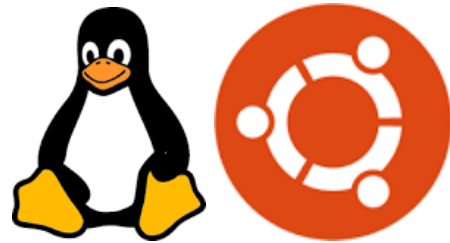


Linux



Introducción

Linux fue creado en 1991 por Linus Torvalds, inspirado en el proyecto Minix, el cual era un clon de Unix. Si bien comúnmente se dice que Linux es un sistema operativo “Unix” en realidad nos referimos a que es “tipo Unix”, ya que no contiene código Unix en él. Aunque fue originalmente desarrollado para computadoras personales, pero actualmente es el principal sistema operativo de servidores.

Ubuntu

Ubuntu es una “distribución” de Linux. Veamos Linux o “el kernel” es el sistema operativo, que en realidad solo sabe administrar procesos, la memoria de estos procesos, y saber cómo interactuar con los dispositivos de la computadora donde corre el S.O. así los procesos pueden trabajar.

Estos procesos son herramientas o utilitarios que nos permiten almacenar datos, ver estos datos, transferirlos de un dispositivo a otro, imprimirlos o borrarlos.

Estos no son uno, son una colección de utilitarios que van desde intérpretes de comandos (bash, csh), utilitarios para manejar impresoras (lp, lpstat), entornos gráficos (kde, gnome, mate) hasta manejadores de base de datos (MySQL, PostgreSQL). Estos utilitarios deben ser empaquetados modularmente para poder ser instalados fácilmente como así también, mantenidos por un equipo de gente para updates de seguridad por ejemplo.

Estos paquetes tienen además una forma de administración para facilitar la instalación y también el update. A este conjunto de paquetes de utilidades, más el S.O: que también es considerado parte de los utilitarios se le llama “distribuciones”, “Una Distro” para los más audaces.

La historia nos cuenta que hay distros que empaquetan solo fuentes y los compilan en el momento de instalación, y otras que empaquetan binarios, compilado de los fuentes. Algunas hacen las 2 cosas.

Dentro de las que empaquetan binarios hay 2 grandes, RPM (Red Hat) y DEB (Debian).

Ubuntu es una distribución basada en Debian.

¿Por qué no usar Windows con WSL o VM?

Si bien a lo largo de los años WSL o el Subsistema de Windows para Linux ha mejorado, sigue sin ser una forma viable de desarrollar software. Tiene muchos problemas de configuración y su funcionamiento puede variar mucho de un sistema a otro, lo que lo hace inviable como una alternativa real a usar Ubuntu.

Si bien Ubuntu en una Máquina Virtual (VM) funciona bien, la cantidad de recursos disponibles hace que tampoco sea una opción viable, aunque puede ser usada para “practicar” la instalación y uso general del sistema operativo.

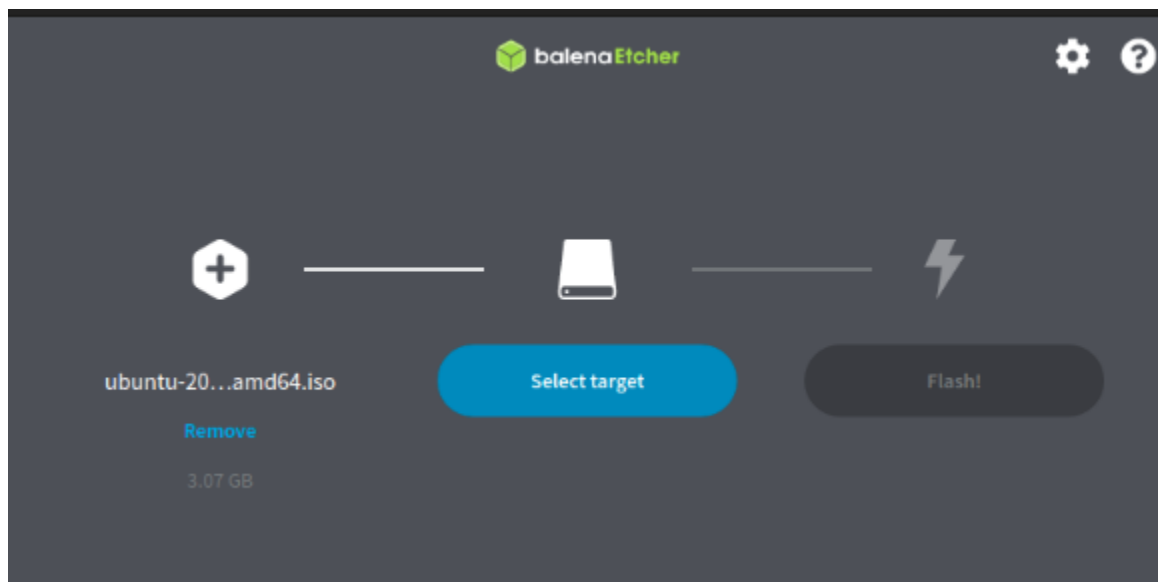
Instalación

Descargar ISO

El primer paso para instalar Ubuntu es dirigirse a la página de descargas (<https://ubuntu.com/download/desktop>) y descargar una imagen ISO de Ubuntu. Lo mejor siempre es descargar la última versión LTS, ya que ésta es la versión de soporte a largo plazo.

Generar pendrive booteable

Luego de descargar la imagen, debemos hacer un pendrive booteable. Para esto usamos balenaEtcher, que podemos descargar de <https://www.balena.io/etcher/>.



Elegimos la ISO que descargamos, luego el pendrive y finalmente creamos el disco booteable.

Bootear con el pendrive

Insertamos el pendrive en nuestra computadora y booteamos desde éste. Si en lugar de esto inicia nuestro sistema operativo normal, será necesario reiniciar la computadora y presionar F12 mientras se inicia para seleccionar el dispositivo USB desde el menú de boot.

