

*Follow the white rabbit*  
Penguin

# Linux rabbit hole

Curso completo de Linux y los temas que lo rodean

Por Ariel Parra.

# 1. ¿Qué es Linux?

## 1.1 Definición

Es un kernel monolítico de código abierto con licencia GNU GPL v2, usado como base de sistemas operativos.

- ¿Qué es un kernel monolítico? El kernel es un software que tiene el control del hardware, administra y optimiza los recursos del sistema como asignación de memoria ram, procesos del CPU, drivers, etc. Actuando, así como un puente entre aplicaciones y el hardware. Este es monolítico cuando todos los servicios del sistema operan en el kernel a través de system calls.
- ¿Qué es el código abierto? Es software donde cualquier persona puede ver, analizar y por ende modificar el código fuente.
- ¿Cuál es la licencia GPLv2? Es una licencia de software donde cualquier modificación al código debe tener la misma licencia y también que por cada archivo binario que sea distribuido tiene que estar disponible su código fuente.

## 1.2 Historia

Fue creado en 1991 por el estudiante de la universidad de Helsinki Linus Torvalds, al mismo tiempo Richard Stallman junto con su proyecto GNU de la fundación de software libre (FSF) ya habían creado las aplicaciones y utilidades de un sistema operativo. Estas aplicaciones y utilidades fueron agregadas al Linux kernel para crear un systema completo llamado GNU/Linux.

## 1.3 Filosofía de linux

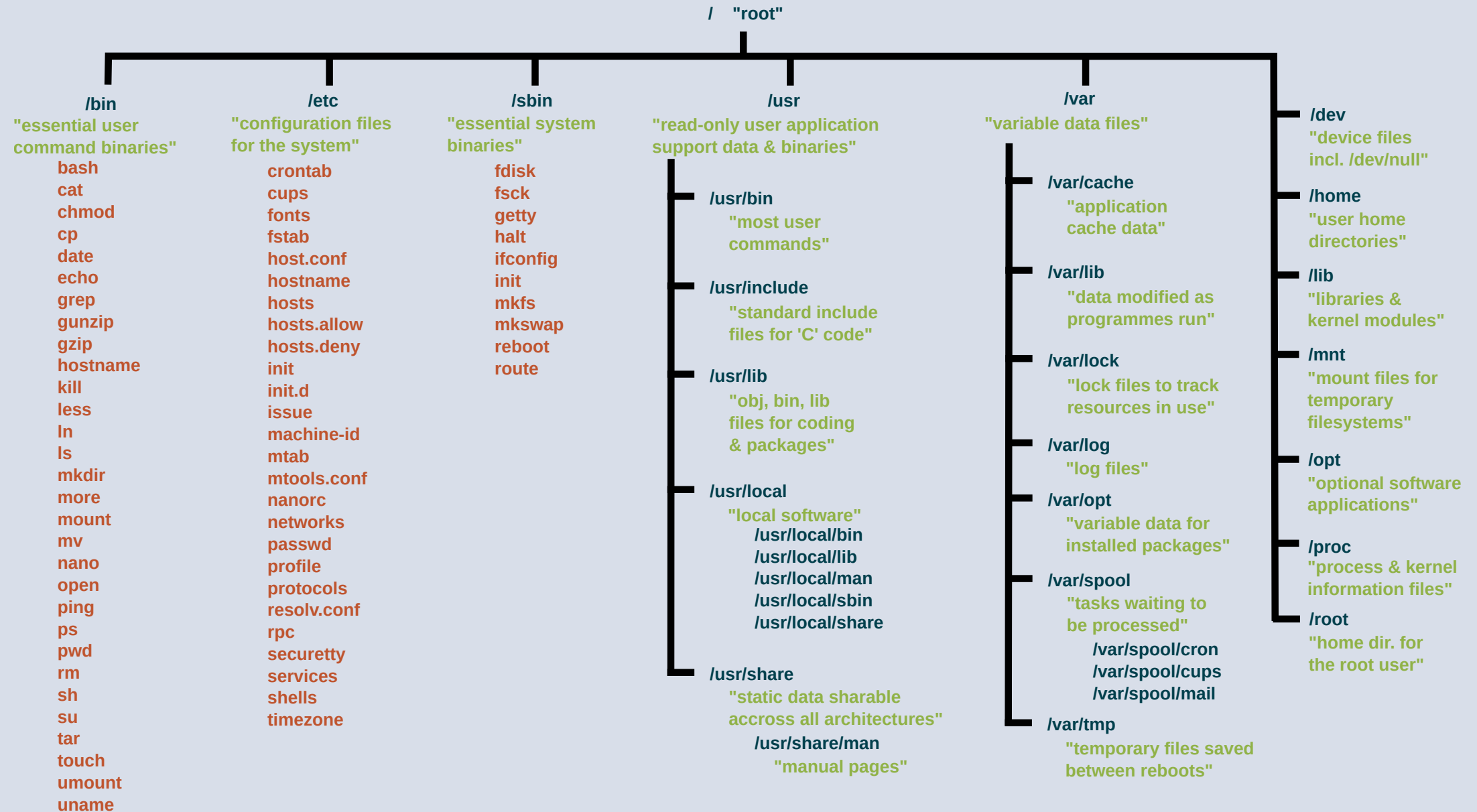
La filosofía de Linux se basa principalmente en la filosofía de UNIX y del open source, donde Linux se diferencia es al darle control completo al usuario, dándole la libertad de elegir, estudiar, modificar y distribuir cualquier parte de su sistema operativo.

El kernel de Linux en si no tiene blobs binarios, sino que estos son distribuidos en el Linux-firmware, esto sirve para inicializar el hardware y los drivers.

Linux tiene estándares como el LSB (Linux Standard Base) que define system interfaces y run times de donde las librerías y aplicaciones dependen.

Los sistemas operativos Linux suelen usar el Filesystem Hierarchy Standard de linux

# 1.4 Linux Filesystem Hierarchy Standard



# 1.5 las capas de Linux

Various layers within Linux, also showing separation between the <b>userland</b> and <b>kernel space</b>						
User mode	User applications	<i>bash, LibreOffice, GIMP, Blender, 0 A.D., Mozilla Firefox, ...</i>				
	System components	<b>init daemon:</b> <i>OpenRC, runit, systemd...</i>	<b>System daemons:</b> <i>polkitd, smbd, sshd, udevd...</i>	<b>Window manager:</b> <i>X11, Wayland, SurfaceFlinger (Android)</i>	<b>Graphics:</b> <i>Mesa, AMD Catalyst, ...</i>	<b>Other libraries:</b> <i>GTK, Qt, EFL, SDL, SFML, FLTK, GNUstep, ...</i>
	<b>C standard library</b>	<i>fopen, execv, malloc, memcpy, localtime, pthread_create ... (up to 2000 subroutines)</i> <i>glibc</i> aims to be fast, <i>musl</i> aims to be lightweight, <i>uClibc</i> targets embedded systems, <i>bionic</i> was written for <i>Android</i> , etc. All aim to be <b>POSIX/SUS-compatible</b> .				
Kernel mode	Linux kernel	<i>stat, splice, dup, read, open, ioctl, write, mmap, close, exit, etc. (about 380 system calls)</i> The Linux kernel <b>System Call Interface</b> (SCI), aims to be <b>POSIX/SUS-compatible</b> <sup>[100]</sup>				
		<b>Process scheduling subsystem</b>	<b>IPC subsystem</b>	<b>Memory management subsystem</b>	<b>Virtual files subsystem</b>	<b>Networking subsystem</b>
		Other components: <i>ALSA, DRI, evdev, klibc, LVM, device mapper, Linux Network Scheduler, Netfilter</i> Linux Security Modules: <i>SELinux, TOMOYO, AppArmor, Smack</i>				
Hardware ( <b>CPU, main memory, data storage devices, etc.</b> )						



## **2. temas relacionadas**

## 2.1 UNIX

UNIX es un sistema operativo creado a principios de los 70s en AT&T por Dennis Ritchie (creador del lenguaje C) y Ken Thompson (creador del lenguaje B).

