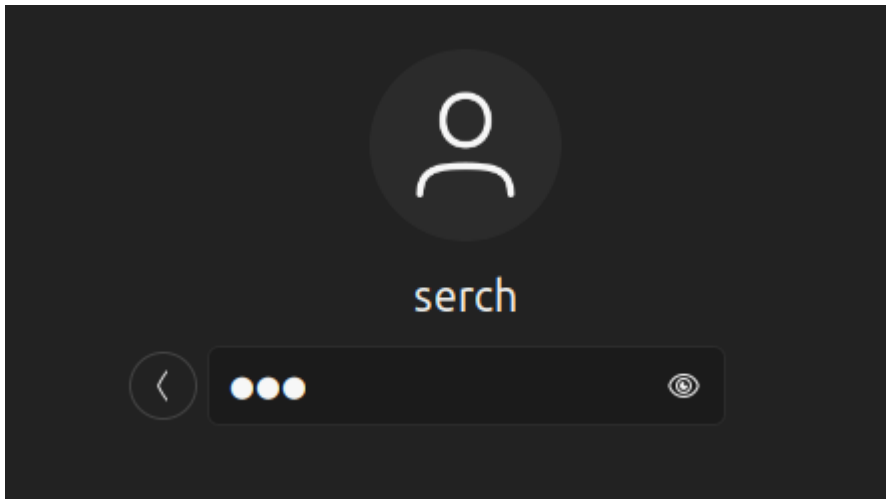
```

sergio:x:1000:1000:Sergio,,,:/home/sergio:/bin/bash
serch:x:1001:1001::/home/serch:/bin/bash
sergio@sergio-VirtualBox:~$ █

```

Ahora aparece el nuevo usuario “Serch”





Si abrimos la terminal y ejecutamos los comandos “whoami” y “pwd” tenemos lo siguiente.

```
serch@sergio-VirtualBox:~$ whoami
serch
serch@sergio-VirtualBox:~$ pwd
/home/serch
serch@sergio-VirtualBox:~$
```

Sin cerrar la sesión de la nueva cuenta cambiamos al usuario original “sergio” y eliminamos al usuario “serch”.

```
sergio@sergio-VirtualBox:~$ sudo userdel -f serch
userdel: user serch is currently used by process 2798
sergio@sergio-VirtualBox:~$ sudo killall -u serch && sudo deluser --remove-home -f serch
No se puede encontrar al usuario serch
```

El usuario fue eliminado exitosamente.

```
sergio:x:1000:1000:Sergio,,,:/home/sergio:/bin/bash
sergio@sergio-VirtualBox:~$
```

Creación y compilación de programas en C usando el editor nano.

Se abre el archivo con el comando nano.

```
s/programasC$ nano holamundo.c  
s/programasC$
```

Se programa el código.

```
#include<stdio.h>  
#include<stdlib.h>  
  
int main()  
{  
  
    printf("Hola Mundo");  
  
    return 0;  
}
```

Se compila y ejecuta el programa.

```
Hola Mundosergio@sergio-VirtualBox:~/Escritorio/Tareas/programasC$ gcc holamundo.c -o holamundo  
sergio@sergio-VirtualBox:~/Escritorio/Tareas/programasC$ ./holamundo  
Hola Mundosergio@sergio-VirtualBox:~/Escritorio/Tareas/programasC$
```

Creación y compilación de programas en C usando el editor gedit.

Se abre el archivo con el comando gedit.

```
holamundo  
gramasC$ gedit holamundo.c
```

Se programa el código.

```
1 #include<stdio.h>  
2 #include<stdlib.h>  
3  
4 int main()  
5 {  
6  
7     printf("Hola Mundo\n");  
8  
9     return 0;  
10 }
```

Se compila y ejecuta el programa.

```
sergio@sergio-VirtualBox:~/Escritorio/Tareas/programasC$ ./holamundo  
Hola Mundo  
sergio@sergio-VirtualBox:~/Escritorio/Tareas/programasC$
```

Conclusiones

Linux es un lenguaje muy versátil que nos permite desarrollar e implementar múltiples algoritmos además de la gestión de procesos como el caso de la creación o eliminación de un usuario del sistema que va ligado a un conjunto de protocolos de procesos.

