

Diseño interno y organización de sistemas operativos: Un estudio comparativo

Internal Design and organization of operating systems: A comparative study

Diego Fernando Marín Sisa

Escuela de Ingeniería de Sistemas e Informática

Universidad Industrial de Santander

diego2204005@correo.uis.edu.co

Johan Sebastián León Peñaloza

Escuela de Ingeniería de Sistemas e Informática

Universidad Industrial de Santander

johan2202037@correo.uis.edu.co

Resumen - En este estudio comparativo de sistemas operativos (SO), se analizan Linux, macOS y Windows desde múltiples perspectivas clave. Linux, conocido por su robustez y configurabilidad debido a su naturaleza de código abierto y características como SELinux, destaca en seguridad y eficiencia en entornos de servidor. Windows ofrece una fuerte compatibilidad de hardware y herramientas integradas como Windows Defender para seguridad, mientras que macOS prioriza la integración del sistema en dispositivos Apple y características de cifrado avanzadas. En términos de rendimiento, Clear Linux y sus distribuciones lidera en tareas intensivas como compresión y descompresión, seguido por Ubuntu que ofrece competencia sólida frente a macOS y Windows en diversas métricas. La elección del SO adecuado depende de las necesidades específicas, ya sea para seguridad, rendimiento o compatibilidad, cada uno con fortalezas distintas que se adaptan a diferentes contextos de uso.

Abstract - In this comparative study of operating systems (OS), Linux, macOS, and Windows are analyzed across multiple critical perspectives. Linux, renowned for its robustness and configurability due to its open-source nature and features like SELinux, excels in security and efficiency in server environments. Windows offers strong hardware compatibility and integrated tools like Windows Defender for security, while macOS prioritizes system integration on Apple devices and advanced encryption features. In terms of performance, Clear Linux leads in intensive tasks such as compression and decompression, followed closely by Ubuntu, which shows strong competition against macOS and Windows across various metrics. Choosing the right OS depends on specific needs, whether for security, performance, or compatibility, each with distinct strengths tailored to different usage contexts.

I. INTRODUCCIÓN

Los sistemas operativos son la columna vertebral de cualquier dispositivo informático. Desde las computadoras personales hasta los dispositivos móviles, dependemos de ellos para gestionar recursos, ejecutar aplicaciones y proporcionar una interfaz entre el usuario y el hardware. Sin embargo, lo que hace que los sistemas operativos sean verdaderamente fascinantes es su diversidad y complejidad.

En palabras de Silberschatz, “Un sistema operativo es un programa que administra el hardware de una computadora. También proporciona las bases de los programas de aplicación y actúa como intermediario entre el usuario y el hardware de la computadora” [1, p.3]. Esta definición resalta la importancia de los sistemas operativos como mediadores entre nosotros y la tecnología subyacente. Pero ¿cómo logran esta tarea tan crucial?

La respuesta es sorprendentemente variada. Existen múltiples enfoques, estructuras y filosofías detrás de los sistemas operativos. Cada uno tiene características, ventajas y desventajas. Así que esta investigación se centra en explorar la organización

interna de los sistemas operativos, reconociendo que cada uno tiene su identidad y forma de funcionar.

El objetivo de esta investigación es analizar la organización interna y el diseño de los sistemas operativos enfocándose en sus arquitecturas, gestión de recursos y experiencias de usuario, para identificar sus ventajas y desventajas.

Los sistemas operativos de referencia a evaluar son los más populares, empezando por el más flexible de estos tres: Linux, continuando con la versión minimalista de Apple: MacOS, y el más popular entre los usuarios de computadoras, el producto de Microsoft: Windows. Los siguientes sistemas para evaluar se describen más detalladamente:

- ❖ **Linux:** Un kernel de código abierto que ha evolucionado desde sus inicios como un kernel monolítico hasta una estructura más similar a un microkernel. Se investigará su estabilidad, seguridad y robustez, así como su presencia dominante en servidores, sistemas embebidos y dispositivos móviles.
- ❖ **macOS:** Desarrollado por Apple, está enfocado en el ambiente de dispositivos Mac. Su diseño elegante, interfaz intuitiva y experiencia de usuario cohesiva son sus principales características. Se analizará cómo macOS se organiza internamente y cómo se diferencia de otros sistemas operativos.
- ❖ **Windows:** El sistema operativo más popular entre usuarios de computadoras personales ofrece una amplia gama de aplicaciones y una interfaz familiar. Se examinará su estructura, componentes clave y su papel en la comunicación entre el usuario y el hardware.

II. ESTADO DEL ARTE

Los sistemas operativos son la base de las computadoras, se han desarrollado y perfeccionado con el tiempo. Este estado del arte revisa los avances y los desafíos en la organización interna de los cuatro sistemas operativos mencionados anteriormente.

En los años 90, Windows dominaba el mercado de sistemas operativos para PC con versiones como Windows 95, 98 y NT. Eran sistemas con una interfaz gráfica amigable, pero con problemas de estabilidad y seguridad introduciendo soporte para aplicaciones nativas de 32 bits y mejoras en la interfaz de usuario. En esa misma década Linux emergió como un sistema operativo de código abierto con gran estabilidad y rendimiento, pero con una interfaz de usuario compleja con versiones como Slackware y Debian. Y que para el sistema operativo de Apple el cual se llamaba Mac OS, era conocido por su interfaz gráfica innovadora pero limitado a equipos Macintosh. Mac OS 7 y 8 se lanzaron en esta época, mejorando la estabilidad y rendimiento de los sistemas operativos de Apple.

En los años 2000, Windows XP y Vista mejoraron la estabilidad y seguridad, pero aún tenían problemas de rendimiento y compatibilidad con hardware. Para el caso de Linux inició con distribuciones como Ubuntu, Fedora y Debian popularizaron Linux con interfaces más amigables, manteniendo su fortaleza en servidores y aplicaciones de alto rendimiento. Sin embargo, Mac OS X (ahora macOS) se basó en Unix, mejorando la estabilidad y rendimiento, mientras mantenía la interfaz característica de Apple.

En la década de 2010, Windows 7, 8 y 10 lograron mayor estabilidad, rendimiento y seguridad, convirtiéndose en los sistemas operativos más utilizados en PCs. Durante este tiempo Linux consolidó su presencia en servidores, supercomputadoras y dispositivos embebidos, con mejoras en compatibilidad de hardware y aplicaciones. Las distribuciones de Linux como Ubuntu y Fedora continuaron mejorando y expandiendo su presencia en servidores y dispositivos embebidos. Luego macOS se consolidó como un sistema operativo de alta calidad, con actualizaciones anuales que mejoraban la integración con dispositivos Apple y aplicaciones nativas.

Actualmente, Windows 10 y 11 son las versiones más avanzadas, con mejoras en la interfaz, integración con servicios en la nube y rendimiento optimizado para hardware moderno. En continuación al desarrollo que ha ido evolucionando durante algunas décadas Linux sigue siendo un sistema operativo muy utilizado en servidores, supercomputadoras y dispositivos IoT, con un ecosistema en constante evolución. Las distribuciones de Linux como Ubuntu y Fedora continuaron mejorando y expandiendo su presencia en servidores, dispositivos embebidos y dispositivos móviles. Para el caso de macOS Ventura es la versión más reciente, con mejoras en seguridad, productividad y experiencia de usuario en equipos Mac.

III. MARCO TEÓRICO

El marco teórico de esta investigación se centra en la comprensión profunda de los sistemas operativos, que son el núcleo de cualquier dispositivo computacional. Estos sistemas, que gestionan los recursos del hardware y proporcionan una interfaz entre el usuario y la máquina, han evolucionado significativamente desde sus inicios. A través de una revisión exhaustiva de la literatura existente y un análisis de los componentes y estructuras fundamentales de los sistemas operativos más populares, este marco teórico proporciona las bases necesarias para el estudio comparativo de Linux, macOS y Windows.

En primer lugar, se definirán los conceptos y funciones esenciales de los sistemas operativos, seguidos de una breve historia de su evolución. Luego, se explorarán los componentes críticos que constituyen un sistema operativo, como el kernel, la gestión de memoria y procesos, y los mecanismos de seguridad. Posteriormente, se detallarán las características específicas de los sistemas operativos seleccionados para este estudio, analizando sus arquitecturas y enfoques de diseño.

1. Definición de Sistemas Operativos

Un sistema operativo (SO) es el software principal que gestiona el hardware y el software de una computadora. Actúa como un intermediario entre los usuarios y el hardware de la computadora, facilitando la comunicación y la ejecución de programas. Sin un sistema operativo, los usuarios no podrían interactuar de manera eficiente con los dispositivos electrónicos.

Concepto y funciones básicas

❖ Gestión de Procesos:

- **Planificación:** Asigna tiempo de CPU a los procesos.
- **Creación y terminación de procesos:** Crea y finaliza procesos según sea necesario.

❖ Sincronización y comunicación:

Asegura que los procesos trabajen juntos sin interferencias.