Una vez iniciada la instalación, llegaremos a esta pantalla:



Desde aquí podremos o instalar ubuntu o probarlo ejecutándolo desde el pendrive. En nuestro caso, vamos a instalarlo. Seguimos los pasos, hasta que el Wizard nos pide elegir cuánto software queremos como base. Podemos hacer una instalación normal que instalara muchos programas de utilidades, como paquete de oficina y reproductores multimedia, pero podemos empezar con una instalación mínima y más adelante instalar estas aplicaciones.

Actualizaciones y otro software

¿Qué aplicaciones le gustaría instalar para comenzar?

☐ Instalación normal

Navegador web, utilidades, paquete de oficina, juegos y reproductores multimedia.

☒ Instalación mínima

Navegador web y utilidades básicas.

Otras opciones

☒ Descargar actualizaciones al instalar Ubuntu

Esto ahorrará tiempo después de la instalación.

☒ Instalar programas de terceros para hardware de gráficos y de wifi y formatos multimedia adicionales

Este software está sujeto a los términos de licencia incluidos en su documentación. Algunos componentes son privativos.

Salir

Atrás

Continuar

Instalar con dual boot con Windows

El siguiente paso, debemos elegir qué tipo de instalación queremos. Si preferimos mantener Windows instalado en nuestro sistema, debemos elegir la opción de “Instalar Ubuntu junto a Windows”.

Tipo de instalación

Este equipo tiene instalado actualmente Windows 10. ¿Qué quiere hacer?

- ☒ **Instalar Ubuntu junto a Windows 10**
Se conservarán todos los documentos, música y demás archivos. Podrá elegir qué sistema operativo iniciar cada vez que arranque su equipo.
- ☐ **Borrar disco e instalar Ubuntu**
Atención: esto eliminará todos sus programas, documentos, fotos, música y demás archivos en todos los sistemas operativos.

Funcionalidades avanzadas... Ninguna seleccionada
- ☐ **Más opciones**
Puede crear particiones, redimensionarlas o elegir varias particiones para Ubuntu.

Salir

Atrás

Continuar


Si instalamos Ubuntu en el mismo disco que Windows (si está instalado en un SSD, sería la opción preferible), debemos ajustar el tamaño de ambos sistemas operativos.

Instalar Ubuntu junto a Windows 10

Seleccione la unidad: SCSI3 (0,0,0) (sda) - 26.8 GB ATA VBOX HARDDISK

14.2 GB ▾

Asigne el espacio en disco arrastrando el siguiente divisor:



Archivos (12.0 GB)
/dev/sda2 (ntfs)
13.1 GB



Ubuntu
/dev/sda3 (ext4)
13.2 GB

Se ocultaron 2 particiones más pequeñas; utilice la [herramienta de particionado avanzada](#) para mayor control

Salir

Atrás

Instalar ahora


Seguimos los siguientes pasos y cuando llegamos a la sección de “¿Quién es usted?”, introducimos nuestros datos

¿Quién es usted?

Su nombre: ✓

El nombre de su equipo: ✓
El nombre que utiliza al comunicarse con otros equipos.

Elija un nombre de usuario: ✓

Elija una contraseña:  **Contraseña fuerte**

Confirme su contraseña: ✓

☐ Iniciar sesión automáticamente

☒ Solicitar mi contraseña para iniciar sesión

☐ Utilizar Active Directory

Deberá proveer el dominio y otros datos en el siguiente paso.

Luego de que la instalación se complete, al reiniciar nos encontraremos con un menú como éste:

GNU GRUB versión 2.06

*Ubuntu

Advanced options for Ubuntu

Memory test (memtest86+.elf)

Memory test (memtest86+.bin, serial console)

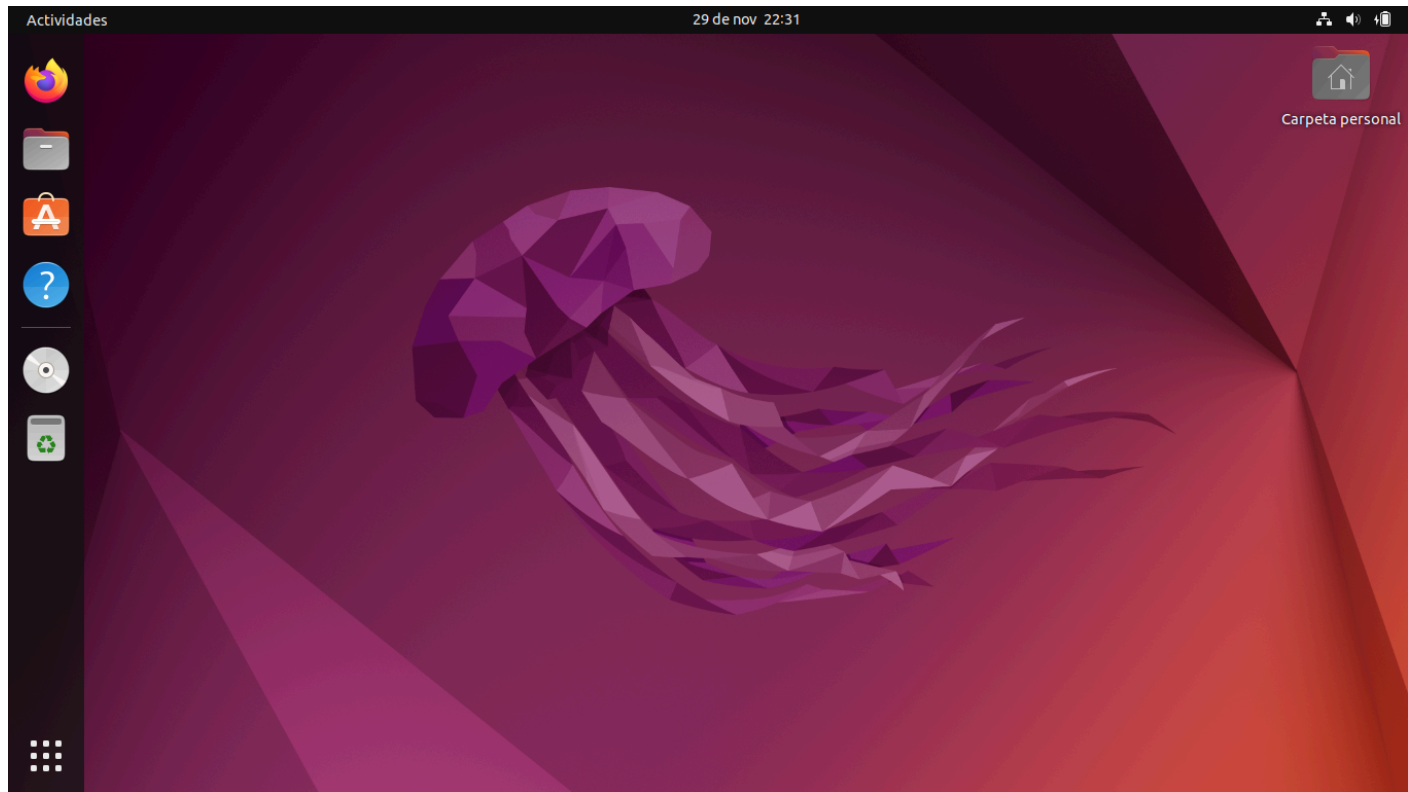
Windows 10 (on /dev/sda1)