- POSIX: "Portable Operating System Interface", son una serie de estándares de uso, impuestos por el IEEE (Institute of Electrical and Electronics Engineers) que incluye el I/O, la terminal, comandos y redes. Estos estándares se suele usar para hacer shell scripts compatibles con muchos sistemas operativos.
- Single UNIX Specification: Es un estándar para sistemas operativos para usar el trademark de UNIX, con interfaces para el lenguaje C, el shell y comandos.
- UNIX philosophy (Doug McIlroy): Escribe programas que hagan una cosa muy bien, que trabajen juntos (input de uno es el output de otro) y \*que manejen streams (serie de bytes para I/O), porque son una interfaz universal.
- Todo es un archivo, los dispositivos, teclados, discos duros, etc. estos son manejados con streams y almacenados en la carpeta /dev

## 2.2 licencias

Las licencias son permisos dados por quien tiene el copyright del contenido, las licencias pueden darte ciertos permisos de uso dependiendo de las licencias.

Copyright: El derecho de autor (copyright) es un tipo de propiedad intelectual que protege las obras originales del autor, dando la libertad de elegir que se hace o que no se hace con su obra.

Copyleft: Es una estrategia para usar el copyright para fomentar y alentar el derecho de copiar, compartir, modificar y mejorar las obras bajo esta licencia.

Dominio Publico: (No Rights Reserved)  es una licencia para deslindarte de los derechos de autor.

Creative commons: son licencias de uso libre pero con atribucion obligatoria, comunmente usados para imagenes, videos y articulos.



Licencias open source:

- [AGPL](#): similar al GPL pero para servicios en linea.
- [Apache](#): incluir la nota de copyright
- BSD [2/3](#): incluir la nota de copyright.
- GPL [v2/v3](#): codigo fuente disponible y copyleft.
- [MIT](#): incluya la nota de copyright.

## 2.3 Filosofía de software libre

La filosofía de software libre tiene relación con Richard Stallman y su proyecto GNU, esta dice que software debe respetar la libertad del usuario, la libertad de distribución, la libertad de uso, la libertad de estudio, etc. Yendo en contra del software privado y con código cerrado, los blobs binarios, las licencias y EULAs no permisivas y el DRM.

## 2.4 BSD

Berkeley Software Distribution, es un sistema operativo creado en la universidad de Berkeley en California, de código abierto y originalmente extensión de AT&T's Research UNIX, a diferencia de Linux este tiene la BSD C library, aunque pueden compartir ciertas utilidades de GNU

Los dos forks más importantes son:

- freeBSD: es un proyecto "grande" con muchos sistemas operativos forkeados apartir de el, como FreeNAS, pfsense, etc.
- OpenBSD: es un proyecto relativamente mas "chico" que freebsd, es inovador y crean nuevas herramientas usadas en toda la comunidad unix, mientras que es de los sistemas más seguros.

estos pueden tener ciertas ventajas y desventajas respecto a Linux.

## 2.5 Privacidad

Recomiendo esta [guia](#) de privacidad o tambien a la youtuber [Naomi Brockwell](#).

- [duckduckgo](#) no es bueno.
- [proton](#) no es confiable.
- las VPN no te dan [privacidad](#).
- la gente esta [desinformada](#) acerca de tor.

## 2.6 Seguridad

Recomiendo esta [guia](#) de seguridad de la arch wiki.

- usa [linux-hardened](#) kernel.
- [selinux](#) o [AppArmor](#) para el control de acceso.
- No uses [kali linux](#), como sistema operativo principal.
- usa OpenBSD, ya que es el sistema unix más [seguro](#).

## 2.7 Minimalismo

El minimalismo en cuestion de Linux, se trata de usar y escribir programas simples siguiendo principios de UNIX y KISS, dos ideas deribadas del minimalismo son:

- **gnuless**: algunos programas de GNU pueden no ser los más eficientes o minimalistas, un video acerca del tema: [Luke Smith: GNU is bloated!](#), donde se habla tambien de la pagina [harmfull cat -v](#) la cual tiene ideas extremas.
- **init freedom**: systemd es el init system más usado, pero este no sigue la filosofia de unix ya que systemd no solo es el init, ya que tiene muchas otras **funciones** y herramientas, por lo que usar **alternativas** es **recomendado**.

Paginas que recomiendan software minimalista: [lukesmith](#) y [suckless.org](#).



## 2.8 Ricing (Personalización)

Como Linux te permite modificar el sistema, puedes configurarlo a todos tus gustos artisticos, se suelen usar paletas de colores como [nord](#) o [Catppuccin](#).

Uno de los mejores lugares para ver, aprender y compartir rices es [r/unixporn](#), donde los usuarios comparten [dotfiles](#) que son los archivos que van en la carpeta de usuario y tienen las configuraciones de los programas que se van a customizar, usualmente son respaldados en un servidor de git.

Tambien puedes usar una distrubucion que ya venga riceada como [archcraft](#) o puedes usar scripts de la comunidad como [LARBS](#).

### **3. Comparación de Sistemas operativos**

# 3.1 Ventajas de Linux

- Es gratuito y open source.
- Superioridad en cuestion de audio con [pipewire](#).
- Puede arrancar [rapidamente](#) el sistema y las aplicaciones.
- Tiempos menores de [compilacion](#)
- Mejores resultados en [geekbench](#).
- Usa menos memoria [RAM](#).
- Es un sistema operativo [portable](#).
- Tiene multiples [comunidades](#) para soporte.
- Tiene drivers opensource oficiales de [AMD](#) y no oficiales de [Nvidia](#).
- Puedes "[revivir](#)" laptops antiguas.
- Promueve [open source](#).
- Tienes libertad de elección sobre cada componente del sistema operativo.

## 3.2 Más referencias de las ventajas de Linux

- [linfo: 25 Reasons to Convert to Linux](#)
- [Joe Collins: Linux vs. Windows | The Fundamental Differences](#)
- [Dave's Garage: Linux vs Windows Round 1: Open Source vs Proprietary - From a Retired Microsoft Dev](#)
- [DistroTube: How Linux Respects Your Privacy \(And How Windows Does NOT!\)](#)
- [Average Linux User: Linux advantages](#)
- [The Linux Foundation: Why are Linux Jobs so In Demand?](#)
- [The Linux Experiment: Windows, macOS & Linux PRIVACY compared: why do they need ALL THIS DATA?!](#)
- [ForrestKnight: Why I Code on Linux Instead of Windows](#)

## 3.3 Desventajas de Linux

- Tienes que aprender temas nuevos, aprender a buscar soluciones y leer.
- No puedes correr ciertos juegos: [rainbow six](#), [roblox](#), etc.
- No puedes correr ciertas aplicaciones: [Adobe Photoshop](#), [MS Office](#), etc.
- Problemas con [impresoras](#).
- Sin soporte para [HDR](#).
- Problemas con [variable refresh rate](#).
- Problemas con [fractional scaling](#).
- Screen tearing en [X11 \(xorg\)](#).
- Problemas al compartir pantalla con audio en [discord](#) y [MS Teams](#).
- El driver open source de nvidia tiene un [peor rendimiento](#) que el oficial.
- La capa de compatibilidad con Windows (wine) puede correr [malware](#).

