

Debian y distribuciones GNU/Linux

Por usuarios y para los usuarios



Autor: Jean Abraham Chacón Candalosa

Licencia: CC BY-SA 4.0

Índice

GNU/Linux Distribuciones

Introducción.....	3	6.11 Comentarios.....	39
1 Conceptos básicos.....	1	6.12 Archivos desktop.....	39
1.1 Tipos de software de computadora.....	2	6.13 AND OR NOT.....	41
1.2 Tipos de distribuciones GNU/Linux.....	4	6.14 Uso de pipe en bash.....	42
1.3 ¿Que es un entorno de escritorio?.....	5	6.15 Uso de grep.....	42
1.4 Antivirus en GNU/Linux.....	6	6.16 Uso de apt.....	43
1.5 Empezar a usar una distro base Debian.....	7	6.17 Uso de dpkg.....	47
1.6 Evitar hacer el distro hopping.....	7	6.18 Agregar repositorios apt non free.....	49
1.7 Carpetas, archivos y rutas.....	8	6.19 Limpieza general.....	51
2 Estructura de carpetas.....	9	6.20 Los Logs/Registros.....	53
2.1 Directorio HOME.....	10	6.21 Servicios Systemd.....	54
2.2 Directorio "usr" y "lib".....	11	6.22 ¿Que es bash script?.....	56
2.3 Directorio "bin" y "sbin".....	11	7 Obtener información del hardware.....	57
2.4 Directorio "boot".....	11	7.1 CPU.....	58
3 Boot y Grub.....	12	7.2 RAM.....	59
3.1 Gestor de arranque.....	12	7.3 Almacenamiento.....	59
3.2 Gestor de arranque Multi Boot.....	13	7.4 Tarjeta gráfica.....	60
3.3 GRUB.....	13	7.5 Sensors.....	61
4 Servidores de paquetes.....	14	7.6 Información general.....	62
4.1 Repositorios APT.....	14	7.7 Información en tiempo real.....	63
4.2 Repositorios apt free.....	16	7.8 Estresar computadora.....	64
4.3 Repositorios apt non free.....	16	8 Servidores gráficos.....	67
4.4 Donde se configuran los repos apt.....	17	8.1 X11.....	67
5 Init System.....	18	8.2 Wayland.....	72
5.1 Systemd Runit OpenRC.....	18	9 Wine.....	73
6 Terminal.....	20	9.1 Instalar Wine.....	73
6.1 Comandos básicos bash.....	21	9.2 Como usar wine.....	74
6.2 Usa HELP.....	26	9.3 Winetricks.....	75
6.3 Variables de entorno.....	26	9.4 DXVK.....	77
6.4 Usa las comillas en la terminal.....	28	9.5 Proton, Protondb y Protontricks.....	79
6.5 Autocompletado.....	29	10 Final.....	81
6.6 Glob con asterisco (*).....	30	Conclusión.....	83
6.7 CHMOD y los permisos.....	31	Glosario.....	84
6.8 Editor de texto nano.....	34	Bibliografía.....	86
6.9 Privilegios sudo.....	35	Licencia.....	89
6.10 Agregar privilegios sudo a usuario.....	36		

Introducción

Muchas veces los usuarios instalan una distribución GNU/Linux, y les gusta y toda la cosa, pero a la hora de querer instalar una aplicación, o configurar el OS, no saben como hacerlo.

En ese tutorial se vera como usar una distribución GNU/Linux, aprenderemos todo lo básico para manejar al cien el sistema operativo.

Todos los ejemplos que se verán, serán utilizando un distribución Debian 12. Pero cualquier distribución base Debian, como Ubuntu, Linux Mint, entre otros, sirven.

Dichos conocimientos te servirán como entrada para poder usar muchas Distribuciones GNU/Linux, no solo los base Debian.

1 Conceptos básicos

Sistemas operativos GNU/Linux

Primero que nada tenemos que saber que es un sistema operativo.

Un sistema operativo es un programa informático que permite administrar los recursos del equipo y facilitar la interacción con el usuario.

Un ejemplo de sistema operativo seria Windows, otro ejemplo es Android.

Es decir; gracias a los sistemas operativos podemos utilizar programas de manera sencilla y eficiente.

¿Qué es Linux?

Linux es un kernel, 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.

Por ejemplo; Gracias a Linux, al presionar una tecla del teclado (vaya la redundancia), esa tecla se escribe.



¿Qué es GNU?

GNU es un sistema operativo de software libre, es decir, respeta la libertad de los usuarios (bonito, ¿verdad?).

Consiste en paquetes de GNU (programas publicados específicamente por el proyecto GNU) además de software libre publicado por terceras partes.

¿Qué es una distribución Linux?

Normalmente se refiere a una distribución Linux al complemento de GNU y Linux, siendo Linux el kernel, y GNU es el sistema operativo. GNU tiene un conjunto de componentes que se complementan con el kernel Linux.

Podríamos decir que llamar distro linux no es lo mas correcto, lo mejor seria decir GNU/Linux, porque existen distribuciones que no usan GNU, pero si Linux, como android; esta no usa GNU, pero si Linux.

Es decir; mejor di “Distro GNU/Linux”. Ya que esto es mas específico.

Además del complemento entre GNU y Linux, una distro GNU/Linux viene con cantidad de aplicaciones útiles, como el navegador, calculadora, y etc. Una distro GNU/Linux viene con una interfaz, con un Inet System, viene todo listo para su uso.

Bueno todo esto puede sonar confuso, pero bueno; GNU y Linux son uno; Ambos se complementan.

Capítulo 1 | Conceptos básicos

¿Porque usar una distro GNU/Linux?

Mayor privacidad, mayor control, y mayor potencial de conocimiento a adquirir.

Ejemplo de mayor control:

En windows no puedes desinstalar Edge, o el explorador de archivos. No los puedes remplazar.

En una distro Linux puedes quitar todo lo que se te de la gana, y remplazarlo por lo que quieras. Esto es como una espada de doble filo (porque puedes romper el sistema), pero sigue siendo mas chido.

En este tutorial veremos como usar GNU/Linux de manera intermedia.

Conocimiento intermedio; con esto podremos hacer de todo. Vamos a parecer hackers, pero nomas parecer. Para ser hacker necesitaras de conocimientos avanzados. Pero de poco en poco mi pana, no te aloques.



1.1 Tipos de software de computadora

Código fuente de un software:

El código fuente es todo el software sin compilar. Pelón. Allí esta todas las funciones, todo su funcionamiento, puedes ver todos los detalles del funcionamiento de este. Se puede modificar, se puede compilar, cualquier cosa.



¿Que caramolas es software libre, y que es el software privativo?

Software libre.

Son las aplicaciones que tienen el principio moral de no acceder a la privacidad del usuario, y no tienen condición de uso. Son gratuitas y su código fuente es publico.

Algunas características chulas del software libre:

- Es gratis.
- Su código fuente es publico y permite su modificación
- Su distribución debe ser gratuita, y sus modificaciones también deben ser gratuitas.
- No invaden la privacidad del usuario.
- La comunidad lo mantiene vivo, lo desarrolla.

Si tienes la habilidad informática suficiente, puedes modificar estas aplicaciones a tu gusto.

Capítulo 1 | Conceptos básicos

Te puedes quedar con esta definición de **software libre**; Es gratis, no te espía, y es moralmente correcto.

Ejemplos de software libre:

- Gimp (*Editor imágenes*)
- Blender (*Modelado 3d*)
- Kdenlive (*Editor de vídeo*)
- SuperTux (*Videojuego chido*)



Software privativo:

Son aplicaciones de paga o gratuitas; Tienen condiciones de uso específicas, entre algunas de sus condiciones, pueden pedir información privada.

Se les dice privativas porque te privan de algo, ya sea privacidad, dinero, o otras cosas.

Son de paga porque son un producto. Y cuando son gratis, normalmente el producto eres tu.

Su código fuente es cerrado, solo ciertas personas pueden acceder a él, el desarrollador y sus socios (Nadie sabe al 100% como jala, solo sus creadores).

Lamentablemente, aunque sean de paga, aun pueden funcionar mal.

Algunas características chulas del software privativo (del software privativo que jala bien):

- Funcionamiento eficiente
- Soporte técnico
- Satisfacción de adquisición

Resumiré al **software privativo** con esto: Puede ser de paga o gratis, tienen condiciones de uso, y no tienes ni la libertad, ni el control total de la aplicación.

Ejemplos de software privativo:

- Minecraft (*Videojuego famoso*)
- Photoshop (*Editor de imagen*)
- 3dsMax (*Modelado 3d*)
- DaVinci Resolve (*Editor de vídeo*)



Un ultimo tipo de software, es un híbrido entre el privativo y el libre.

Software open source:

Este tiene el código fuente publico, de la versión mas reciente hecha o de alguna versión previa.

Es gratis, pero la versión complicada oficial puede privarte de alguna privacidad.

Tiene mas condiciones de distribución (como las privativas).

Es decir; Es gratis, tiene el código libre, pero la versión compilada oficial puede privarte de alguna libertad.

Capítulo 1 | Conceptos básicos

Ejemplos de software open source:

- Chromium (*La base de chrome*)
- Android (*Ta loco*)
- Ubuntu (*El canijo tiene telemetría*)



Bueno eso es lo que tienes que saber de los tipos de software.

1.2 Tipos de distribuciones GNU/Linux

Sistemas operativos GNU/Linux

Existen muchísimas distribuciones GNU/Linux. Veremos de las mas canijas a las mas sencillas.

Distros GNU/Linux avanzadas o super avanzadas

Existen distros para usuarios avanzados, que para usarlas bien necesitas tener conocimientos intermedios-avanzados para su uso. Hasta a veces tienes que programar y toda la cosa.

Ejemplos:

- Gentoo. *Se tiene que compilar todo, hasta la hora.*
- Arch Linux. *Ta canijo; mucha terminal.*



Distros GNU/Linux intermedias

Tienes que tener conocimientos básicos del funcionamiento de GNU/Linux, tendrás que usar terminal en muchas ocasiones.

Ejemplos:

- Debian.
- Open Suse.



Distros GNU/Linux sencillas

Tiene la intención de ser puro click, nada de terminal, nada de comandos. Todo bonito y intuitivo.

Ejemplos:

- Linux Mint.
- Ubuntu.
- Q4OS. *Su interfaz parece windows.*
- Elementary OS. *Se parece al Mac.*



Lo mejor para empezar, pos son las distros sencillas. Pero bueno, al final si tienes conocimientos suficientes cualquier distro te sirve. Ya que en una distro puedes hacer todo lo que se te de la gana (si sabes como).

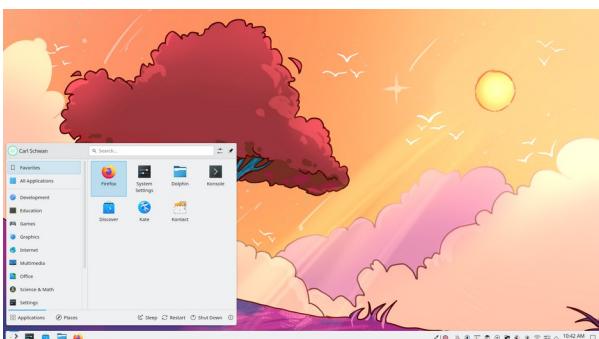
1.3 ¿Qué es un entorno de escritorio?

Es un conjunto de programas que ofrecen al usuario un entorno amigable. Es una interfaz gráfica que ofrece configuraciones y accesibilidad, incluyendo barras de herramientas, integración con aplicaciones, utilidades como arrastrar y soltar, etc.

Un entorno de escritorio, le da ese look al OS. Su visual y funcionalidad es muy importante. Los entornos de escritorio hacen que las personas sin conocimientos avanzados en computadoras puedan usar el pc. Y que los usuarios avanzados no se vuelvan locos con tanto comando.

Algunos de entornos de escritorios usados en GNU/Linux:

- **Xfce4:** <https://xfce.org/>
- **Gnome:** <https://www.gnome.org/>
- **KDE Plasma:** <https://kde.org/es/plasma-desktop/>
- **Trinity:** <https://www.trinitydesktop.org/>
- **LXDE:** <https://www.lxde.org/>



1.4 Antivirus en GNU/Linux

Constantemente se habla del peligro de que al pc se le pague un virus.

Pero en GNU/Linux esta posibilidad es considerablemente menor. Esto es así porque la mayoría de virus son creados para computadoras con Windows o con Mac OS.

Además el sistema de permisos de archivos que tienen las distros GNU/Linux, hace que los virus tengan mas barreras a pasar.

Lo mejor para evitar virus es "no usar fuentes de descarga desconfiables". Un antivirus usa recursos de la computadora y puede dejarla lenta.

Conociendo esto, es opción tuya el instalar un antivirus o no.

El antivirus mas conocido para Linux es ClamAV. Es gratuito y open source. Pero se tiene que configurar y no es para principiantes, tienes que meterle mano para que jale.

Ejemplos de antivirus de uso sencillo:

ClamAV + ClamTk (Antivirus + Cliente sencillo de usar):

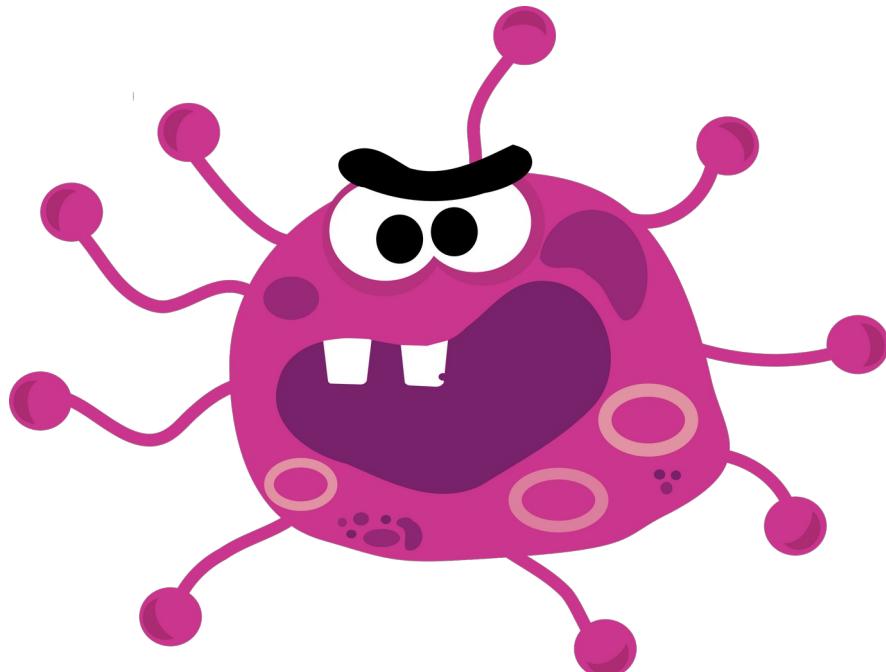
- ClamAV: <https://www.clamav.net/>
- ClamTk: <https://github.com/dave-theunsub/clamtk>

Comodo (Gratis limitado o de paga Full):

- <https://antivirus.comodo.com/>

ESET NOD32 para Linux (Gratis limitado o de paga Full):

- <https://www.eset.com/mx/empresas/descargas/endpoint-antivirus-linux/>



Capítulo 1 | Conceptos básicos

1.5 Empezar a usar una distro base Debian

Ahora si ya, mucho quilombo. Vamos a empezar a usar una distro GNU/Linux base Debian.

Usaremos la distribución Linux Mint. Es base Ubuntu, y a su vez Ubuntu es base Debian. El diseño visual de Linux Mint se parece a Windows.

Linux Mint: <https://www.linuxmint.com/>

Este OS viene con varios sabores/versiones, una mas pesada que la otra, pero para no complicarnos la vida yo recomiendo la mas ligerona:

Linux Mint Xfce Edition: <https://www.linuxmint.com/download.php>

Te recomiendo ver un vídeo de como instalar Linux Mint.

O podemos ir a su documentación de “como instalar”:

<https://linuxmint-installation-guide.readthedocs.io/en/latest/>

O las dos cosas al mismo tiempo (vídeo y documento). Una vez instalado pos vemos como anda la cosa, como se ve, como se mueve, juega con el OS.

1.6 Evitar hacer el distro hopping

El distro hopping o distro jumping es; instalar y instalar distros GNU/Linux.

Porque te gusta probar y probar, y tienes la intención de encontrar la mas chida.

Puede que quieras cambiar de distro GNU/Linux por cualquier problema, y si lo haces a cada rato, eso es malo porque ninguna distro GNU/Linux es perfecta. Como la hicieron humanos y los humanos no son perfectos, siempre tendrá sus fallitas el OS.

Lo recomendable es no hacer esto, porque no te aporta nada, solo pierdes mucho tiempo, lo mejor es quedarte con lo que instalas.

Lo que tienes que hacer es quedarte con el OS que instalas y que aprendas a solucionar todos sus errores. Añadir todo lo que te guste. Personalizar tu OS.

Recomiendo que escribas todas las soluciones que logres hacer. Escribas todo lo que consideres importante documentar. Para que en el futuro leas eso y digas; Ah, ¡así se hace esto!

Mínimo quédate un año con el OS. Y si no te convence pos busca otro. Créeme que las distros GNU/Linux son super personalizables y versátiles, solo es buscar como hacer las cosas.

Además puedes probar distros desde el internet, con el sitio web [distrosea](#).

Aprovecha al máximo la distro GNU/Linux.

Capítulo 1 | Conceptos básicos

1.7 Carpetas, archivos y rutas

Carpetas en el pc:

Una carpeta es una ubicación virtual dentro de la estructura de archivos de computadora.

Pos que te sirve para tener todo mas ordenado, allí puedes meter archivos.



Archivos en el pc:

Cuando nos referimos a archivo, pues es a una sola cosa; un texto, un vídeo, o básicamente cualquier cosa que no sea una carpeta. Un archivo contiene información para algún programa.



Ejemplos de archivos:

- archivo
- archivo.sh
- archivo.mp4
- archivo.extencionInventada
- archivo.exe

Rutas

Bueno cuando estamos en el pc, la ruta es una carpeta o un conjunto de carpetas como esto:

- /carpeta/carpeta2/carpeta3/
- /carpeta/



Bastante sencillo de comprender, este tipo de conceptos aunque sean sencillos, son los que te suben de level.

2 Estructura de carpetas

El Árbol de Directorios de Linux; Conoce las Principales Carpetas

La estructura de los directorios de Linux, así como su contenido y funciones, viene definida en el denominado “*Filesystem Hierarchy Standard*” o FHS por sus siglas en inglés, que en otras palabras viene a ser el “*Estándar de jerarquía para los sistemas de archivos en sistemas Linux*” y otros derivados de UNIX.

Es decir que esta es una estructura que se usa mucho en diferentes sistemas operativos. Es un *estándar de como ordenar los archivos*. Todos los árboles tienen una raíz, y este árbol de directorios tiene una, y es “/”. Si, una diagonal loca.

