

# Sistemas Operativos Linux, sus Componentes y su Aplicación en la Vida Cotidiana

Brandon Sánchez Porras

*Escuela de Ingeniería en Sistemas de  
Computación  
Universidad Fidélitas  
Heredia, Costa Rica  
bsanchez20389@ufide.ac.cr*

Jason Solís Mata

*Escuela de Ingeniería en Sistemas de  
Computación  
Universidad Fidélitas  
Heredia, Costa Rica  
jsolis80423@ufide.ac.cr*

Isaac Hidalgo Vargas

*Escuela de Ingeniería en Sistemas de  
Computación  
Universidad Fidélitas  
Heredia, Costa Rica  
ihidalgo80219@ufide.ac.cr*

Sebastián Valverde Hernandez

*Escuela de Ingeniería en Sistemas de Computación  
Universidad Fidélitas  
Heredia, Costa Rica  
email@example.com*

**Abstract—** Este proyecto busca explicar el funcionamiento interno de los sistemas operativos basados en Linux y su aplicación en la vida cotidiana para difundir la importancia de los mismos y sus múltiples usos. Explicando la correcta instalación y el uso del SO en múltiples áreas como lo son los sistemas operativos embebidos, los servidores web y más popularmente; los dispositivos Android.

**Index terms—**Linux, Sistemas Operativos, Ingeniería, Informática, Servidores

## I. INTRODUCCIÓN

### A. Antecedentes del problema de investigación

El sistema operativo UNIX inició su desarrollo en 1969, a manos de Bell Labs de AT&T en los Estados Unidos [1]

Durante el desarrollo de UNIX AT&T estaba en una posición dominante como un monopolio en la industria de la telefonía en EEUU, el departamento de Justicia de EEUU declaró un decreto de consentimiento que prohibió que AT&T se involucrara en actividades comerciales fuera del campo de su servicio de telefonía. Por lo tanto, UNIX fue distribuido con código fuente a las universidades con una licencia académica que permitía el estudio, modificación y redistribución del código [2], lo cual facilitó la distribución del mismo y su crecimiento en popularidad entre instituciones académicas y negocios.

En 1984, AT&T se deshizo de sus compañías de operación regional, y fue liberada de su obligación de no entrar en el negocio de las computadoras, por lo que empezaron a vender UNIX como un producto propietario o de código cerrado, en

el que los usuarios no estaban legalmente autorizados a modificar su código [3].

Debido a la conversión de UNIX en un producto de código cerrado, se crearon 3 proyectos importantes; The GNU Project, The GNU General Public License (GNU GPL) y MINIX [4, 5, 6, 7].

The GNU Project, creado por Richard Stallman en 1983 es un sistema operativo inicialmente pensado para tener todas las utilidades para escribir y ejecutar programas en C, como por ejemplo un editor, un shell, un compilador, etc más un kernel, el cuál no tuvo mucho éxito. Además, buscaba tener compatibilidad con programas para UNIX [4], [5].

Minix fue un sistema operativo “UNIX-like” minimalista creado por Andrew S. Tanenbaum y lanzado en 1987 dirigido a estudiantes y otros que deseaban aprender principios de sistemas operativos [6]

Con esta información aclarada, podemos abordar la creación del kernel Linux.

En el otoño de 1990, Linus Torvalds, estudiante de la Universidad de Helsinki, tomó un curso de UNIX. En este otoño, Linus encontró un problema, el hardware de su universidad no podía soportar más de 16 usuarios a la vez, por lo que, en palabras de Linus, “Tenías que hacer fila para poder utilizar una terminal”. Debido a esto y a que leyó el libro requerido en su curso; “Operating Systems: Design and Implementation”, el cuál sirvió de guía para crear sistemas similares a UNIX. Al leer este libro, Linus decidió comprar su primera computadora personal y le instaló MINIX el cual modificó. Frustrado con la licencia de MINIX, que solo permitía su uso para propósitos educativos, decidió crear Linux, el cual fue lanzado en 1991. [8]

Actualmente, se crean distribuciones basadas en el kernel Linux como por ejemplo Ubuntu, Debian y Arch Linux.