❖ Gestión de Memoria

- **Asignación y liberación de memoria:** Controla qué parte de la memoria está en uso y qué parte está libre.
- **Paginación y segmentación:** Maneja la memoria virtual y física para optimizar el uso de recursos.

❖ Gestión de Dispositivos de Entrada/Salida

- **Controladores de dispositivos:** Interactúa con el hardware a través de controladores específicos.
- **Administración de dispositivos:** Coordina y controla el uso de dispositivos de entrada y salida.

❖ Gestión de Archivos

- **Sistema de archivos:** Organiza y almacena datos en el disco.
- **Operaciones de archivos:** Proporciona mecanismos para crear, leer, escribir y eliminar archivos.

❖ Seguridad y Protección

- **Control de acceso:** Limita el acceso a los recursos del sistema a usuarios autorizados.
- **Autenticación:** Verifica la identidad de los usuarios.
- **Protección de recursos:** Evita que procesos no autorizados accedan o modifiquen recursos críticos.

❖ Interfaz de Usuario

- **Interfaz gráfica de usuario (GUI):** Ofrece una manera visual de interactuar con el sistema operativo.
- **Interfaz de línea de comandos (CLI):** Permite a los usuarios introducir comandos para realizar tareas específicas.

❖ Gestión de Recursos

- **Asignación de recursos:** Distribuye recursos del sistema, como CPU, memoria y dispositivos de E/S.
- **Monitorización de recursos:** Rastrea el uso de recursos para optimizar el rendimiento del sistema.

2. Historia y Evolución de los Sistemas Operativos

Primera Generación (1940 a principios de 1950): Las primeras computadoras no incluían sistemas operativos, y los usuarios escribían programas en lenguaje de máquina para realizar cálculos básicos.

Segunda Generación (1955 – 1965): Apareció el primer sistema operativo, GMOSIS, desarrollado por General Motors para computadoras IBM, basado en procesamiento por lotes.

Tercera Generación (1965 – 1980): Se introdujo la multiprogramación, permitiendo la ejecución simultánea de múltiples tareas, y aparecieron los mainframes y minicomputadoras como el DEC PDP-1.

Cuarta Generación (1980 – Presente): Marcada por la evolución de las computadoras personales y el desarrollo del sistema operativo Windows por Microsoft, junto con el uso continuado de minicomputadoras y la popularidad de sistemas operativos como Windows 95, 98, XP, 7 y 10, y la relevancia de Apple.

• No OS – (Años 0 a 1940)

Como sabemos, antes de los años 1940 no se usaban sistemas operativos. En esa época, las personas no tenían un sistema operativo en sus computadoras, por lo que debían escribir manualmente las instrucciones para cada tarea en lenguaje de máquina (lenguaje basado en 0-1). En ese entonces, era muy difícil para los usuarios implementar incluso una tarea simple. Era muy lento y nada amigable, ya que no todos tenían el nivel de comprensión necesario para entender el lenguaje de máquina, que requería un entendimiento profundo.

• Sistemas de Procesamiento por Lotes - (1940s a 1950s)

Con el paso del tiempo, los sistemas de procesamiento por lotes llegaron al mercado. Los usuarios podían escribir sus programas en tarjetas perforadas y cargarlas al operador de la computadora. Luego, el operador agrupaba diferentes lotes de trabajos similares y los servía uno por uno a la CPU. La CPU primero ejecutaba los trabajos de un lote y luego pasaba a los trabajos de otro lote en secuencia.

• Sistemas Multiprogramados - (1950s a 1960s)

La multiprogramación fue el primer sistema operativo donde realmente comenzó la revolución. Permitía a los usuarios cargar múltiples programas en la memoria y asignar una porción

específica de memoria a cada programa. Cuando un programa estaba esperando alguna operación de E/S (que tomaba mucho tiempo), el SO permitía a la CPU cambiar del programa anterior a otro programa (el primero en la cola de listos) para la ejecución continua de los programas con interrupciones.

• Sistemas de Tiempo Compartido - (1960s a 1970s)

Los sistemas de tiempo compartido son una versión extendida de los sistemas multiprogramados. Aquí se añadió una característica extra para evitar el uso de la CPU durante mucho tiempo y dar acceso a la CPU a todos los programas tras un intervalo de tiempo determinado. Básicamente, el SO cambia de un programa a otro después de un intervalo de tiempo determinado para que todos los programas puedan acceder a la CPU y completar su trabajo.

• Introducción de la GUI - (1970s a 1980s)

Con el paso del tiempo, llegaron las Interfaces Gráficas de Usuario (GUIs). Por primera vez, los sistemas operativos se volvieron más amigables y cambiaron la forma en que las personas interactuaban con las computadoras. La GUI proporcionó elementos visuales al sistema informático, lo que hizo que la interacción del usuario con la computadora fuera más cómoda y amigable. El usuario podía clicar en elementos visuales en vez de escribir comandos. Algunas características de la GUI en Windows de Microsoft incluyen íconos, menús y ventanas.

• Sistemas en Red - (1980s a 1990s)

En los años 1980, la popularidad de las redes de computadoras alcanzó su punto máximo. Se necesitaba un tipo especial de sistemas operativos para gestionar la comunicación en red. Sistemas operativos como Novell NetWare y Windows NT fueron desarrollados para gestionar la comunicación en red, proporcionando a los usuarios la facilidad de trabajar en un entorno colaborativo y facilitando el intercambio de archivos y el acceso remoto.

• Sistemas Operativos Móviles - (Finales de 1990s a Principios de 2000s)

La invención de los smartphones creó una gran revolución en la industria del software. Para manejar las operaciones de los smartphones, se desarrollaron tipos especiales de sistemas operativos. Algunos de ellos son: iOS y Android, etc. Estos sistemas operativos fueron optimizados con el tiempo y se volvieron más poderosos.

• Integración de la IA - (2010s a la actualidad)

Con el paso del tiempo, la inteligencia artificial entró en escena. Los sistemas operativos integraron características de la tecnología de IA como Siri, Google Assistant y Alexa, y se volvieron más poderosos y eficientes de muchas maneras. Estas características de IA con el sistema operativo crean

características completamente nuevas como comandos de voz, texto predictivo y recomendaciones personalizadas.

3. Componentes Fundamentales de un Sistema Operativo

a. KERNEL

El kernel es el componente central y más fundamental de un sistema operativo, que actúa como un intermediario entre el hardware y el software. Tiene la responsabilidad de gestionar y coordinar todos los recursos del sistema, como la memoria, los procesos, los dispositivos de entrada/salida y la seguridad, para proporcionar un entorno de ejecución estable y eficiente a los programas de usuario. El kernel es el primer programa que se carga en la memoria cuando se inicia el sistema operativo y permanece en ejecución hasta que se apaga el sistema. Tiene un acceso directo y privilegiado al hardware, lo que le permite controlar y gestionar todos los componentes del sistema, como la CPU, la memoria RAM, los dispositivos de almacenamiento, las interfaces de red, etc. Una de las principales funciones del kernel es la gestión de la memoria. Asigna y libera espacio de memoria para los diferentes procesos, evitando conflictos y garantizando que cada proceso tenga acceso a la memoria que necesita. También se encarga de la paginación y la segmentación de la memoria, lo que permite a los procesos utilizar más memoria de la que está físicamente disponible en el sistema.

Otra función crucial del kernel es la gestión de procesos. Determina qué procesos pueden utilizar la CPU, cuándo y durante cuánto tiempo. Coordina la ejecución de múltiples procesos de manera concurrente, asegurando que cada proceso reciba el tiempo de CPU que necesita y evitando que un proceso monopolice el procesador. El kernel también actúa como un intermediario entre el software y el hardware, proporcionando una interfaz de llamadas al sistema (SCI) a través de la cual los procesos pueden solicitar servicios al kernel, como abrir archivos, enviar datos a través de la red, o acceder a dispositivos de entrada/salida. El kernel procesa estas solicitudes, interactúa con el hardware correspondiente y devuelve los resultados a los procesos. Además, el kernel se encarga de la seguridad del sistema, controlando los permisos de acceso a los recursos del sistema por parte de los procesos. Esto evita que los procesos accedan a áreas de memoria o dispositivos a los que no deberían tener acceso, lo que podría comprometer la integridad y la estabilidad del sistema. El kernel es el componente central de un sistema operativo que actúa como puente entre el hardware y el software. Existen diferentes tipos de kernel con distintas arquitecturas:

Kernel monolítico: Es un kernel grande que se encarga de todas las tareas fundamentales como la gestión de memoria, administración de procesos y coordinación de operaciones de entrada/salida. Sistemas operativos como Linux, macOS y Windows utilizan este tipo de kernel.

Microkernel: Es un kernel diseñado intencionalmente de un tamaño pequeño para evitar que un fallo paralice todo el sistema operativo. Para asumir las mismas funciones que un kernel monolítico, el microkernel se divide en varios módulos. Hasta ahora, solo el componente Mach de macOS utiliza esta arquitectura de microkernel.