Use las teclas ↑ y ↓ para seleccionar la entrada marcada.
Pulse «Intro» para arrancar el SO seleccionado, «e» para editar
los órdenes antes de arrancar o «c» para una línea de órdenes.
La entrada marcada se ejecutará automáticamente en 28 s.

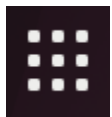
Esto nos permitirá elegir entre Ubuntu y Windows para operar.

El desktop de Ubuntu

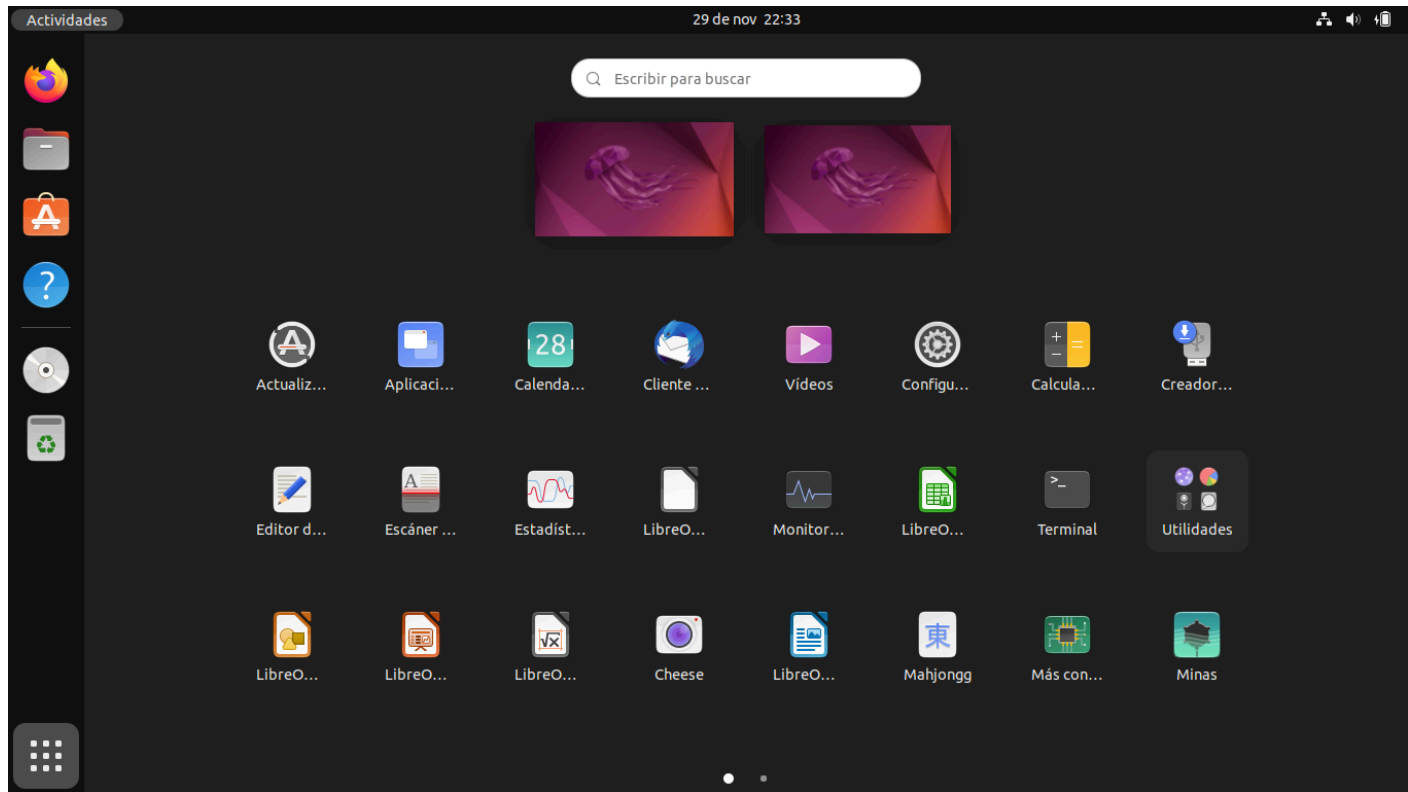
El escritorio de Ubuntu es bastante sencillo de usar. A la izquierda, tenemos el dock con las aplicaciones ‘favoritas’ o en ejecución.



Para acceder al resto de las aplicaciones, hay que hacer clic en el botón de abajo.