## 3.4 Ventajas MacOS

- Tiene interfaces con diseños **inovadores**.
- Tiene una certificación **posix**.
- **Xcode** para desarrollar aplicaciones de iOS/MacOS.
- Juegos nativos como **roblox** o juegos de Blizzard como **WOW**.
- Aplicaciones nativas como **Adobe Photoshop**, **MS Office**, **FL Studio**, etc.
- Compatibilidad y uso de **heic/hecv** integrado al sistema.
- Tienen el **ecosistema de apple**.
- Tienen de los mejores **procesadores arm** para escritorio y soporte de aplicaciones de iOS.
- Soporte de **HDR**.
- Puedes compartir pantalla con sonido en **discord** y **MS Teams**.
- Buen rendimiento en juegos DX12 con **D3DMetal** y **game porting toolkit**.

## 3.5 Desventajas MacOS

- Appple ID es un requisito para instalar MacOS.
- No puede correr ciertos **juegos**.
- Quitaron el soporte de **32-bits** y con eso muchas **apps**.
- Hardware **oficial** y **no oficial** limitados.
- Las computadoras Mac son **costosas**.
- La existencia de **OCSP**.
- Múltiples problemas de **privacidad**.
- Los **contratos de licencia de software**.
- Posibles conexiones con NSA e implicación con **PRISM**.
- No suele haber drivers open source.
- La capa de compatibilidad con Windows (wine) puede correr **malware**.
- No tiene controles de volumen por **aplicación**.

## 3.6 Recomendaciones al usar MacOS

- No uses una cuenta de Apple ID con tus datos personales.
- Usa macOS en una vm con NAT activado.
- Lee esta [guia](#) de privacidad en MacOS.
- Usa [alacritty](#) como emulador terminal.
- Usa [brew](#) como package manager.
- Evita aplicaciones nativas de MacOS.



## 3.7 Ventajas de Windows

- Compatibilidad perfecta con [DirectX](#).
- [Compatibilidad](#) con aplicaciones antiguas de windows.
- Las empresas les importa la distribución del [mercado](#) de sistemas operativos, por lo que ciertas aplicaciones y juegos están pensados solo para Windows.
- Soporte de [HDR](#).
- Windows Subsystem for Linux ([WSL](#)) y Windows Subsystem for Android ([WSA](#)).

## 3.8 Desventajas de Windows

## 3.8.1 Desarrollador

- `Java updater` esta siempre activo en segundo plano.
- Ejecuta programas de manera lenta (a tirones) aunque sea con powershell.
- Hyper-V y Windows hypervisor platform solo disponibles para `Windows Pro`.
- Virtual Machine Platform, Hyper-V, Windows hypervisor platform y WSL2 interfieren con aplicaciones: `dell alienware control`, `throttle stop`, etc.
- Los hosts en Windows son `read-only` y modificarlo puede alertar al `antivirus`.
- Para correr un "`hello world!`" de rust ocupas que descargas visual studio community junto con 1gb para los c++ build tools y el Windows SDK.
- Visual studio community usa como `10gb` o `más`.
- Problemas con librerías, a veces tienes que compilarlas manualmente como `pycrypto` o en mi caso librespot de rust.
- Problemas de velocidad con `I/O` y `ntfs`.

## 3.8.2 Uso general

- Las apps por defecto se actualizan individualmente ya sea abriendo la aplicación o teniendo que descargar la nueva versión de la página del creador.
- Para usar winget ocupas una cuenta Microsoft para descargar [App-Installer](#) de la microsoft store y para usarlo ocupas constantemente aceptar los [UAC](#) prompts, aparte de aceptar la [CLA](#).
- No puedes nombrar archivos con ciertos [nombres](#).
- Windows pro cuesta [\\$6399](#) pesos MXN.
- windows 11 [requiere](#) a tener una cuenta Microsoft y usarla en tu sistema.
- El sistema operativo se expande hasta más de [40gb](#).
- El buscador de windows es [ineficiente](#).
- Prefetch y superfetch (SysMain), suelen tener el disco duro al [100%](#) de uso.

- Suele haber más [virus](#) que macOS o Linux
- aplicaciones freeware pueden tener virus como [atubecatcher](#), etc.
- La Microsoft store tiene aplicaciones más aplicaciones [.appx](#) que [.exe](#)
- Tienes que pagar para poder usar archivos [HEVC](#)
- Windows incluye [bloatware](#)
- Los antivirus pueden bloquear aplicaciones/servicios en el [firewall](#)
- Windows Antimalware Service puede usar [100%](#) del CPU
- Problemas en pantallas con [high dpi](#) dando lugar a [apps borrosas](#)
- La solución a la mayoría de errores es [reinstalar](#) el sistema operativo
- Las [apps](#) se ejecutan al inicio o crean servicios que se ejecutan al inicio
- Fondos de pantalla en slidehow puede no [funcionar](#)
- Pocas opciones open source de [customizacion](#) y algunas opciones privadas cuestan [wallpaperengine](#), [taskbarX](#), [start11](#), etc.

### 3.8.3 Hardware

- Device encryption ocupa hardware moderno (InstantGo/AOAC/HSTI-compliant), basicamente de 2014 en adelante.
- Los requisitos de Windows 11, procesador de 64 bits, UEFI, TPM v2.0 y una grafica compatible con DirectX12.
- Los discos duros con NTFS tienen que defragmentarse.
- Los drivers oficiales de Nvidia tienen telemetría.
- No suele haber drivers open source.

### 3.8.4 Juegos

- Descargas **muchas versiones** de microsoft visual c++ redistributable por juego.
- Los juegos de la microsoft store (xbox app) suelen correr **peor** o dar **input lag**.
- No puedes acceder a las **carpetas** de juegos de la microsoft store (xbox app) aun siendo administrador.

## 3.8.5 Privacidad

- Opciones anti privacidad activados por defecto.
- Windows phone home.
- Windows esta implicado con la NSA y PRISM.
- Datos de diagnóstico obligatorios.
- Edge y office envían telemetría.
- Delivery Optimization esta activado por defecto.



## 3.9 Más referencias de las desventajas de windows

- GNU: Microsoft's Software is Malware
- Jody Bruchon Tech: Windows 11 Must Be Stopped - A Veteran PC Repair Shop Owner's Dire Warning
- reddit: why many people dislike windows as an development environment?
- reddit: what is your worst experience with ms windows?
- reddit: why so much hate to windows?
- Wikipedia: Criticism of Microsoft
- installgentoo wiki: Windows
- installgentoo wiki: Windows 10
- itvision blog: why windows 10 sucks
- itvision blog: why windows 11 sucks

## 3.10 Recomendaciones al usar windows

- Usa Windows 10 sin iniciar sesión con una cuenta de Microsoft.
- Descarga apps de la ms store con <https://store.rg-adguard.net/> .
- Usa [PowerToys](#) para mejorar la experiencia de uso.
- Si usas Windows en una vm con NAT usa [WinApps](#).
- Si usas una gráfica Nvidia, instala drivers sin telemetría con [nvcleaninstall](#).
- Si es para jugar, preinstala todos los Visual C++ desde [techpowerup](#).
- Usa [Chris Titus Tech's Windows Utility](#) para desactivar fácilmente telemetría o también para activar UTC si se usa Windows en dual-boot.
- Bloquea las direcciones a donde se envían telemetrías en el archivo de [hosts](#).
- Prueba que el software descargado de internet no tenga virus con [virustotal](#).
- Usa aplicaciones con posibles virus en [sandboxie plus](#)
- [autoruns](#) para desactivar apps/servicios al inicio del sistema.
- Usa [alacritty](#) como emulador terminal y usa [scoop](#) como package manager.