Kernel híbrido: Es una combinación del kernel monolítico y el microkernel. El kernel es más compacto y modular, permitiendo cargar dinámicamente otras partes del kernel. Sistemas como Linux y macOS utilizan este tipo de arquitectura híbrida.

❖ Funciones del Kernel

- **Gestión de la memoria:** El kernel supervisa cuánta memoria se utiliza y dónde se almacena cada elemento. Asigna y libera memoria de manera eficiente para los diferentes procesos.
- **Gestión de procesos:** El kernel determina qué procesos pueden utilizar la CPU, cuándo y durante cuánto tiempo. Coordina la ejecución de múltiples procesos de manera concurrente.
- **Controladores de dispositivos:** El kernel actúa como un intermediario o intérprete entre el hardware y los procesos, permitiendo que el software se comunice con el hardware.
- **Seguridad y llamadas al sistema:** El kernel recibe y procesa las solicitudes de servicio de los procesos a través de una interfaz de llamadas al sistema (SCI). También se encarga de la seguridad, controlando los permisos de acceso a recursos.

b. SISTEMAS DE ARCHIVOS

El sistema de archivos es el subsistema del sistema operativo encargado de la gestión y organización de los datos almacenados en los dispositivos de memoria secundaria, como discos duros y unidades de estado sólido (SSD). Su principal objetivo es permitir al usuario acceder a los archivos de manera rápida y sencilla, independientemente de su ubicación física en el dispositivo de almacenamiento. El sistema de archivos define cómo se almacenan los datos, cómo se accede a ellos y cómo están organizados en el dispositivo de almacenamiento. Utiliza una estructura de árbol que parte de un directorio raíz, desde el cual se

ramifican el resto de las carpetas, subcarpetas y archivos. Las principales funciones del sistema de archivos son:

- Crear, borrar y modificar archivos
- Permitir el acceso a los archivos para lectura y escritura
- Automatizar la gestión de la memoria secundaria
- Permitir referenciar un archivo por su nombre simbólico
- Proteger los archivos frente a fallos del sistema
- Permitir el uso compartido de archivos a usuarios autorizados

Existen diversos sistemas de archivos estándar para los diferentes sistemas operativos, como NTFS (Windows), ext (Linux), HFS+ (macOS) y FAT (compatible con varios sistemas). Aunque presentan similitudes, en general son incompatibles entre sí, a menos que se utilicen programas de terceros. Cada sistema de archivos tiene sus propias características, ventajas y desventajas. Por ejemplo, NTFS ofrece funciones avanzadas como permisos de seguridad, registro diario y compresión, mientras que FAT es más sencillo, pero carece de algunas de estas características. La integridad del sistema de archivos es crucial, ya que la destrucción de la información almacenada puede tener consecuencias catastróficas. Para proteger los datos, se utilizan mecanismos como la detección y gestión de sectores defectuosos en el disco, y la realización de copias de seguridad (backups) periódicas

La estructura del sistema de archivos es fundamental para la organización y gestión eficiente de los datos en un sistema operativo. Los principales aspectos de la estructura de los sistemas de archivos son:

- **Estructura Jerárquica de Directorios:** El sistema de archivos se organiza en una estructura de árbol invertido, con un directorio raíz "/" del que cuelgan subdirectorios y archivos. Esta estructura jerárquica permite una búsqueda eficiente de archivos y la agrupación lógica de los mismos. Los usuarios pueden acceder a los archivos mediante rutas de acceso absolutas o relativas al directorio de trabajo actual.
- **Tipos de Sistemas de Archivos:** Existen diversos sistemas de archivos estándar para los diferentes sistemas operativos, como NTFS (Windows), ext (Linux), HFS+ (macOS) y FAT (compatible con varios sistemas). Cada sistema de archivos tiene sus propias características, ventajas y desventajas en cuanto a funciones avanzadas, rendimiento, compatibilidad, etc.
- **Estructura Interna del Sistema de Archivos:** A nivel interno, el sistema de archivos se compone de estructuras como el superbloque, la tabla de i-nodos y los bloques de datos. Estas estructuras permiten al

kernel gestionar la asignación de memoria, los metadatos de los archivos y el acceso a los datos almacenados.

- **Gestión de Archivos:** El sistema de archivos proporciona operaciones básicas para crear, borrar, leer y escribir archivos, así como atributos como nombre, tipo, ubicación, tamaño y permisos de acceso. Además, permite el acceso concurrente a los archivos por parte de múltiples procesos y la protección de los datos ante fallos del sistema.

c. GESTIÓN DE PROCESOS

La gestión de procesos en sistemas operativos es el conjunto de técnicas y algoritmos utilizados por el sistema operativo para administrar y coordinar la ejecución de múltiples procesos de manera eficiente. Algunos aspectos clave de la gestión de procesos incluyen:

- **Planificación:** La planificación es un aspecto fundamental en la gestión de procesos en sistemas operativos. Algunas de las principales características de la planificación de procesos son:
 - **Objetivos de la planificación**
 - Decidir qué proceso ejecutar de entre los que están listos para ejecución
 - Lograr equidad, eficiencia, y minimizar el tiempo de respuesta y el tiempo de proceso global
 - Maximizar el número de trabajos procesados, aunque estos objetivos pueden ser contradictorios
 - **Algoritmos de planificación**
 - Planificación por turno rotatorio: asigna un intervalo de tiempo (cuanto) a cada proceso
 - Planificación por prioridad: ejecuta los procesos de mayor prioridad primero
 - Planificación a dos niveles: un planificador de alto nivel maneja procesos en memoria y disco, y uno de bajo nivel maneja la ejecución en memoria
 - **Sincronización y comunicación de procesos**
 - Es necesaria la sincronización cuando varios procesos acceden a recursos compartidos, para evitar problemas como condiciones de carrera

- Los procesos deben poder comunicarse e intercambiar información, a través de mecanismos como tuberías, colas de mensajes, memoria compartida, etc.
- **Sincronización:** La sincronización es un aspecto fundamental en la gestión de procesos en sistemas operativos. Permite coordinar la ejecución de múltiples procesos que acceden a recursos compartidos, evitando problemas como condiciones de carrera.
- **Mecanismos de sincronización**
Algunos de los principales mecanismos de sincronización utilizados en la gestión de procesos incluyen:
 - **Semáforos**
Los semáforos permiten controlar el acceso a recursos compartidos mediante un contador que indica cuántos procesos pueden acceder simultáneamente. Esto garantiza la exclusión mutua en secciones críticas.
 - **Monitores**
Los monitores son estructuras de datos que encapsulan variables compartidas y procedimientos de acceso, asegurando que solo un proceso pueda estar dentro del monitor a la vez.
 - **Variables condicionales**
Las variables condicionales permiten que un proceso se bloquee hasta que una cierta condición se cumpla, como que otro proceso libere un recurso. Esto facilita la sincronización entre procesos.
 - **Señales**
Las señales son mecanismos de comunicación asíncrona entre procesos que permiten que un proceso notifique a otro la ocurrencia de un evento relevante, facilitando su sincronización.
 - **Problemas de sincronización**
Algunos problemas comunes que pueden surgir en la sincronización de procesos incluyen:
 - **Interbloqueo (deadlock):** cuando dos o más procesos se bloquean mutuamente esperando recursos que nunca serán liberados.
 - **Inanición (starvation):** cuando un proceso nunca obtiene acceso a un

recurso compartido debido a que otros procesos lo acaparan continuamente.

- **Condición de carrera:** cuando el resultado de un cómputo depende del orden relativo de ejecución de los procesos.

Estos problemas deben ser cuidadosamente evitados mediante el diseño adecuado de los mecanismos de sincronización.

- **Comunicación entre procesos**

La comunicación entre procesos es un aspecto fundamental en la gestión de procesos en sistemas operativos. Algunas de las principales características de la comunicación entre procesos incluyen:

- **Mecanismos de comunicación**
 - Señales: Permiten que un proceso notifique a otro sobre la ocurrencia de un evento.
 - Tuberías (pipes): Permiten que un proceso hijo herede un canal de comunicación con su proceso padre, transfiriendo datos entre ellos.
 - Memoria compartida: Ofrece una forma rápida de comunicar datos entre procesos que han mapeado la misma región de memoria.
 - Colas de mensajes: Listas de mensajes almacenados en el kernel, que permiten la comunicación asíncrona entre procesos.
- **Tipos de comunicación**
 - Comunicación directa: Los procesos deben nombrarse explícitamente entre sí.
 - Comunicación indirecta: Los procesos se comunican a través de algo que comparten, como una mailbox.
- **Importancia de la comunicación**
 - Permite que los procesos cooperen y sincronicen sus acciones, evitando problemas como condiciones de carrera.
 - Facilita la modularidad, al permitir que procesos especializados se comuniquen entre sí.

d. GESTIÓN DE MEMORIA

La gestión de memoria es un aspecto fundamental en los sistemas operativos modernos. Permite presentar a los

procesos un espacio de memoria contiguo, optimizando su utilización al mantener en memoria principal solo los datos e instrucciones con los que se está trabajando activamente, descargando el resto a memoria secundaria cuando sea necesario.