### B. Justificación del problema de investigación

El propósito de este estudio es comprender, analizar y sintetizar información referente a los sistemas operativos basados en Linux y sus aplicaciones en la vida cotidiana con el fin de distribuir la misma y hacer conciencia sobre la importancia de los sistemas operativos basados en Linux, ya que el pensamiento del público general sobre el mismo sigue siendo atribuido a la programación o creación de código complejo, donde esto es algo incierto puesto que puede ser utilizado como cualquier otro sistema tanto para uso diario como profesional. Según datos de la web StatCounter [9], para enero del presente año 2023, un 1.15% de los dispositivos en el mundo son regidos por Linux, encabezando esta lista con un 43.01% de dispositivos administrados por Android como sistema operativo como se puede apreciar en Figure 1

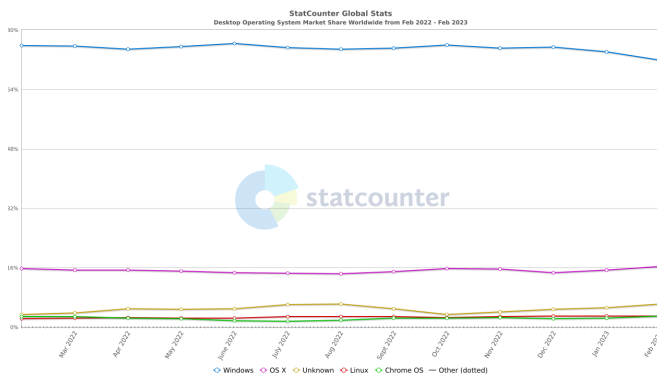


Figure 1: Gráfico del uso de sistemas operativos en computadores de escritorio

### C. Formulación del problema de investigación

¿De qué manera puede influir el conocimiento relacionado a los sistemas operativos poco comunes como lo puede ser Linux en la población general acostumbrada a los más comerciales?

### D. Objetivo General

Incentivar a los usuarios de dispositivos tecnológicos sobre la importancia y conocimiento de lo complejos que pueden llegar a ser los Sistemas operativos y sus diferentes funcionalidades en la vida cotidiana, enfocándose específicamente en los sistemas operativos basados en GNU+Linux.

### E. Objetivos Específicos

- Explicar como puede ser instalado Linux en un dispositivo inteligente.
- Instruir sobre el funcionamiento de los sistemas operativos basados en Linux.
- Aclarar qué ventajas y desventajas conlleva la utilización de sistemas operativos complejos a simple vista.

### F. Hipótesis

Se piensa que este tema será de fundamental ayuda explicativa para aquellos estudiantes a quienes interese este tema por el hecho de que el tema trata de enfocar y especificar el uso e historia de los sistemas operativos basados en Linux.

## II. MARCO TEÓRICO

El proyecto busca informar a los usuarios de dispositivos tecnológicos sobre la complejidad y usabilidad de los sistemas operativos, enfocándose específicamente en los sistemas operativos basados en GNU+Linux. Por lo tanto, es necesario definir varios conceptos de suma importancia para el entendimiento de los sistemas operativos y los sistemas operativos basados en GNU+Linux.

A continuación se exploran múltiples conceptos importantes para el propósito de esta investigación:

### A. Software

El software es la parte lógica del ordenador, las instrucciones que el hardware ejecuta para completar las tareas que necesitamos que realice.

El software es también lo que conocemos como programas o aplicaciones que son como secuencias de instrucciones (órdenes) que debe realizar el ordenador, algo así como una receta de cocina con todos sus pasos especificados [10].

Dentro de las múltiples clasificaciones del software, las más relevantes para esta investigación son las detalladas a continuación:

1) *Software de Sistema*: Es aquel que aporta herramientas y utilidades de apoyo para el mantenimiento del equipo, entre los que se encuentran el sistema operativo, los controladores de dispositivos (pequeños programas que permiten la conexión satisfactoria del ordenador con dispositivos de entrada y salida), etc [10].