Filesystem Hierarchy Standard | FHS

Carpeta	Función
/	Root/Raíz. Acá se guarda absolutamente todo. Todas las carpetas que se mencionan acá, están adentro de esta.
<u>bin</u>	Binarios de usuario.
<u>boot</u>	Ejecutables y archivos requeridos para el arranque.
<u>dev</u>	Archivos de información de todos los volúmenes.
<u>etc</u>	Archivos de configuración del sistema y de aplicaciones.
<u>home</u>	Directorio personal con las carpetas del usuario.
<u>lib</u>	Bibliotecas necesarias para la ejecución de binarios.
<u>media</u>	Directorio de montaje de volúmenes extraíbles.
<u>opt</u>	Ficheros de aplicaciones externas que no se integran en /usr.
<u>proc</u>	Ficheros de información de procesos.
<u>root</u>	Directorio personal de superusuario.
<u>sbin</u>	Binarios del sistema.
<u>srv</u>	Archivos relativos en servidores web, FTP, etc.
<u>tmp</u>	Ficheros temporales.
<u>usr</u>	Archivos de programas y aplicaciones instaladas.
<u>var</u>	Archivos de variables, logs, emails de los usuarios del sistema, etc.

Capítulo 2 | Estructura de carpetas

Bueno, estas son todas las carpetas, todo el árbol de directorios en Linux. Debian usa este estándar.

Todos estos directorios son súper necesarios. Pero nosotros como usuarios estaremos haciendo todas nuestras actividades en */home*.

Y pos también */usr*, que es donde tenemos instaladas las apps. Ya las apps que usemos se encargaran de usar las carpetas que necesiten.

Por ejemplo cualquier app que usemos, estará en */usr*, y esta app usara la carpeta */lib* para usar lo que necesite.

2.1 Directorio HOME

Es el directorio de los usuarios estándar, y por lo tanto es el destinado a almacenar todos los archivos del usuario, como documentos, fotos, videos, música, plantillas, y etc.

Dentro “*/home*” están todos los directorios personales de los usuarios, nombrados según el nombre de usuario utilizado. Así por ejemplo; Supongamos que en un sistema hay dos usuarios, denominados “User1” y “User2”, la estructura será así (sus rutas al usuario):

- */home/User1*
- */home/User2*

Cada directorio de usuario contiene varias carpetas para ayudarlo a clasificar la información. Estas generalmente son: */Documentos*, */Imágenes*, */Música*, */Plantillas* y */Vídeos*

Así se vería el árbol completo:

/home/user/

- */Documentos*
- */Descargas*
- */Imágenes*
- */Música*
- */Plantillas*
- */Vídeos*



Básicamente así jala la carpeta “home”.

2.2 Directorio “usr” y ”lib”

usr

El directorio “/usr” viene de <<User System Resources>> y sirve para almacenar todos los archivos de solo lectura y relativos a las utilidades de usuario, incluyendo todo el software instalado a través de los gestores de paquetes de cada distribución.

Contiene puros programas, y las dependencias necesarias para que jalen.

lib

En este directorio se almacenan todas las librerías necesarias para que funcionen los programas necesarios para que jale el OS. Hasta tiene lo necesario para jale el Kernel. En sistemas operativos de 64 bits, con compatibilidad de 32 bits; Existen las carpetas “lib”, “lib32”, y “lib64”.

Resumen:

“usr” Tiene los programas instalados por el usuario. No son necesarios para que funcione el OS.

“lib” Tiene todos los programas necesarios para que jale el OS y el Kernel.

“lib64” Lo mismo que lib. Pero puro programa de 64 bits.

2.3 Directorio “bin” y “sbin”

bin

Acá se almacenan todas las funciones básicas a nivel de usuario. Solo almacena ejecutables de usuario.

Estos son algunos de los ejecutables que almacena:

cat, cd, cp, grep, gzip, kill, ls, mv, rm, ping, su, ps, tar y vi.

sbin

Todos los binarios necesarios para tareas administrativas gestionadas por el usuario root o super usuario del sistema se encuentran en el directorio sbin.

Este también almacena ejecutables, pero ejecutables que solo se pueden usar con permisos root. Sirven para gestionar tareas del sistema operativo. Funciones internas.

2.4 Directorio “boot”

La carpeta “boot” Incluye todo lo necesario para que funcione el proceso del arranque. Esta carpeta es usada mientras se inicia el sistema operativo. Acá se almacenan los gestores de arranque, como GRUB.

GRUB guarda todo lo chido acá. Configuración, librerías, programillas, todo eso.

Este directorio sirve para el uso del programa de gestión de inicio del sistema operativo.

3 Boot y Grub

Boot

El **boot**, es el proceso de inicialización de un dispositivo electrónico. Es cuando inicia la computadora.

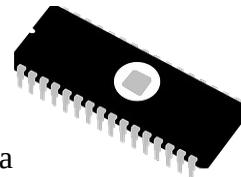
Selección de boot en la computadora

La computadora tiene un gestor del boot, y esta en la BIOS. Esta se encarga de seleccionar entre varias opciones de boot/inicio, y arrancar la seleccionada y descartar las demás.

En la BIOS se puede configurar que boot seleccionar. Dicho esto de un modo coloquial: Decirle al PC que tiene que botear.

BIOS

La BIOS es lo primero que se ejecuta al encender el PC. La BIOS es una secuencia de códigos de ejecución (software) almacenada en un chip de la placa base (hardware) que permite a la misma reconocer qué está conectado a ella; procesador, tarjeta gráfica, RAM, unidades de almacenamiento... y cualquier otra tarjeta adicional PCIe. Sin la BIOS tan solo tendríamos una placa base, no un ordenador.



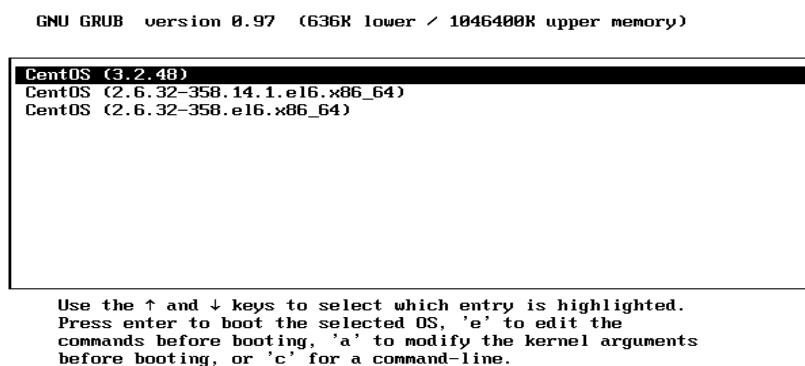
3.1 Gestor de arranque

¿Que es un gestor de arranque?

El gestor de arranque sirve para que arranque el sistema operativo. Como su nombre lo indica, gestiona el arranque, el arranque del OS, carga todo lo necesario para que este funcione. Sirve para cargar el sistema operativo o el entorno de ejecución, necesario para agregar programas a la memoria y proporcionar acceso a los componentes.

Es un programa que se comunica con la BIOS, y a su vez la BIOS se comunica con el hardware, y la info que obtiene se la da al gestor de arranque. Una comunicación continua.

En GNU/Linux el gestor de arranque seleccionado por excelencia es GRUB.



3.2 Gestor de arranque Multi Boot

Llamado **multi boot** o **dual boot**; Esta es una capacidad que tienen algunos gestores de arranque. GRUB es uno de esos.

Esta función del **multi-boot/dual-boot**. Permite gestionar varios gestores de arranque en un mismo gestor de arranque.

GRUB por ejemplo, te permite tener un menú de selección donde puedes seleccionar entre windows o una distro linux.

Gracias a esta función del multi-boot/dual-boot. Se pueden tener varios sistemas operativos instalados en un mismo disco duro.

Otro ejemplo; el gestor de arranque de windows 10, permite la selección de diferentes windows instalados en el mismo disco duro.

Nota:

Esta función del multi boot no solo sirve para seleccionar OS instalados en el mismo disco duro, sino que también sirve para seleccionar OS en otros discos duros. Solo que no lo mencione porque eso ya se puede hacer desde la BIOS.

3.3 GRUB

Definición oficial:

GNU GRUB es un multiboot cargador de inicio. Se derivaba de GRUB, el GRand Unificado Bootloader, que fue diseñado e implementado originalmente por Erich Stefan Boleyn.

GRUB 2 ha sustituido a lo que anteriormente se conocía como GRUB (es decir, versión 0.9x), que, a su vez, se ha convertido en el legado GRUB. Se siguen haciendo mejoras a GRUB, pero las versiones actuales de lanzamiento son bastante utilizables para una operación normal.

Su nombre código es: grub2

Este es el gestor de arranque mas usado en GNU/Linux.
Una de sus principales bellezas, es que permite el multi boot.



Algunas de sus funciones curiosas.

- Puede detectar resolución máxima de la pantalla.
- Se le puede poner aspecto al grub, para que se vea mas chido y personal.
- Puedes cargar tanto distros GNU/Linux como windows. Muy bonito.
- Se le personaliza el tiempo de espera de selección.

Sitio web de grub: <https://www.gnu.org/software/grub/>

4 Servidores de paquetes

¿Dónde se descargan los programas?. Respuesta: En los servidores de paquetes.

Todo programa que descargamos, está almacenado en un servidor, que su función es almacenar esos programas. Por ejemplo *Steam*, tiene un servidor de paquetes, donde almacena todos los juegos.

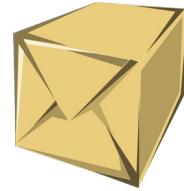
Por ejemplo, en *Debian* existen los repositorios para apt.

Estos almacenan todos los programas que ya están ready para usarse en Debian. Esta bien chido, porque puedes instalar bien fácil un montón de programas. Incluye librerías, utilerías, de todo. Estos repositorios están en servidores que se pueden usar por internet.

¿Qué es un repositorio?

Es un espacio centralizado donde se almacena, organiza, mantiene y difunde información digital, habitualmente archivos informáticos.

Los repos APT almacenan puros programas, con sus configuraciones y sus dependencias. Muy bonito y fácil de usar.



4.1 Repositorios APT

El programa APT

Debian tiene sus servidores de paquetes. Y se les llaman repositorios para APT.

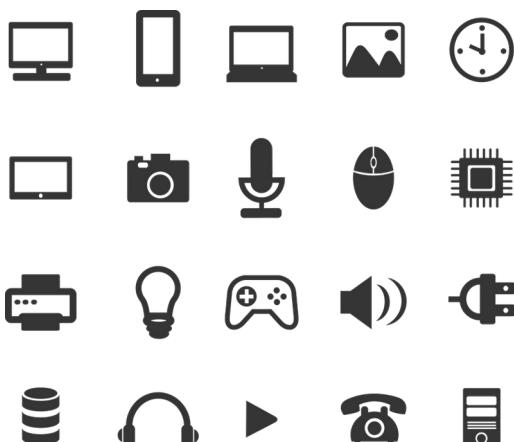
APT es una abreviación, y su nombre completo es: **Advanced Packaging Tool**.

Este es un programa que gestiona el uso del servidor de paquetes, los repos apt.

Repositorios APT

Estos repositorios no son más que directorios que contienen dos tipos de archivos:

- Packages.gz
Esto contiene puros binarios instalables. Son los programas
- Sources.gz
Esto es el puro código sin compilar. Sirve para ver como jala el programa, pero necesita ser compilado para su uso normal.



Capítulo 4 | Servidores de paquetes

Secciones de repositorios APT en Debian

main: Contiene solamente los paquetes que cumplan los requisitos de la licencia del OS. Tiene la intención de almacenar programas para servir para el uso general del OS.

Acá todas las dependencias para el funcionamiento de los programas están cubiertas. Esta sección de "main" no es dependiente de ninguna otra.

contrib: Acá se almacenan paquetes suplementarios, pensados para funcionar en Debian, requieren software fuera de la distribución (fuera de main) para compilar y/o funcionar. Estos programas deben de ser de uso libre, deben ser programas libres. Para eso deberán de cumplir con **DFSG**.

non free: Contienen programas para funcionar en Debian, pero no son libres, no cumplen con DFSG. Programas como "fuentes de texto de microsoft", "winrar", "drivers de nvidea", ese tipo de programas. Los linuxeros los usamos para instalar drivers. Los drivers son necesarios para que jale chido el pc.

sources: Puro código sin compilar. Sirve para aprender, para cacharrear.

DFSG:

The Debian Free Software Guidelines

Normas para que un paquete sea considerado como libre; Normas hechas por Debian.

Dependencia de programas:

Los programas muchas veces necesitan del funcionamiento de otros programas. A eso se le llama dependencia de programa.

Estos programas tienen “*funciones para uso general*” que se pueden usar otros programas. Los programas que funcionen con otros programas, son dependientes.

Ejemplo:

Un *programaX*, depende del *programaY* y el *programaZ* para funcionar.

programaY y *programaZ*: son las dependencias del *programaX*.

Nota:

Por defecto, Debian tiene como repos a; main, contrib, y sources.

Pero no tiene por defecto a "non-free". Ya que esta sección no es de uso libre, pero es bueno saber que existe. Esta sección nos sirve para instalar programas y drivers que necesitemos.

Resumen:

- Los repositorios de programas, almacenan programas.
- Los repositorios apt, almacenan programas.
- Los repositorios apt nos sirven para instalar programas de manera sencilla.

4.2 Repositorios apt free

Estos repos, en realidad serian los de la sección de apt: “*main* y *contrib*”. Estos repos almacenan los programas que cumplen con DFSG, son programas libres.

Ejemplos de programas almacenados en apt como free:

- gimp
- neofetch
- htop
- supertux
- libre office



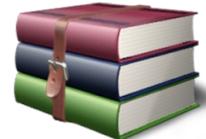
La mayoría de programas default instalados en nuestro sistema operativo GNU/Linux, serán free.

4.3 Repositorios apt non free

Estos repos, en realidad son los de la sección "main non-free" y "contrib non-free": Son puros programas que no son libres, son privativos. Tienen acuerdos de uso y esas cosas.

Ejemplos de programas non free:

- Microsoft font text
- Winrar
- Drivers Nvidea
- Drivers AMD
- Drivers Realtek



Estos son programas que nos sirven para solucionar problemas con el pc, o aplicaciones específicas como Winrar. Generalmente un linuxero usa los drivers y poco mas.

4.4 Donde se configuran los repos apt

¿Donde se configuran los repos apt en Debian?

Se configura desde un archivo de configuración, el cual es un archivo de texto.

El archivo de configuración de los repositorios apt esta en la siguiente ruta: `/etc/apt/`

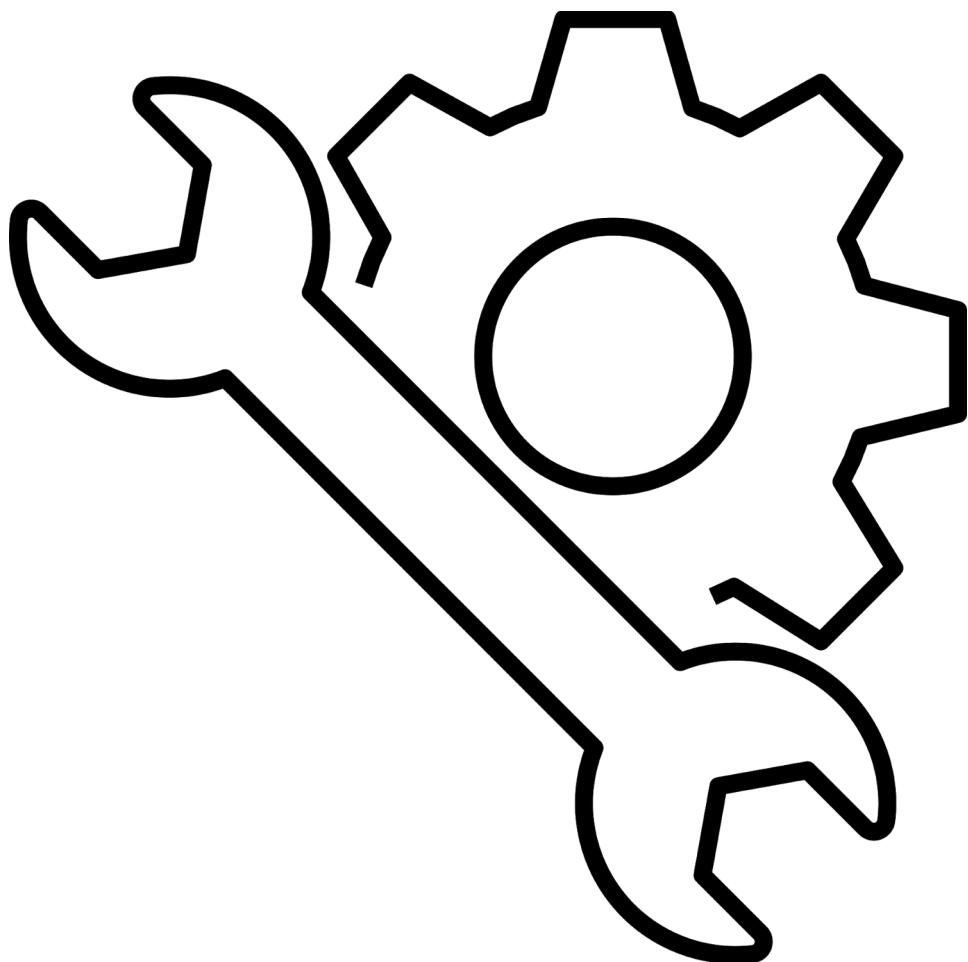
El archivo se llama: `sources.list`

Y la ruta completa, incluyendo el nombre del archivo seria: `/etc/apt/sources.list`

¿Para que me sirve configurar los repos apt?

Te sirve para agregar mas repos, incluso repos que no son de Debian, son repos de tercero que jalan con apt.

Podrías agregar los “repos non-free”, para poder obtener drivers necesarios para el funcionamiento del pc. Eso es lo que generalmente se hace.



5 Init System

¿Que es un Init System?

Init (abreviatura de initialization) es el primer proceso en ejecución tras la carga del núcleo y el que a su vez genera todos los demás procesos.

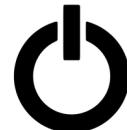
Init System Estilo BSD:

El init de BSD ejecuta el script de inicialización en `/etc/rc`, de forma parecida a como se hacía en el Unix original de Bell Labs. No hay niveles de ejecución (runlevels), el archivo `/etc/rc` determina qué programas se ejecutan por init. La ventaja de este sistema es que es simple y fácil de editar manualmente, aunque sujeto a errores, pues un simple error en ese script podría inutilizar el procedimiento de arranque del sistema.

Init System Estilo System V:

En GNU/Linux es el fichero encargado de establecer los runlevels disponibles, para que pueda ser leído por init. A continuación se muestra un ejemplo del inicio de este fichero, en el cual se establece el runlevel 5.