## **4. Distribuciones de Linux**

## 4.1 ¿Qué es una distribucion de Linux?

Una distribución (distro) de Linux es un sistema operativo completo con muchos componentes open source, utilidades e interfaces.

## 4.2 ¿Qué compone una distribución de Linux?

Los componentes y características que construyen a una distribución de linux son:

- Filosofía: KISS/minimalismo, free/libre, bsd userspace, compilación, etc.
- Uso: juegos, media, privacidad, seguridad, escolar, etc.
- Release model: fix, rolling release, lts, bleeding edge
- Boot Loader: GRUB, Syslinux, LILO, systemd-boot, etc.
- Tipo de kernel: Stable (vanilla), zen, libre, LTS, Realtime, etc.
- Init system / daemon manager: runit, OpenRC, systemd, SysVinit, s6, dinit, etc.
- Sistema de archivos (File system): ext2/3/4, Btrfs, XFS, ZFS, etc.
- Jerarquía de archivos: Linux File Hierarchy Structure o modificación de este.
- Librería de c: glibc, musl, uClibc, etc.
- Utilidades básicas: gnu core utils, busybox, util-linux, bin-utils, lspci-utils, etc.

- Interfaz de shell: bash, zsh, fish, dash, ash, etc.
- Servidor grafico: Xorg (X11) / protocolo grafico: Wayland.
- Tecnología de audio: ALSA, pulseaudio, Jackaudio, Pipewire.
- Login/Display manager: sddm (KDE), GDM (GNOME), LightDM, etc.
- Window manager: dwm, sway, i3, xmonad, Hyprland, etc.
- Desktop environment: Gnome, Plasma, xfce4, etc.
- Graphics Toolkits: GTK+ , Qt, etc.
- Gestor de paquetes (package manager): apt, DNF, pacman, portage, yay, etc.
- Formato de paquetes: .deb, .rpm, .pkg.tar.zst, source code, install scripts, etc.

- Repositorio de paquetes de la distribución.
- Aplicaciones incluidas:
  - Terminal (CLI/TUI):
    - Editor de textos: nano, vim, neovim, emacs, etc.
    - Información: neofetch, htop, etc.
    - Etc.
  - Graficas (GUI):
    - Navegador: Firefox, LibreWolf, Chromium, etc.
    - Office Suite: Libreoffice, Onlyoffice, Calligra etc.
    - Emulador de terminal: konsole, gnome-terminal, alacritty, kitty, st, etc.
    - Editor de textos: kate, Gnome text editor, gvim, etc.
    - Explorador de archivos: nautilus (gnome), dolphin (KDE)
    - Visualizador de imágenes: feh, gwenviewer, etc.
    - Etc.

## 4.3 Categorías de las distribuciones

Hay distribuciones hechas con un uso específico estas pueden ser conjuntos de programas y configuraciones que adecuan el sistema a ese uso, aunque sigues teniendo la libertad de poder modificarlo a tu gusto:

- Juegos: SteamOS, ChimeraOS, popOS, Fedora Games spin, etc.
- Routing: openWRT, LibreCMC, ipfire, alpine, etc.
- Media: Ubuntu Studio, Fedora Design Suite, etc.
- Escolar: Uaubuntu, Edubuntu, Zorin OS 16 Education, etc.
- Privacidad/Seguridad: Whonix y/o Qubes, Tails, etc.

Hay dos grandes categorías de distribuciones:







- Independientes: tienen su propio package manager y repositorio
- Derivados: tienen compatibilidad con quien se derivaron y pueden crear una sinergia: debian <-> ubuntu, SUSE <-> openSUSE, RHEL <-> fedora.









- Tipos de distribuciones derivadas:
- Flavour: con otros escritorios, temas, y/o aplicaciones, siendo reconocidas por la distribución, por ejemplo, para Ubuntu hay flavours como: edubuntu,ubuntu,lubuntu, etc.
- Remix/Respin: modificación ofreciendo algo nuevo, pero sin ser reconocidos por la distribución.
- Fork: hacen cambios grandes, cambiando el uso, filosofía, etc.
- Clone/Repackage: copian el código y tienen otra infraestructura para la distribución, por ejemplo, para RHEL hay repackages como Rocky linux, Oracle Linux, etc.






Hay muchísimas **distribuciones** para elegir, pero hay cuatro componentes principales que diferencian a una distribución de linux, estos son: el sistema de arranque (init system/daemon), gestor de paquetes (package manager), la librería de c y las core utils.

## 4.3.1 Distribuciones independientes

Logo	distribucion	init	pkg manager	libreria c	core utils
	Alpine	openrc	apk	musl	busybox
	Arch	systemd	pacman	glibc	gnu
	Debian	systemd	apt (dpkg)	glibc	gnu
	Slackware	sysVinit	slackpkg	glibc	gnu
	Gentoo	OpenRC/systemd	portage	glibc/musl	gnu
	Void	runit	xbps	glibc/musl	gnu

Logo	distribucion	init	pkg manager	libreria c	core utils
	Nixos	systemd	nix	glibc	gnu
	RHEL	systemd	DNF (yum)	glibc	gnu
	SUSE LE S/D	systemd	Zypper	glibc	gnu
	LFS	systemd/sysV	-	glibc	gnu
	Kiss	runit	kiss	musl	busybox
	Chimera-linux	Dinit	apk	musl	BSD-derived

## 4.3.2 Distribuciones Derivadas

Logo	distribucion	init	basado en	libreria c	core utils
	Artix	OpenRC/runit/etc.	arch	glibc	gnu
	Devuan	sysV/OpenRC/etc.	debian	glibc	gnu
	Ubuntu	systemd	debian	glibc	gnu
	Uaubuntu	systemd	ubuntu	glibc	gnu
	OpenSUSE	systemd	SUSE	glibc	gnu

Logo	distribucion	init	basado en	libreria c	core utils
	Fedora	systemd	RHEL	glibc	gnu
	Endeavour	systemd	arch	glibc	gnu
	Archcraft	systemd	arch	glibc	gnu
	PopOS	systemd	ubuntu/debian	glibc	gnu
	Mint	systemd	ubuntu/debian	glibc	gnu
	Elementary	systemd	ubuntu/debian	glibc	gnu
	Zorin	systemd	ubuntu/debian	glibc	gnu

### 4.3.3 Recomendadas por la FSF (linux-libre kernel)

Logo	distribucion	init	basado en	pkg manager	libreria c	core utils
	Parabola	systemd/OpenRC/etc.	arch	pacman	glibc	gnu
	Hyperbola	OpenRC	debian/arch	pacman	glibc	gnu
	Guix	shepherd	independiente	Guix	glibc	gnu
	Trisquel	systemd	ubuntu/debian	apt	glibc	gnu

## 5. Comandos

A las aplicaciones con CLI/TUI (Command Line Interface/ Terminal User Interface) y utilidades de terminal se les llama comandos, estos como en las funciones de programación, tienen argumentos y/o parámetros con los que se usan, siendo los argumentos los archivos o directorios usados y los parámetros las opciones del comando.

Categorice los comandos que creo más importantes en su función y los marque en funcion del conjunto de utilidades con seis letras mayúsculas siendo C de "GNU core utils", U de "Util-linux", G de "GNU Software", P de "procps", I de "iproute2 o iputils", S de "SHELL" y O de "Otros".