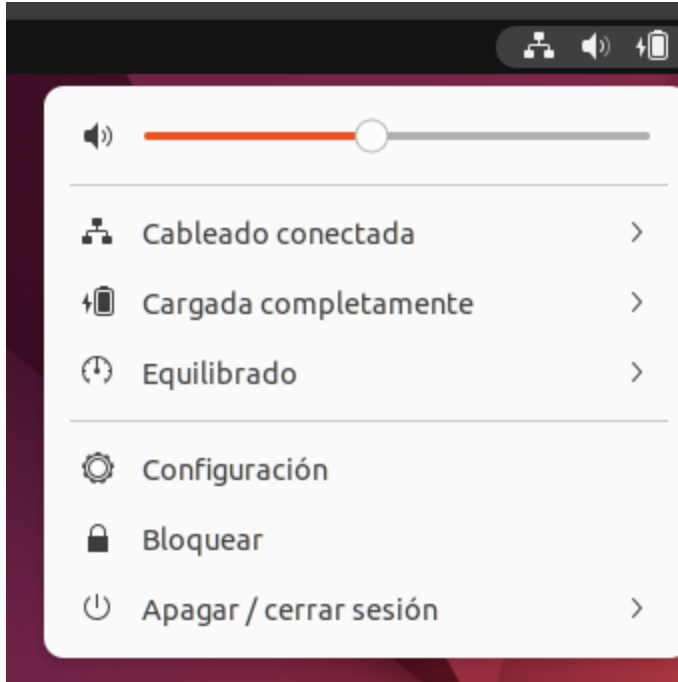


Al hacerle clic se mostrará el listado de aplicaciones instaladas. Si queremos hacer que alguna de estas aplicaciones aparezca en el dock aunque no esté abierta, será necesario agregarla a favoritos. Podemos hacer esto haciendo clic derecho y luego “Agregar a favoritos”.



Configuración

Haciendo clic en el grupo de iconos que se ven arriba a la derecha, se abre un menú de configuración. Desde aquí podemos configurar energía, Internet y demás. También podemos apagar, suspender y bloquear el equipo.

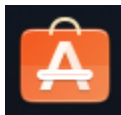


Desde el botón “**Configuración**” podemos editar opciones más avanzadas, desde cambiar el fondo de pantalla y apariencia general, hasta conectar dispositivos bluetooth y mucho más.

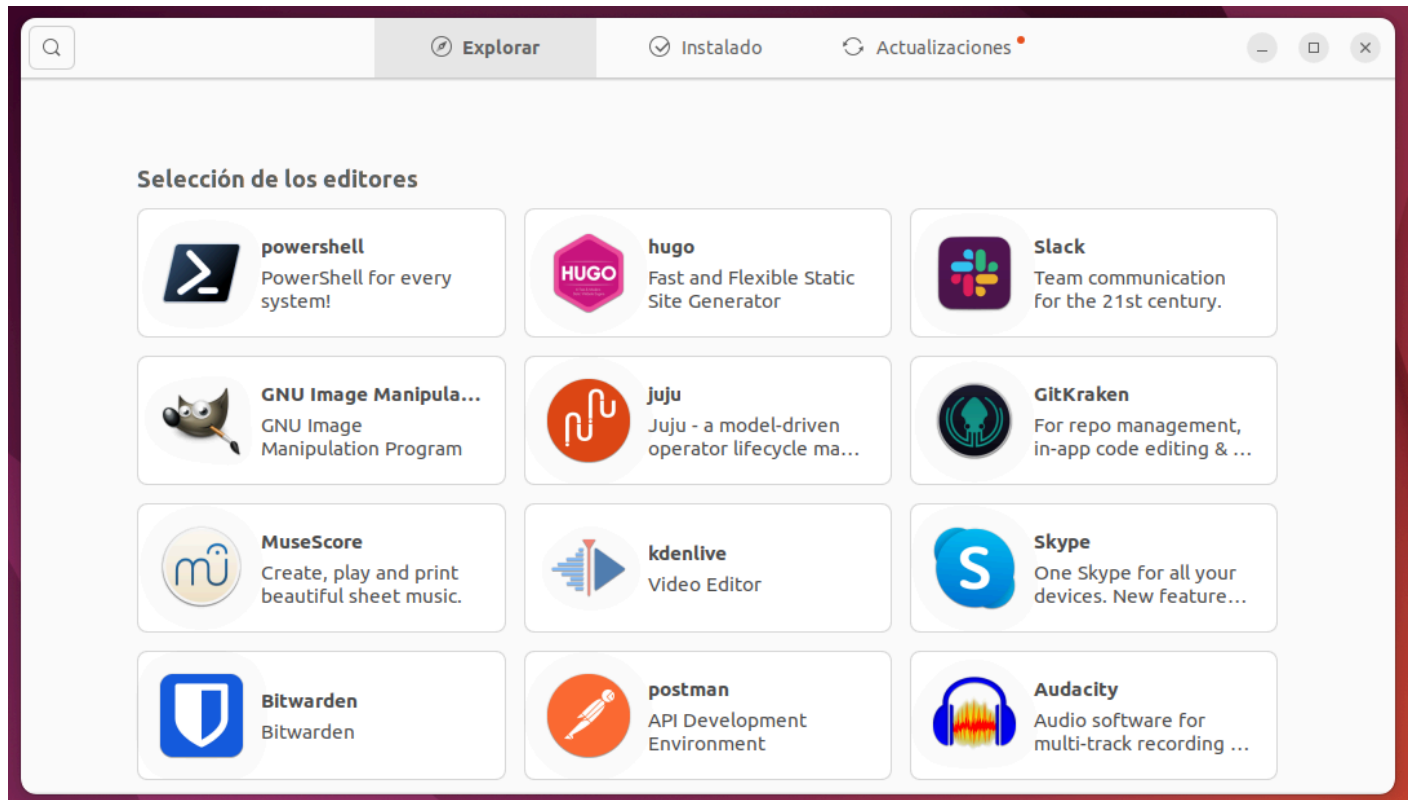
Instalación de aplicaciones

Instalación con la tienda

Se pueden instalar toda clase de aplicaciones desde la tienda de Ubuntu.