- **Objetivos de la gestión de memoria**

- Asignar memoria a los procesos cuando la necesitan y liberarla cuando terminan
- Evitar fallos de página, es decir, necesitar algo que está en memoria secundaria, lo que obliga a esperar a que se traiga a memoria principal
- Utilizar la memoria principal disponible para acelerar la entrada/salida, mediante buffers y caché

- **Memoria Virtual:** La memoria virtual es un aspecto fundamental de la gestión de memoria en los sistemas operativos modernos. Algunas de las características clave de la memoria virtual son:

- Permite a los procesos tener la ilusión de disponer de un espacio de direcciones contiguo y mucho mayor que la memoria física real disponible. Esto facilita el trabajo de los compiladores.
- Utiliza técnicas como paginación y swapping para mantener en memoria principal solo las páginas de código y datos que se están utilizando activamente, descargando el resto a memoria secundaria (disco).
- Ofrece al proceso más memoria de la que hay físicamente disponible, mejorando el rendimiento al evitar tener que cargar y descargar constantemente procesos completos.
- Permite el uso eficiente de la memoria mediante técnicas como copy-on-write y el mapeo de archivos a memoria.
- Requiere la colaboración entre el software del sistema operativo y el hardware de gestión de memoria (MMU) para traducir las direcciones virtuales a físicas.

- **Paginación:** La paginación es una técnica fundamental en la gestión de memoria de los sistemas operativos modernos. Permite presentar a los procesos un espacio de direcciones virtuales más grande que la memoria física disponible, mejorando la eficiencia en el uso de la memoria.

- **Funcionamiento de la paginación**

- La memoria física se divide en bloques de tamaño fijo llamados marcos (frames).
- Los procesos se dividen en unidades lógicas del mismo tamaño llamadas páginas.
- Cada página de un proceso se asigna a un marco de memoria física de manera dinámica.
- Una tabla de páginas mantiene el mapeo entre las páginas lógicas de cada proceso y los marcos físicos donde residen.

- **Ventajas de la paginación**

- Permite que los procesos tengan más memoria de la físicamente disponible, mediante el uso de memoria virtual.
- Facilita la asignación y liberación de memoria a los procesos, al trabajar con unidades de tamaño fijo.
- Evita la fragmentación externa de la memoria, al poder ubicar las páginas de un proceso de manera dispersa.
- Simplifica la protección de memoria, al poder controlar el acceso a cada página individualmente.

- **Paginación y memoria virtual**

- La paginación se combina con la memoria virtual para mantener en memoria principal solo las páginas que se están utilizando activamente.
- Cuando un proceso necesita una página que no está en memoria, se produce un fallo de página que el sistema operativo maneja trayendo esa página desde el espacio de intercambio (swap).
- Los algoritmos de reemplazo de páginas, como LRU o óptimo, deciden qué página sacar de memoria principal cuando se necesita espacio.

- **Segmentación**

La segmentación de memoria es una técnica de gestión de memoria utilizada en sistemas operativos que permite dividir el espacio de direcciones lógicas de un proceso en segmentos de tamaño variable. Algunos aspectos clave de la segmentación incluyen:

- **Organización lógica de la memoria:** Un programa se divide en unidades lógicas llamadas segmentos, como procedimientos, funciones, pila, tablas de símbolos, etc.
 - Cada segmento tiene un identificador y consiste en un punto de inicio y un tamaño asignado.
 - Una dirección lógica se expresa como una tupla <número de segmento, desplazamiento dentro del segmento>.
- **Traducción de direcciones**
 - La tabla de segmentos realiza la traducción de direcciones lógicas bidimensionales a direcciones físicas lineales.
 - Cada entrada de la tabla de segmentos contiene la base (dirección física del inicio del segmento) y el límite (tamaño del segmento).
 - La CPU consulta la tabla de segmentos para traducir cada dirección lógica.
- **Implementación**
 - Cada proceso tiene su propia tabla local de descriptores (LDT) con una entrada por cada segmento.
 - Existen también tablas globales de descriptores (GDT) para objetos compartidos por todos los procesos.
 - Los registros STBR (puntero a la tabla de segmentos) y STLR (longitud de la tabla) permiten acceder a la tabla de segmentos.
- **Ventajas**
 - Mejor correspondencia con la visión del programador sobre la estructura del programa.
 - Permite compilar y cargar módulos de manera independiente.
 - Facilita la modificación y compartición de módulos.
- **Desventajas**
 - Complejidad adicional en el manejo de memoria virtual.
 - Mayor costo de hardware y software.
 - Aparición de fragmentación externa al asignar segmentos de tamaño variable.

e. SEGURIDAD Y PROTECCIÓN

La **seguridad y protección** son componentes esenciales de cualquier sistema operativo, ya que

garantizan la integridad, confidencialidad y disponibilidad de los datos y recursos del sistema. A continuación, se detallan los aspectos clave de la seguridad y protección en un sistema operativo:

❖ Control de Acceso

- **Listas de control de acceso (ACLs):** Especifican los permisos para cada usuario o grupo en relación con archivos y otros recursos.
- **Modelos de control de acceso:** Incluyen el control de acceso discrecional (DAC), el control de acceso obligatorio (MAC) y el control de acceso basado en roles (RBAC).

❖ Autenticación

- **Verificación de identidad:** Métodos para confirmar que un usuario es quien dice ser. Incluyen contraseñas, tarjetas inteligentes, biometría y autenticación de dos factores.
- **Gestión de credenciales:** Almacenamiento seguro y manejo de credenciales de usuario.

❖ Autorización

- **Permisos y privilegios:** Determinan lo que los usuarios autenticados pueden hacer dentro del sistema. Esto incluye permisos de lectura, escritura y ejecución para archivos y recursos.

❖ Protección de Recursos

- **Aislamiento de procesos:** Evita que los procesos interfieran entre sí. Cada proceso opera en su propio espacio de memoria.
- **Protección de memoria:** Utiliza técnicas como la paginación y segmentación para proteger áreas de memoria y evitar accesos no autorizados.

❖ Cifrado

- **Cifrado de datos en reposo:** Protege los datos almacenados en disco contra accesos no autorizados.
- **Cifrado de datos en tránsito:** Asegura que los datos transmitidos por la red no sean interceptados ni alterados.

❖ Auditoría y Registro:

- **Registros de auditoría:** Mantienen un registro de las actividades del sistema, incluidos accesos y modificaciones a los archivos y recursos.
- **Detección de intrusos:** Sistemas que monitorean y analizan los registros de

auditoría para identificar comportamientos sospechosos.

❖ **Seguridad de Red:**

- **Cortafuegos:** Filtran el tráfico de red para prevenir accesos no autorizados.
- **VPN (Redes Privadas Virtuales):** Proveen canales de comunicación seguros a través de redes inseguras, como Internet.

• **Protección contra Malware:**

- **Antivirus y Antimalware:** Detectan y eliminan software malicioso que puede comprometer la seguridad del sistema.
- **Actualizaciones y parches de seguridad:** Mantienen el sistema operativo actualizado para proteger contra vulnerabilidades conocidas.

❖ **Modelos de Seguridad:**

- **Modelo Bell-LaPadula:** Enfocado en la confidencialidad de los datos.
- **Modelo Biba:** Enfocado en la integridad de los datos.
- **Modelo de Confianza Dinámica:** Permite ajustar dinámicamente los niveles de confianza y permisos de los usuarios y procesos en el sistema.

IV. DESARROLLO

En el campo de la ingeniería de sistemas, los sistemas operativos juegan un papel crucial al servir como el puente entre el hardware y las aplicaciones de software. Este estudio comparativo se centra en tres de los sistemas operativos más utilizados y representativos hoy: Linux, macOS y Windows. Cada sistema operativo se diseñó con diferentes objetivos y características, haciendo adecuados para distintos tipos de dispositivos y entornos de uso.

El propósito de esta sección es proporcionar un análisis detallado de estos sistemas operativos desde varias perspectivas críticas, incluyendo la gestión de recursos, experiencia del usuario, seguridad y rendimiento. Al investigar y comparar estos aspectos, buscamos identificar las fortalezas y debilidades de cada sistema operativo, así como su idoneidad para diversas aplicaciones.

Para lograrlo, se ha realizado una investigación exhaustiva utilizando varias fuentes, incluyendo literatura académica, documentación oficial, estudios de caso y pruebas de rendimiento. Esta investigación nos permitirá no solo describir las

características técnicas de cada sistema operativo, sino también evaluar su impacto práctico en escenarios de uso reales.