## 5.1 Comandos de navegacion de archivos

comando	uso común	comando	uso común
(C) touch	touch <archivo>	(G) gzip	gzip (-d) <archivo>
(C) cp	cp <origen> <destino>	(G) tar	tar (c/x)zfv <archivo>.tar.gz <dir>
(C) ls	ls <dir>, ls -l -a, ls -la <dir>	(C) pwd	pwd
(C) rm	rm <archivo>, rm -r -d -f <dir>	(C) cd	cd <dir>
(C) mkdir	mkdir <dir>	(C) ln	ln -s <origen> <destino>
(C) rmdir	rmdir <dir>	(C) chmod	chmod +x <archivo>
(C) mv	mv <archivo> <destino>	(C) chown	chown <usuario>:<grupo> <archivo>



## 5.2 Comandos de Lecto-escritura de archivos

comando	uso común	comando	uso común
(C) cat	cat <archivo>	(G) sed	sed -i 's/este/por-este/' <archivo>
(G) less	less <archivo>	(G) awk	<comando>   awk '{print \$1,\$4}'
(C) head	head <archivo>	(O) vi	vi <archivo>
(C) tail	tail <archivo>	(C) sort	sort <archivo>

## 5.3 Comandos acerca de procesos

comando	uso común	comando	uso común
(S) jobs	jobs	(U) kill	kill <pid>
(S) bg	bg <job id>	(N) killall	killall <app>
(S) fg	fg <job id>	(P) pkill	pkill <app>
(P) ps	ps, ps aux	(O) top	top
(O) pstree	pstree	(O) htop	htop

## 5.4 Comandos para encontrar cosas

comando	uso común
(G) find	find <archivo>
(S) history	history
(G) grep	<comando>   grep <string>
(O) pgrep	pgrep <comando>
(O) locate	locate <archivo>
(U) wheris	wheris <archivo>

## 5.5 Comandos de almacenamiento

comando	uso común	comando	uso común
(C) df	df	(U) mount	mount -t ntfs-3g <dev> <destino>
(C) du	du -h <archivo>	(U) umount	umount <destino>
(U) fsck	fsck -y <dev>	(U) lsblk	lsblk
(U) fdisk	-l <dev>	(U) blkid	blkid
(G) cfdisk	cfdisk		

## 5.6 Comandos de redes

comando	uso común	equivalente a net-tools
(I) ip	ip address, ip a	ifconfig
	ip link set/add/remove <interfaz>	ifconfig
	ip route	route, netstat
	ip neigh	arp
(I) ss	ss	netstat

comando	uso común	comando	uso común
(I) traceroute	traceroute <pagina web>	(O) wget	wget <pagina web>
(I) ping	ping <pagina web>	(O) curl	curl -o <archivo> <pagina web>
(O) ssh	ssh <usuario>@<ip>	(O) git	git clone <servidor.git>
(O) scp	scp <origen> <usuario>@<ip>: <destino>	(U) rfkill	rfkill (un)block all

## 5.7 Comandos de informacion

comando	uso común	comando	uso común
(U) dmesg	dmesg -t	(O) which	which <comando>
(O) neofetch	neofetch	(C) whoami	whoami
(C) uname	uname	(C) who	who
(G) time	time <programa>	(C) uptime	uptime
(G) date	date +"%D"	(O) lsusb	lsusb
(C) hostname	hostname	(U) lscpu	lscpu
(P) free	free -h	(O) xinput	xinput list
(O) lspci	lspci	(C) env	env

## 5.8 Comandos extras

comando	uso común	comando	uso común
(O) man	man <comando>	(U) su	su, su <usuario>
(O) whatis	whatis <comando>	(C) wc	wc <archivo>
(C) echo	echo '<string>', echo \$<variable>	(G) diff	diff <archivo> <archivo>
(S) exit	exit	(O) xclip	xclip -sel clip
(C) printf	printf "entero: %i" "\$<variable>\n"	(G) gpg	gpg --verify <archivo>.sig/.gpg
(O) sudo	sudo <comando>	(C) sha512sum	sha512sum -c <archivo>.sha512sum



## **6. Funciones específicas**

## 6.1 Funciones de shell

función	ejemplo de uso	función	ejemplo de uso
*	find /*	!	!<num>
=	<variable>=<valor>	!!	!!
\$()	cowsay \$(fortune)		fortune   cowsay
{}	echo {1..10}		echo 'try'    echo 'catch'

función	ejemplo de uso	función	ejemplo de uso
&	<comando> &	&&	<comando> && <comando>
>	echo "ola" > <archivo>	;	<comando> ; <comando>
>>	echo "ola" >> <archivo>	alias	alias nombre=<comando>
2>&1	firefox >>/dev/null 2>&1 &	export	export

## 6.2 Shortcuts de terminal

### LINUX TERMINAL SHORTCUTS CHEATSHEET



Automatically complete the file, directory, or command you're typing.



Erases the complete line.



Kill the current foreground process running in terminal.



Erase the part of the line after the cursor.



Suspend the current foreground process running in terminal.



Erase the word before the cursor.



Delete the character at the cursor location.



Paste the last thing you cut from the clipboard.



Clears the screen just like clear command.



Go to the previous command in the command history.



Move the cursor to the beginning of the line.



Go to the next command in the command history.



Move the cursor to the end of the line.



Search for a command in command history.

## 6.3 Funciones especificos de la distro

- **WIFI**: NetworkManager, IWD, ConnMan, wpa\_supplicant.
- **Firewalls**: iptables, uwd, firewallld.
- **Package Manager**: apt, pacman, portage, YUM, Zypper etc.
- **Init system**: poweroff, shutdown, poweroff, halt.
  - **daemon management**: systemctl, sv, rc-service, etc.
- **Cron jobs**: cronie, fcron, dcron, etc.
- **shell scripts**: Shebang, compatibilidad posix.

## 6.4 Más informacion acerca de comandos

- [The linux Command Handbook](#)
- [arch wik: core utilities](#)
- [decoded gnu core utils](#)
- [Gentoo wiki: util-linux, procps](#)
- [iproute2](#)
- [iputils](#)
- [net-tools.](#)
- [Learn Linux: Linux Commands for Beginners](#)
- [freeCodeCamp.org: The 50 Most Popular Linux & Terminal Commands - Full Course for Beginners](#)
- [NetworkChuck: 60 Linux Commands you NEED to know \(in 10 minutes\)](#)

## **7. Aplicaciones**

Con la filosofía de Linux de usar programas interconectados, las aplicaciones suelen depender de otros programas llamados dependencias, los encargados de mantener las relaciones entre aplicaciones y dependencias son los Package Managers. Aun así las aplicaciones pueden llegar a tener problemas de compatibilidad si la aplicación o la dependencia esta desactualizada, para solucionar este problema se han creado formatos de paquetes que son portables, donde la aplicación es distribuida junto con la dependencias, esto puede ocasionar tamaños de aplicaciones más grandes, a costa de una mayor "estabilidad", los tres grandes distribuidores de aplicaciones portables son: [AppImage](#), [flatpak](#) y [Snapcraft](#)



# 7.1 Aplicaciones recomendadas

- Editor de texto/código: [vim](#), [vsodium](#)
- Ofimática en markdown: presentaciones con [marp](#), documentos con [pandoc](#)
- Navegador: [Librewolf](#)/[Firefox](#) + [Arkenfox.js](#) o [ungoogled-chromium](#)
- Extensiones de navegador: [uBlock origin](#), [bitwarden](#), [libRedirect](#), [decentraleyes](#), [ClearURLs](#)
- Motor de búsqueda: [SearXNG](#)
- Emulador de terminal: [Alacritty](#) o [st](#)
- Monitor de recursos: [btopy](#)
- Visualizador de imágenes: [feh](#)
- Reproductor de video: [mpv](#)
- Descargar videos de Youtube: [yt-dlp](#)
- Respaldos: [timeshift](#) ([rsync](#), [btrfs snapshots](#)) y [git](#)
- Android en Linux: [waydroid](#)

## 8. Hardware

## 8.1 Procesadores x86 (i686)/x64 (x86\_64, amd64)

Es una familia de procesadores con arquitectura CISC (complex instruction set computing), creados originalmente por intel en 1978. Tienen un ISA (Instruction Set Architecture) close source. En cuestión de Linux estos suelen necesitar de un microcode para un mejor rendimiento del procesador.