Desde la página de inicio de la tienda podemos, elegir una de las aplicaciones seleccionadas por los editores, usar la lista de categorías, o usar la lupa para encontrar lo que estamos buscando.



Seleccionamos la aplicación que queremos y luego hacemos clic en el botón **“Instalar”**. Una vez descargada la aplicación estará disponible en el menú de aplicaciones.

Instalación con .deb

También puede ser que queramos instalar una aplicación que no está en la tienda. Como es el caso, por ejemplo, del navegador Chrome. Para eso, debemos descargar el archivo .deb desde la página de Chrome en <https://www.google.com/chrome/>.



El navegador creado por Google

Descargar Chrome

Elegimos "Descargar Chrome" y luego, como tenemos Ubuntu, el .deb.

Descargar Chrome para Linux

Debian/Ubuntu/Fedora/openSUSE.

Selecciona tu paquete de descarga:

- ☒ .deb de 64 bits (para Debian/Ubuntu)
☐ .rpm de 64 bits (para Fedora/openSUSE)

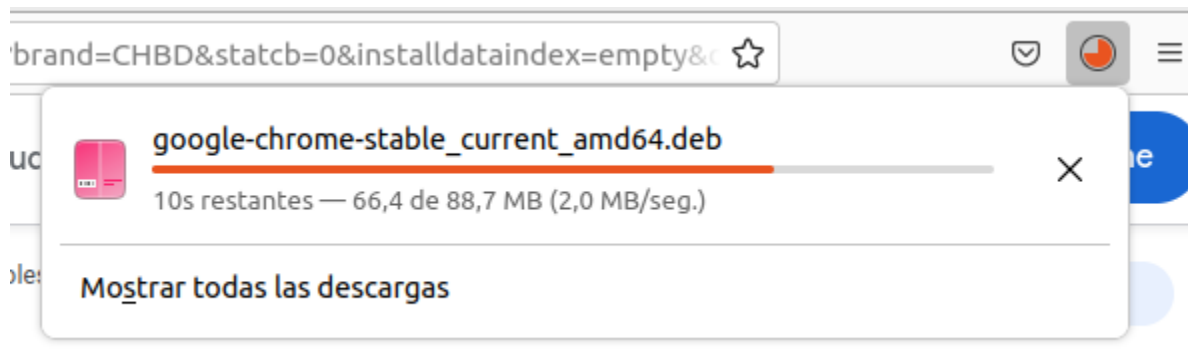
Not Debian/Ubuntu or Fedora/openSUSE? There may be a community-supported version for your distribution [here](#).

Al descargar Google Chrome, aceptas los [Términos del Servicio de Google](#) y los [Términos del Servicio Adicionales de Google Chrome y ChromeOS](#).

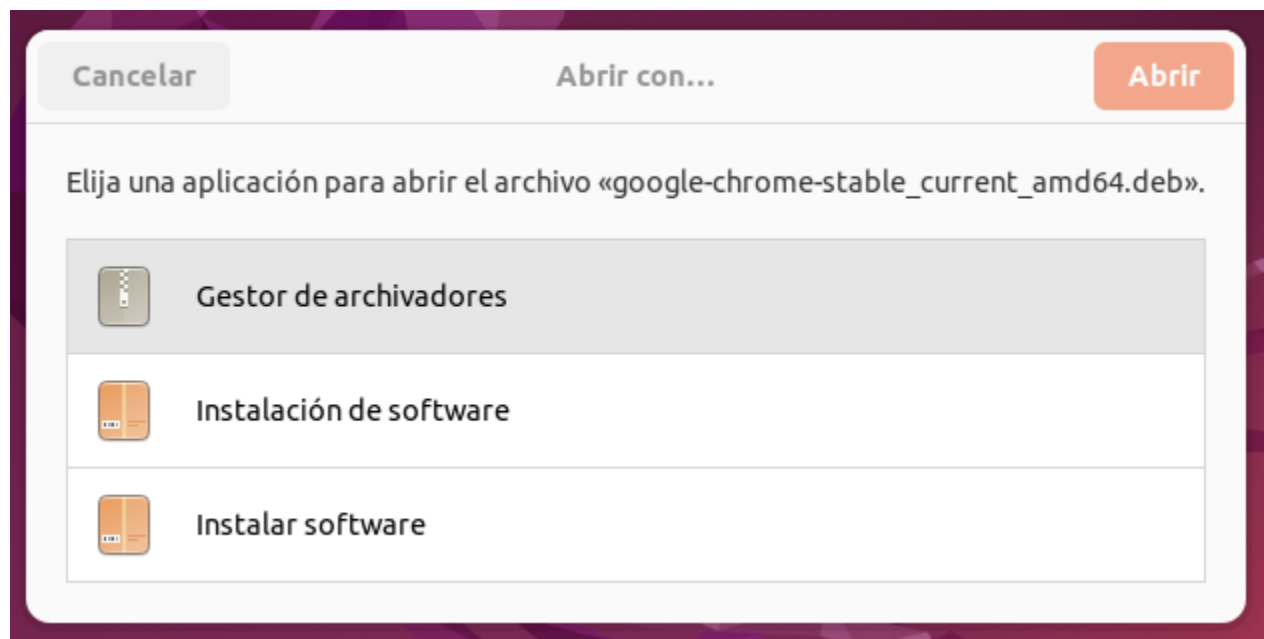
Nota: Al instalar Google Chrome, **se añadirá el repositorio de Google**, lo que hará que tu sistema mantenga actualizado Google Chrome de forma automática. Si no quieres utilizar el repositorio de Google, ejecuta el comando "sudo touch /etc/default/google-chrome" antes de instalar el paquete.