2) *Software de Programación*: Conjunto de programas que permiten a los desarrolladores la creación de estos, haciendo uso de diferentes lenguajes (como lo son C, C++, C#, F#, Haskell, Python, etc.). Entre ellos, destacan los depuradores y enlazadores, entre otros [10].

### B. Sistemas Operativos

Un sistema operativo es un software que administra los procesos entre el Hardware y Software de un equipo informático y proporciona servicios a las aplicaciones que se ejecutan en el, cumpliendo una función como intermediario entre el Hardware y las diferentes aplicaciones que se ejecutan en el mismo.

A lo largo del tiempo los sistemas operativos han evolucionado y adaptándose a nuevas tecnologías, desde los primeros basados en cinta perforada, hasta los más modernos

a día de hoy como los pueden ser Windows, MacOS yLinux, contando cada uno con características y ventajas propias permitiendo a los usuarios optar por el que más se ajuste a sus necesidades.

Los sistemas operativos pueden dividirse en distintas categorías, como por ejemplo según el tipo de arquitectura que posea, existen los Sistemas operativos de un solo usuario diseñados para ser utilizados por una sola persona con un equipo informático personal, como lo pueden ser Windows, MacOS o ChromeOS. Luego están los Sistemas operativos de varios usuarios que permiten a un grupo de personas utilizar un mismo computador al mismo tiempo para que los equipos de trabajo puedan desarrollar proyectos de manera más ordenada y controlada. Por último los Sistemas operativos de tiempo real fueron diseñados para controlar aplicaciones que requieran de una respuesta rápida y concisa a eventos en tiempo real como por ejemplo lectura de sensores, activación de dispositivos o comunicación con otros sistemas.

Los sistemas operativos también pueden dividirse en categorías según el tipo de código fuente que tengan, como lo pueden ser los sistemas operativos de código cerrado los cuales son desarrollados y mantenidos por una empresa y bloqueando al público el acceso a su código fuente, esto quiere decir que los usuarios no pueden manipular o personalizarlo según sus preferencias, en su lugar, dependen de las actualizaciones proporcionadas por la compañía propietaria de dicho sistema. Los sistemas operativos de código abierto son aquellos cuyo código fuente está disponible públicamente y puede ser utilizado y modificado libremente por el usuario adaptando el software a sus necesidades, generalmente suelen ser basados en una comunidad de desarrolladores y usuarios que colaboran en mejorar el software, suelen ser gratuitos, por ejemplo Linux, FreeBSD o Ubuntu [11].

### C. Componentes de un sistema operativo GNU+Linux

Un sistema operativo GNU/Linux suele presentarse en diversas distribuciones pero en general se conforma principalmente por el kernel de Linux más un conjunto de programas y herramientas para el manejo de archivos, administración de usuarios y gestores de paquetes, entre otros

1) *Kernel*: El *Kernel* de Linux, del alemán que significa núcleo, es la parte fundamental del sistema operativo, desarrollado por Linus Tovalds, puede administrar, brindar una interfaz de usuario y contener la lista de todos los dispositivos conectados. Un sistema operativo con núcleo monolítico, concentra todas las funcionalidades posibles (planificación, sistema de archivos, redes, controladores de dispositivos, gestión de memoria, etc) dentro de un gran programa. El mismo puede tener un tamaño considerable, y deberá ser re-compilado por completo al añadir una nueva funcionalidad. Todos los componentes funcionales del núcleo tienen acceso a todas sus estructuras de datos internas y a sus rutinas. Un error en una

rutina puede propagarse a todo el núcleo. Todos sus componentes se encuentran integrados en un único programa que ejecuta en un único espacio de direcciones. En este tipo de sistemas, todas las funciones que ofrece el sistema operativo se ejecutan en modo supervisor.