El **Init System**, se encarga de cargar todos los servicios, cargar todos los procesos. Administrar todos los servicios del sistema operativo.



5.1 Systemd | Runit | OpenRC

En las distribuciones GNU Linux, existen varios init system. Algunos de los mas famosos son:

- Systemd
- Runit
- OpenRC

Runit:

Runit es un esquema de init y gestión de servicios para sistemas operativos tipo Unix que inicializa, supervisa y finaliza procesos en todo el sistema operativo.

OpenRC:

OpenRC es un sistema init basado en dependencias para sistemas Unix-like que mantiene la compatibilidad con el init system ya provisto por el sistema, que normalmente se encuentra en `/sbin/init`.

Systemd

Systemd es el administrador de servicios y sistemas en Linux, y la estandarización de la mayoría de distribuciones de Debian y Red Hat. Systemd fue desarrollado con el objetivo de encargarse de arrancar todo lo que está por debajo del Kernel.

Systemd es un conjunto de bloques básicos de construcción para un sistema Linux. Proporciona un sistema y gestor de servicios que funciona como PID 1 y comienza el resto del sistema.

Capítulo 5 | Init System

Otra de las funciones de systemd linux, es el de apoyar la restauración del sistema e implementar un sistema de gestión de dependencias fundamentado en un control lógico de los servicios.

Systemd, no solo es un init system, hace mas cosas. Y algunos programas dependen de este para funcionar.

Systemd es el que mas se usa, es el Init System que usa Debian.

En Debian base, cuando esta iniciando el OS se ve un montón de texto; ese texto es enviado por el Init System, indica todos los procesos que se cargan. También muestra su estado y esas cosas.

Ejemplo de servicios cargados por Systemd:

```
[ OK ] Reached target basic.target - Basic System.
      Starting accounts-daemon.service - Accounts Service...
[ OK ] Started anacron.service - Run anacron jobs.
      Starting avahi-daemon.service - Avahi mDNS/DNS-SD Stack...
      Starting blueman-mechanism.service - Bluetooth management mechanism...
      Starting bluetooth.service - Bluetooth service...
[ OK ] Started cron.service - Regular background program processing daemon.
      Starting dbus.service - D-Bus System Message Bus...
      Starting e2scrub_reap.service - Remove Stale Online ext4 Metadata Check Snapshots...
      Starting lm-sensors.service - Initialize hardware monitoring sensors...
[ OK ] Started low-memory-monitor.service - Low Memory Monitor.
      Starting mount-disks_with-label.service - Montar discos con label al iniciar sesión...
      Starting polkit.service - Authorization Manager...
      Starting power-profiles-daemon.service - Power Profiles daemon...
      Starting smartmontools.service - Self Monitoring and Reporting Technology (SMART) Daemon...
      Starting switchoo-control.service - Switchoo Control Proxy service...
      Starting sysstat.service - Resets System Activity Logs...
      Starting systemd-logind.service - User Login Management...
      Starting udisks2.service - Disk Manager...
[ OK ] Finished lm-sensors.service - Initialize hardware monitoring sensors.
[ OK ] Finished sysstat.service - Resets System Activity Logs.
[ OK ] Finished e2scrub_reap.service - Remove Stale Online ext4 Metadata Check Snapshots.
[ OK ] Started dbus.service - D-Bus System Message Bus.
[ OK ] Finished networking.service - Raise network interfaces.
      Starting NetworkManager.service - Network Manager...
      Starting wpa_supplicant.service - WPA supplicant...
[ OK ] Started accounts-daemon.service - Accounts Service.
[ OK ] Started polkit.service - Authorization Manager.
[ OK ] Started power-profiles-daemon.service - Power Profiles daemon.
[ OK ] Started avahi-daemon.service - Avahi mDNS/DNS-SD Stack.
[ OK ] Started switchoo-control.service - Switchoo Control Proxy service.
[ OK ] Started bluetooth.service - Bluetooth service...
```

6 Terminal

¡Ahora si viene lo chido!

Para ser usuarios linuxeros potentes tenemos que usar la terminal vato. Primero unos conceptos súper necesarios. Fundamentales; como dicen los intelectuales.

Terminal:

Nos referimos realmente a emulador virtual, que consiste en un programa que emula las especificaciones de un terminal estándar. Nos sirve para dar ordenes al pc. Meterle comandos.

Terminal estándar:

Dispositivo electrónico o electromecánico que se utiliza para interactuar con un computador.

Comando:

Mensaje del usuario enviado al ordenador, y que va a provocar una respuesta en el ordenador. Los comandos son en realidad órdenes, pues indican al dispositivo informático que debe hacer o que debe ejecutar a continuación (según el comando que se le envíe). Instrucciones para el pc.

Parámetros:

Un parámetro es un valor que se le pasa una función, método o programa.

Por ejemplo;

Si tenemos la función "suma", y el nombre en consola de esta función es también "suma", y en esta función podemos sumar infinitos valores, la usaremos así:

```
suma 1 2 3 4
```

Resultado:

```
10
```

Acá los parámetros serían los números "uno, dos, tres, y cuatro". Y el comando sería todo esto "suma 1 2 3 4"

Sintaxis:

Una sintaxis es la estructura de uso de un programa. Es decir; como se ordenan los parámetros, para que el programa jale bien. El orden se respeta si o si, de lo contrario no jala el programa.

Por ejemplo, supongamos que existe el programa "cacahuete"; pide que primero pongas el color del cacahuete, después el sabor, y finalmente la persona que se lo comerá.

Así se usaría este programa:

```
cacahuete rojo chocolate Jonas
```

Esta sería su sintaxis:

```
cacahuete color sabor nombre
```

Una sintaxis genérica:

```
program syntaxis
```

Programa:

Un programa de computadora o programa informático es una compleja secuencia de instrucciones digitales diseñadas para su reproducción en el procesador central de una computadora.

Un programa puede ser una función, una aplicación con interfaz visual, y entre otros. La pc esta llena de programas adentro de otros programas, que funcionan para que jale otro programa. El propio sistema operativo es un programa.

Ejemplos de programas:

La función "suma" o la función "cacahuete", que te mencione antes.

La aplicación de "emulador de terminal" (todas las distros deben tener).

Firefox. Calculadora. Ya tu sabe mi bro.

6.1 Comandos básicos bash

Bash en la terminal

En la terminal todos los comandos jalan con bash. bash es bonito y la terminal lo agradece.

¿Para que caramba las me sirve aprender comandos?

Pues para instalar apps, entender el OS, hacer cosas mas personalizadas y chidas. No siempre lo necesitaras, pero es bueno saberlo, porque tendrás que usar comandos de terminal.

Bueno antes de lo divertido tenemos que aprender a usar la terminal.

¿Para que es terminal?

Es sencillo, la terminal te permite darle instrucciones al sistema operativo; Por medio de texto, donde ese texto representa comandos; Instrucciones para el sistema operativo.

Abrir la terminal:

Para abrir la terminal, presionamos "**ctrl + alt + t**", o "**win + t**", o la **buscamos en el menú de aplicaciones**.

Comando: instrucción en texto para la computadora.

En la terminal podemos escribir comandos. **Para poner un comando, primero escribimos el comando y después presionamos enter.** Recuerda que los comandos tienen que ser escritos con cero errores, o si no, no jalan, respetando mayúsculas y espacios.

Capítulo 6 | Terminal

Escribe esto en terminal, y una vez escrito presiona enter:

```
echo $HOME
```

Te mostrara algo asi:

```
/home/user
```

Donde en vez de "user" se mostrara tu nombre de usuario.

"**echo \$HOME**" fue el comando. Y el enter que le diste, fue la ejecución del comando.

Ahora ya sabes como poner comandos, pero ahora necesitas entender los programas de terminal mas usados, entender cuando y porque usarlos.

Programas de consola que probablemente mas tengas que usar

Programa	Descripción
man	<i>Muestra el manual del comando que indiquemos</i>
sudo	<i>Dar permisos de administrador, permisos de super usuario. Se tiene que usar de forma precavida.</i>
ls	<i>Listar los archivos y carpetas</i>
cd	<i>Cambiar de carpeta/directorio/ruta</i>
pwd	<i>Mostrar la ruta al directorio actual</i>
mv	<i>mover archivos</i>
rm	<i>Borrar archivos</i>
mkdir	<i>Crear un directorio</i>
echo	<i>Mostrar en pantalla</i>
nano	<i>Editor de texto.</i>
touch	<i>Crear un archivo de texto</i>
apt	<i>Advanced Packaging Tool</i>
dpkg	<i>Gestor de paquetes de Debian</i>

Capítulo 6 | Terminal

Probemos usar algunos de estos comandos en terminal para entender mejor:

Los resultados serán similares a estos, pero no iguales.

Solo escribimos lo que esta a la derecha del "\$"

Paso 1 Vemos los archivos:

```
user@pc:~$ ls
```

Resultado:

```
Carpeta      Capeta2      Capeta3      Capeta4  
Archivo.txt
```

Paso 2 Nos movemos de carpeta (Escribimos alguna de las carpetas que nos mostró el "ls"):

```
user@pc:~$ cd Carpeta
```

Resultado:

```
user@pc:~/Carpeta$
```

Paso 3 Creamos un archivo de texto:

```
user@pc:~/Carpeta$ touch texto.txt
```

Resultado:

```
user@pc:~/Carpeta$
```

Si no se ve nadota, si se creo el archivo.

Paso 4 Volvemos a ver los archivos:

```
user@pc:~/Carpeta$ ls
```

Resultado:

```
archivo archivo archivo...  
texto.txt
```

Entre los archivos que se muestran, busca el "texto.txt"

Capítulo 6 | Terminal

Paso 5 Editamos el texto:

```
user@pc:~/Carpeta$ nano ./texto.txt
```

Resultado:

```
Escribiendo algo...
```

*Se abre el editor de texto nano, intenta escribir algo.
Y luego guarda presionando "ctrl + s", y luego sal con "ctrl + x"*

Paso 6 Vemos la ruta completa en la que estamos:

```
user@pc:~/Carpeta$ pwd
```

Resultado:

```
/home/user/Carpeta
```

Copiamos la ruta seleccionando eso y presionando “ctrl + shift + c”

Paso 7 y final.

Abrimos el explorador/gestor de archivos, y pegamos en la ruta de el resultado de pwd con “ctrl + v”. (El gestor de archivos búscalos en el menú de aplicaciones)

Muy bien hemos usado varios comandos.

ls, cd, touch, nano y pwd.

Al principio puede parecer confuso pero le agarras la onda rápido.

Notas importantes sobre el manejo básico en la terminal.

(El shift es la misma tecla que el mayus)

En la terminal se copia y pega con:

Copiar: "ctrl + shift + c"

Pegar: "ctrl + shift + v"

Para cancelar el proceso de un comando en terminal puedes precionar: "ctrl + c"

Cancelar proceso: "ctrl + c"

Recordar no, practicar si

Ahora, es normal que no recuerdes para que sirven todos los comandos, o recordar que comandos existen.

Así que es bueno tener esta pagina de comandos de bash:

https://es.wikipedia.org/wiki/Comandos_Bash

Capítulo 6 | Terminal

Y recordar que a la mayoría de comandos (por no decir todos) les puedes meter el parámetro **-h**, o el parámetro **--help**. Ambos son pa lo mismo.

Ejemplos de uso de parametro -h y --help:

```
sudo -h  
ls -h  
cd --help  
pwd -h  
touch --help  
nano -h
```

Da igual si usa --help o -h, es tu preferencia usar uno u otro. Pero a algunos programas solo les jala el -h, o el --help.

Si solo escribes el "help" en consola, te dará como resultado los comandos mas usados.

```
help
```

Resultado:

```
Lista de comandos mas usados....
```

Ejemplos de uso de ls:

```
ls /ruta/
```

Ejemplo de uso de cd:

```
cd /ruta/
```

Ejemplo de uso de mkdir:

```
mkdir /ruta/nombre-carpeta-que-quiero-crear
```

Bueno ahora ya tienes una idea de como usar comandos.

6.2 Usa HELP

Tienes que usar el parámetro help vato. Este parámetro lo tienen muchos comandos. Es un estándar. Ta chido porque te da información de como jala el programa.

Normalmente se pone el parámetro help, de estas maneras:

```
-h  
--help
```

Esto depende del programa. Pero esta es la manera mas común de poner help.

Ejemplos:

```
apt --help  
ls -h  
dpkg --help
```

En dpkg, solo se puede ver su ayuda con "--help", no se puede con "-h".

Y no olvides que el puro "help" en la terminal tiene un resultado. Muestra una ayuda general.

Alternativamente a “help”, puedes usar el programa “man”, este te da información de casi todos los programas. **Ejemplos de uso de man:**

```
man program  
man dpkg  
man man
```

Resumen:

Help esta en todos lados, úsallo.

Cuando tengas una duda, el parámetro help te ayudara. Es para eso vato.

6.3 Variables de entorno

¿Qué es una variable?

Una variable es una ubicación que permite almacenar un valor que se puede cambiar según las condiciones o la información del programa.

Ejemplo de variable, pon esto en la terminal:

Primero:

```
variable=0809
```

Después:

```
echo $variable
```

Esto mostrara el valor de dicha variable

Capítulo 6 | Terminal

¿Qué son las variables de entorno de GNU/Linux?

Las variables de entorno son valores dinámicos que afectan los programas o procesos que se ejecutan en un servidor. Existen en todos los sistemas operativos y su tipo puede variar. Las variables de entorno se pueden crear, editar, guardar y eliminar.

En GNU/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.

Ejemplos de variables de entorno en GNU Linux:

`HOME, USER, BASH, BASH_VERSION`

Ejemplo de uso de variables de entorno, pon primero el signo de "\$" seguido de la variable de entorno. **Pon esto en terminal:**

```
echo $HOME
```

```
echo $USER
```

Esto mostrara el valor de dichas variables.

Puedes acceder a estas variables en todo momento.

Ver todas las variables de entorno disponibles

Para ver todas las variables de entorno del tito GNU/Linux, **pon el comando "set" en terminal:**

```
printenv
```

Resultado:

```
XDG_GREETER_DATA_DIR=/var/lib/lightdm/data/jean_abraham
XDG_MENU_PREFIX=xfce-
XDG_RUNTIME_DIR=/run/user/1000
XDG_SEAT=seat0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
XDG_SESSION_CLASS=user
XDG_SESSION_DESKTOP=lightdm-xsession
XDG_SESSION_ID=2
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SESSION_TYPE=x11
XDG_VTNR=7
    _='5.2.15(1)-release'
vblank_mode=0
...
```

Capítulo 6 | Terminal

Resumen, quédate con este comando: `set`
Y por lo pronto Memorizare esta variable de entorno: `$HOME`

6.4 Usa las comillas en la terminal

El uso de las comillas de la terminal, ayuda a hacer comandos mas legibles y bonitos.

Porque cuando tenemos un archivo con un nombre con espacios, como este:
`archivo con muchos espacios y cosas locas.txt`

No podemos escribirlo en terminal así:

```
nano archivo con muchos espacios y cosas locas.txt
```

No podemos esperar un resultado bueno. Pasara esto:

```
ERORR
```

Esto es así porque los comandos no detectan los espacios tan fácil, tienes que poner otro carácter feo en lugar del espacio.

Para poder escribirlo con espacios tienes que ponerlo en comillas, así:

```
nano "archivo con muchos espacios y cosas locas.txt"
```

Resultado:

```
Eso si jala.
```

Puedes usar comillas simples o comillas normales. Pero no combines, usa unas u otras, así:

```
"archivo"
```

```
'archivo'
```

No así:

```
"archivo'
```

Resumen:

Esto no: archivo con muchos espacios y cosas locas.txt

Esto si: "archivo con muchos espacios y cosas locas.txt"

Puedes usar comillas simples o comillas normales. Lo que mejor se te acomode mi pana.

6.5 Autocompletado

Ahora el uso del autocompletado en las terminales. Esto nos sirve para ahorrarnos tiempo, y ser mas felices.

**Si tenemos un archivo que tiene un nombre todo feo y largo, como este:
archivo_nombre-feo_largo-23607aca.txt**

Y tenemos que escribirlo, pos podemos presionar la tecla TAB para autocompletar

Ejemplo:

Supongamos que tenemos el archivo "archivo_nombre-feo_largo-23607aca.txt" en la siguiente ruta:

```
/home/Documentos/
```

Pos para escribirlo en terminal hacemos esto

```
/home/Documentos/archivo_nombre
```

(escribimos solo una parte del nombre. Que hueva escribir todo)

Después presionamos TAB y listo:

```
/home/Documentos/archivo_nombre-feo_largo-23607aca.txt
```

Siempre usa el autocompletado, te facilita la vida. No solo sirve para archivos.txt, sirve para cualquier archivo o carpeta.

6.6 Glob con asterisco (*)

¿Como usar el glob?

Bueno este es una función muy util y se puede hacer de varias maneras, pero acá solo veremos una. Es una de las mas chidas. Para usar este glob solo necesitamos teclear la tecla "*". El asterisco.

Ejemplo

Estamos en una carpeta donde tenemos un montón de archivos, entre ellos unos .mp4. Queremos listar esos archivos.mp4.

Estos son los archivos:

```
/ruta/carpeta/
    archivosloco
    archivosloco1
    archivosloco2
    archivosloco3
    video1.mp4
    video2.mp4
    video3.mp4
    video99.mp4
    videoSi.mp4
    videoNo.mp4
```

Para mostrar solo los .mp4, en terminal tecleamos:

```
ls /ruta/carpeta/*.mp4
```

Resultado:

```
video1.mp4
video2.mp4
video3.mp4
video99.mp4
videoSi.mp4
videoNo.mp4
```

Que chimba, solo nos mostró los mp4.
Esta es una forma de usar el glob asterisco.

Capítulo 6 | Terminal

Ahora otro ejemplo:

Tenemos archivos.txt, que comienzan siempre con esto "texto-chido_"; Seguido de cualquier otra cosa, mira:

```
texto-chido_popo.txt  
texto-chido_cocos.txt  
texto-chido_calcio.txt  
texto-chido_rechorcholis.txt  
texto-chido_cacahuete.txt
```

Queremos borrar todo porque somos unos destroyer man.

```
rm texto-chido_*
```

Resultado:

```
Te borro todos los archivos que comienzen con "texto-chido_"
```

También puedes ser mas específico, y solo borrar los que comienzen con "texto-chido_", y que terminene con ".txt"

```
rm texto-chido_*.txt
```

Así te aseguras que solo borre los .txt

6.7 CHMOD y los permisos

Este es un tema importantísimo. Uno de los motivos de que GNU/Linux se dice que es mas seguro, es el tema de los permisos de los archivos.

Los permisos:

- Los archivos para que puedan ser leídos necesitan permisos de lectura.
- Los archivos para que puedan ser modificados necesitan permisos de escritura.
- Los archivos para que pueden ser ejecutados necesitan permisos de ejecución.