Presentaremos un análisis comparativo detallado para cada sistema operativo mencionado, comenzando con su gestión de recursos y avanzando por los otros aspectos clave. Este análisis servirá como una guía comprensiva para entender cómo cada sistema operativo maneja las tareas fundamentales y cuáles son sus capacidades y limitaciones en el contexto moderno de la informática.

Preferencia según atributos de calidad del sistema operativo

Este estudio “Estudio comparativo de los atributos de los sistemas operativos [10]” recapitula los resultados de preferencia de los usuarios según ciertos atributos de calidad del sistema, estos atributos son:

- **GUI**
- **Seguridad**
- **Compatibilidad de hardware**
- **Manejo de memoria**
- **Interfaz en modo texto**
- **Arquitectura**
- **Sistema de archivos**
- **Portabilidad**
- **Manejo de procesos**

Los resultados se enuncian a continuación:

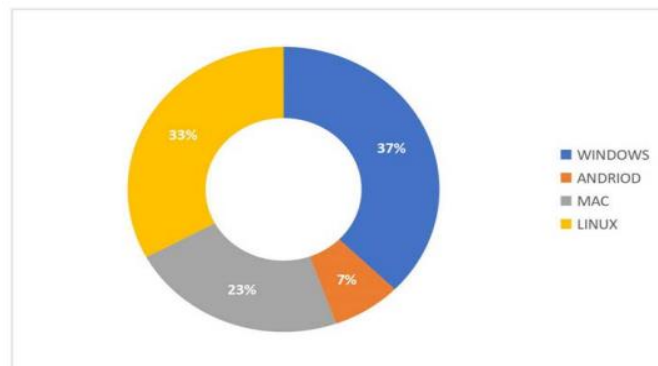


Figura 1: Diagrama de pie de la preferencia de usuario del sistema operativo.

Como se puede apreciar existe una clara preferencia sobre Windows como sistema operativo, sin embargo, Linux, también es muy preferido por los usuarios, estos dos últimos componen el 70% de la preferencia de los usuarios de sistemas operativos.

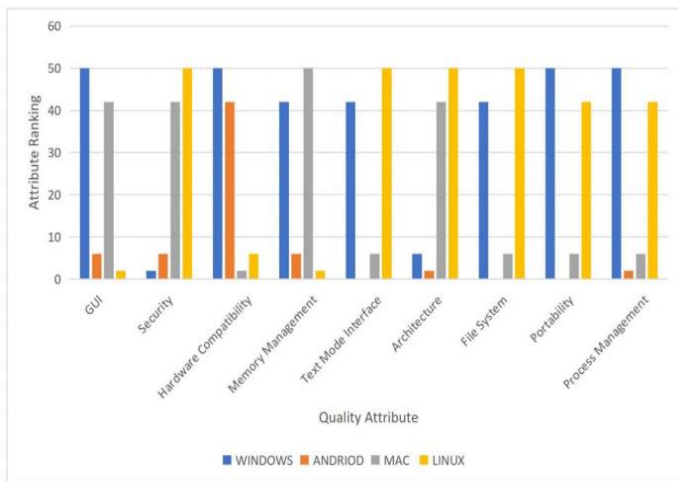


Figura 2: Diagrama de barras de las métricas de evaluación.

Windows posee más capacidad para GUI, liderando este atributo, en cuanto a seguridad el estudio descrito en la sección de seguridad determinó que Linux ofrecía mucha más robustez que Windows y que comparado con MacOS eran similares. Para nadie en su secreto que casi cualquier computadora o tarjeta madre es compatible con Windows, caso contrario con Linux que está más enfocada en la parte de servidores, en cuanto a manejo de memoria macOS es muy eficiente en esta gestión de memoria que de hecho Linux en los 3 comparados en este estudio es el peor. En cuanto arquitectura habría una lucha clara entre macOS y Linux, ambos basados en Unix, ligeramente Linux supera a macOS en cuestiones de arquitectura y en manejo de archivos bueno, Linux en este estudio demostró ser competente a diferencia de macOS.

Los resultados de este estudio son variados, los sistemas operativos tienen puntos fuertes y débiles, se podría concluir que la elección de un sistema operativo requiere de un análisis profundo asociado a la preferencia del usuario, pero que, si se busca es seguridad, flexibilidad, arquitectura y manejo de archivos, Linux es el apropiado. Interfaz de usuario y compatibilidad de hardware, entonces Windows es la elección. Por último, no sería tan fundamental escoger un sistema operativo por su manejo de memoria, pero este estudio demostró que macOS era el indicado.

Gestión de Recursos

Esta comparación se centra en la gestión de procesos y de memoria de los 3 sistemas operativos objetos de evaluación. Las

condiciones de la realización de esta prueba son las siguientes: Primero se requiere de un Hypervisor o monitor de máquina virtual, para la prueba específica se usó Hypervisor VMware, posteriormente se crean las máquinas virtuales donde se instalaron cada uno de los sistemas operativos a analizar. Los sistemas operativos son: Windows 7 Profesional, Ubuntu 10.10 y Mac OS X 10.16. En estos sistemas operativos se instala un servidor web apache que es el proceso que mide el rendimiento de los OS. Este servidor se verá afectado por distintas configuraciones con la proyección de evidenciar cambios de rendimientos producidos por este. Por último, para las mediciones de rendimiento se usaron herramientas de benchmarking, estas herramientas son JMeter y ab.

Se hará un énfasis particular en las pruebas realizadas con JMeter ya que son más significativas. La prueba de JMeter sigue una secuencia que deberá repetirse desde cada sistema operativo, esta secuencia es la siguiente: Primero, se accederá a la página principal de WordPress desde el servidor de apache. Segundo, se ingresará con usuario y contraseña a la aplicación web. Tercero, se creará una nueva entrada dentro de WordPress. Cuarto, se publicará dicha entrada. Quinto, se verá la nueva entrada. Y, por último, se saldrá de WordPress. Esta secuencia se realizará en dos ambientes de prueba. En la primera, se simulan 10 usuarios concurrentes realizando 100 peticiones. En la segunda, se simulan 50 usuarios concurrentes realizando 500 peticiones, llevando así el servidor al límite. Los resultados de las pruebas son métricas medias de la realización de 5 repeticiones de las mismas condiciones de prueba, es decir, cada prueba se realizó 5 veces, proporcionando robustez a los resultados.

Los resultados de la primera prueba (Figura 1) enuncian que tanto Windows 7 como Mac OS X, hacen más uso de su memoria para la realización de la primera prueba. Estos dos tienen un rendimiento menor en lo que se refiere a procesar una solicitud por segundo, por último, el tiempo de respuesta por petición también lo lidera Ubuntu (basado en Linux) con 0.139 s.

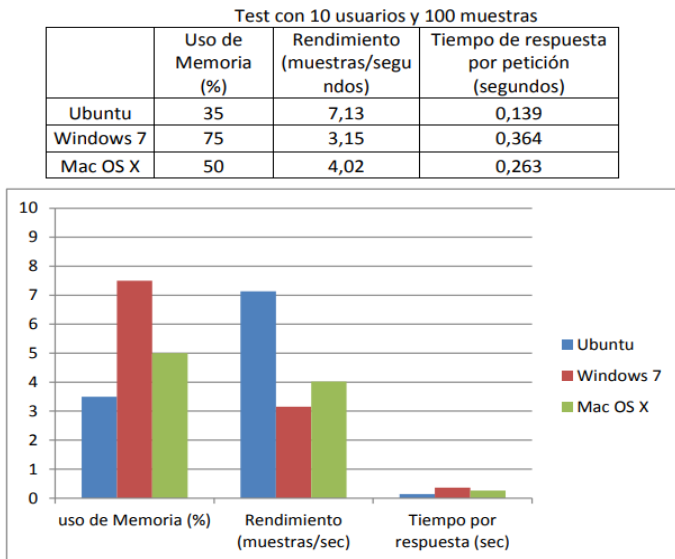


Figura 3: Tabla y gráfico de resultado de la prueba 1.

En cuanto a la segunda prueba, los resultados (Figura 2) indican que tanto Ubuntu como Mac OS X usan más memoria frente a una prueba que proporciona más carga, pero, en términos de rendimiento Ubuntu basado en Linux, sigue siendo el sistema con mejor rendimiento para esta prueba. Como lo menciona Serrano F “En este sentido encontramos un caso curioso, y es que Ubuntu obtiene un mayor rendimiento respecto a Mac OS X, siendo ambos sistemas prácticamente iguales en cuanto a que los dos se basan en la arquitectura de UNIX. Incluso el uso de memoria es menor. La explicación se puede encontrar en la compilación del kernel. No siendo el estudio del mismo parte de este trabajo, sí se ha podido constatar tras la búsqueda de información que el núcleo de Mac OS X tiene problemas de rendimiento provocados por el Kernel Mach” [7]. Por lo tanto, Ubuntu es más eficiente en lo que en términos de la compilación de kernel, lo que lo hace tan rápido e ideal para la elección de trabajo con servidores, su respuesta es naturalmente más rápida que los otros dos. Para finalizar, en términos de estrés Ubuntu mostró un tiempo de respuesta de 0.145 s por petición. Todos los sistemas operativos usaron el 100% de CPU al emplear estos test, así que en términos de CPU estaban en las mismas condiciones.