El hecho de que Linux no fuera desarrollado siguiendo el diseño de un micronúcleo (diseño que, en aquella época, era considerado el más apropiado para un núcleo por muchos teóricos informáticos) fue asunto de una famosa y acalorada discusión entre Linus Torvalds y Andy Tanenbaum.

A diferencia de los núcleos monolíticos tradicionales, los micronúcleos tienen la cualidad de que pueden cargar los controladores y extensiones al núcleo de forma más sencilla, sin importar que el sistema esté funcionando.

Actualmente Linux es un núcleo monolítico híbrido. Los controladores de dispositivos y las extensiones del núcleo normalmente se ejecutan en un espacio privilegiado conocido como anillo 0, con acceso no restringido al hardware, aunque algunos se ejecutan en espacio de usuario. A diferencia de los núcleos monolíticos tradicionales, los controladores de dispositivos y las extensiones al sistema operativo se pueden cargar y descargar fácilmente como módulos, mientras el sistema continúa funcionando sin interrupciones.

También, a diferencia de los núcleos monolíticos tradicionales, los controladores pueden ser revocados (detenidos momentáneamente por actividades más importantes) bajo ciertas condiciones. Esta habilidad fue agregada para gestionar correctamente interrupciones de hardware, y para mejorar el soporte de Multiprocesamiento Simétrico [12].

2) *Gestores de paquetes*: Los gestores de paquetes se encargan de integrar al equipo el software necesario para el usuario, pues se dedican a actualizar sus repositorios (paginas web donde pueden descargar los paquetes), descargar los paquetes (conjunto de archivos que además de tener programas también contienen manuales, archivos de configuración para el gestor y dependencias) e integrar sus dependencias (son conjunto de programas u otros paquetes necesarios para el óptimo funcionamiento del paquete de interés) [12].

3) *File System o Sistema de Archivos*: En Linux (como en cualquier Sistema Operativo) se puede diferenciar el sistema de archivos que está almacenado en las particiones del disco y otros dispositivos, y el sistema de archivos tal y como lo presenta al usuario un equipo ejecutando Linux como software de base. En Linux se monta un sistema de archivos de disco en el sistema de archivos del equipo en funcionamiento. Un disco duro no es más que un gran espacio sobre el que se puede escribir unos y ceros. Un sistema de archivos impone cierta estructura en ello, y hace que parezca que hay archivos dentro de directorios, y directorios dentro de directorios. Cada archivo se representa por un i-nodo, que contiene información de a quién pertenece el archivo, cuándo se creó

y dónde encontrar su contenido. Los directorios también se representan por medio de i-nodos, pero estos contienen información sobre dónde encontrar los i-nodos de los archivos que están en el directorio. Por ejemplo, si el sistema quiere leer /home/maría/imagen1.jpg primero busca el i-nodo del directorio raíz "/" , luego busca el i-nodo del directorio home en el contenido de "/", luego busca el i-nodo del directorio maría en el contenido de /home, luego el i-nodo de imagen1.jpg que le dirá qué bloques del disco leer. La facultad que tienen los directorios de poder contener, a su vez, otros directorios (denominados subdirectorios) permite una organización o estructura jerárquica con la forma de un árbol invertido.

El directorio inicial de esa jerarquía se denomina directorio raíz y se simboliza con una barra de división (/).

En Linux la palabra raíz se utiliza para denominar dos conceptos. Uno es el usuario raíz (user root) la persona con control absoluto en la administración del sistema. El otro es el concepto de directorio raíz: un directorio específico que se considera la base del sistema de archivos. Cada file system pertenece a una clase o tipo que define la característica administrativa. Así como Windows 9x, usa FAT, y Windows NT hace lo suyo con NTFS, el LINUX en particular adopta el sistema de archivo Ext2 (Más actualmente, ext4, btrfs, zfs, entre otros) . Aparte con este sistema de archivos Linux puede manejar los antiguos minix y ext y otros sistemas no propios de Linux como los sistemas Fat de MS-DOS y Windows y el sistema iso9660 de los CD-Rom [13].