Estos son los tipos procesadores más **usados** en computadoras de escritorio, laptops y servidores.

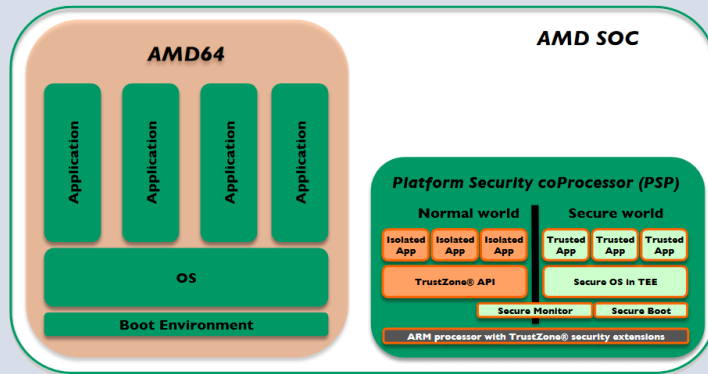
los procesadores de las laptops pueden alcanzar altas temperaturas y para mitigar esto se solía hacer undervolting, pero es una **vulnerabilidad**, por lo que se recomienda utilizar **frequency scaling** y desactivar **Turbo Boost**.

Hay dos grandes [vulnerabilidades](#) para este tipo de arquitecturas que actúan desde la capa -3 de la seguridad del procesador: Intel ME y AMD PSP.

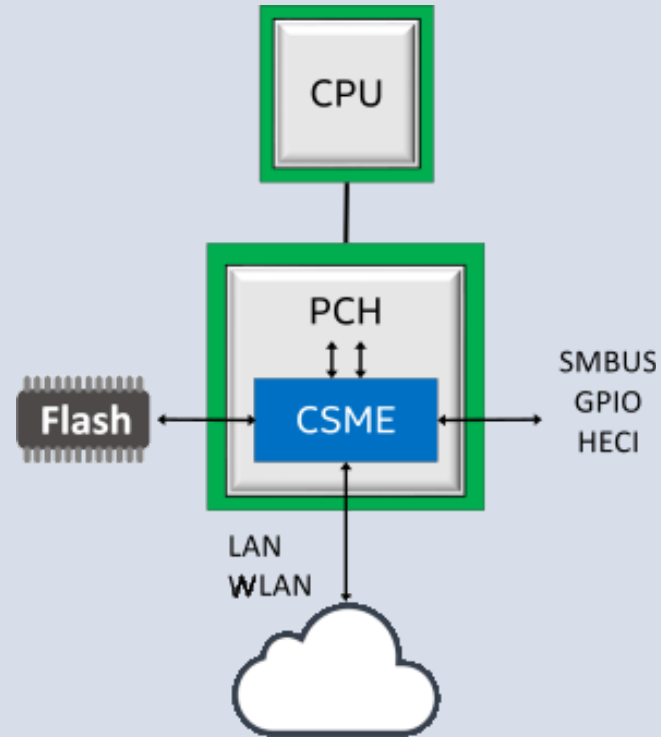
Intel ME es un sistema operativo que siempre está activo mientras la placa madre tenga corriente, este puede acceder a la tarjeta de red, lo que lo convierte en una gran [backdoor](#). Este se puede deshabilitar en una gran parte con [me\\_cleaner](#). Pero si se deshabilita Intel ME en procesadores de la 11 generación (Tiger-Lake) en adelante, se usa el [triple de energía](#) en estado de suspensión.

AMD PSP: Es un procesador ARM independiente del procesador x86, el cual ejecuta su propio firmware y este puede llegar a tener [vulnerabilidades](#). Se pueden deshabilitar ciertos módulos en algunas [motherboards](#)

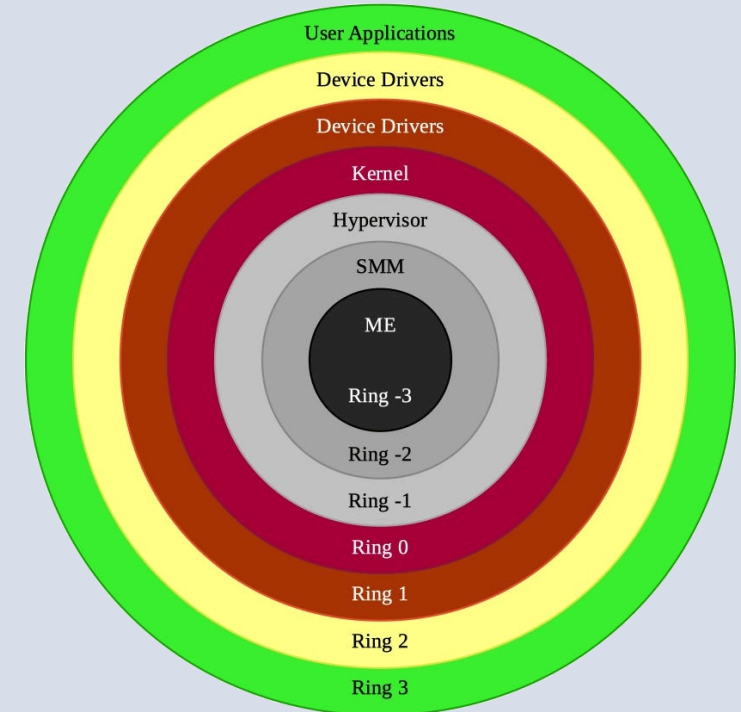
La mejor forma de usar un procesador con esta arquitectura es con un BIOS/UEFI parchado con [Libreboot](#) o [Coreboot](#) ya que estos te permiten mitigar los efectos de Intel ME o AMD PSP.



Arquitectura de AMD PSP. Imagen recuperada de [freundschafter](#).



Arquitectura de Intel ME. Imagen recuperada de [black hat](#).



CPU security rings. Imagen recuperada de [RealWorldCyberSecurity](#).

## 8.2 Procesadores arm

Es una familia de procesadores con arquitectura RISC (Reduce instruction set computing) con un ISA close source; este tiene un TEE (Trusted Execution Environment) close source llamado TrustZone la cual puede tener vulnerabilidades

Es el procesador más usado en el mundo ya que este se encuentra en los celulares, aunque también se pueden encontrar en laptops, computadoras de escritorios, servidores y single board computers. La razón de esto es su eficiencia energética.

## 8.3 Procesadores Risc-V

Es una familia de procesadores con arquitectura RISC, con un ISA open source, por el momento estos tienen menor potencia computacional que arm por lo que hay menos **usos**; este también tiene un TEE llamado keystone, la diferencia es que este es open source.