Aceptar e instalar

En Firefox se mostrará la descarga en progreso y una vez terminada, podremos abrir el archivo .deb.



Ubuntu nos preguntará con qué programa debe abrir el archivo .deb, luego de elegir “Instalación de software” y “Abrir”, pasaremos a una pantalla similar a las de la tienda de Ubuntu.



Desde aquí, hacemos clic en “Instalar” y Chrome quedará instalado en nuestra computadora.



google-chrome-stable

Instalar

The web browser from Google

Google Chrome is a browser that combines a minimal design with sophisticated technology to make the web faster, safer, and easier.

Trabajando con bash

Ya vimos cómo trabajamos e instalando herramientas usando herramientas gráficas, pero hay una forma de trabajar que nos permite controlar todo el S.O. sin usar gráficos, si no vía comandos.

Para poder hacerlo necesitamos un intérprete de comandos, en Ubuntu el intérprete de comandos básicos se llama **bash** (**B**ourne **A**gain **S**H). **sh** significa shell, que en se puede traducir como cáscara, y es la cáscara que rodea al S.O. y es la forma que le damos órdenes, Aun existe **sh** en un linux común, pero **bash** tiene una serie de ventajas mayores.

Pero vamos a la base de todo, **bash** se ejecuta desde la interfaz gráfica abriendo o llamando al comando **Terminal**, aparece de un montón de formas pero lo que ejecuta es **bash**. También cuando no instalamos la interfaz gráfica y solo nos aparece en la pantalla **login**, una vez autenticado lo que se ejecuta es **bash**.

Si lo ejecutamos de la interfaz gráfica vemos:



O si entramos en forma no grafica:

```
charlie@dalma:~$ █
```

Eso que vemos `charlie@dalma:~$` lo llamamos prompt, y nos indica que bash, el intérprete de comandos, está esperando que pongamos un comando y apretamos enter, para ejecutarlo.

Lo básico que nos puede dar es `$`, el resto es una personalización del prompt que da por defecto Ubuntu y se puede cambiar, lo veremos más adelante. El de este prompt en especial es que: `charlie` es el nombre de usuario, `dalma` es el nombre de la máquina y el símbolo `~` indica en qué directorio estamos en este caso `~` es el directorio raíz de nuestro usuario, también llamado **HOME** (casa).

Siempre nos da eso cuando está esperando comandos y podemos ejecutar cosas, por ejemplo:


```
charlie@dalma:~$ pwd
/home/charlie
charlie@dalma:~$ free
              total        used        free      shared  buff/cache   available
Mem:      20235392    14499516     731276     3208392     5004600     1857828
Swap:     48812028     2948096    45863932
charlie@dalma:~$ date
Tue 24 Jan 2023 10:46:12 AM -03
charlie@dalma:~$ whoami
charlie
charlie@dalma:~$ hostname
dalma
charlie@dalma:~$
```

Ejecuté 5 comandos: `pwd`, `free`, `date`, `whoami` y `hostname`. Y todos ellos tienen su respuesta, o feed-back

Existen más intérpretes de comandos, como dijimos `sh`, `ash`, `zsh`, `csh`, `tcsh`, las diferencias están en la forma que interactúan con el usuario pero todos, exactamente todos solo ejecutan comandos. En cualquiera de ellos podría ejecutar los comandos anteriores y la respuesta sería la misma.

Comandos, Entrada y Salida (command and Input & Output)

Todo comando que ejecutamos en cualquier intérprete de comando, tiene una entrada (input) y una salida (output). Lo más simple es ver la salida

Introducción a la línea de comandos

La línea de comandos es donde pasaremos la mayor parte del tiempo al conectarse a otros servidores y servicios en la nube. Gracias a ésta, se pueden ejecutar comandos complejos sin necesidad de una interfaz gráfica. Es importante saber usar la terminal ya que nos permite realizar acciones que serían muy complejas o de plano imposibles de hacer de forma gráfica. También podemos automatizar tareas y organizar recursos que de otra forma tomarían mucho tiempo realizar.

No es necesario saber de memoria todos los comandos, pero es importante saber en general cómo moverse en la consola y cómo obtener información de comandos que necesitemos. En el desarrollo de software es común el uso de comandos para instalar dependencias, construir aplicaciones y realizar tests, entre otras actividades.

Para abrir la terminal, podemos ir al listado de aplicaciones y hacer clic en el icono de la aplicación:




Alternativamente, simplemente presionando **CTRL+ALT+T** se abrirá una ventana nueva de la terminal.

Copiar y pegar en la consola

Los atajos de teclado en la consola para copiar y pegar son ligeramente diferentes a lo que estamos acostumbrados. **CTRL+SHIFT+C** sirve para copiar y **CTRL+SHIFT+V** para pegar. También podemos hacer uso del botón central del mouse para pegar.



Autocompletar

La tecla **Tab**  sirve para autocompletar. Esto nos permite tipear menos para conseguir lo que queremos.

Por ejemplo, si queremos ir a un directorio "~/Documentos/notas/ubuntu", podemos comenzar a tipear "Do" y con solo pulsar la tecla "Tab" la consola sabrá que queremos decir "Documentos". Si hay otra carpeta que cumple los requisitos, nos dará una lista a elegir y usar tab nos permitirá movernos por la lista hasta llegar a la que queremos. Entonces, para llegar a esa carpeta, solo tendremos que tipear: "Do", Tab, "n", Tab, "u" Tab. Si solo hay una opción posible, incluso podemos hacerlo más rápido, simplemente pulsando Tab nos llenará automáticamente el nombre de la carpeta. La función de autocompletar también ayudará al usuario a la hora de ejecutar comandos con argumentos y opciones.

Uso de las teclas de dirección

Las teclas de dirección son extremadamente útiles a la hora de manejarse en la consola. El mouse tiene poco protagonismo, por lo que tenemos que valernos más del teclado para ejecutar acciones.