4) *Core Utils o Utilidades principales*: Las utilidades principales de GNU son las utilidades básicas de manipulación de archivos, texto y shell, provenientes del sistema operativo GNU, el cuál fué mencionado en los antecedentes del problema. Se espera que estas utilidades existan en todos los sistemas operativos. Algunas de las coreutils más comúnmente utilizadas incluyen:

- chmod - cambiar los modos/permisos de un archivo
- chown - cambiar el usuario/grupo dueño del archivo o carpeta
- chroot - Cambia el directorio raíz a utilizar
- cp - copiar archivos y carpetas
- dd - copiar y convertir archivos
- df - muestra el espacio libre en sistemas de archivos
- du - muestra el uso de almacenamiento en sistemas de archivos
- ln - crea enlaces de archivos
- ls - muestra una lista con los contenidos del directorio actual (cwd o current working directory)
- mkdir - crea directorios/carpetas
- mv - mueve o renombra archivos
- rm - eliminar archivos o carpetas
- rmdir - elimina carpetas vacías
- touch - crear archivos vacíos [14]

Estas herramientas son vitales para el uso del sistema operativo, ya que sin ellas, no se podría llevar a cabo las funciones más básicas del mismo, como lo es crear archivos y manipularlos.

5) *Shell o interfaz de linea de comandos*: La interfaz con el sistema operativo se denomina shell.

El shell es la capa más externa del sistema operativo. Los shells incorporan un lenguaje de programación para controlar procesos y archivos, además de iniciar y controlar otros programas. El shell gestiona la interacción entre el usuario y el sistema operativo solicitando la entrada, interpretando dicha entrada para el sistema operativo y gestionando cualquier resultado de salida procedente del sistema operativo.

Los shells ofrecen un método para comunicarse con el sistema operativo. Esta comunicación tiene lugar de forma interactiva (la entrada desde el teclado se ejecuta inmediatamente) o como un script de shell. Un *script de shell* es una secuencia de mandatos del shell y del sistema operativo que se almacena en un archivo.

Cuando se inicie la sesión en el sistema, éste localiza el nombre de un programa de shell a ejecutar. Después de ejecutarse, el shell visualiza un indicador de mandatos. Este indicador suele ser un \$ (signo de dólar). Cuando escriba un mandato en el indicador y pulse la tecla Intro, el shell evaluará el mandato e intentará ejecutarlo. Según las instrucciones del mandato, el shell grabará la salida del mandato en la pantalla o lo dirigirá a la salida. A continuación, restaura el indicador de mandatos para que escriba otro mandato.

Una línea de mandatos es la línea en la que escribe. Contiene el indicador del shell. El formato básico de cada línea es el siguiente:

```
$ mandato argumento(s)
```

El shell considera que la primera palabra de una línea de mandatos (hasta el primer espacio en blanco) es el primer mandato y que todas las palabras posteriores son argumentos. [15]

Por ejemplo:

```
$ cd ~/Downloads
# ^ ^
# | |
# mandato argumento
```

El código contenido en el bloque anterior utiliza el o comando o mandato "cd" (*change directory* o "cambiar directorio" en español) y este mandato toma el argumento " ~/Downloads", el cual significa "/home/user/Downloads", o más comúnmente, el directorio de Descargas.

### III. MARCO METODOLÓGICO

#### A. Tipo de investigación

El modelo de estudio de la presente investigación acerca de los sistemas operativos Linux es de tipo exploratorio e investigativo, ya que el enfoque al cual queremos llevar, es una introducción completa a las funciones y características de dicho sistema operativo, para que nuevos usuarios sin ninguna experiencia puedan comprender la utilidad de sistemas operativos alternativos, para que así, los usuarios tengan un abanico de posibilidades al momento de adquirir un equipo informático.