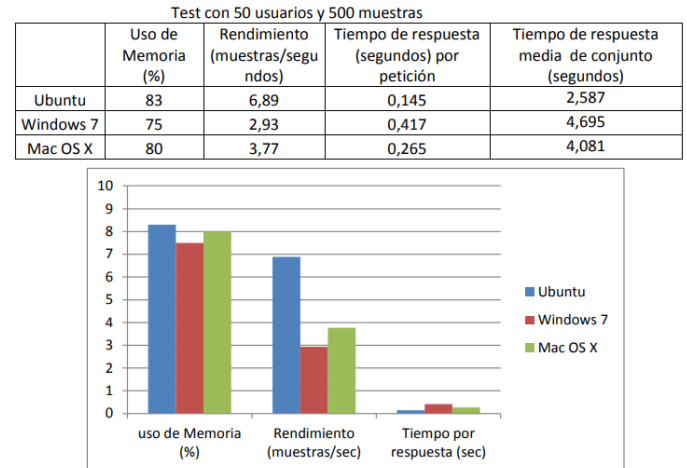


Figura 4: Tabla y gráfico de resultado de la prueba 2.

Las demás pruebas que se realizaron abarcan modificaciones directas sobre el servidor de apache, se consideró que estas pruebas que están descritas más sobre el software que sobre el comportamiento de los sistemas operativos no fuese considerada dentro de este análisis, y que sería suficiente con las pruebas descritas anteriormente, esta sección entonces resalta la eficiencia de Ubuntu en su manejo de memoria y de procesos que garantizan que esta secuencia de ejecución use menos memoria y rinda más que los demás sistemas operativos.

Seguridad

Para la comparativa de seguridad se tomará en dos partes debido a la disponibilidad de la información, primero un análisis de seguridad entre el sistema operativo Linux y el sistema operativo Windows, y posteriormente las opiniones recolectadas en foros de personas experimentadas en el nicho que responden a la pregunta qué tan seguro es el sistema operativo macOS comparado con el sistema operativo Linux.

El primer ensayo analiza las características de seguridad de los sistemas operativos Windows y Linux, y desarrolla perspectivas sobre cuál de ellos ofrece las mejores características de seguridad. El estudio concluye que la privacidad es un factor esencial que las organizaciones deben considerar. Los atacantes tienden a buscar información que puedan utilizar con propósitos maliciosos, lo que hace necesario tomar la seguridad del sistema con seriedad [8].

Características de Seguridad de Windows OS

- **Windows Defender Smart Screen:** Protege contra sitios web con phishing y malware, evitando la descarga de archivos maliciosos.
- **Windows Defender Application Guard:** Utiliza la virtualización para proteger contra amenazas avanzadas dirigidas.
- **User Account Control (UAC):** Controla y autoriza acciones en el sistema, limitando las actividades a los administradores.
- **Windows Defender Device Guard:** Aplica listas blancas de aplicaciones y controla qué aplicaciones pueden ejecutarse.
- **Windows Defender Exploit Guard:** Controla cómo se acceden y auditan las carpetas, protegiendo contra amenazas dentro de la red.
- **Microsoft Bitlocker:** Encriptación de disco completo que protege archivos y sistemas.
- **Windows Defender Credential Guard:** Utiliza seguridad basada en virtualización para aislar secretos y proteger contra aplicaciones no autorizadas.

Características de Seguridad de Linux OS

- **Discretionary Access Control (DAC):** Modelo de control de acceso que permite al propietario de un objeto definir políticas de seguridad.
- **Security-Enhanced Linux (SELinux):** Implementa políticas de seguridad definidas en un lenguaje especializado, controlando el acceso a procesos y archivos.
- **Access Control Lists (ACLs):** Utilizados para implementar el control de acceso extendido.
- **Namespaces:** Permite a los usuarios crear namespaces de usuario, ejecutando código no confiable como root sin riesgo para el host.
- **Linux Cryptography Features:** Utiliza algoritmos de cifrado como el Data Encryption Standard (DES) y módulos de seguridad del kernel (LSMs) para soportar varios modelos de seguridad.
- **Integrity Management Subsystem:** Utiliza la arquitectura de medición de integridad (IMA) para mantener una lista de medidas de integridad en tiempo de ejecución, asegurando que los datos no sean comprometidos.

Para finalizar, estas son las conclusiones obtenidas de este estudio:

- Linux se considera más seguro debido a su naturaleza de código abierto, acceso limitado y variedad de distribuciones. Windows, aunque ampliamente utilizado, enfrenta desafíos de seguridad debido a su popularidad y modelo cerrado.
- Tanto Windows como Linux implementan diversas estrategias para proteger los sistemas contra amenazas como programa maligno y phishing, crucial para la privacidad y confidencialidad de los datos en entornos corporativos y personales.

Para la segunda comparación Linux vs macOS, se obtendrá 3 opiniones y se hará un análisis crítico de los 3 puntos de vista, frente a cuál sistema es más seguro. Se realizará un resumen de cada opinión o punto de vista de la discusión planteada en este foro. Las respuestas buscan solventar la siguiente duda.
Pregunta: ¿Qué tan seguro es macOS en comparación con Linux? ¿Los usuarios pueden estar tan seguros con macOS como lo estarían con Linux al navegar por Internet, etc.?

Opinión 1: Se aborda la seguridad de macOS comparada con Linux en dos partes: la comparación de los kernels (XNU vs Linux) y la comparación de los sistemas completos (macOS vs distribuciones Linux). En resumen:

- ❖ **Comparación de kernels:** Se considera que ambos kernels son igualmente seguros, ya que heredan políticas de gestión de usuarios, permisos de archivos y acceso a hardware de sus raíces Unix.
- ❖ **Comparación de sistemas completos:** Algunas distribuciones Linux (como Fedora, RHEL, CentOS) se consideran ligeramente más seguras que macOS debido a SELinux y actualizaciones de seguridad más frecuentes. Sin embargo, otras distribuciones Linux (como Ubuntu, openSUSE, SLED) podrían ser ligeramente menos seguras que macOS en configuraciones predeterminadas.
- ❖ Se menciona que macOS y Linux pueden ser igualmente seguros en escenarios del mundo real, y la seguridad no debería ser el factor decisivo al elegir entre ambos sistemas operativos.

Opinión 2: Esta respuesta se centra en detalles técnicos específicos de implementaciones de seguridad en macOS y Linux:

- ❖ Se discuten las diferencias en la implementación de ACLs entre macOS y Linux, destacando cómo estas diferencias afectan la seguridad percibida y real en ambos sistemas.
- ❖ Se menciona que, debido a su arquitectura y políticas de seguridad, macOS y Linux tienen enfoques diferentes hacia la gestión de privilegios locales y la seguridad en profundidad.
- ❖ Se proporciona un análisis detallado de las prácticas de seguridad específicas en macOS, como el cifrado de disco completo, la deshabilitación de contenido Flash y la ejecución de código firmado.

Opinión 3: En esta respuesta, se expresa una opinión general sobre la seguridad comparativa entre macOS y Linux:

- ❖ Se señala que MacOS puede verse como menos seguro que Linux por su naturaleza cerrada y popularidad, lo que lo convierte en un objetivo mayor para los ataques.
- ❖ Se menciona que BSD, en el que se basa macOS, es generalmente considerado más seguro que Linux, pero se advierte sobre la selección de versiones de código más antiguas en macOS.
- ❖ Se destaca la capacidad de Linux para ser altamente personalizable en seguridad, algo que no es tan accesible en macOS por su naturaleza cerrada.

Después de analizar críticamente las tres respuestas, se puede concluir que la seguridad de macOS frente a Linux depende de varios factores:

- ❖ **Implementación y configuración:** macOS y Linux pueden ser seguros cuando se configuran correctamente con las mejores prácticas de seguridad y se mantienen actualizados.
- ❖ **Características y enfoques de seguridad:** Ambos sistemas operativos ofrecen capas de seguridad avanzadas, como el cifrado de disco completo y la gestión de permisos avanzada. La diferencia radica en cómo estas características se implementan y administran.
- ❖ **Personalización y control:** Linux ofrece una mayor capacidad de personalización y control sobre la configuración de seguridad, lo que puede ser crucial para entornos donde se requiere un nivel muy específico de protección.

En última instancia, la elección entre macOS y Linux en términos de seguridad debería basarse en las necesidades específicas del usuario, las políticas de seguridad de la organización y la

capacidad para implementar y mantener las mejores prácticas de seguridad. Ambos sistemas operativos pueden proporcionar un nivel alto de seguridad, pero la implementación correcta es clave para maximizar esta seguridad en cada caso particular.