En esta práctica se observó cómo es que linux implementa sus comandos y es sensible a las minúsculas, ya que la mayoría de los comandos suelen ir en minúsculas y pueden ir acompañados de parámetros que de igual forma cambia su comportamiento. Además de que tiene sus protocolos de seguridad como el caso de la existencia de un super usuario con el que se puede acceder a múltiples opciones en la línea de comandos.

Referencias

- [1] "Máquina virtual," Wikipedia, La enciclopedia libre. 27-Oct-2021. [Online]. Available: https://es.wikipedia.org/wiki/M%C3%A1quina_virtual#cite_note-1. [Accessed: Mar. 13, 2023].
- [2] A. G. Sánchez, "Módulo 02 - Introducción a GNU/Linux," Universidad Nacional del Sur, Bahía Blanca, Argentina, 2018. [Online]. Available: <http://cs.uns.edu.ar/~ags/linux/material/M%C3%B3dulo-02.pdf>. [Accessed: Mar. 13, 2023].
- [3] "Linux Installation: Hardware Requirements & Distribution," Study.com, [Online]. Available: <https://study.com/academy/lesson/linux-installation-hardware-requirements-distribution.html>. [Accessed: Mar. 13, 2023].
- [4] J. Garrison, "The Year of the Linux Desktop," J. Garrison, 02-Feb-2021. [Online]. Available: <https://justingarrison.com/blog/year-of-linux-desktop/>. [Accessed: Mar. 13, 2023].
- [5] "Global Desktop Operating System Market Share | StatCounter Global Stats." [Online]. Available: <https://gs.statcounter.com/os-market-share/desktop/worldwide>. [Accessed: Mar. 13, 2023].
- [6] J. M. Fernández, "GNOME vs KDE ¿Cuál es el mejor escritorio?," Maslinux, 04-Oct-2017. [Online]. Available: <https://maslinux.wordpress.com/2017/10/04/gnome-vs-kde-cual-es-el-mejor-escritorio/>. [Accessed: Mar. 13, 2023].
- [7] "Diferencia entre CLI y GUI," Gadget-Info, [Online]. Available: <https://es.gadget-info.com/difference-between-cli>. [Accessed: Mar. 13, 2023].
- [8] "Introducción a la línea de comandos," The Carpentries, [Online]. Available: <https://swcarpentry.github.io/shell-novice-es/01-intro/>. [Accessed: Mar. 13, 2023].
- [9] "Tipos de usuarios y grupos, directorios e IDs," TICARTE, [Online]. Available: <https://www.ticarte.com/contenido/tipos-de-usuarios-y-grupos-directorios-e-ids>. [Accessed: Mar. 13, 2023].

- [10] "Rutas relativas y rutas absolutas en Linux," Zeppelinux, [Online]. Available: <https://www.zeppelinux.es/rutas-relativas-y-rutas-absolutas-en-linux/>. [Accessed: Mar. 13, 2023].
- [11] "Redireccionamiento en Linux," Linux Total, [Online]. Available: <https://www.linuxtotal.com.mx/index.php?cont=redireccionamiento-en-linux>. [Accessed: Mar. 13, 2023].
- [12] B. Ward, "A. Clasificación de comandos," in How Linux Works, 2nd ed., San Francisco, CA: No Starch Press, 2015, pp. 537-540. [Online]. Available: <http://etutorials.org/Linux+systems/how+linux+works/Appendix+A+Command+Classification/>. [Accessed: Mar. 13, 2023].
- [13] "Variables de entorno en Linux," Hostinger, [Online]. Available: <https://www.hostinger.mx/tutoriales/variables-de-entorno-linux>. [Accessed: Mar. 13, 2023].