Ni las carpetas se salvan de los permisos, ¡las carpetas necesitan de permisos para incluso poder ver lo de adentro!

Normalmente los archivos del OS ya tienen los permisos que requieren. Pero, aveces vamos a querer cambiar los permisos.

Por ejemplo; para usar apps portátiles que descargamos. Y pos lo descargamos y todo eso, pero puede que no lo podamos abrir. Hasta puede que nos salga una ventanilla que nos diga que el archivo no tiene permisos de ejecución.

Con chmod desde la terminal; Veremos la forma básica de dar permisos a los archivos y carpetas.

Capítulo 6 | Terminal

Recuerda que puedes usar "**chmod --help**". Para ver como jala el vato.

Ver permisos del archivo o capreta con ls

Para los archivos

```
ls -l /ruta/archivo
```

Para las carpetas

```
ls -ld /ruta/carpetilla
```

Resultado de ejemplo:

```
-rw-r--r-- 1 user group 1024 Month 1 12:00 ruta/archivo
```

Dar permisos de lectura en archivos o carpetas

Con "r" damos permisos de lectura.

```
chmod +r /ruta/archivo-o-carpeta
```

Dar permisos de escritura

Con "w" damos permisos de escritura

```
chmod +w /ruta/archivo-o-carpeta
```

Dar permisos de lectura y escritura de una pasada

Con "rw" damos eso al mismo tiempo.

```
chmod +rw /ruta/archivo-o-carpeta
```

Dar permisos de ejecución

Con "+x" damos permisos de ejecución

```
chmod +x /ruta/archivo
```

Dar permisos de lectura, escritura y ejecución con chmod. (todo al mismo tiempo)

Con rwx damos todos los permisos. En terminal se tiene que escribir

```
chmod +rwx /ruta/programa.sh
```

(Suponiendo que tu programa es un .sh)

Otro ejemplo:

```
chmod +rwx /ruta/programa.AppImage
```

Otro:

```
chmod +rwx /ruta/programa.bin
```

Capítulo 6 | Terminal

Fíjate que en los ejemplos anteriores tenemos el carácter de mas, este carácter indica a chmod que le añada ese permiso.

Ahora mira estas igualdades para chmod:

0 = --- = sin acceso
1 = --x = ejecución
2 = -w- = escritura
3 = -wx = escritura y ejecución
4 = r-- = lectura
5 = r-x = lectura y ejecución
6 = rw- = lectura y escritura
7 = rwx = lectura, escritura y ejecución

Para dar permisos a los archivos, en vez de usar letras y guión bajo/símbolo menos, podemos usar números así:

(Mira las equivalencias de los números)

Lectura:

```
chmod 444 /ruta/archivo-o-carpeta
```

Escritura:

```
chmod 222 /ruta/archivo-o-carpeta
```

Lectura y escritura:

```
chmod 666 /ruta/archivo-o-carpeta
```

Ejecución:

```
chmod 111 /ruta/archivo-o-carpeta
```

Lectura escritura y ejecución

```
chmod 777 /ruta/archivo-o-carpeta
```

Resumen:

*R de Read. W de Write. X de Execute.
Read; Leer. Write; Escribir. Execute; Ejecutar*

Memoriza esto: r--, -w-, --x, rwx

Memoriza este comando (Te dice como usar chmod): chmod +rwx archivo.sh

6.8 Editor de texto nano

¿Qué es Nano?

Nano es un editor de texto simple y fácil de usar en la terminal.

Cómo Usar Nano (Uso básico)

1. Abrir o crear un archivo

```
nano nombre-del-archivo.txt
```

(Si no existe, lo crea)

2. Escribir o editar texto

```
nano nombre-del-archivo.txt
```

Simplemente escribe en el teclado.

Usa las flechas del teclado para moverte.

Usa las teclas home, end, page up, page down, para moverte súper rápido.

3. Atajos importantes (combinaciones con Ctrl)

Ctrl + o Guardar (Write Out)

Ctrl + x Salir (eXit)

Ctrl + k Cortar línea actual

Ctrl + u Pegar lo cortado

Ctrl + w Buscar texto

Ctrl + \ Reemplazar texto

Ctrl + g Ayuda (muestra todos los atajos)

Ejemplo Práctico:

Primero crea un archivo de texto

Después escribe algo. Guarda con “ctrl + o”. Sal con “ctrl + x”

Puedes acceder al contenido del archivo, volviendo a escribir el primer de creación del archivo de texto.

The screenshot shows a terminal window titled "GNU nano 4.8" with the file name "Main.gd". The code in the buffer is:

```
#Crea los enemigos
func NewEnemy():
    x = 80
    while ii <= 6: #6 filas
        while i <= 7: #7 naves
            var MultiEnemigos = Enemy.instance()
            MultiEnemigos.position = Vector2(x, y)
            x += 126
            add_child(MultiEnemigos)
            i = i + 1

    get_tree().call_group('GrupoEnemigos', 'CambioDeNave')
    i = 1
    ii = ii + 1
    y -= 89
    x = 80

#Movimiento de las naves enemigas
func _on_Timer_timeout():
    get_tree().call_group('GrupoEnemigos', 'mover')
```

A cyan arrow points from the bottom right towards the status bar, which contains the following key bindings:

G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex^J Justificar^C Posición
X Salir ^R Leer fich.^\\ Reemplazar^U Pegar ^T Ortografía^I Ir a linea

```
nano hola_mundo.txt
```

6.9 Privilegios sudo

¿Que es sudo?

El programa sudo (super user do, en inglés) es una utilidad de los sistemas operativos tipo Linux. Permite a los usuarios ejecutar programas con los privilegios de seguridad de otro usuario (normalmente el usuario root) de manera segura, convirtiéndose así temporalmente en super usuario.



Usuario Root:

Root es el nombre de usuario o la cuenta que tiene acceso a todos los comandos y archivos en Linux. Puede hacer todo lo que se le de la gana.

¿Para que sirve sudo?

Permite implementar un control de acceso granulado de que usuarios ejecutan comandos. Pos en Debian sirve para hacer de todo, cualquier cosa, tienes todo el poder. Puedes hasta romper todo el sistema sin mucho esfuerzo.

No somos usuarios root por defecto

Al instalar un OS no somos usuarios root. Hacer todo como root es peligroso, por eso mismo se necesita del uso de sudo, para solo hacer lo mínimo necesario como root.

Hacer actividades root

Para empezar a hacer algo como root, tenemos que usar "su". Pon esto en terminal:

```
su
```

Te pedirá contra y se la pones. (Si no se ve lo que escribes, es normal, igual se pone lo que escribes).

Resultado:

Ahora eres root. Todo lo que hagas desde ahora sera como root.

Para comprobar que somos root, pon en terminal:

```
sudo ls
```

Resultado:

Si te muestra texto, es porque si eres root/admin/super usuario.

Ahora sal del root con:

```
exit
```

Capítulo 6 | Terminal

Resultado:

Seles como root. Ahora eres usuario normal.

Sintaxis de como usar sudo:

```
sudo comando
```

Ejemplo:

```
sudo ls
```

6.10 Agregar privilegios sudo a usuario

1. Entrar como super usuario. Abrimos terminal y entramos como root. Con:

```
su
```

Te pedirá contra y se la pones. (Si no se ve lo que escribes, es normal, igual se pone lo que escribes). (Si no se ve es normal, igual se pone lo que escribes).

1. Resultado parecido:

```
root@usuario:/home/usuario#
```

Se tiene que ver algo así ahora, fíjate que diga “root” al principio.

2. Ver usuarios con los permisos para usar sudo:

Después de haber entrado como root ahora vemos los usuarios con permisos sudo, con sudo -l:

```
sudo -l
```

2. Resultado parecido:

```
Matching Defaults entries for user on pc:  
    env_reset, mail_badpass,  
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/  
/sbin\:/bin, use_pty
```

```
User user may run the following commands on pc:  
(ALL : ALL) ALL
```

Si ves tu nombre, probablemente no necesites permisos root, pero igual ve el proceso.

Capítulo 6 | Terminal

3. Ponemos el comando visudo:

```
sudo visudo
```

3.1 O si no jala el visudo, ponemos el comando:

```
sudo nano /etc/sudoers
```

sudo da permisos necesarios para la edición de archivos root. nano es el editor de texto. /etc/sudoers es el directorio

3. Resultado:

Se ve texto.

4. Buscamos la siguientes lineas:

Para movernos con nano en el texto usamos:

Las teclas: home, fin, page up, page down, flechas del teclado.

Estas teclas nos sirven para subir, bajar, y moverse de izquierda o derecha por el texto.

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
```

Abajo de root o abajo de "# User privilege specification", escribimos:

```
nombre_usuario    ALL=(ALL:ALL) ALL
```

Recuerda que tu nombre se ve en la terminal al lado izquierdo del arroba:

```
nombre_usuario@pc:~$
```

Asegúrate de escribir el nombre igual como esta, o si no no jala.

5. Guardamos y cerramos:

Precionamos ctrl+s

Precionamos ctrl+x

Ahora somos podemos usar sudo.

Si estamos en terminal y seguimos viendo

```
root@nombre_usuario
```

Ponemos un exit:

```
exit
```

Así salimos del modo root.

Si usamos "sudo visudo" cerramos sesión y volvemos a entrar. O reinicia el pc.

Capítulo 6 | Terminal

Verificar que somos root

Primero tenemos que abrir una terminal como usuario normal, no como root.

Y para ver si somos root escribimos

```
sudo -l
```

Resultado:

Nos pide contra y nos mostrara un texto; Allí debería estar nuestro nombre de usuario.

Ya por el simple hecho de que nos pide contra, ya significa que somos usuarios con permisos sudo

Ejemplo de resultado:

```
Matching Defaults entries for nombre_usuario on pc:  
    env_reset, mail_badpass,  
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/  
/sbin\:/bin, use_pty  
  
User nombre_usuario may run the following commands on pc:  
    (ALL : ALL) ALL
```

Resumen:

Pon el comando "sudo visudo"

O pon el comando "sudo nano /etc/sudoers"

Busca las lineas:

```
# User privilege specification  
root    ALL=(ALL:ALL) ALL
```

Y abajo de esa linea pon:

```
nombre_usuario    ALL=(ALL : ALL) ALL
```

6.11 Comentarios

Comentarios en bash:

Los comentarios en bash se indican con el carácter "#". Si, el gatito.

Para un comentario, se escribe primero el gato "#", seguido del comentario. Los comentarios no hacen absolutamente nada en la computadora, sirven para decirle a los desarrolladores/usuarios comentarios, información.

Ejemplo:

```
# Esto es un echo, se pone con la intención de saltar una linea  
echo
```

También lo puedes poner al lado derecho del echo

```
echo # Este es el clasico echo.
```

Prueba usar comentarios, ejecuta esto en la terminal:

```
echo "Hola mundo, aca hay un comentario pero no se vera"  
# Este es un echo classic. Que bonito.
```

Los comentarios no hacen nada, es solo info para el usuario que lo ve.

```
# Comentario.
```

6.12 Archivos desktop

Los "archivos.desktop". Estos archivos son accesos directos, son ejecutables. La mayoría de distribuciones GNU/Linux los pueden ejecutar. Debian es compatible.

Todos los iconos que ves en el menú de aplicaciones son "archivos.desktop".

Estos archivos dicen que programa ejecutar, a que categoría pertenece el programa, si se ejecutara en terminal, la versión, entre otras cosillas.

Vamos a hacer un "archivo.desktop" para que veas como jala.

Pon los siguientes comandos en terminal en orden:

*Explicación del lo que haremos; Estableceremos dos variables, la ruta, y el archivo.
La ruta, es la carpetilla, y el archivo es el "archivo.desktop"*

paso 1:

```
ruta=$HOME/.local/share/applications
```

paso 2:

```
archivo=$ruta/prueba-feliz.desktop
```

Capítulo 6 | Terminal

paso 3:

```
mkdir $ruta
```

paso 4:

```
touch $archivo
```

paso 5:

```
nano $archivo
```

paso 6:

```
# Dentro de nano: Copia esto y pegalo
# se pega en terminal con "ctrl + shift + v"
[Desktop Entry]
Encoding=UTF-8
Version=1.0
Type=Application
Name=Desktop default
Comment=Comentario default
Path=/home
Exec=man yes
Icon=
Terminal=true
Categories=System;
```

paso 7:

guardamos en nano con "ctrl + s". salimos de nano con "ctrl + x"

paso 8:

```
chmod 777 $archivo
```

paso final:

```
$archivo
```

paso final alternativo:

*Si no te abre nada el paso final, entonces; Entra a la ruta con tu explorador de archivos favorito.
\$HOME/.local/share/applications*

Busca "prueba-feliz.desktop" y dale dos clicks.

Explicación de lo hecho.

Capítulo 6 | Terminal

Establecemos las variables “ruta” y “archivo”, ruta es la carpeta donde se guardan los iconos locales y archivo es “prueba-feliz.desktop”.

Creamos la carpeta. Creamos el archivo.desktop. Editamos el archivo.desktop. Le dimos permisos al archivo.desktop. Y ejecutamos el archivo.desktop

Nota: Algunos entornos de escritorio o sistemas operativos, te pedirán marcar el archivo como ejecutable desde lo gráfico, por eso esta el paso final alternativo.

Resumen: Los archivos "desktop", son ejecutables para ejecutar otros ejecutables. Accesos directos.

6.13 AND OR NOT

El operador lógico AND.

Este operador te permite poner varios comandos al mismo tiempo, en una sola linea. Se utiliza el carácter "&&". Dos andpersan. Pones un comando, seguido de otro, ejemplo:

```
comando1 && comando2
```

Puedes poner tantos comandos como quieras. Solo asegúrate de poner los andpersan "&&". Y fíjate que al final del ultimo comando, ya no se pone andpersan, se deja el puro espacio.

Con esto puedes hacer cosas mas rápidas y chulas. **Ejemplo de uso:**

```
echo $HOME && echo "Hola" && ls
```

En la terminal, este es el operador que mas usarás.

Existen mas operadores, pero esos no los veremos, solo los mencionaremos:

- **OR:** Operador lógico "hacer esto o hacer esto". Se pone con "||", dos pipes.
- **NOT:** Operador lógico "no hacer esto". Se pone con "!", signo de admiración de cierre.

Usa AND, siempre que lo veas util "&&".

Otra forma de poner un "and", es con el punto y coma. Sintaxis:

```
Comando; ComandoDos; OtroComando;
```

Ejemplo:

```
clear; echo "Hola, mira estos archivos:"; pwd; ls
```

6.14 Uso de pipe en bash

¿Que es un pipe?

Pipe es una función que toma la salida de un comando, y lo usa como entrada para otro comando. El símbolo para el pipe es '|'. El pipe une dos comandos.

Comando usando pipe "|":

```
ls | cat
```

Acá la salida de "ls", es usada como entrada para "cat". cat muestra la salida de "ls".

Es decir en la derecha del pipe "|" se pone el comando que da una salida, y en la derecha del "|" se pone el comando que usara la salida.

Comando usando pipe:

```
ls | cat | echo $** "Tamos usando pipe perros"
```

Acá la salida la proporcionan "ls | cat", y ya vimos que sale de allí.

Y esa salida el comando 'echo \$** "Tamos usando pipe perros"' la recibe de entrada.

Resumen: Los pipes te permiten realizar procesos de manera mas rápida. Realmente útil.

6.15 Uso de grep

¿Que es grep?

grep es un programa, cuya función es buscar PATRONES en cada ARCHIVO. PATRONES consistirá en uno o más patrones separados entre si por un salto de línea.

grep no solo se limita a buscar patrones en archivos, sino también en buscar palabras o cadenas de texto en archivos o en la entrada estándar de la terminal. grep hace muchas cosillas. grep pertenece a la familia Unix.

:Cómo utilizar el comando grep en la terminal?. Sintaxis de uso de grep:

```
grep [opciones] pattern [archivo]
```

Recuerda que puedes ver sus opciones con –help:

```
grep --help
```

Ejemplo usándolo en archivo de texto:

```
grep bash $HOME/.profile
```

Acá se buscara la palabra "bash", en el archivo de texto \$HOME/.profile

Capítulo 6 | Terminal

Ejemplo usando pipe, en vez de usar archivo, el puro string:

```
help | grep gnu
```

Lo que devuelva help, se usara como entrada para grep, y grep buscara entre ese texto, la palabra gnu. Devolverá las lineas que tengan gnu.

Ejemplo usando parámetro -i:

```
man man | grep -i manless
```

Con el -i, busca "texto", sin importar si tienen mayúsculas o minúsculas. Por defecto busca todo bien específico, que coincida mayúsculas y minúsculas.

Ejemplo conteo de palabras que coinciden con la búsqueda:

```
grep -c bash $HOME/.profile
```

Cuenta las veces que sale la palabra "bash" en el archivo de texto "\$HOME/.profile"

Ejemplo buscar varias palabras:

```
grep bash $HOME/.profile && grep run $HOME/.profile
```

Buscamos las palabras bash y run, en el archivo "\$HOME/.profile"

Ejemplo encontrar palabra en varios archivos a la vez:

```
grep system /etc/*.conf
```

Busca la palabra system, en todos los archivos.conf

```
grep -l system /etc/*.conf
```

Busca la palabra system, en todos los archivos.conf. Pero solo muestra el nombre del archivo que contiene esa palabra.

Cambiar grep con otros comandos (usando el pipe):

```
cat /etc/sudo.conf | grep log
```

Resumen: Usa grep para buscar palabras o archivos. Muy rápido y eficiente.

6.16 Uso de apt

¿Como instalar aplicaciones, drivers, entre otros softwares?

Para esto usaremos **dpkg** y **apt**. Ambos se complementan; **apt** usa internet para bajar paquetes, pero también jala sin internet, pero pos no podría bajar paquetes, solo podría gestionar los paquetes locales.

Bueno para usar apt tendrás que usar "sudo", "sudo" permite ejecutar comandos con permisos de super usuario. Un super usuario permite leer, modificar, eliminar o todo esos archivos root.

Archivos con permisos root. Algunos de estos archivos son necesarios para el buen funcionamiento del OS, por lo que su modificación requiere de permisos sudo.

Capítulo 6 | Terminal

Primero que nada pon en terminal

```
apt -h
```

Y luego:

```
dpkg - -h
```

Estos comandos te dirán que funciones tiene apt y dpkg respectivamente.

Usaremos cinco de sus funciones de apt:

update, upgrade, search, install, uninstall

¿Que pasa cuando usamos sudo?

Pos la terminal nos pedirá contraseña de root. Deberíamos saber la contra. Y la escribimos y presionamos enter. Lo normal es que no se vea lo que escribimos, pero si se esta poniendo el texto. En algunas distros si se ve, pero no en todas. Son detallitos bonitos, pero no necesarios.

Actualizar el OS

Para actualizar el OS necesitamos de los permisos sudo.

Paso 1. Este comando sirve para que apt sepa que programas necesitan una actualización:

```
sudo apt update
```

Paso 2. Este comando ahora si ya actualiza todo morocho:

```
sudo apt upgrade
```

Resultado:

```
....  
....  
¿Continuar? S/N
```

(o quizás te pregunte en english)

```
¿Continue? Y/N
```

Paso 3 y final. Pos tu escribes "s" o "y" (dependiendo si te pone "s/n" o "y/n"). Y presionas el enter. Puede que te ponga varias veces eso del S/n, y pos tu le mandas que si "s".

Así se actualiza el OS.

Capítulo 6 | Terminal

¿Como buscar una aplicación con apt?, Así:

```
sudo apt search app-que-quiero-buscar
```

No nos hace falta el sudo, pero vamos ponerlo porque queremos buscar en modo loco.

Ejemplo:

```
sudo apt search supertux
```

Recuerda que acá en apt todas las apps se escriben de seguido, sin espacios, lo que si pueden tener son guiones/signos de menos, o barras bajas. Así: app-feliz, no app feliz.

Retomando el comando “apt search supertux”. Seguro te salio algo así entre las opciones:

```
supertux/stable 0.6.3-2 amd64  
Classic 2D jump 'n run sidescroller with Tux
```

Vamos a instalarlo

¿Como instalar apps?

Para instalar apps necesitamos de los permisos sudo. Sintaxis: `sudo apt install app`

Ejemplo: `sudo apt install supertux`

¿Como desinstalar apps?

Sintaxis de como desinstalar app con apt: `sudo apt uninstall app`

También se puede con: `sudo apt purge app`

Ejemplos:

```
sudo apt uninstall supertux
```

```
sudo apt purge supertux
```

Y si te gusta el supertux (esta en el menu de apps), pos instalalo de nuevo. ¡Si que si!