## 8.4 Hardware recomendado

Linux es compatible con la mayoría de hardware, pero es recomendado usar hardware especializado para tener una compatibilidad perfecta con Linux donde se respete tu privacidad y libertad de uso.

vendedores de laptops/computadoras con Coreboot:

- [Starlabs](#)
- [Tuxedo](#)
- [System76](#)

hardware y computadoras sin software propietario:

- [Libreboot list](#)
- [H-node](#)
- [FSF Recommended Complete Systems](#)
- [Respects Your Freedom Certification](#)



## 9. Servidores

## 9.1 Software para servidores

Las distribuciones [más usadas](#) en servidores son las basadas en Debian como: ubuntu server y Debian o las empresariales como: RHEL y SLES, aunque también se usa Proxmox como hypervisor para VMs o containers, pero si se usara una SBC recomendando [armbian](#).

Los file systems más usados son ZFS (openZFS) o XFS, pero recomendando este [video](#) de Chris Titus para elegir el file system para tu servidor.

containers: permiten la agrupación y aislamiento de aplicaciones con todo su entorno de ejecución, para mantener la funcionalidad. Las soluciones más usadas son [Docker](#), [Podman](#), [Skopeo](#), [Buildah](#) y [Containerd](#).

Container Helpers: [Kubernetes](#), [Portainer](#), [Rancher](#), [Yatch](#), [LazyDocker](#) y [CasaOS](#).

El web server más usado es nginx con el [34.4%](#) del total de páginas web y este tiene [ventajas](#) sobre Apache en varios aspectos.

## 9.2 Hardware para servidores caseros

Laptops: thinkpads (T60s, T400s, X200s), etc.

All in One (AIO)/Mini PCs: Dell OptiPlex, HP elitedesk, HP elite mini, ThinkCentre, Beelink Mini, ZimaBoard, etc.

Single Board Computers (SBC)/System on a chip (SoC) con arm: la [FSF](#) recomienda procesadores Rockchip, FreeScale y Allwinner. Yo recomiendo los procesadores Rockchip con 4 núcleos o más: [orange Pi 5 Plus](#), [orange pi 5](#), [ROCKPro64](#) y [Quartz64 Model A](#).

Ventajas de usar laptops, SBCs y AIOs: Tamaños pequeños, bajos db de ruido, precios accesibles en primera y segunda mano, suelen tener un bajo consumo energético, las laptops tienen pantallas y teclados para mantenimiento y pueden usar sus baterías en caso de un corte de luz, los AIO pueden usar ECC RAM y/o tarjetas gráficas y los SBC tienen GPIO (General-purpose input/output) pin headers.

Desventajas: puertos limitados, los gráficos integrados pueden no ser óptimos para media servers, los SBC suelen usar tarjetas micro SD que tienen **bajas velocidades de lecto-escritura** y son propensas a **corrupción de datos**.

Soluciones/actualizaciones: **USB en vez de microSD para SBCs**, no break como respaldo de energia, clusters, load balancer, USB 2.5gb Ethernet Adapter, adaptadores sata a USB, USB hub con energia externa, HDD/SSD en el puerto de DVD de laptops, puertos de tarjeta red (M.2 key/mini PCIe) o express card de laptop: eGPU, ethernet card, usb 3.0, sata hub, ssd.

## 9.3 Servicios para servidores

Guía de seguridad para servidores por [LiveOverflow](#)

VPS: Virtual Personal Server, son servidores en la "nube", yo recomiendo [vultr](#), pero hay alternativas gratuitas como [AWS Free tier](#) (por 12 meses) y [Oracle Cloud Free Tier](#) (para siempre)

Página web sin servidor ni dominio: [Github Pages](#) y [neocities](#).

DDNS: los ISP pueden cambiar tu dirección ip pública en cualquier momento, las soluciones a esto que recomiendo son: [cloudflare API](#), [duckDNS](#) y [freeDNS](#).

Dominios Web gratis por un año c/u, con [GitHub Student Developer Pack](#):  
[.tech](#), [.me](#), [.engineer](#), [.software](#), [.live](#) etc.

Certificados SSL: son los certificados que te hacen que una página web tenga https, yo recomiendo: [Let's Encrypt](#)

Reverse proxy: es un servidor que redirecciona el tráfico al servidor de la página web, con mayor seguridad y velocidad. Las soluciones que recomiendo son: [cloudflare tunnels](#) y [Nginx Proxy Manager](#)

NAS: Network Attached Storage, es una forma de acceder a discos duros a través de una red ya sea en casa o desde fuera, existen sistemas operativos como freeNAS y equipos dedicados, pero conviene mejor hacer tu propio NAS.

RAID: Redundant Array of Independent Disks, storage, los discos duros son más baratos que los ssd pero estos pueden fallar y ser más lentos, las soluciones que hay son: [RAIDZ](#) con ZFS o [MergerFS](#) + [SnapRAID](#), RAID5, RAID6, RAID10, etc.

## 9.4 Proyectos para servidores

- Página web con [nginx](#)
- Password manager con [vaultwarden](#)
- Almacenamiento en la nube con [nextcloud](#)
- Nube con IA para fotos con [photoprism](#)
- Search engine con [searxng](#)
- VPN con [wireguard](#)
- Controlador de DNS con [pihole](#)
- Servidor de email con [postfix](#)
- [etc.](#)

## **10. Aprender más acerca de Linux**



# 10.1 Guías y Paginas

- Arch wiki
- Gentoo wiki
- Hack the box: learn Linux
- EndevoarOs wiki
- Machtelt Garrels: Introduction to Linux
- Linux journey
- Wikipedia: Linux portal
- Wikibooks: how Linux works
- IBM: Learn Linux 101
- Linux foundation: introduction to Linux
- Kernel.org: Documentation
- Y más

## 10.2 Youtubers de Linux que recomiendo

- The linux experiment: noticias semanales
- DistroTube: reviews de distros, opiniones y tutoriales
- Luke Smith: opiniones, recomendaciones y tutoriales
- Mental Outlaw: opiniones y noticias
- Michael Horn: tips de Linux
- Wolfgang's: tips para servidores
- Hardware Haven: hardware para servidores
- Y más

# 11. Referencias

- Adekotujo, A., Odumabo, A., Adedokun, A., & Aiyeniko, O. (2020). *A Comparative Study of Operating Systems: Case of Windows, UNIX, Linux, Mac, Android and iOS*. [https://www.researchgate.net/profile/Adedoyin-Odumabo/publication/343013056\\_A\\_Comparative\\_Study\\_of\\_Operating\\_Systems\\_Case\\_of\\_Windows\\_UNIX\\_Linux\\_Mac\\_Android\\_and\\_iOS/links/61f2b50a9a753545e2fe8300/A-Comparative-Study-of-Operating-Systems-Case-of-Windows-UNIX-Linux-Mac-Android-and-iOS.pdf](https://www.researchgate.net/profile/Adedoyin-Odumabo/publication/343013056_A_Comparative_Study_of_Operating_Systems_Case_of_Windows_UNIX_Linux_Mac_Android_and_iOS/links/61f2b50a9a753545e2fe8300/A-Comparative-Study-of-Operating-Systems-Case-of-Windows-UNIX-Linux-Mac-Android-and-iOS.pdf)
- arm. (s.f.). \* CPU Architecture: A Foundation for Computing Everywhere\*. <https://www.arm.com/architecture/cpu>
- arm. (s.f.). *TrustZone for Cortex-A*. <https://www.arm.com/technologies/trustzone-for-cortex-a>
- Boschs, P. (2019) *Introduction to the Intel Management Engine OS (Part 1)*. [https://pbx.sh/intelme-sw1/#hasarfaty\\_2019](https://pbx.sh/intelme-sw1/#hasarfaty_2019)
- Britannica, T. Editors of Encyclopaedia (2023). *Linux*. Encyclopedia Britannica. <https://www.britannica.com/technology/Linux>
- Cloudflare. (s.f.). *What is a reverse proxy? | Proxy servers explained*. <https://www.cloudflare.com/learning/cdn/glossary/reverse-proxy/>
- Cloudflare. (s.f.). *What is dynamic DNS (DDNS)?*. <https://www.cloudflare.com/learning/dns/glossary/dynamic-dns/>
- copyright.gov. (s.f.). *What is Copyright?*. <https://www.copyright.gov/what-is-copyright/>
- copyright.org. (s.f.). *What is copyleft?*. <https://copyleft.org/>
- coreboot. (2017). 1. *AMD Platform Security Processor (PSP) Firmware Integration Guide*. [https://doc.coreboot.org/soc/amd/psp\\_integration.html#platform-security-processor-psp-overview](https://doc.coreboot.org/soc/amd/psp_integration.html#platform-security-processor-psp-overview)