#### B. Población o Muestra

Para este proyecto hemos decidido no llevar a cabo ningún tipo de censo o muestreo probabilístico, ya que el propósito es explorar y sintetizar conceptos ya antes explorados por otras personas, con el fin de educar a las personas con menos conocimiento al respecto

#### C. Descripción de los instrumentos

- Se utilizará una computadora o máquina virtual para la instalación de una distribución de Linux minimalista, con el propósito de explicar el uso de cada uno de los programas de manera individual.
- Se utilizará la distribución de Linux ArchLinux, disponible en <https://archlinux.org> debido a su extremo minimalismo, el cual nos permite explicar paso por paso la instalación de la misma, y el uso de cada componente.
- Se hará uso de la wiki de instalación de ArchLinux, que es escrita y actualizada por la comunidad de la distribución, disponible en <https://wiki.archlinux.org>.

#### D. Tratamiento de la información

Para la ejemplificación de la información se realizará una instalación de linux en un dispositivo físico o virtual, con el propósito de explicar el uso de la mayoría de elementos instalados y llegar a 2 fases importantes:

1. Sistema operativo listo para uso en servidores.
2. Sistema operativo listo para uso diario por un usuario final.

Con esto se quiere mostrar a la población dirigida, los beneficios o accesibilidades que ofrecen los sistemas operativos basados en Linux así como también el conocimiento sobre cómo estos son instalados, bajo qué requerimientos y en específico la manera adecuada y más segura de instalarlos.

## IV. RESULT ANALYSIS

## V. RECOMMENDATIONS AND CONCLUSIONS

### REFERENCES

- [1] S. Talens-Oliag, "Breve historia de Linux y el movimiento del Software Libre," Debian, 2004. (<https://people.debian.org/~sto/BHLSL/BHLSL.pdf>)
- [2] H. Meeker, "Open source licensing: What every technologist should know," 2017. (<https://opensource.com/article/17/9/open-source-licensing>)
- [3] C. J. Tozzi, *For Fun and Profit: A History of the Free and Open Source Software Revolution*, Cambridge, MA: The MIT Press, 2017.
- [4] R. Stallman, "Initial Announcement," 1983. (<https://www.gnu.org/gnu/initial-announcement.html>)
- [5] A. (-C. Tai, "The History of the GPL," 2001. ([https://www.free-soft.org/gpl\\_history/](https://www.free-soft.org/gpl_history/))
- [6] A. Tanenbaum, "Some Notes on the "Who wrote Linux" Kerfuffle, Release 1.5," 2004. (<https://web.archive.org/web/20100818183310/http://www.cs.vu.nl/~ast/brown/>)
- [7] C. Tozzi, "Open Source History: Why Did Linux Succeed?," 2016. (<https://web.archive.org/web/20170817205211/http://thevarguy.com/open-source-application-software-companies/050415/open-source-history-why-did-linux-succeed>)
- [8] G. Moody, "The Greatest OS That (N)ever Was," 1997. (<https://www.wired.com/1997/08/linux-5/>)
- [9] S. GlobalStats, "Operating System Market Share Worldwide," 2022. (<https://gs.statcounter.com/os-market-share>)
- [10] Gabriel Carmona Romera, *Sistema Operativo, Búsqueda De Información Internet/intranet Y Correo Electronico*, IC Editorial, 2021. (OCLC: 1339055519)
- [11] N. J. Cura, *Fundamentos De Sistemas Operativos*, Córdoba: Jorge Sarmiento Editor - Universitas, 2020. (OCLC: 1283844134)
- [12] N. de Anda, "Componentes primordiales de GNU/Linux," 2018. (<https://www.factor.mx/portal/base-de-conocimiento/componentes-gnu-linux/>)
- [13] M. Serna, *Sistemas Operativos*, Córdoba: Jorge Sarmiento Editor - Universitas, 2020. (OCLC: 1283843398)
- [14] D. Wiki, "Coreutils," *Debian Wiki*, May 2020.
- [15] \. I. , "Shells del sistema operativo," 2021. (<https://www.ibm.com/docs/es/aix/7.2?topic=administration-operating-system-shells>)