```
sudo apt install supertux
```

¿Como reinstalar apps con apt?

Y después de instalado vamos a reinstalarlo porque estamos locos:

```
sudo apt reinstall supertux
```

Capítulo 6 | Terminal

Limpieza con apt

```
sudo apt clean
```

Elimina todos los .deb descargados

```
sudo apt autoremove
```

Solo borra los .deb antiguos/no actualizados

```
sudo apt purge $(dpkg -l | grep '^rc' | awk '{print $2}')
```

Buscar y eliminar configuraciones residuales

Explicación:

- “dpkg -l | grep ‘^rc’” *Lista paquetes desinstalados con configs residuales.*
- “awk ‘{print \$2}’” *Extrae solo los nombres.*
- “apt --purge” *Elimina completamente (paquete + configs).*

Solo descargar programas y guardarlos.

Sintaxis: `apt download programa`

Ejemplo: `cd $HOME && apt download nano`

Nota importante:

`sudo apt update` *Nos permite obtener los paquetes disponibles.*

Por lo que si no jala el upgrade o el install/reinstall, es porque necesitamos poner primero sudo apt update.

Pon siempre “sudo apt update” antes de hacer cualquier cosa que implique usar internet con apt.

En el próximo capítulo veremos como usar dpkg porque este es mas canijo de usar, apt usa dpkg por nosotros. apt es buena onda, dpkg también pero este es mala leche.

Resumen: *Usa apt para instalar la mayoría de las apps.*

6.17 Uso de dpkg

Primero que nada, **dpkg** no usa internet, es local, para poder usar **dpkg** necesitas de un paquete instalado o de un paquete a instalar. **dpkg** gestiona la instalación y desinstalación de aplicaciones de debían.

En debían, los paquetes de aplicaciones son **archivos.deb**. Es decir las aplicaciones que podemos instalar con debian son **archivos.deb**

Buscar aplicaciones instaladas.

```
con --list o -l    dpkg --list    dpkg -l
```

Acá no hace falta usar sudo, porque no necesitas permisos de root para ver los paquetes instalados.

```
dpkg --list | grep text
```

Podemos buscar apps instalados con grep, y solo mostrar lo que tenga algo de lo indicado en el grep. Tenemos que usar el pipe "|" luego el grep, seguido de lo que queremos "filtrar"

Ejemplos:

```
dpkg --list | grep  
python
```

```
dpkg -l | grep gtk
```

Para instalar aplicaciones hacemos lo siguiente

La sintaxis: **sudo dpkg --install archivo.deb**

Ejemplo:

Supongamos que descargamos "chrome.deb", y pos esta en la ruta "\$HOME/descargas/"

Pos lo instalamos asi:

```
sudo dpkg --install $HOME/descargas/chrome.deb
```

O también se puede nomas con -i:

```
sudo dpkg -i $HOME/descargas/chrome.deb
```

Puedes instalar varias apps (usando el asterisco vato):

```
sudo dpkg -i $HOME/descargas/*.deb
```

Capítulo 6 | Terminal

Como desinstalar aplicaciones

Usaremos `--remove -r`, o también `--purge -P`. Usa `remove` para asegurarte de solo borrar. Acá solo podemos borrar los paquetes instalados en dpkg.

Acuérdate de como buscar paquetes en dpkg

Ejemplos:

```
sudo dpkg --remove nombre-de-paquete  
sudo dpkg -r nombre-de-paquete  
sudo dpkg --purge nombre-de-paquete  
sudo dpkg -P nombre-de-paquete
```

Diferencias entre `remove` y `purge`.

- **Remove:** Elimina el paquete, pero deja sus archivos de configuración en el sistema
- **Purge:** Elimina el paquete junto con sus archivos de configuración.

Verificar la integridad de un paquete

Sirve para asegurarnos de que el paquete tiene todo lo necesario para jalar, si no tiene lo necesario, puede que no jale o que jale mal.

Usaremos el parámetro: `-V` o `--verify`

Ejemplos:

```
sudo dpkg --verify nano  sudo dpkg -V grep
```

Situación; Paquete roto. ¿Que hacer cuando un paquete de dpkg esta todo roto y no jala?:

```
sudo dpkg --configure -a nombre-de-paquete
```

Esto fuerza la configuración de paquetes a medio instalar o en estado inconsistente.

Limpieza en dpkg

Con este comando se limpian archivos que no tienen sentido que existan:

```
sudo dpkg --purge $(dpkg -l | grep '^rc' | awk '{print $2}')
```

Explicación:

```
dpkg -l | grep '^rc'  
awk '{print $2}'  
dpkg --purge
```

Lista paquetes desinstalados con configs residuales.

Extrae solo los nombres.

Elimina completamente (paquete + configs).

Esto es la información básica del uso de DPKG

Capítulo 6 | Terminal

Recuerda usar --help si necesitas mas info: `dpkg --help`

Resumen: Usa `dpkg` para instalar programas fuera de los repos apt. Con este programa gestionas todos los programas instalados, de forma local.

6.18 Agregar repositorios apt non free

Como agregar repositorios non free

Repositorios apt non free:

Estos contienen aplicaciones no open source, aplicaciones con permisos de uso y todo eso. Es posible que te priven de muchas libertades.

Estos repos son necesarios para instalar drivers privativos, entre otros programas privativos. La mayoría de linuxeros usan estos repos para instalar drivers.

Agregar estos repos solo te da la posibilidad de instalar programas privativos desde apt. Tenerlos disponibles en si, no te instala nada, eso lo tienes que hacer vos chavalin/dubalin.

Ver repositorios actuales

```
sudo apt-add-repository --list
```

Resultado:

(Acuérdate que sudo te pide contra) (Te mostrara algo asi)

```
deb http://deb.debian.org/debian/ bookworm main
deb http://security.debian.org/debian-security main
deb http://deb.debian.org/debian/ bookworm-updates main
```

Observa lo que esta al final de las lineas, "main", puede que tu tengas solo "main" o tengas mas palabras como: contrib, non-free-firmware

Si tienes contrib y non-free-firmware, en las tres lineas, es porque ya tienes los repos non-free, pero igual vamos a ponerlos (si ya están no te agrega nada mas, lo dejaría igual).

Agregar repositorios apt non free

Para hacerlo, necesitamos usar permisos de super usuario.

Mostraremos dos métodos, el **método 1 "el mas fácil y repidote"**, El **método 2 "mas lento pero es bueno saberlo hacer"**.

Elije el método que quieras, pero si no te decides por cual, elige el método 1. (*Todos los pasos son comandos*)

Agregar repositorios apt non free; Método 1:

Capítulo 6 | Terminal

Paso 1: `sudo apt-add-repository contrib`

Paso 2: `sudo apt-add-repository non-free`

Paso 3: `sudo apt-add-repository non-free-firmware`

Paso 4: `sudo apt-add-repository --list`

Vemos el resultado, si salen las palabras non-free, contrib y non-free-firmware. Lo logramos chavales.

Agregar repositorios apt non free; Método 2:

Paso 1: `sudo nano /etc/apt/sources.list`

Paso 2; Editamos el archivo:

Dentro del archivito, vemos buscamos las siguientes líneas:

```
deb http://deb.debian.org/debian/ bookworm main
deb-src http://deb.debian.org/debian/ bookworm main

deb http://security.debian.org/debian-security bookworm-security
main
deb-src http://security.debian.org/debian-security bookworm-
security main

deb http://deb.debian.org/debian/ bookworm-updates main
deb-src http://deb.debian.org/debian/ bookworm-updates main
```

Al final de cada linea (a la derecha de main) le ponemos esto: contrib non-free non-firmware non-free

Paso 3: Guardamos con "ctrl + s", y luego salimos con "ctrl + x"

Paso 4: `sudo apt-add-repository --list`

Vemos el resultado, si salen las palabras non-free, contrib y non-free-firmware. Lo logramos chavales.

Como deshabilitar los repositorios

Capítulo 6 | Terminal

Paso 1:

```
sudo apt-add-repository --remove contrib non-free non-free-firmware
```

Paso 2: **sudo apt-add-repository --list**

Verificamos que ya no estén las palabras que le dijimos que removiera.

Nota:

En debian 12, "non-free" ya no existe ahora se llama "non-free-firmware", pero igual nosotros lo agregamos porque si. Si pones algo que no existe, no pasa nada, nomas el apt lo ignora. Pero de preferencia pon todo bien.

Si estas en debian 12 y quieres dejar todo clean pos nomas remueve el "non-free" y solo deja el "non-free-firmware".

Esos de debian 12 se pusieron a cambiar nombres. Pero bueno, seguro les sirve para tener todo ordenado y desarrollar mas agusto, y eso pos ta bien.

Resumen:

Agregar repos nunca fue tan fácil. Que bonito, entre mas usuarios tenga Linux todo es mucho mejor y mas perron.

Aggrega los repos "main contrib non-free" y "non-free-firmware"

Estas son los repos mas bonitos de debian 12. Siuuu!

6.19 Limpieza general

Limpieza general del sistema operativo Debian. Los programas generan varias archivos basura, estos archivos se pueden llegar a pesar mucho. Estos archivos tienen múltiples funciones, pero no son dependencias para que jalen los programas. Por lo que es bueno eliminarlos una que otra vez, digamos; “Cada vez que te te acuerdes”.

Algunos ejemplos de archivos basurilla:

- cache de apps
- miniaturas de apps
- temporales duplicados
- temporales que no se eliminaron porque son rebeldes

Comando apt limpiar cache: **sudo apt clean**

Comando apt limpiar dependencias que ningún programa usa: **sudo apt autoremove**

Capítulo 6 | Terminal

Limpiar archivos temporales y duplicados con la app Fslint: `sudo apt install fslint`

Esta app elimina todo el mugrero. Es bonita y facil de usar. No temas en usar esta app, esta no puede romper tu sistema operativo.



Para hacer todo la demas limpieza, puedes usa bleachbit: `sudo apt install bleachbit`

O desde su sitio web: <https://www.bleachbit.org/>

Una vez instalado selecciona que se limpiara

Lee muy los descripciones de las opciones, porque esta app elimina de todo. Pero nada que sea de gran dependencia, es decir, esta app no puede romperte el sistema operativo.



Administrar uso de programas

Usar un gestor de tareas. Busca en tu menú de apps, alguna app que diga "gestor de tareas", "monitor de recursos", o algo asi, o sino usa "**htop**"

Estas apps te permiten ver que apps usan mas memoria, o mas cpu, o cualquier cosa. Dependiendo de el uso de la app, pues veras que heces con esa app. **htop**, es sin duda alguna de las mejores, aunque no es la mas bonita, es la mas conocida y a demostrado tener datos precisos.

Ya como ultimo, seria eliminar los kernels viejos, pero eso ni te voy a mencionar como se hace. La verdad eso lo dejo a que lo busques tu, porque eliminar kernels viejos, suele ser una mala idea, estos no pesan tanto y sirven como backup.

Resumen:

La lentitud del sistema operativo puede ser porque el pc no jala bien, o no tiene el power para el OS. Pero eso ya es harina de otro costal, eso ya es tema hardware.

Limpiar el sistema operativo, es eliminar archivos innecesarios. Detener servicios o programas innecesarios. Un sistema operativo super limpio, es un sistema operativo con su máxima velocidad.

La suciedad no siempre es culpa del sistema operativo, muchas veces es por algún programa o varios programas, el usuario es el que debe gestionar todo lo que instala para garantizar que el sistema operativo este limpio y reluciente.

Pero; la posibilidad de que el sistema operativo sea lento por si solo, es posible. Recuerda ver los requisitos de uso del sistema operativo.

Siendo sincero, la limpieza del sistema operativo es compleja, porque requieres de comprender muchas cosas, tanto de hardware como de software; Pero generalmente con estos programas y comandos debería ser mas que suficiente.

6.20 Los Logs/Registros

Los logs o también llamados los registros. Se refieren a un montón de información del funcionamiento del programa. Es pura información para analizar.

Gracias a estos archivos podemos verificar el buen o mal funcionamiento de los procesos de los programas.

Los logs no tienen función para el programa, tienen función para el desarrollador y para el usuario. Su función es informar a la gente.

Supongamos que usas el bluetooth para conectar unos audífonos, pero de repente te sale un error, y se te desconectan los audífonos. Bueno para ver el error que paso, podrías buscar el log del bluetooth y ver que onda.

Los logs suelen ser muy descriptivos, pero a veces te pueden decir "sucedió el error 22867 cuando se desconectó el dispositivo 232461". Para leer algunos de estos logs locos tienes que buscar e informarte que significan esos números/códigos de error.

Quédate con lo siguiente; Los logs o registros, son archivos de texto que contienen información relacionada con el funcionamiento de los procesos de los programas.

Estos archivitos de log, se almacenan acá: `/var/log/`

Comando para ver el Log del inicio de sesión: `last -f /var/log/wtmp`

Este comandito, te devuelve el las veces que iniciates sesión, con el reboot

Comando para ver lo que a hecho dpkg: `sudo cat /var/log/dpkg.log`

Comando para ver la chamba que a hecho el karnel: `sudo dmesg`

Comando para los servicios iniciados para el inicio de sesión:

`sudo cat /var/log/boot.log`

Comando para ver todos los logs, porque estamos en modo curioso:

`sudo cat /var/log/*.log`

Esto es tanta info que nomas es algo para probar ver logs.

Ver logs de con interfaz gráfica, botones y esas cosas. Con la app: gnome-system-log

La puedes buscar, en apt: `sudo apt search gnome-system-log`



Si existe, se instalaría con el comando: `sudo apt install gnome-system-log`

Capítulo 6 | Terminal

Resumen: Los archivos.log, tienen información util, esta información sirve para determinar el funcionamiento de los procesos de los programas.

Existen muchos mas archivos.log, si quieres saber de su existencia puedes darle un ls a la ruta "/var/log/" .

6.21 Servicios Systemd

¿Que es systemd?

Es un conjunto de herramientas, libreras y servicios diseñados para administrar y configurar de manera centralizada todos los servicios en los sistemas operativos con kernel Linux.

Systemd, viene con varias herramientas para administrarlo:

- **systemctl**: sirve para controlar tanto el sistema, como los servicios del administrador.
- **journalctl**: Administra el sistema de registro.
- **hostnamectl**: sirve para controlar el nombre de host
- **localectl**: configuración de la localidad y teclado.
- **timedatectl**: para poner hora y fecha al os.
- **systemd-cgls**: te dará la posibilidad de mostrar el contenido de cgroup
- **systemadm**: interfaz para el comando systemctl

Dabian usa Systemd. La mayoría de OS GNU/Linux tienen de init system Systemd.

Acá solo usaremos systemctl. Esta herramienta nos servirá para gestionar los servicios de systemd.

Usaremos systemd, para ver servicios activos, ver servicios disponibles, y habilitar o deshabilitar servicios.

Estos son solo sintaxis de uso, no son comandos funcionales, pero te sirven para entender como se usa.

Te recomiendo no usar los comandos de habilitar o deshabilitar al menos que sepas cien por cien que servicio habilitar, deshabilitar, iniciar o cerrar. Limítate a solo ver como jala el systemctl. Los comandos que no tengan que ver con esto, los puedes probar.

Los nombre .service, se les puede poner solamente el nombre, el .service, ya lo mete solito el systemctl. Es decir puedes poner nombre-de-servicio, o nombre.service. Las dos formas son correctas.

Ver los servicios disponibles: `systemctl list-units --all | grep .service`

Recargamos servicios:

```
sudo systemctl daemon-reload  
# Sirve por si algún servicio fue modificado, quitamos o agregamos servicios.
```

Ver estado de servicio: `sudo systemctl status nombre.service`

Capítulo 6 | Terminal

Habilitar servicio:

```
sudo systemctl enable nombre-servicio  
# El proximo inicio de sesion, se abrirá lo que diga el servicio habilitado
```

Deshabilitar servicio:

```
sudo systemctl disable nombre.service  
# El proximo inicio de sesion, no se abrirá lo que diga el servicio
```

Iniciar/Probar servicio:

```
sudo systemctl start nombre-servicio  
# Abrir el servicio de una. Pero este comando no lo habilita para que en el proximo inicio de sesion se abra solito.
```

Cerrar/Detener servicio:

```
sudo systemctl stop nombre.service  
# Cerrar el servicio de una. Pero este comando no lo desabilita para el proximo inicio de sesion.
```

Ver como funciona el servicio: `sudo systemctl cat nombre-servicio`

Instrucciones de ejemplo para crear tu propio servicio.

En la ruta “/etc/systemd/system/”, creamos un `archivo.service`.

Comando:

```
sudo touch /etc/systemd/system/ejemplo-feliz.service  
sudo nano /etc/systemd/system/ejemplo-feliz.service  
# Creamos y editamos archivo
```

Después, le metemos texto con esta estructura:

```
[Unit]  
Description=Ejemplo de uso  
  
[Service]  
Type=oneshot  
ExecStart=comando  
RemainAfterExit=true  
  
[Install]  
WantedBy=multi-user.target
```

Lo probamos:

```
sudo systemctl daemon-reload  
sudo systemctl start ejemplo-feliz.service
```

Lo agregamos como servicio:

```
sudo systemctl daemon-reload  
sudo systemctl enable ejemplo-feliz.service
```

Capítulo 6 | Terminal

O después de agregado lo quitamos:

```
sudo systemctl daemon-reload  
sudo systemctl enable ejemplo-feliz.service
```

Resumen: *Systemd es un conjunto de herramientas útiles para el OS, entre esas herramientas hay una que permite la gestión de servicios del sistema operativo.*

Los servicios del sistema operativo son archivos con instrucciones que se ejecutan al iniciar el pc. Estos archivos generalmente abren programas que permiten el buen funcionamiento de otros programas.

6.22 ¿Qué es bash script?

Bash script; este es una forma de escribir comandos ordenada y automatizada. Este te permite poner sentencias lógicas, y hacer procesos mas complejos. Es como cuando pones comandos en la terminal. Pero con un bash script haces todo mas avanzado, mas mecanizado.

Un bash script, es un "archivo .sh", este archivo contiene el script; todo el código de dicho script.

Ejemplo de lo contenido en un archivo.sh:

```
#!/bin/bash  
echo "¡Hola Mundo!"
```

Estos archivos necesitan de permisos de lectura, escritura y ejecución, para poder ser usados.

Resumen: *Bash script te permite meter comandos en bash. Te permite hacer todo mas personalizado y automatizado.*

Para aprender a usar bash script al máximo power, tienes que aprender a programar en bash script. Como es su sintaxis y que funciones tiene.

7 Obtener información del hardware

Los componentes del PC. La computadora tiene componentes de software, y componentes de hardware, acá hablaremos del componente hardware. El hardware es toda la parte física de la computadora.

Componente hardware del PC: Es un dispositivo electrónico, que es una pieza, un modulo para la computadora, el cual hace una función específica.

Los componentes de computadora que veremos serán los siguientes:

- Motherboard
- Procesador CPU
- Memoria RAM
- Tarjeta gráfica
- Almacenamiento

Los conceptos que se verán son para “entender de manera superficial”, la función que tienen, estos componentes son bastante mas complejos, su funcionamiento es ridículamente amplio.

Explicación de la función de los componentes:

- **Motherboard:** Es la base para poner todos los demás componentes, esta se encarga de administrar la comunicación entre todos los componentes, contiene todo lo necesario para que la comunicación entre los componentes suceda. Por eso se le dice que es la placa madre, la placa base; todos esos nombres se refieren a lo mismo, a la motherboard.
- **Procesador CPU:** El cerebro de la computadora, este componente es el encargado de hacer todos los procesos de análisis complejos, este componente recibe información y la procesa.

CPU es la abreviación de "Central Processing Unit", en spanish: Unidad Central de Procesamiento

- **Memoria RAM:** Este componente se encarga de almacenar información de manera temporal. Es a información la envía a una velocidad realmente rápida a los componentes que lo requieran. Los programas la utilizan constantemente enviar mucha info.

Al apagar el pc, toda la información almacenada en la RAM sera eliminada. La información que puede contener esta limitada, pero como esta en constante agregación y eliminación de información, pos jala bien.

Mientras este encendido el PC la ram sera usada, por lo que es necesario siempre apagar el PC, nada de suspender o hibernar, lo mejor es apagar.

- **Tarjeta Gráfica:** Este componente tiene la función específica de enviar imagen a la pantalla. Renderizar todo lo que cualquier programa requiera mostrar. También es conocida como; Placa de vídeo y Procesador de vídeo.

Capítulo 7 | Obtener información del hardware

- **Almacenamiento:** Este componente tiene la función de poder gestionar información de forma permanente. Lo que guardes se quedara. Aquí es donde se guarda todo, desde el sistema operativo, hasta sus programillas y archivos personales.

Algunos dispositivos de almacenamiento:

- Solid Static Disk. Disco estático sólido. SSD
- Hard Drive. Disco Duro

Actualmente (2025), lo que mas se usa es el SSD, ha demostrado ser superior al disco duro en muchas cuestiones.

Resumen: La motherboard es el componente base donde se instalan todos los demás componentes. El procesador procesa cualquier información, la ram se encarga de enviar información de forma muy rápida, las tarjetas gráficas procesan el vídeo. El componente de almacenamiento, es donde se guarda todo.

7.1 CPU

Ahora vamos a aprender como obtener información de los componentes. Obtener información de los componentes nos permitirá determinar si andan jalando bien o mal.

Como obtener información del procesador: Con el comando "lscpu", este te mostrara la información principal de tu procesador. **lscpu**

El archivo que dice información de tu procesador: También podemos probar leer el archivito "/proc/cpuinfo". Este archivo dice información de tu porcesador. **cat /proc/cpuinfo**

Si nomas quieres ver su nombre: **cat /proc/cpuinfo | grep "model name"**

Aplicación CPU-X

Página web de CPU-X: <https://thetumultuousunicornofdarkness.github.io/CPU-X/>

Esta es una aplicación hecha para jalar en sistemas operativos GNU-Linux. Esta aplicación proporciona información de tu procesador, su nombre, su uso en tiempo real. Tiene interfaz gráfica, y su uso es visual.

Esta aplicación tiene otras funciones secundarias que son;

- Benchmark para procesador.
- Ver información de: Procesador, Motherboard, Memoria RAM, Tarjeta grafica

Su nombre en código es "cpu-x"

Este programita esta en debían, se instala con: **sudo apt install cpu-x**

Capítulo 7 | Obtener información del hardware

En esta aplicación no hay pierde, solo se abre y ya puedes ver la info del procesador. Para obtener información mas privilegiada puedes abrirlo en terminal con sudo: `sudo cpu-x`

Resumen: Lee el archivo "/proc/cpuinfo" para ver información de tu cpu, o usa el comando "lscpu". Usa cpu-x para ver info del procesador, tarjeta gráfica, y memoria ram. Para obtener información adicional, puedes abrirlo con sudo.

7.2 RAM

Obtener información de la memoria ram. Utilidad "free". Esta te permite obtener la información de uso de la memoria ram, y el espacio swap.

Comando: `free -h` El menos -h es para que se vea la información mas bonita/legible.

Archivito "/proc/meminfo", este contiene la información de la ram. Comando obtener mucha info de la ram: `cat /proc/meminfo`

Comando solo obtener la capacidad total de memoria: `grep MemTotal /proc/meminfo`

Comando obtener información de la memoria ram con el programa: dmesg

Comando para obtener capacidades de memoria: `sudo dmesg | grep "Memory"`

Resumen: Puedes usar los programas free y dmesg, para obtener info de la memoria ram. Puedes leer el archivo "/proc/meminfo", para ver información de la memoria ram.

7.3 Almacenamiento

Programa df: Para ver el almacenamiento podemos usar el comando "df -h"

Comando: `df -h` Con el parámetro -h, hacemos que el resultado sea mejor presentable.

Programa du: Este te permite mostrar el peso de todas las carpetillas del OS.

Comando: `du -h | sort -h`

"du -h" indica el peso de todos los archivos y carpetas. "sort -h", acomoda de menor a mayor. Mero arriba los menores y mero abajo los mayores.

Capítulo 7 | Obtener información del hardware

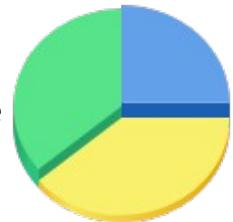
Programa lsblk: Este programa te permite hacer de todo todito. Permite ponerle muchos parámetros. **Comando:** `lsblk -a -o LABEL,SIZE,MOUNTPOINT,FSUSED,FSUSE%`

El `-a` muestra todos los dispositivos de almacenamiento que encuentre el OS. El `-o` contiene las columnas que se mostraran. Las columnas son las siguientes: Muestra su etiqueta `LABEL`, su tamaño `SIZE`, su punto de montaje `MOUNTPOINT`, su uso en bytes `FSUSED`, y su porcentaje de uso `FSUSE%`

Analizador de uso de disco Baobab

Este programa final, lo tienes que instalar, es el "**Baobab**", este programa te permite ver información de tu almacenamiento. Créeme, te muestra la información del almacenamiento de manera muy bonita, una aplicación bastante útil. **Sitio web:** <https://apps.gnome.org/es/Baobab/>

Instalación con apt: `sudo apt install baobab`



Resumen: Puedes usar los programas "df", "nu", para ver el uso de bytes de tu almacenamiento. Puedes usar "lsblk", para ver cualquier información relacionada con el funcionamiento del almacenamiento. Puedes usar la aplicación Baobab, para ver información del almacenamiento de manera gráfica y visual.

7.4 Tarjeta gráfica

lspci: Este programa te permite obtener información relacionada con los dispositivos pci, aca esta la información de la tarjeta gráfica. Lista todos los dispositivos PCI.

Comando lspci para ver nombre de la tarjeta gráfica: `lspci | grep VGA`

"lspci" lista todos los dispositivos PCI, el grep VGA, muestra solo las líneas que tengan la palabra "VGA"

Resultado de ejemplo:

08:00.0 VGA compatible controller: Advanced Micro Devices, Inc.
[AMD/ATI] Navi 24 [Radeon RX 6400/6500 XT/6500M] (rev c7)

Acá el nombre de la gráfica seria: Navi 24 [Radeon RX 6400/6500 XT/6500M] (rev c7)

Comando lspci para mas información de la gráfica: `lspci -vnn | grep VGA -A 12`

Capítulo 7 | Obtener información del hardware

Versión OpenGL, herramienta glxinfo. Esta programa es posible que no lo tengas instalado, pero es solo de instalarlo. Este te ayuda a ver que versión de opengl tiene la tarjeta gráfica. Muestra todo lo relacionado con la implementación GLX.

Comando para ver información versión de opengl: `glxinfo | grep OpenGL`

Comando para ver información de tarjeta gráfica: `glxinfo -B`

Versión Vulkan, herramienta vulkaninfo. Para poder usar esta herramienta necesitas instalar "vulkan-tools", el cual contiene vulkaninfo, entre otras herramientas:

`sudo apt install vulkan-tools`

Si no esta, busca un paquete equivalente: `apt search vulkaninfo`

Comando ver versión de vulkan: `vulkaninfo | grep -i version`

Te muestra un montón de info del vulkan, pero con el "grep version", solo te muestra las líneas que contengan la palabra "versión"

Aplicaciones para ver mucha información de la tarjeta gráfica GPU:

glz:

<https://www.geeks3d.com/glz/>

GPU Viewer:

<https://github.com/arunsivaramanneo/GPU-Viewer>

<https://flathub.org/apps/io.github.arunsivaramanneo.GPUVIEWER>

Resumen: Usa `lspci`, `glxinfo`, `vulkaninfo`, `glz`, `gpu viewer`, para obtener información de la gráfica. `lspci`; esta preinstalado en el OS. `glxinfo`, `vulkaninfo`, `gl-z`, y `gpu viewer`; necesitan ser instalados para ser utilizados.

7.5 Sensores

Herramienta **sensors**, esta te sirve para obtener la información de los sensores de los componentes de la computadora, su temperatura vato.

Comando para obtener temperatura de componentes: `sensors`

Capítulo 7 | Obtener información del hardware

lm-sensors, este programa te permite detectar los sensores del pc.

Necesita ser instalado, y se instala con: `sudo apt install lm-sensors`

Comando para obtener info de sensors: `sudo sensors-detect --auto`

psensor. Este programa, es una **interfaz gráfica para el programa lm-sensors**. Te permite ver información de los sensores de los componentes del pc. Se tiene que instalar con:

`sudo apt install psensor`

Comando para abrir psensor: `psensor`

También lo puedes buscar en el menú de aplicaciones.

Resumen: Ver la información de los sensores te permite ver la temperatura de los componentes, la utilidad "sensors" viene preinstalada. Para mas información de sensores se pueden instalar la aplicación lm-sensors. psensor, te permite usar lm-sensors de manera gráfica y sencilla (puros botones).

7.6 Información general

Veremos aplicaciones para obtener información general de los componentes de la computadora.

neofetch; Programa de terminal.

Permite ver la información de la distribución utilizada, y información de los componentes principales del pc. Comando: `neofetch`

Se tiene que instalar, y se puede instalar con: `sudo apt install neofetch`

Sitio web de neofetch: <https://github.com/dylanaraps/neofetch>

Capítulo 7 | Obtener información del hardware

Ejemplo de resultado del comando neofetch:

```
met$$$$$gg .                jean_abraham@cm-debian
,g$$$$$$$$$$$$$$$$$P .      -----
 ,g$$P"      ""Y$$. .      OS: Debian GNU/Linux 12 (bookworm) x86_64
,$$P'          '$$ .       Host: A520M K V2 -CF
`,$$P      ,ggs .      `$$b: Kernel: 6.1.0-28-amd64
`d$$'      ,P" ' .      $$: Uptime: 20 mins
$$P      d$'      ,     $$P Packages: 3549 (dpkg), 18 (flatpak)
$$:      $$.-      ,d$$' Shell: bash 5.2.15
$$;      Y$b.      ,d$P' Resolution: 1920x1080
Y$$.      .`"Y$$P" ' DE: Xfce 4.18
`$$b      "-. __ WM: Xfwm4
`Y$$
`Y$$.      WM Theme: Default
`$b.      Icons: Adwaita [GTK2], Breeze [GTK3]
`Y$b.      Terminal: xfce4-terminal
`"Y$b.      Terminal Font: Monospace 11
`"__"      CPU: AMD Ryzen 5 3500X (6) @ 3.600GHz
              GPU: AMD ATI Radeon RX 6400/6500 XT/6500M
              Memory: 3031MiB / 15908MiB
```

hardinfo; Aplicación con interfaz gráfica.

Permite ver la información general de todos los componentes del pc. Es muy intuitiva de usar, y se ve bien (puros botones, bien bonito).

Se tiene que instalar, y se puede instalar con:

```
sudo apt install hardinfo
```

Sitio web de hardinfo: <https://github.com/lpereira/hardinfo>

Resumen: Puedes usar hardinfo para obtener información general de todos los componentes del pc, incluyendo tanto hardware como software. Puedes usar neofetch para obtener información sobre los principales componentes del pc, incluyendo tanto hardware como software.

7.7 Información en tiempo real

¿Que es obtener información en "tiempo real"?: Que la información que se ve esta en constante actualización, cambiando la información si es necesario. Cada x milisegundos la aplicación/programa obtiene información y la muestra. Todo esto de forma automática, no es necesario que se involucre el usuario.

Capítulo 7 | Obtener información del hardware

cpu-x: Anteriormente mencionado, te da información en tiempo real sobre el funcionamiento del procesador, ram y tarjeta grafica. Pero principalmente se enfoca en mostrar información del procesador.

baobab: Anteriormente mencionado, te da información relacionada con el almacenamiento.

htop: Muestra información relacionada con los procesos del sistema operativo, el uso de ram, el uso de cpu, el nivel de prioridad de los procesos. El uso de hardware cada proceso, incluso permite gestionar los procesos, por ejemplo; detener procesos.

Aplicación Gestor de Tareas / Administrador de tareas

Esta aplicación te permite gestionar todos los procesos del sistema operativo, el uso de ram, el uso de cpu, el nivel de prioridad de los procesos. El uso de hardware cada proceso, incluso permite gestionar los procesos, por ejemplo; detener procesos.

Lo mas probable es que tu sistema operativo GNU-Linux tenga ya un Gestor de Tareas, y pos se llama así "Gestor de tareas", o un nombre parecido. Búscalas.

Y por si no tienes una, puedes usar:

Stacer: Optimizador y monitorización del sistema operativo. Es un gestor de tareas bonito y funcional.

Lo puedes instalar con apt: `sudo apt install stacer`

Recuerda darle un update al apt. Y si no te jala el comando, busca si existe la app en los repos apt.

Sitio web: <https://oguzhaninan.github.io/Stacer-Web/>

Resumen: Para ver los recursos usados del OS, puedes usar un gestor de tareas, este te permite esto y gestionar los procesos del sistema operativo. htop y stacer son ejemplos de gestores de tareas.

7.8 Estresar computadora

En resumidas cuentas; Esto se hace para probar el power del pc, su estabilidad, su durabilidad, y su calentura.

Uso normal del pc: El uso normal de la computadora, se refiere a usar programas/videojuegos. Por ejemplo; Cuando ves videos, tus actividades diarias en el pc, usar aplicaciones ofimáticas, jugar al supertuxkart, etc.

Estresar el pc: Cuando fuerzas el uso de uno o mas componentes al 100%. Tener al componente al límite; esto se logra utilizando programas que tienen como objetivo "estresar el pc". También puede pasar que por "x" o "y" motivo un programa utilice demasiados recursos y suceda el "estrés en el pc".

Capítulo 7 | Obtener información del hardware

¿Para que sirve estresar la computadora?: Sirve para analizar el funcionamiento de los componentes cuando están al límite. Su rendimiento mas bruto. Obtener estadísticas de su rendimiento. Esto nos permite determinar si un componente esta funcionando bien, o esta funcionando mal.

Por ejemplo; Si fuerzas al 100% el procesador durante cinco minutos, y pasado ese tiempo el pc se apaga, es probable que el procesador no este funcionando bien.

Nota: Una aplicación que sirve para estresar el PC, no necesita de permisos sudo. El estrés del pc no necesita de permisos sudo. Evita usar sudo cuando estresas el PC, esto es para evitar inconvenientes.

Peligros de estresar la computadora

- **Desgatado de componentes.** Si el componente ya esta defectuoso, esto lo puede dañar aun mas. Reducción de vida del componente. Riesgo de sobrecarga si tenemos componentes defectuosos.

Ya de una te digo que evites estresar la pc todos los días, esto solo se hace para probar su funcionamiento, no para hacerlo a cada rato, le reduce la vida a los componentes.

Principalmente evita estresar la RAM y el almacenamiento, estos son especialmente debiluchos.

- **Mayor gasto energético.**

Estos peligros/riesgos no deberían pasarte, ya que los componentes del pc están hechos para ser fuertes. El estrés del pc solo hazlo para realizar análisis de rendimiento, not for fun.

Aplicación GtkStressTesting

Utilidad del sistema diseñada para estresar y monitorizar varios componentes de hardware. Tiene interfaz Gtk, y es muy fácil de usar. Todo es muy visual. Esta es especialmente buena para estresar el CPU.

Sitio Web: <https://github.com/kaisenlinux/GtkStressTest>

Aplicación FurMark

Para estresar gráfica. Ofrece una prueba de estrés de GPU muy intensiva. Es para probar la tarjeta gráfica. Esta es una aplicación con interfaz grafica, facil de usar.

Sitio Web: <https://geeks3d.com/furmark/>

Aplicación stress-ng

Acá veremos como estresar RAM. Cabe aclarar que stress-ng sirve para estresar absolutamente cualquier cosa, pero es mas complicada de usar.

Es por puros comandos, pero te permite estresar cualquier cosa, pero como ya cubrimos como estresar el CPU, y como estresar la gráfica, te mencionare como estresar la RAM con stress ng.

Ejemplo; Como estresar RAM al 90% con stress ng.

(al 90% para evitar congelamientos de pantalla, que es normal cuando se usa el 100% de ram)