- Cray. (2016). *The final ISA showdown: Is ARM, x86, or MIPS intrinsically more power efficient?*. <https://community.arm.com/arm-community-blogs/b/architectures-and-processors-blog/posts/the-final-isa-showdown-is-arm-x86-or-mips-intrinsically-more-power-efficient>
- creativecommons. (s.f). *About The Licenses*. <https://creativecommons.org/licenses/>
- creativecommons. (s.f.). *About The Licenses*. <https://creativecommons.org/licenses/>
- David Both. (2014). *The impact of the Linux philosophy*. <https://opensource.com/business/14/12/linux-philosophy>
- Dayzerosec. (2023). *Reversing the AMD Secure Processor (PSP) - Part 1: Design and Overview*. <https://dayzerosec.com/blog/2023/04/17/reversing-the-amd-secure-processor-psp.html>
- Dodge, C., Irvine, C., & Nguyen, T. (2005). *A study of initialization in Linux and OpenBSD*. <https://core.ac.uk/download/pdf/36700709.pdf>
- Eric Steven Raymond. (2003). *Basics of the Unix Philosophy*. <http://www.catb.org/~esr/writings/taoup/html/ch01s06.html>
- ExplainingComputers. (2022). *Explaining RISC-V: An x86 & ARM Alternative* [Video]. <https://yewtu.be/watch?v=Ps0JFsyX2fU>
- ezeelinux. (2018). *Linux Distributions Deconstructed*. <https://www.ezeelinux.com/news/linux-distributions-deconstructed/>
- Fireship. (2022). *Linux in 100 Seconds* [Video]. YouTube. <https://yewtu.be/watch?v=rrB13utjYV4>
- freeBSD. (2023). *Explaining BSD*. <https://docs.freebsd.org/en/articles/explaining-bsd/>

- freundschafter. (2018). *About AMD TrustZone, AMD Platform Security Processor (PSP), AMD Secure Technology*. <https://freundschafter.com/research/about-amd-trustzone-amd-platform-security-processor-psp-amd-secure-technology/>
- GeeksforGeeks. (2019). *Linux Tutorials | Getting Started | Introduction | GeeksforGeeks* [Video]. [https://yewtu.be/watch?v=0EDwEQoui\\_g](https://yewtu.be/watch?v=0EDwEQoui_g)
- GNU. (2023). *GNU Coreutils*. <https://www.gnu.org/software/coreutils/manual/coreutils.pdf>
- GNU. (2023). *GNU General Public License, version 2*. <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>
- GNU. (s.f.). *What is Free Software?*. <https://www.gnu.org/philosophy/free-sw.html>
- IEEE. (2017). *IEEE Std 1003.1™-2017*. <https://pubs.opengroup.org/onlinepubs/9699919799/>
- Intel. (2017). *What is Intel® Management Engine?*. <https://www.intel.com/content/www/us/en/support/articles/000008927/software/chipset-software.html>
- Keystone. (2021). *What is Keystone Enclave?*. <https://keystone-enclave.org/>
- Libreboot. (s.f.). *Intel Management Engine (ME)*. <https://libreboot.org/faq.html#intelme>
- Linus Torvalds. (1997). *Linux: a Portable Operating System*. [https://www.cs.helsinki.fi/u/kutvonen/index\\_files/linus.pdf](https://www.cs.helsinki.fi/u/kutvonen/index_files/linus.pdf)
- Linux Foundation. (2015). *Filesystem Hierarchy Standard*. [https://refspecs.linuxfoundation.org/FHS\\_3.0/fhs-3.0.html](https://refspecs.linuxfoundation.org/FHS_3.0/fhs-3.0.html)
- Linux Foundation. (2015). *Linux Standard Base*. <https://refspecs.linuxfoundation.org/lsb.shtml>
- Machtelt Garrels. (2008). *Introduction to Linux*. <https://tldp.org/LDP/intro-linux/intro-linux.pdf>

- Michael Tunnell. (2023). *Explaining Linux Family Trees: Why Are There So Many Linux Distro?* [Video]. <https://yewtu.be/watch?v=kF8CRt05s6A>
- NGINX. (s.f). *What Is a Reverse Proxy Server?*. <https://www.nginx.com/resources/glossary/reverse-proxy-server/>
- Opensource.com. (s.f.). *What is Linux?*. <https://opensource.com/resources/linux>
- Opensource.com. (s.f). *What is open source?*.<https://opensource.com/resources/what-open-source>
- Open Source Initiative. (2007). *The Open Source Definition*. <https://opensource.org/osd/>
- Open source initiative. (s.f). *OSI Approved Licenses*. <https://opensource.org/licenses/>
- Red Hat. (2019). *What is the Linux kernel?*. <https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel>
- Red Hat. (2023). *Understanding containers*. <https://www.redhat.com/en/topics/containers>
- Roch, B. (2004). *Monolithic kernel vs. Microkernel*. [https://web.cs.wpi.edu/~cs3013/c12/Papers/Roch\\_Microkernels.pdf](https://web.cs.wpi.edu/~cs3013/c12/Papers/Roch_Microkernels.pdf)
- Seagate. (s.f ). *What is NAS*. <https://www.seagate.com/blog/what-is-nas-master-ti/>
- SUSE. (s.f.). *Linux Distribution*. <https://www.suse.com/suse-defines/definition/linux-distribution/>
- The Linux Experiment. (2018). *Linux DISTRIBUTION: explained* [Video]. <https://yewtu.be/watch?v=6ggLWTSz6ck>
- The Linux Information Project. (2005). *Flavors of UNIX Definition*. <http://www.linfo.org/flavors.html>
- The Linux Information Project. (2006). *What is Linux?*. <http://www.linfo.org/newbies.html>

- The Linux Kernel Organization. (2019). *About Linux Kernel*. <https://www.kernel.org/linux.html>
- The Linux Kernel Organization. (2019). *Is Linux Kernel Free Software?*. <https://www.kernel.org/category/faq.html>
- The Open Group. (2008). *The Single UNIX Specification, Version 4*. <https://unix.org/version4/overview.html>
- University of Pittsburgh. (2022). *Copyright and Intellectual Property Toolkit*. <https://pitt.libguides.com/copyright/licenses>
- Westerndigital. (s.f.). *RAID Storage*. <https://www.westerndigital.com/solutions/raid>
- Wheeler, D. (2003). *History of Unix, Linux, and Open Source / Free Software*. <https://tldp.org/HOWTO/Secure-Programs-HOWTO/history.html>