Rendimiento

En las pruebas de rendimiento realizadas en la comparación fue el último Windows 10 contra Ubuntu 21.10 y el propio Clear Linux de Intel contra Apple macOS 12.1. Todas las pruebas se realizaron en el mismo Mac Mini 2018 con procesador Intel Core i7 8700B "Coffee Lake" con gráficos UHD 630, 8 GB de RAM y SSD de 250 GB. Se utilizó el mismo sistema a lo largo de las pruebas: la frecuencia de la CPU y las diferencias de memoria informadas en la tabla del sistema automatizado equivalen a la forma en que cada sistema operativo expone diferentes detalles, etc.

macOS 12.1

- En la prueba de V-Ray, macOS 12.1 tuvo un rendimiento similar a Windows 10 y algo inferior a Clear Linux.
- En la prueba de Primesieve, macOS 12.1 tuvo un rendimiento ligeramente inferior a Windows 10 y Clear Linux.
- Sin embargo, macOS 12.1 no pudo ejecutar las pruebas de Selenium Web Benchmarks debido a que no pudo encontrar el binario de Google Chrome.
- **WebP: Default:** 1.375
- **WebP: Quality 100:** 2.320
- **WebP: Quality 100, Lossless:** 6.775
- **compress-zstd-1.4.5 - Compression Speed:** 686.4 MB/s
- **compress-zstd-1.4.5 - Decompression Speed:** 760.1 MB/s
- **phoronix-test-suite:** Generalmente, los resultados de macOS tienden a ser intermedios, no liderando en la mayoría de las categorías, pero tampoco siendo los más bajos.

macOS 12.1 tiene un rendimiento intermedio, generalmente quedando en tercer lugar en muchas pruebas.

Ubuntu 21.10

- Ubuntu 21.10 no se incluyó en los resultados presentados, por lo que no hay información sobre su rendimiento en comparación a los otros sistemas.
- **WebP: Default:** 1.245

- **WebP: Quality 100:** 2.185
- **WebP: Quality 100, Lossless:** 6.332
- **compress-zstd-1.4.5 - Compression Speed:** 740.5 MB/s
- **compress-zstd-1.4.5 - Decompression Speed:** 767.4 MB/s
- **phoronix-test-suite:** Ubuntu muestra un rendimiento competitivo, superando a macOS en varias pruebas y quedando cerca de Clear Linux en muchas.

Ubuntu 21.10 también muestra un rendimiento sólido, a menudo superando a macOS y Windows.

Clear Linux 35390

- En la prueba de V-Ray, Clear Linux 35390 tuvo el mejor rendimiento de los tres sistemas operativos.
- En la prueba de Primesieve, Clear Linux 35390 también superó ligeramente el rendimiento de macOS 12.1 y Windows 10.
- **WebP: Default:** 1.324
- **WebP: Quality 100:** 2.131
- **WebP: Quality 100, Lossless:** 6.258
- **compress-zstd-1.4.5 - Compression Speed:** 748.8 MB/s
- **compress-zstd-1.4.5 - Decompression Speed:** 772.1 MB/s
- **phoronix-test-suite:** Clear Linux frecuentemente lidera en las pruebas, mostrando un rendimiento superior en muchas categorías, especialmente en tareas de compresión y descompresión.

Clear Linux tiende a ser el líder en rendimiento en la mayoría de las pruebas, destacándose especialmente en tareas de compresión y descompresión.

Windows 10 Pro

- En las pruebas de V-Ray y Primesieve, el rendimiento de Windows 10 Pro fue comparable al de macOS 12.1, aunque ligeramente inferior al de Clear Linux 35390.
- Al igual que macOS 12.1, Windows 10 Pro no pudo ejecutar las pruebas de Selenium Web Benchmarks.
- **WebP: Default:** 1.827
- **WebP: Quality 100:** 2.455
- **WebP: Quality 100, Lossless:** 6.692
- **compress-zstd-1.4.5 - Compression Speed:** 655.0 MB/s
- **compress-zstd-1.4.5 - Decompression Speed:** 752.6 MB/s

- **phoronix-test-suite:** Windows 10 Pro generalmente tiene resultados más bajos en comparación con los sistemas operativos basados en Linux, aunque sigue siendo competitivo en ciertas pruebas.

Windows 10 Pro tiende a tener un rendimiento inferior en comparación con los sistemas Linux, aunque sigue siendo competitivo en algunas categorías específicas.

En resumen, según los resultados mostrados, Clear Linux 35390 es el líder en muchas pruebas, especialmente en tareas de compresión, descompresión y en el rendimiento de la CPU, tiene un rendimiento superior al de macOS 12.1 y Windows 10 Pro en las pruebas realizadas. macOS 12.1 y Windows 10 Pro tuvieron un rendimiento similar en algunas pruebas, pero presentaron problemas de compatibilidad con ciertas herramientas. Pero hay que considerar que son resultados de una sola prueba y varían según las configuraciones y cargas de trabajo específicas.

macOS 12.1 vs. Linux vs. Windows				
ptsli	macOS 12.1	Ubuntu 21.10	Clear Linux 35296	Windows 10 Pro
webp: Default	1.375	1.426	1.330	1.492
webp: Quality 100	2.330	2.269	2.134	2.451
webp: Quality 100, Lossless	18.208	17.034	16.134	17.522
webp: Quality 100, Highest Compression	6.141	6.082	6.388	7.303
webp: Quality 100, Lossless, Highest Compression	37.703	36.429	37.410	40.671
compress-idx-1 - Compression Speed	6586.25	6540.12	7629.82	6555.82
compress-idx-1 - Decompression Speed	49.87	49.84	51.28	51.86
compress-idx-3 - Compression Speed	7621.9	8150.1	8202.1	7821.0
compress-idx-3 - Decompression Speed	49.87	49.82	49.37	50.20
compress-idx-9 - Compression Speed	7988.1	8187.3	8225.9	7786.0
compress-idx-9 - Decompression Speed	1916.4	1910.8	1936.5	1971.2
compress-zstd-1 - Compression Speed	3373.2	3335.4	3411.9	3080.9
compress-zstd-1 - Decompression Speed	156.1	156.3	159.2	158.5
compress-zstd-3 - Compression Speed	3447.9	3426.8	3474.8	3181.0
compress-zstd-3 - Decompression Speed	17.8	18.7	19.4	18.2
compress-zstd-19 - Compression Speed	2963.8	2996.0	3015.1	2757.1
compress-zstd-19 - Decompression Speed	1071.1	964.6	1073.4	1236.3
compress-zstd-3, Long Mode - Compression Speed	107.7	126.3	122.0	123.4
compress-zstd-3, Long Mode - Decompression Speed	3634.1	3613.8	3723.1	3366.3
compress-zstd-19, Long Mode - Compression Speed	15.5	16.3	16.8	15.3
compress-zstd-19, Long Mode - Decompression Speed	3158.8	3110.8	3044.4	2859.5
libraw: Post-Processing Benchmark	10.34	20.97	47.82	9.36
lucorenderer: DLSS - CPU	1.36	1.08	1.07	1.92
lucorenderer: Rainbow Colors and Prism - CPU	5.47	4.82	4.75	2.45
emtime: Pathtracer - Crown	6.5864	6.7478	6.7100	6.4269
emtime: Pathtracer (IPC - Crown)	7.0307	7.0722	7.0848	7.0487
emtime: Pathtracer - Awan Dragon	7.8792	7.9642	8.0479	7.9249
emtime: Pathtracer - Awan Dragon Obj	7.0144	7.1640	7.2777	6.8359
emtime: Pathtracer (IPC - Awan Dragon)	8.8305	8.8932	8.7284	8.4971
emtime: Pathtracer (IPC - Awan Dragon Obj)	7.3618	7.5329	7.6592	7.3807
kvaasar: Boophorus 4K - Slow	3.96	4.21	4.41	3.78
kvaasar: Boophorus 4K - Medium	3.98	4.26	4.47	3.83
kvaasar: Boophorus 1080p - Slow	21.20	21.67	22.86	18.63
kvaasar: Boophorus 1080p - Medium	21.64	22.34	23.89	20.09
kvaasar: Boophorus 4K - Very Fast	9.22	9.94	12.25	8.92
kvaasar: Boophorus 4K - Ultra Fast	18.48	17.59	21.07	15.77
kvaasar: Boophorus 1080p - Very Fast	42.96	45.95	58.83	47.70
kvaasar: Boophorus 1080p - Ultra Fast	75.83	80.74	97.70	74.46
rvr1k-1	6.383	6.108	6.107	6.068
rvr1k-5	2.918	1.910	1.888	1.813
rvr1k-6	2.708	2.101	2.100	2.165
rvr1k-10	7.448	6.970	7.065	6.365
odin: RT_rdr_gfx_nm_3840x2160	0.23	0.22	0.23	0.23
odin: RT_rdr_gfx_nm_3840x2160	0.23	0.22	0.22	0.23
odin: RT_lightmap_hdr_4096x4096	0.11	0.11	0.11	0.11
compress-7zip: Decompression Rating	37225	36458	41404	36568
compress-7zip: Decompression Rating	36891	36373	36704	37361
vlccoffin: Total Time	14812437	15481298	16831881	14813028
avifenc: 1024 Head Memory, 26 Depth	19633863	19633863	19960894	19633490
avifenc: 0	62.467	66.607	61.324	172.584
avifenc: 2	48.924	49.007	67.757	58.888
avifenc: 6	16.322	16.824	15.544	18.905
avifenc: 10	4.714	4.193	2.782	4.989
avifenc: 6, Lossless	85.476	92.031	86.205	86.308
avifenc: 10, Lossless	7.812	6.860	5.728	7.80
c-ray: Total Time - 4K, 16 Rays Per Pixel	153.398	107.868	83.480	116.005
compress-1z: Compressing ubuntu-18.04.3-server-2018.07	60.830	54.991	48.823	60.781
encode-raw: WAV To FLAC	17.008	14.469	13.884	16.250
atencio: Medium	9.5087	9.4207	9.0592	9.1969
atencio: Thorough	9.5010	9.9968	9.7511	11.2281
atencio: Exhaustive	87.4885	86.8183	93.3317	106.7752
blender: BMW27 - CPU-Only	236.49	249.00	245.92	297.86
blender: Classroom - CPU-Only	635.97	673.07	675.12	651.26
blender: Fluffy Cat - CPU-Only	329.26	344.42	342.90	421.73
blender: Pibratton Barabara - CPU-Only	824.06	848.12	832.73	964.69
indigobench: CPU - Bedroom	1.183	1.152	1.304	1.091
indigobench: CPU - Superior	2.829	2.689	2.743	2.819
pybench: Total Per Average Twd Time	1266	975	943	1341
appleseed: Empty	564.843623	629.781488	618.659352	667.0729
appleseed: Disney Material	313.75824	378.611881	380.734072	405.483
appleseed: Material Texter	313.804891	354.400143	346.788384	373.859
phpbench: PHP Benchmark Suite	642911	611864	1068953	483425
encode-ffmpeg: WAV To MP3	14.446	14.407	13.319	14.180
git: Time To Complete Common Git Commands	71.552	52.237	52.330	62.722
v-ray: CPU	6369	6392	6300	5901
prime95: 1912 Prime Number Generation	36.988	37.494	38.200	37.071
seleium: JRE8- Firefox	38.69	46.49	45.19	43.64
seleium: JRE8- Firefox	136.3	111	114	61.8
seleium: JRE8- Firefox	67.213	78.228	56.605	66.373
seleium: Speedometer - Firefox	130.8	114	123	119
seleium: JRE8-4 - Google Chrome	39.14	16.34		17.36
seleium: JRE8-4 - Google Chrome	106.7	34.0		35.8
seleium: JRE8-4 - Google Chrome	58.025	138.768		134.417
seleium: Speedometer - Google Chrome	130.1	143		160
seleium: WA-DM ImageCompuate - Firefox	33.1	31.6	33.8	28.8
seleium: WA-DM collisionDetection - Firefox	507.6	486.1	485.9	485.7
seleium: WA-DM ImageCompuate - Google Chrome	32.0	45.47		38.98
seleium: WA-DM collisionDetection - Google Chrome	630.7	385.59		373.78