```
stress-ng --vm 1 --vm-bytes 90% --timeout 300s --verbose
```

Bueno, ahora que ya sabes que es estresar el PC, te menciono que no es necesario ni recomendable estresar el PC durante horas/días, con unos 30 minutillos es suficiente.

Resumen: Lo mejor para estresar cada componente, es utilizar una app que se especialice en estresar dicho componente. GtkStressTesting, para estresar CPU, FurMark para estresar GPU, y stress-ng para la RAM. Si nomás te quieres quedar con un todo en uno, tienes que elegir stress-ng; este es el mas versátil.

Pero te menciono de nuevo que; lo recomendable es tener una app de stress especializada en estresar un componente, un app por cada componente que quieras estresar.

8 Servidores gráficos

Servidor gráfico: Un servidor gráfico o servidor de ventanas es un programa cuya tarea principal es coordinar la entrada y la salida de sus clientes hacia y desde el resto del sistema operativo, el hardware, y otros. El servidor gráfico se comunica con sus clientes con el protocolo de servidor gráfico. Este se comunica con la GPU, el servidor gráfico y la GPU son compas. Trabajan juntos.

En GNU Linux, se usan principalmente dos servidores gráficos:

X11 y **Wayland**. Donde **X11** es el legacy, y **Wayland** es el new. Ambos son potentes, **Wayland** se dice ser mas potente, pero no le creas mucho.

8.1 X11

Definición de X11 oficial: El proyecto X.Org ofrece una implementación de código abierto del sistema X Window. El desarrollo se lleva a cabo en colaboración con la comunidad [freedesktop.org](https://www.freedesktop.org). La Fundación X.Org es la organización educativa sin fines de lucro cuya Junta Directiva contribuye a esta iniciativa y cuyos miembros lideran esta labor. **Se creo en 1987.**



Sitio web: <https://www.x.org/wiki/>

Algunos de los escritorios que usan X11 son; xfce4 y lxde.

xrandr: Se utiliza para configurar el tamaño, la orientación y/o el reflejo de las salidas de una pantalla. También puede configurar el tamaño de la pantalla.

Donde esta el X11

- Esta en la capeta: `/etc/X11/`
- Donde configurar cualquier cosa relacionada a X11, en esta carpeta: `/etc/X11/xorg.conf.d/`
- Donde obtener log de X11: `/var/log/Xorg.0.log`

Como saber que tipo de gráfica tienes, según X11: `grep drivers /var/log/Xorg.0.log`

Obtener información de la configuración actual: `xrandr --verbose`

Capítulo 8 | Servidores gráficos

Problemas con el screen tearing

Este es un problema visual, hace que la pantalla se corte, si tienes X11 es posible que lo tengas. Para verificar si lo tienes puedes buscar vídeos de "screen tearing test".

Si este es un problema que no te importa, puedes dejarlo así, no es un error y no alenta el pc.

Ver información relacionada con el triple buffer:

```
glxinfo | grep "GL_EXT_framebuffer_object"
```

Este problema del screen tearing en X11 se soluciona activando el triple buffer. Ahora los pasos para activar el triple buffer.

Paso 1: Para saber si tenemos activado o desactivado el triple buffer:

```
xrandr --verbose | grep TearFree
```

Paso 2: Tenemos que saber que tipo de gráfica tenemos, sea AMD, Intel o Nvidea:

```
grep drivers /var/log/Xorg.0.log
```

Resultado de ejemplo:

```
[      10.892] (II) Loading /usr/lib/xorg/modules/drivers/amdgpu_drv.so
```

A mi me salio amdgpu, entonces eso es lo que tengo, una amdgpu.

Dependiendo de la GPU que te salga; Podemos crear un `archivito.conf` en `"/etc/X11/xorg.conf.d/"` para configurar el triple buffer.

Pero no vamos a batallar y mejor solo pondremos un comando.

Comando para activar triple buffer en todas las pantallas que se encuentren conectadas:

```
xrandr | grep ' connected' | cut -f 1 -d ' ' | while read display; do xrandr --output $display --set TearFree on; done
```

Comando para desactivar el triple buffer:

```
xrandr | grep ' connected' | cut -f 1 -d ' ' | while read display; do xrandr --output $display --set TearFree off; done
```

Si nomas quieras habilitarlo/desactivarlo en una sola pantalla:

```
xrandr --output DisplayPort-0 --set TearFree on
```

DisplayPort-0, es la pantalla default.

Desactivar:

```
xrandr --output DisplayPort-0 --set TearFree off
```

Capítulo 8 | Servidores gráficos

Script para activar triple buffer

Si quieres que esto se establezca de forma permanente, puedes hacer un script, y mételo en el inicio como servicio, con `systemctl`.

Primero; Crear y editar script.sh:

```
sudo nano /usr/local/bin/xrandr_tearfree-on.sh
```

El texto:

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

```
# Para metele el triple buffer a todas las pantallitas.  
xrandr |grep ' connected'|cut -f 1 -d ' '|while read display;do  
xrandr --output $display --set TearFree on;done
```

Segundo; Permisos de ejecución:

```
sudo +x /etc/init.d/xrandr_tearfree-on.sh
```

Tercero; Crear servicio para iniciar el script:

```
sudo nano /etc/systemd/system/xrandr_tearfree-on.service
```

El texto:

```
[Unit]  
Description=Activar el tear free  
  
[Service]  
Type=oneshot  
ExecStart=/etc/init.d/xrandr_tearfree-on.sh  
RemainAfterExit=true  
  
[Install]  
WantedBy=graphical.target
```

Cuarto; Probamos que jale:

```
sudo systemctl start xrandr_tearfree-on.service
```

Quinto; Añadirlo al inicio:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable xrandr_tearfree-on.service
```

Capítulo 8 | Servidores gráficos

Fin; Probamos que todo este bien (tiene que estar en en enabled):

```
sudo systemctl status xrandr_tearfree-on.service
```

Desactivar Vsync en X11

Variables de entorno, para desactivar o activar; **vsync**. Esta configuración del **vsync** reduce tus fps a los hz de tu pantalla, intenta mantener una estabilidad en los frames, reduce errores visuales. Suena bonito, pero esta función reduce rendimiento, yo recomiendo desactivarla.

Esta config de normal no esta habilitada, pero quien sabe puede que tu si la tengas activada. Depende de lo que haga tu OS, aveces se ponen locos y te habilitan cualquier cosa por defecto.

Para asegurarte de desactivar el vsync puedes hacer esto

Editar/Crear el archivo: `sudo nano /etc/environment`

Le agregamos este texto:

```
vblank_mode=0  
# Esta config deja todo default, el vsync se activara solo si un  
programa lo activa, de lo contrario no se activara. Como debe ser.
```

AMD, activar o desactivar free-sync:

El freesync solo jala en pantallas compatibles, la mayoría de pantallas modernas lo soportan.

Esta función esta chida, pero puede darte input lag, y lag a secas también. Pero si quieras activar o desactivarla acá te digo como.

Para saber si tu pc soporta free-sync: `xrandr --props | grep vrr_capable`

Si esta en 1, es compatible y si esta en 0, no es compatible.

Para ver si esta habilitado o deshabilitado el FreeSync:

```
grep VariableRefresh /var/log/Xorg.0.log
```

En AMD Creamos/Editamos el archivito:

```
sudo nano /etc/X11/xorg.conf.d/20-amdgpu.conf
```

Y ponemos:

```
Section "Device"  
    Identifier "AMD"  
    Driver "amdgpu"  
    Option "VariableRefresh" "true"  
EndSection  
  
# (Pon true si quieres activar, pon false si quieres desactivar)
```

Capítulo 8 | Servidores gráficos

Sensibilidad de Mouse:

Para quitar la aceleración en GNU Linux, Debian 12, X11, Xfce4, LightDM: Crear el archivo de texto "50-mouse-acceleration.conf", en la ruta "/etc/X11/xorg.conf.d/". Se requerirán permisos de sudo/administrador/root.

Utilizaremos nano para hacerlo (se puede usar cualquier editor de texto que permita sudo):

```
sudo nano /etc/X11/xorg.conf.d/50-mouse-acceleration.conf
```

El archivo contendrá lo siguiente:

```
Section "InputClass"
    Identifier "Mouse Settings"
    MatchIsPointer "yes"
    Option "AccelerationProfile" "-1"      # Desactiva la aceleración
    Option "AccelerationScheme" "none"       # Utiliza un perfil lineal
    Option "AccelSpeed" "0"                  # AccelSpeed Sensibilidad
# Sensibilidad (puedes probar valores entre -1 y 1, Por defecto 0)
EndSection
```

Lo guardamos y probamos reiniciando el entorno gráfico o el pc.

Reiniciar el entorno gráfico lightdm: `sudo systemctl restart lightdm`

O Reiniciar el PC: `systemctl reboot`

Puedes verificar que si jala este archivo de configuración, si te devuelve algo este comando:

```
cat /var/log/Xorg.0.log | grep "Mouse Settings"
```

Resumen: X11 es legacy, pero aun se usa mucho, sirve para mostrar todo. En X11 puedes tener problemas con configuraciones como; triple buffer, tearfree y vsync. La distro que instalamos, configurara el X11, pero nosotros también podemos configurarlo.

8.2 Wayland

Wayland es un reemplazo para el protocolo y la arquitectura del sistema de ventanas X11 con el objetivo de que sea más fácil de desarrollar, ampliar y mantener. **Se creo en 2008.**



Este funciona diferente a X11, pero tiene intenciones de ser lo mas compatible posible con X11.

Wayland no solo es un servidor gráfico, sino que también es una arquitectura de sistema. Un estándar para hacer un servidor gráfico. Para que el desarrollo sea mas eficiente y ordenado.

Sitio web de Wayland: <https://wayland.freedesktop.org/>

Wayland funciona especialmente bien con las tecnologías mas nuevas. Este servidor gráfico tiene gran soporte con las gráficas sacadas de los años; 2016-2025.

Problemas con tarjetas gráficas antiguas

Con gráficas del 2013 para abajo, wayland no es tan bueno usándolas, muchas veces funcionan mal.

Cambiar sensibilidad de mouse:

Se tiene que utilizar "libevdev-tweak-device" del paquete "libevdev-tools".

```
sudo libevdev-tweak-device --abs ABS_X --res 99 /dev/input/event99 &&
sudo libevdev-tweak-device --abs ABS_Y --res 99 /dev/input/event99
```

Donde debes reemplazar 99 por la resolución deseada y event99 por el dispositivo de entrada.

Resumen: Si tienes una gráfica del 2013 para arriba, lo mas probable es que funcione bien en Wayland. Si tienes una gráfica viejilla del 2013 para atrás, lo mas probable es que no jale tan bien.

Pero Wayland constantemente hace cambios, así que puede que el soporte con gráficas antiguas mejore.

9 Wine

¿Que es Wine?

Es un programa; **Wine** te permite correr software de Windows en otros sistemas operativos. Con **Wine**, puede instalar y correr estas aplicaciones igual que lo haría en Windows.



Código Abierto y Guiado por el Usuario

Wine siempre será software libre. Aproximadamente la mitad del código fuente de **Wine** es escrito por voluntarios. El resto es patrocinado por intereses comerciales, especialmente de **CodeWeavers** quienes venden una versión de **Wine** con soporte.

Si tienes mucho dinero, y estas dispuesto a usar **Wine** al máximo power, puedes pagar el soporte **CodeWeavers**, de lo contrario te recomiendo aprender a usar **Wine**.

Resumen: *Wine te sirve para abrir ejecutables.exe. Esta en constante mejora y este avanza gracias a los esfuerzos de los desarrolladores y los usuarios.*

9.1 Instalar Wine

Básicamente, tienes que entrar en la pagina web de wine: <https://www.winehq.org/>

Darle a descarga, y seleccionar la distribución que tengas instalada. Para ver que os tenemos instalado usamos el comando: `cat /etc/os-release`

Supongamos que tienes Linux Mint base Ubuntu 24, bueno entonces seleccionas eso "Ubuntu". Se verán instrucciones de instalación.

Te recomiendo ver detenidamente las instrucciones, son puro copy paste de comandos, pero tienes que seleccionar los comandos correctos.

Acá un descripción de ejemplo de lo que harás:

- Agregamos compatibilidad con repos de 32 bits (Esto siempre si o si siempre se hace):
`sudo dpkg --add-architecture i386`
- Descargas y agregas la clave del repositorio. Recuerda asegurarte de agregar el repositorio que sea para la distribución que tienes instalada, que coincida nombre y versión numérica de distro.
- Un apt update y después instalamos la rama Stable. La otras ramas están todas locas, la Stable es como dice; Estable.

No te muestro exactamente como se instala, porque Wine esta en constante cambio y su método de instalación puede variar un poco.

Capítulo 9 | Wine

Comprobar su funcionamiento de Wine:

Una vez instalado, puedes comprobar si jala con el siguiente comando:

```
wine --version
```

Se tendría que ver la versión del wine, algo así: "wine-10.0"

Iniciar wine para el uso.

Una vez comprobando que jala, podemos inicializar wine, para que ya esté ready:

```
winecfg
```

Si nos sale una ventanita, le damos instalar, instalar y así hasta que termine. Cuando esto suceda, ya estará listo wine.

Wine (sin abrirlo como root) esta instalado aquí; \$HOME/.wine/

Te recomiendo nunca abrir wine como root. Un programa no debería necesitar esto.

Dale un ls: `ls $HOME/.wine/*` Con eso vemos como está estructurado el wine.

Resumen: Para instalar wine, sigue las instrucciones de instalación de la página web oficial. Son puros comandos, pero puro copy and paste, re easy.

9.2 Como usar wine

Obtener información de wine: `man wine | cat`

Configuración general de wine: `winecfg` También sirve para "inicializar" wine.

Por si quieres establecer la ruta donde se creará el wine:

```
WINEPREFIX=/ruta/carpeta winecfg
```

Abrir consola wine, como la de windows: `wineconsole`

El equivalente al Panel de control en wine: `wine control`

Desinstalar aplicaciones: `wine uninstaller`

Como ejecutar una aplicación: `wine start "/ruta/programa.exe"`

También puedes darle dos clicks al programa y se abrirá.

Ejecutar con librerías nativas wine (las más estables, pero también las más lentillas)

Comando de ejemplo:

```
Exec=env WINEDLLOVERRIDES="d3d11,d3d10,d3d9,dxgi=b" wine app.exe
```

Capítulo 9 | Wine

Depurar aplicación; Hay que usar winedbg, ve su información: `man winedbg | cat`

Así se depura un programa: `winedbg "/ruta/programa.exe"`

Dentro de winedbg

Matamos el proceso del programa: `kill`

Volvemos a ejecutar el programa: `run`

Salimos del winedbg: `quit`

Así es como se usa wine, realmente es sencillo de usar. Tiene mas cosas, pero con estas ya manejas chido el wine.

9.3 Winetricks

Este es un programa, que recopila un montón de librerías compatibles para Wine, para que funciones los programas que necesiten de librerías que no están en Wine. **winetricks** te puede servir para abrir programas que no jalan en wine default.



Instalar winetricks

winetricks esta en los repos de debian.

Primero actualizamos la lista de paquetes, ya tu sabe: `sudo apt update`

Lo podemos instalar con: `sudo apt install winetricks`

Si no instala nada puede que sea porque le cambiaron el nombre, puedes buscarlo:

`apt search winetricks`

Si quieras la ultima versión de winetricks, tendrás que seguir las instrucciones de su github. La ultima versión esta fuera de apt. Pero créeme, lo mas seguro es que no necesites de la ultima versión, la versión de apt debería ser suficiente.

Como usar winetricks

Bueno primero que nada, recuerda que puedes ver su información con:

`winetricks -h`

O para aun mas información: `man winetricks`

Capítulo 9 | Wine

Winetricks tiene muchas cosas que puedes descargar y configurar, las puedes ver con:

winetricks k list *Lita de opciones*

Listar dlls disponibles a descargar: **winetricks dlls list**

Listar lo instalado: **winetricks list-installed**

Listar otras cosas:

- **wintetricks apps list**
- **winetricks benchmarks list**
- **winetricks fonts list**
- **winetricks games list**
- **winetricks settings list**
- **winetricks settings list**
- **winetricks list-cached**
- **winetricks list-download**
- **winetricks list-manual-download**

Listar todo porque estamos locos: **winetricks list-all**

Comando para setup visual: **winetricks**

Usar winetricks sin argumentos adicionales permite a los usuarios acceder a una interfaz gráfica que simplifica la gestión de entornos Wine.

Comando establecer todo lo indicado en winetricks en una ruta específica:

WINEPREFIX=/ruta/programa winetricks

Cambiando esta variable de entorno, todo lo que haga winetricks sera hecho en esa ruta.

Por defecto wineprefix mete todo de manera global a wine. Muy recomendado especificar el programa que necesita de winetricks y no hacerlo de manera global.

Capítulo 9 | Wine

Instalar una dll: `winetricks package-dll`

Esto agrega una dll al wine de manera global.

Acuérdate que puedes buscar dlls, con "dlls list"

Ejemplo de uso:

`mkdir $HOME/wine-test` Crear carpeta de prueba

`winetricks dlls list | grep -i dx9` Vemos los archivitos

`WINEPREFIX=$HOME/wine-test winetricks d3dx9`

Establecemos ruta en donde se bajaran las cosillas. Como es una prueba, lo meto en la home. Descargamos el archivito. Puedes borrar \$HOME/wine-test si quieres, es solo una prueba.

Resumen: *winetricks es un complemento para wine, entre sus principales funciones; te permite establecer configuraciones y dlls de manera sencilla.*

9.4 DXVK

Implementación basada en Vulkan de D3D8, 9, 10 y 11 para Linux/Wine

Sitio web: <https://github.com/doitsujin/dxvk>

Es un conjunto de dlls que transforman instrucciones de direct a instrucciones de vulkan. Vulkan en Linux y en otros OS, a demostrado ser poderoso y eficiente. Muchas veces da mas fps.

DXVK da un rendimiento increíble en la mayoría de videojuegos. Esta en constante mejora y es bastante estable.

Nota súper importante, asegúrate de que tu gráfica tenga vulkan vato. Y el mesa de preferencia el mesa mas nuevo.

Comando para ver nombre de tu gráfiquiña: `lspci | grep VGA`

Con ese nombre busa info en internet.

Ver versión de vulkan y otras cosillas, con vulkaninfo: `vulkaninfo | grep -i version`

Capítulo 9 | Wine

Instalación de DXVK en el Wine

(Los métodos de instalación que se verán, es para pc de 64 bits)

Instalación método github: Te voy a decir instalarlo, pero te recomiendo ver las instrucciones de instalación del sitio web.