Las flechas horizontales   sirven para movernos un carácter a la vez en la instrucción que estamos tipeando. Si usamos CTRL+← o CTRL+→ nos moveremos palabra por palabra horizontalmente. También podemos usar las teclas HOME/INICIO y END/FIN para movernos al comienzo y al final.

Por otro lado, las flechas verticales $\uparrow\downarrow$ sirven para navegar instrucciones que ejecutamos anteriormente, esto es particularmente útil cuando debemos ejecutar un mismo comando en repetidas ocasiones.

Navegación

Existen varios comandos que nos ayudan a ubicarnos en un directorio en particular y obtener información del mismo. El primero a tener en cuenta es `pwd`. Este comando nos mostrará nuestra ubicación actual.

```
pwd
/home/xacademy
```

El directorio por defecto será el de nuestro directorio de inicio, que se ubica en `/home/{nuestro nombre de usuario}`.

Si queremos movernos a otro directorio, hacemos uso del comando `cd`. Por ejemplo, digamos que queremos movernos al directorio raíz y luego ejecutamos `pwd` para verificar que estemos realmente en ese directorio.

```
cd /
pwd
/
```

Si queremos regresar a nuestro directorio de inicio, podemos hacerlo usando dos atajos, primero la variable `$HOME` y segundo el carácter `~` (que en un teclado en español se ejecuta haciendo **ALTGR+Ñ**)

```
cd $HOME
pwd
/home/xacademy
cd /
pwd
/
cd ~
/home/xacademy
```

También podemos movernos un directorio arriba usando `cd ..`

```
cd ..  
pwd  
/home
```

Otro comando útil es el **ls** que nos mostrará todos los archivos en un directorio.

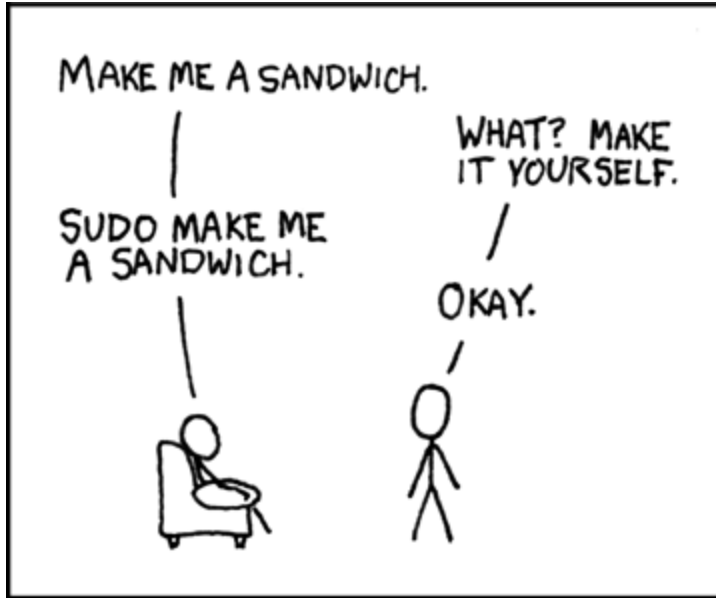
```
cd ~  
ls  
Descargas  Escritorio  Música      Público  Vídeos  
Documentos  Imágenes    Plantillas  snap
```

Este comando admite varias opciones, las más comunes son **ls -l**, que mostrará los archivos con el formato largo:

```
ls -l  
total 36  
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Descargas  
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Documentos  
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Escritorio  
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Imágenes  
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Música  
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Plantillas  
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Público  
drwx----- 3 xacademy xacademy 4096 dic  7 14:10 snap  
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Vídeos
```

Uso de "sudo"

Al agregar "sudo" antes de un comando, le permite a un usuario ejecutar ese comando como el súper usuario. Existen comandos y archivos que para ser accedidos o ejecutados, requieren permisos especiales y sin estos, el sistema operativo no nos permitirá accederlos.



Por ejemplo, un comando útil es “less”, el cual nos permite leer los contenidos de un archivo. Digamos que queremos ver el registro de booteo, que se guarda en el archivo `/var/log/boot.log`.

```
less /var/log/boot.log
/var/log/boot.log: Permission denied
```

Como este archivo tiene permisos solo de lectura para el usuario root, necesitamos sudo para siquiera leerlo.

```
sudo less /var/log/boot.log
[sudo] password for xacademy:
```

Al usar sudo, nos pedirá la clave del usuario (que al tipearla, no mostrará ni asteriscos, ni nada) y si la ingresamos correctamente, accederemos al archivo en cuestión.

Instalar paquetes usando la terminal

Para agregar funcionalidades a nuestro sistema operativo desde la terminal podemos hacer uso del comando `apt`, el cual provee una interfaz para el manejo de paquetes de sistema. Por defecto, debemos hacer uso de **sudo** para poder ejecutar `apt` ya que requiere permisos elevados.

El primer comando que ejecutaremos es **sudo apt update**, esto actualizará el listado de paquetes y versiones disponibles para ser instaladas. NO actualizará los paquetes ya instalados en el sistema (para eso, podemos usar el comando **sudo apt upgrade**).

sudo apt update

```
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [656 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [577 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [781 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4732 B]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [963 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [624 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [8150 B]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [956 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [7278 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [3520 B]
```

Luego de actualizar el listado, podremos instalar un paquete, en este caso instalaremos **curl** que es un comando útil para el desarrollo de software en general.