IV. CONCLUSIONES

Diferencias Clave entre Sistemas Operativos

- ❖ Linux, macOS y Windows son sistemas operativos con enfoques distintos debido a sus objetivos de diseño y características específicas.
- ❖ Linux se destaca por su robustez en seguridad, flexibilidad y eficiencia en servidores.
- ❖ Windows es preferido por su interfaz de usuario amigable y amplia compatibilidad de hardware.
- ❖ macOS sobresale en gestión de memoria y experiencia del usuario, especialmente en entornos creativos.

Gestión de Recursos y Rendimiento

- ❖ En las pruebas de rendimiento, Ubuntu mostró ser superior en manejo de procesos y eficiencia de memoria frente a Windows y macOS.
- ❖ Clear Linux lideró en pruebas de compresión y descompresión, destacándose en rendimiento de CPU.
- ❖ macOS y Windows 10 presentaron resultados competitivos, pero con limitaciones específicas en ciertas pruebas, como problemas de compatibilidad.

Seguridad y Privacidad

- ❖ Linux, con su naturaleza de código abierto y herramientas de seguridad como SELinux, ofrece una mayor seguridad percibida en comparación con macOS y Windows.
- ❖ Windows ofrece características robustas como Windows Defender, mientras que macOS se beneficia de una base BSD y prácticas de seguridad implementadas.

Preferencia del Usuario y Adaptabilidad

- ❖ La elección entre sistemas operativos depende de las necesidades específicas del usuario, desde preferencias de interfaz hasta requisitos de seguridad y rendimiento.
- ❖ Linux y Windows dominan la preferencia de los usuarios, cada uno por sus fortalezas en diferentes áreas, mientras que macOS es popular entre los usuarios creativos y en entornos corporativos específicos.

Recomendaciones Finales

- ❖ Para entornos que requieren alta personalización y seguridad en servidores, Linux es la elección ideal.
- ❖ Windows es preferido en entornos de consumo y ofimáticos debido a su facilidad de uso y compatibilidad.
- ❖ macOS destaca en diseño gráfico y gestión de medios gracias a su eficiencia en manejo de memoria y optimización del sistema.

V. REFERENCIAS

[1] CSeguir, C. (s/f). *SISTEMAS OPERATIVOS WINDOWS, LINUX, MACOS, ANDROID*. SlideShare. Recuperado el 18 de junio de 2024, de <https://es.slideshare.net/slideshow/sistemas-operativos-windows-linux-macos-android/37215422>

[2] (S/f). Xunta.gal. Recuperado el 18 de junio de 2024, de https://www.edu.xunta.gal/centros/iesblancoamorculledo/aula_virtual/pluginfile.php/25655/mod_page/content/30/SistemasOperativos_JoaRanieri_AlvaroRodriguez_SergioVillar.pdf

[3] Valero, C. (2019, julio 23). *Más allá de Windows, Linux o macOS: 10 sistemas operativos alternativos*. ADSLZone. <https://www.adslzone.net/2019/07/23/sistemas-operativos-alternativos-windows-linux-macos/>

[4] Reinoza, J. (2020, mayo 7). *Windows, macOS y Linux – Características, diferencias y curiosidades*. TecnoTrono. <https://tecnotrono.com/software/sistemas-operativos/257/>

[5] Saldaña, J. R. (2023, junio 2). Las diferencias entre Windows, Linux y MAC OS. *Deusto Enterprise Services - En Deusto Enterprise Services Nos Especializamos En Servicios de Consultoría Informática. Contamos Con Alta Experiencia Desde 2003*. <https://deustoes.com/en/2023/06/02/las-diferencias-entre-windows-linux-y-mac-os/>

[6] GeeksforGeeks. Evolution of Operating System. Accedido el 3 de julio de 2024. [En línea]. Disponible: <https://www.geeksforgeeks.org/evolution-of-operating-system/>

[7] F. Serrano. "Comparative study of the attributes of operating systems". Accedido el 3 de julio de 2024. [En línea]. Disponible: <https://openaccess.uoc.edu/bitstream/10609/8179/1/fserranocaTFC0611.pdf>

[8] International Journal of Modern Information Retrieval and Management. "Security features of Windows OS". Accedido el 3 de julio de 2024. [En línea]. Disponible: <https://ijmirm.com/index.php/ijmirm/article/view/19/18>

[9] Quora. "How secure is macOS compared to Linux? Can users be as safe with macOS as they would be with Linux while surfing the internet, etc". Accedido el 3 de julio de 2024. [En línea]. Disponible: <https://www.quora.com/How-secure-is-macOS-compared-to-Linux-Can-users-be-as-safe-with-macOS-as-they-would-be-with-Linux-while-surfing-the-internet-etc>

[compared-to-Linux-Can-users-be-as-safe-with-macOS-as-they-would-be-with-Linux-while-surfing-the-internet-etc](https://www.quora.com/How-secure-is-macOS-compared-to-Linux-Can-users-be-as-safe-with-macOS-as-they-would-be-with-Linux-while-surfing-the-internet-etc)

[10] IOP Science. "Comparative study of operating system attributes". Accedido el 3 de julio de 2024. [En línea]. Disponible: <https://iopscience.iop.org/article/10.1088/1757-899X/1107/1/012061/pdf> [11] "Historia de Linux - Evolución del Sistema Operativo". frikidelhacking.com. Accedido el 3 de julio de 2024. [En línea]. Disponible: <https://frikidelhacking.com/sistemas-operativos/linux/historia/> [11] J. E. Magen. "The coming era of OS kernel modularity". DEV Community. Accedido el 3 de julio de 2024. [En línea]. Disponible: <https://dev.to/yonkeltron/the-coming-era-of-os-kernel-modularity-5emc>