Descomprimir el archivo descargado (El ultimo Lanzamiento/Release):

- Busca en lo descomprimido las carpetas /x64 y /x32. Estas contienen "archivos.dll"
- Lo contenido en la carpeta x64, lo mandamos a la ruta de "wine/drive_c/windows/system32"
- Lo contenido en la carpeta x32, lo mandamos a la ruta de "wine/drive_c/windows/syswow64"

Instalación de archivos DXVK, con comandos (de seguro te jalará):

```
export WINEPREFIX=$HOME/.wine
```

Wineprefix debe ser la ruta del wine, por lo regular es la home del user. Seguido de la carpeta ".wine"

```
cp x64/*.dll $WINEPREFIX/drive_c/windows/system32
```

```
cp x32/*.dll $WINEPREFIX/drive_c/windows/syswow64
```

Copiamos los archivos de 64 bits y los de 32 bits, como corresponde

winecfg Inicializamos winecfg y añadimos las librerías. Por ejemplo: d3d9, d3d10, etc...

Te puedes quedar con esto:

- Las dll x64, lo mandamos a la ruta "drive_c/windows/system32"
- Lo dll x32, lo mandamos a la ruta "drive_c/windows/syswow64"

Instalación metodo winetricks:

Comando: **winetricks dxvk**

Nota: Si, así de fácil vato. La verdad te recomiendo seguir el tuto de github, pero si te da hueva pos el winetricks te pueda jala clean.

Capítulo 9 | Wine

Final, Configurar wine para usar dxvk. Para que te jale dxvk tienes que abrir la app grafica "winecfg"

Comando: **winecfg**

Aca, vamos a la pestaña Librerias/Libs, y agregamos los dll contenidos en dxvk: d3d10_core, d3d11, d3d8, d3d9, dxgi

Si no te jala dxvk, entra a winecfg y elimina lo agregado.

Ejecutar con librerías nativas wine (sin dlls agregadas por voce)

Esto serviría para abrir el programa sin DXVK. O abrirlo con cualquier dll agregada por el usuario. (Aprueba de errores)

Comando de ejemplo:

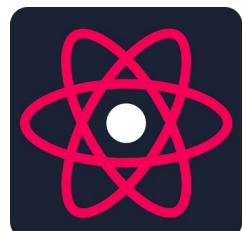
```
Exec=env WINEDLLOVERRIDES="d3d11,d3d10,d3d9,dxgi=b" wine app.exe
```

Resumen: DXVK sirve para dar mas fps en los juegos. Se tiene que instalar en wine, no viene con este.

9.5 Proton, ProtonDB y Protontricks

¿Que es Proton?

Proton es una nueva herramienta lanzada por Valve Software que se ha integrado con **Steam** Play para que jugar juegos de Windows en Linux sea tan sencillo como pulsar el botón Jugar en **Steam**. En su interior, **Proton** incluye otras herramientas populares como **Wine** y **DXVK**, entre otras, que de otra manera; un jugador tendría que instalar y gestionar estas herramientas por sí mismo.



Source Code: <https://github.com/ValveSoftware/Proton>

¿Como usar Proton?

La verdad la manera mas sencilla de usar proton es usando steam, esta sería la manera automática. La manera manual es mas complicadete. Los pasos para usar proton serian; **seleccionar el juego, entrar a propiedades y forzar compatibilidad con proton**. En steam puedes incluir juegos locales fuera de steam, no solo los juegos de steam.

ProtonDB

Es la base de datos de Proton, y también existe una pagina que muestra la base de datos de Proton, dicha base de datos indica el nivel de compatibilidad de los juegos en Linux, ChromeOS, y MacOS. Allí dice si "x" juego de Windows, jala bien o mal, o si siquiera jala.

Sitio web: <https://www.protondb.com/>

Capítulo 9 | Wine

Protontricks

Un contenedor que hace cosas de winetricks para juegos compatibles con Proton requiere de Winetricks.

GitHub: <https://github.com/Matoking/protontricks>

Resumen: *Proton es creado por Steam, para usarse en Steam, sirve para ejecutar videojuegos de windows en Linux, de la manera mas eficiente posible. Su código fuente esta disponible, y los usuarios avanzados lo agradecen. ProtonDB es un sitio web que recopila los videojuegos compatibles con Proton, y su nivel de compatibilidad.*

10 Final

Hemos llegado al final de este curso, felicidades. Ahora usa mucho tu distribución favorita.

Todos los conocimientos vistos son para usar distribuciones GNU/Linux a nivel intermedio. Con todo esto ya puedes hacer muchas cosas.

Si te interesa subir de nivel, y quieres volverte avanzado, necesitas masterizar todo lo aprendido y aprender aun mas cosas, te mencionare algunas:

- **Personaliza la interfaz gráfica a tu gusto.** Esto te permitirá moverte por el OS lo mas rápido y agusto posible.
- **Estudiar la función de GNU.** Comprender la importancia de GNU, y como este usa Linux.
- **Manejo de ip's.** El pc tiene ip, los dispositivos tienen ip, el wifi tiene ip. Todo tiene ip.
- **Profundizar en conceptos complejos.** Conceptos complejos como: Init System, Kernel, Drivers, API's, Bios, Uefi, Servidor Gráfico, entre otros (Busca mas, de hecho una vez estés aprendiendo sobre "conceptos complejos", lo mas probable es que se hablen de aun mas conceptos. Estúdialos). Esto te permitirá entender como funciona internamente el sistema operativo, también te ayudara a entender mas sobre el hardware.
- **Mecanografía:** Pura practica, es fácil. Esto te permitirá escribir súper rápido en el PC. Sin ver, y con incluso menos errores gramaticales y de ortografía. La practica hace al maestro, seguro lo dominaras chaval.

Esta pagina web te enseña de todo, sirve muy bien para ver la posición de los dedos y manos: <https://agilefingers.com/es>

- **Programación bash script.** Este es fácil de aprender, y es muy útil. Esto te permitirá automatizar muchas tareas.
- **Instalación de drivers.** Esto te permitirá obtener el mejor rendimiento del hardware en el PC.
- **Configuración del servidor gráfico.** Esto te permitirá obtener la mejor configuración para tu disfrute visual. Ya sea que se vea bonito, o que se vea muy rápido.

A grandes rasgos, estamos hablando de como configurar Wayland o x11. Aprende el que use tu OS. O capaz no usa x11 o Wayland, tal vez usa otro, por eso es bueno estudiar los servidores gráficos.

- **Uso de APT; Agregar repos APT y crear repos APT.** APT, es súper potente, acá solo vimos su uso básico, es bueno saber como usarlo al full power. Aprenderás a usar repos APT de terceros, y a hacer los tuyos propios.

Capítulo 10 | Final

- **Gestión de archivos.** Gestionar los archivos como todo un profesional, con las utilerías bash por defecto, te hará ahorrar muchísimo tiempo.
- **Gestión de archivos comprimidos.** Muchas cosas en internet, por no decir todo en internet; están comprimidos. Es bueno saber como descomprimir y comprimir. Recomiendo aprender como usar 7zip. Probablemente el mas potente, incluso mas que Winrar. Opcionalmente puedes investigar "7zip vs Winrar".
- **Configuración de GRUB2.** Perro ese GRUB es súper útil, entender como configurarlo te dará mucho power.
- **Utilerías SystemD.** SystemD, el Init System mas usado. Tiene una cantidad de utilerías absurdas, tan alta que incluso la gente linuxera se queja, dicen SystemD no sigue la filosofía Unix. Pero bueno, este ha demostrado ser eficiente, y puede que no necesariamente el mas rápido, pero lo suficiente rápido como para ser usado por casi todos los distros.

Sus utilerías te permitirán gestionar todo lo relacionado con los servicios. Y estos están muy ligados con el OS, con el Boot, con todo vato.

- **Programación python, java, c++, c**

Con conocimiento de programación, entenderás aun mas el funcionamiento de los programas. Python en particular se usa mucho en GNU/Linux, de hecho de seguro viene instalado por defecto en tu distribución GNU/Linux.

- Python; Desarrollar programas de manera muy rápida.
- Java; Desarrollar programas con alta compatibilidad.
- C++, C; Hacer programas complejos con alto rendimiento, programación mas lenta pero eficiente.

Básicamente te diría que aprendieras en este orden: Python, Java, C++, y C.

Nota; Podrías remplazar el Java por el C#, y el C por el Rust. Con un C like y un python ta bien.

Sería de mas fácil a mas difícil. No es necesario que aprendas todos, pero sin duda alguna te darán mucho conocimiento.

- **Uso de Git:** Control de versiones con git, este te permitirá comprender como los programas gestionan sus versiones. Importante para mantener un programa bien ordenadito.
- **Compilación de paquetes/programas:** Esto se ve en la programación. Pero se puede hacer sin saber de esta, tendrías que conocer que son las dependencias de un programa, y después establecer las instrucciones de compilación correctas.
- **Aprender inglés.** Esto te permitirá leer la información de los programas de manera mas rápida. No hace falta que seas un profesional del inglés, con que sepas leer un poco de inglés ta bien. Por eso mismo te recomiendo usar cursos gratis de inglés. Duolingo, es divertido y se puede usar en Web o en el Celular. Diría que un "nivel MCER B1 en inglés", es mas que suficiente.

Conclusión

Si eres un usuario no programador y te gusta GNU/Linux, siendo programador te va a gustar mas.

La verdad en GNU/Linux nunca dejas de aprender, hay demasiada información, y encima GNU/Linux a cada rato crece mucho mas. No te abrumes por no saber todo, es demasiada información, y siempre estarás aprendiendo algo nuevo, hasta sin querer queriendo.

Y no te voy a mentir, lo que te puse para aprender a ser usuario avanzado, te lleva años, de 1 a 6 años. Un año si te la pasas estudiando todo el día, y seis años si estudias todos los días un poquillo.

Y luego encima, podríamos decir que hay niveles de "usuarios avanzados", es decir; hay usuarios súper avanzados, que hacen parecer a los avanzados novatos. Pero bueno; igual esto no es una competencia, si tu estas agusto con lo que sabes no hay problema, como digo; muchas veces aprendemos cosas nuevas hasta sin querer queriendo.

Glosario

Archivos

- **El Estándar de Jerarquía de Directorios o FHS:** La estructura de los directorios de Linux, así como su contenido y funciones, viene definida en el denominado Filesystem Hierarchy Standard o FHS por sus siglas en inglés, que en otras palabras viene a ser el estándar de jerarquía para los sistemas de archivos en sistemas Linux y otros derivados de UNIX.
- **Carpetas:** Una carpeta es una ubicación virtual dentro de la estructura de archivos de computadora. Sirve para tener todo mas ordenado, allí puedes meter archivos.
- **Archivos:** Es una sola cosa; un texto, un vídeo, o básicamente cualquier cosa que no sea una carpeta. Un archivo contiene información para algún programa.
- **Gestor de archivos:** Es una aplicación que te permite moverte por los archivos y carpetas del OS. Gestionar el sistema de archivos.
- **Sistema de archivos:** Un sistema de archivos se encarga de estructurar y manejar todos los datos. El acciona los "mueve", "copia", "borra", entre otras funciones. Un sistema de archivos es necesario para poder utilizar los dispositivos de almacenamiento.

Boot

- **Boot:** Es el proceso de inicialización de un dispositivo electrónico. Es cuando inicia la computadora.
- **Dual Boot/Multi Boot:** Permite gestionar varios gestores de arranque en un mismo gestor de arranque.
- **Grub:** GNU GRUB es un multiboot cargador de inicio. Se derivaba de GRUB, el GRand Unificado Bootloader, que fue diseñado e implementado originalmente por Erich Stefan Boleyn.
- **Booteear:** Es la acción de iniciar el sistema operativo.
- **USB Booteable:** Es un unidad de almacenamiento USB, que permite ser inicializada por la BIOS.
- **USB with Boot:** Lo mismo que USB Booteable.

Programación

- **Código:** Refiriéndonos específicamente al código de un programa, este representa todas las instrucciones necesarias para que dicho programa funcione. Estas instrucciones son escritas para la posterior interpretación y acción del pc.
- **Código fuente:** El código fuente es todo software sin compilar. Allí están toda su funcionalidad, indicada con lujo de detalle. Dicho código se podría modificar, compilar, cualquier cosa.

- **Programa compilado:** Un programa compilado esta listo para su uso general, no tiene a la vista su código, simplemente trae las instrucciones al lenguaje maquina, que dicho sea de paso, no es fácil de interpretar para un humano, solo es fácil para el computador

Coloquiales y Populares

- **Cacharrear:** En el contexto de la informática, 'cacharrear' se refiere a experimentar, trastear o jugar con un sistema operativo, software o hardware para entender su funcionamiento, personalizarlo o modificarlo. Es una actividad común entre los entusiastas de GNU/Linux.
- **Distrohopper:** Persona que instala una distro nueva cada semana. Sufre de 'distro-jumping' crónico.
- **Linuxeros:** Término utilizado para referirse a las personas que utilizan y/o son entusiastas de GNU/Linux. A menudo implica un cierto nivel de conocimiento técnico y una filosofía de software libre.
- **Jacker:** Persona que cree o parece ser hacker, pero no lo es.
- **Dubalin:** Persona que vive en dubai o que es adicta a comer paletas dubalin. Naa, la verdad me invente esta palabra.

Bibliografía

Las carpetas

- [El árbol de directorios de Linux](#)

BIOS

- [Para que sirve la bios](#)

BOOT

- [El boot](#)
- [Bootloader](#)
- [GNU GURB Website](#)
- [GURB](#)

APT y repositorios

- [apt-sources](#)
- [APT y repositorios](#)
- [Repositorios debian](#)
- [Rélicas de Debian](#)
- [DFSG](#)
- [Activar los repositorios privativos non-free en Debian](#)
- [Cómo habilitar los repositorios Contrib y Non-Free en Debian 12](#)

Init System

- [Init](#)
- [Runit](#)
- [OpenRC](#)
- [Systemd](#)

Variables de entorno

- [Hostinger](#)
- [linux-console](#)

Nano

- [DeepSeek Chat](#)

Uso de pipes

- [Pipes en bash](#)

Uso de grep

- [Comando Grep de Linux: cómo utilizarlo y ejemplos útiles](#)

Limpieza general

- [Limpiar linux](#)
- [Liberar espacio en caché en Debian y derivadas](#)

Logs y registros

- [Linux ver registro de inicio de sesión](#)
- [syslog debian 12](#)

Servicios Systemd

- [Administrar y crear servicios con systemd en Linux](#)
- [Servicios systemd](#)

CPU

- [Información de cpu](#)
- [Obtener información de CPU en Linux](#)

RAM

- [RAM usada y libre en Linux](#)

Almacenamiento

- [Guía para monitorear el uso del disco en Linux](#)
- [Cómo comprobar la información de tus discos en GNU/Linux](#)

Tarjeta gráfica

- [Cómo obtener información de la tarjeta gráfica en Linux](#)
- [Cómo verificar qué GPU está instalada en Linux](#)

Estresar computadora

- [Stress Test CPU and Memory \(VM\) On a Linux](#)
- [Pruebas de estrés en sistemas GNU/Linux](#)

Servidor gráfico

- [Wikipedia Servidor gráfico](#)

X11

- [Variables de Entorno en Linux](#)
- [Variable Refresh Rate/VRR](#)
- [HOWTO fix screen tearing](#)
- [HOWTO enable Adaptive Vertical Sync \(Freesync\) on AMD GPUs](#)

Wayland

- [How to set device-specific mouse settings in Wayland under Libinput \(Debian Gnome\)?](#)

Wine

- [Wine official website](#)

Winetricks

- [How to use the command 'winetricks'](#)

Licencia

Attribution-ShareAlike 4.0 International

Creative Commons Corporation ("Creative Commons") is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an "as-is" basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

Considerations for licensors: Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material not subject to the license. This includes other CC-licensed material, or material used under an exception or

limitation to copyright. More considerations for licensors:
wiki.creativecommons.org/Considerations_for_licensors

Considerations for the public: By using one of our public licenses, a licensor grants the public permission to use the licensed material under specified terms and conditions. If the licensor's permission is not necessary for any reason--for example, because of any applicable exception or limitation to copyright--then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. More considerations for the public:

wiki.creativecommons.org/Considerations_for_licensees

=====

Creative Commons Attribution-ShareAlike 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-ShareAlike 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 -- Definitions.

- a. Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.
- b. Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.
- c. BY-SA Compatible License means a license listed at creativecommons.org/compatiblelicenses, approved by Creative Commons as essentially the equivalent of this Public License.
- d. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.
- e. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.
- f. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights

that applies to Your use of the Licensed Material.

- g. License Elements means the license attributes listed in the name of a Creative Commons Public License. The License Elements of this Public License are Attribution and ShareAlike.
- h. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.
- i. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.
- j. Licensor means the individual(s) or entity(ies) granting rights under this Public License.
- k. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.
- l. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.
- m. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 -- Scope.

a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:
 - a. reproduce and Share the Licensed Material, in whole or in part; and
 - b. produce, reproduce, and Share Adapted Material.
2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.
3. Term. The term of this Public License is specified in Section 6(a).
4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a) (4) never produces Adapted Material.
5. Downstream recipients.

- a. Offer from the Licensor -- Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.
 - b. Additional offer from the Licensor -- Adapted Material. Every recipient of Adapted Material from You automatically receives an offer from the Licensor to exercise the Licensed Rights in the Adapted Material under the conditions of the Adapter's License You apply.
 - c. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.
6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

- 1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.
3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:
 - a. retain the following if it is supplied by the Licensor with the Licensed Material:
 - i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
 - ii. a copyright notice;
 - iii. a notice that refers to this Public License;
 - iv. a notice that refers to the disclaimer of warranties;

- v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
 - b. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
 - c. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.
2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.
3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

b. ShareAlike.

In addition to the conditions in Section 3(a), if You Share Adapted Material You produce, the following conditions also apply.

- 1. The Adapter's License You apply must be a Creative Commons license with the same License Elements, this version or later, or a BY-SA Compatible License.
- 2. You must include the text of, or the URI or hyperlink to, the Adapter's License You apply. You may satisfy this condition in any reasonable manner based on the medium, means, and context in which You Share Adapted Material.

3. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, Adapted Material that restrict exercise of the rights granted under the Adapter's License You apply.

Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

- a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;
- b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material, including for purposes of Section 3(b); and
- c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 -- Disclaimer of Warranties and Limitation of Liability.

- a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION,

WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.

- b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES, COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.
- c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 -- Term and Termination.

- a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.
- b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:
 1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

- c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.
- d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 -- Other Terms and Conditions.

- a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.
- b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 -- Interpretation.

- a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.
- b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License

without affecting the enforceability of the remaining terms and conditions.

- c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.
 - d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.
-

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the "Licensor." The text of the Creative Commons public licenses is dedicated to the public domain under the CC0 Public Domain Dedication. Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at creativecommons.org/policies, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at creativecommons.org.