apt install curl

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 49 not upgraded.
Need to get 194 kB of archives.
After this operation, 453 kB of additional disk space will be used.
Get:1 http://ar.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64
7.81.0-1ubuntu1.6 [194 kB]
Fetched 194 kB in 1s (247 kB/s)
Selecting previously unselected package curl.
(Reading database ... 212828 files and directories currently installed.)
Preparing to unpack .../curl_7.81.0-1ubuntu1.6_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.6) ...
Setting up curl (7.81.0-1ubuntu1.6) ...
Processing triggers for man-db (2.10.2-1) ...
```

Obtener documentación de un comando

La herramienta más completa para obtener documentación de un comando en particular es el comando “man”. Nos sirve para leer el manual del comando que queremos usar y trae información útil sobre lo que querramos hacer. Por ejemplo `man ls` nos mostrará el manual del comando “ls”, con información útil sobre las opciones disponibles para usarlo. Para salir del manual solo hace falta pulsar la tecla “q”.

Otra opción es usar el sitio cheat.sh. Podemos usarlo directamente desde la línea de comandos ejecutando `curl cheat.sh/comando`. Por ejemplo:

```
curl cheat.sh/ls
# To display everything in <dir>, excluding hidden files:
ls <dir>

# To display everything in <dir>, including hidden files:
ls -a <dir>

# To display all files, along with the size (with unit suffixes) and timestamp:
ls -lh <dir>

# To display files, sorted by size:
ls -S <dir>

# To display directories only:
ls -d */ <dir>
...
```

Podemos probar usar man y cheat.sh con otros comandos útiles, como “cp”, “mv”, “rm” “mkdir”, “touch” y “top”

Vim

Esta aplicación es de las más poderosas, y si bien es algo difícil de dominar, los comandos básicos son sencillos de recordar. Primero lo instalamos usando apt.

```
sudo apt update
...
sudo apt install vim
...
```

Ahora vamos a movernos a la carpeta Documentos en nuestro directorio de inicio y crear un archivo “holamundo” con vim.

```
cd ~/Documentos/  
vim holamundo
```

Se abrirá un archivo vacío:



Para poder editar este nuevo archivo, usamos la tecla “i” y pasamos al modo de insertar. Vamos a tipear “Hola mundo”



Una vez que terminemos de insertar, pulsamos la tecla **ESC** y luego, para guardar nuestros cambios usamos **:wq** (la w significa que vamos a escribir el archivo y la q, que vamos a salir de vim).

[illegible]

Pulsamos **ENTER** para confirmar.

Para verificar que efectivamente se editó nuestro archivo, podemos usar el comando “cat”.

```
cat holamundo
Hola mundo
```

Crear un script

Una de las ventajas de usar la línea de comandos es el nivel de control y automatización que nos permite. Podemos crear scripts de bash para simplificar y automatizar tareas.

Vamos a crear un archivo automatizar.sh (Si bien la extensión “sh” nos sirve para identificar archivos ejecutables, no es realmente necesario).

```
vim automatizar.sh
```

Como primera línea, agregamos un “shebang”, que sirve para informar de que se debe usar el intérprete bash al ejecutar este archivo:

```
#!/bin/bash
```

Digamos que queremos crear un directorio de manera interactiva, es decir, el script le preguntará al usuario el nombre del directorio a crear.

```
#!/bin/bash
echo "¿Nombre del directorio?"
read directorio
mkdir $directorio
```

El comando “**read**” leerá lo que escriba el usuario y lo guardará en la variable “directorio”, que luego puede ser llamada usando **\$directorio**.

Luego de guardar, podemos probar el script usando el comando bash

```
bash automatizar.sh
¿Nombre del directorio?
prueba
cd prueba
pwd
/home/xacademy/prueba
```

Digamos que, además de crear un directorio, queremos que nuestro script cree 5 archivos y cada uno de esos archivos tenga por contenido un número aleatorio.

```
#!/bin/bash
echo "¿Nombre del directorio?"
read directorio
mkdir $directorio
touch ./${directorio}/archivo{1..5}.txt
for i in {1..5}
do
    echo $RANDOM > ./${directorio}/archivo$i.txt
done
```

Ahora agregamos el comando touch, que crea archivos vacíos, en este caso cinco archivos con nombres **archivo1.txt** al **archivo5.txt**. Finalmente, usamos un bucle “for” del 1 al 5 para llenar esos archivos vacíos con un número aleatorio usando la variable “**\$RANDOM**”.

Al salir y ejecutar nuevamente nuestro script, podemos validar los contenidos de esos archivos usando **cat**, **less** o **vim**.

El último punto en cuestión es poder ejecutar el script sin necesidad de anteponer “bash”. Para esto, debemos modificar el archivo “automatizar.sh” para hacarlo ejecutable. Para esto, hacemos uso del comando “chmod” y usamos la opción “+x” que es la bandera para hacer ejecutable el archivo.

```
chmod +x automatizar.sh
./automatizar.sh
prueba3
cat ./prueba3/archivo1.txt
19060
```

SSH

Un cliente SSH es un programa para conectarse de manera segura a una máquina remota y ejecutar comandos desde ahí como si fuera una máquina local. Es muy común usarla para conectarse a servidores o servicios cloud.

La forma más básica de usar ssh es **ssh {nombre de usuario}@{servidor}**.

Para probar SSH podemos crear una cuenta gratuita en <http://sdf.org/>. Puede demorar un poco, pero nos llegará a nuestra casilla de mail la información para conectarse. En nuestro caso, creamos un usuario xacademy y para conectarnos usamos el siguiente comando (cambiando xacademy por tu usuario).

```
ssh xacademy@tty.sdf.org
The authenticity of host 'tty.sdf.org (205.166.94.4)' can't be established.
ED25519 key fingerprint is SHA256:Zjwb07AU8rHJExYrmZS2LqGZ7WfdoELfMrF54W92PYA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'tty.sdf.org' (ED25519) to the list of known hosts.
xacademy@tty.sdf.org's password:
```

Una vez conectado al servidor se pueden realizar diversas acciones. Los comandos “help” y “what” pueden servir para orientarse.

Próximos pasos

- Permisos de archivos
- Customizar la línea de comandos
- Usuarios y grupos
- Servicios
- Programación de trabajos con Cron
- Tubos y filtros
- Variables de ambiente
- Transferir archivos por red
- Logs
- Manejo de red
- Seguridad