

# Comparación de rendimiento de ejecución de tareas en distintos sistemas operativos: una revisión sistemática

## Comparison of task execution performance on different operating systems: a systematic review

Recibido: noviembre 01 de 2024 | Revisado: noviembre 28 de 2024 | Aceptado: diciembre 12 de 2024

CHRISTIAN MANUEL RODRÍGUEZ CHILET<sup>1</sup>  
MANUEL ALONSO CARRANZA AVELLANEDA<sup>1</sup>  
ADRIAN OLULO VERAMENDI<sup>1</sup>  
IVÁN CARLO PETRLIK AZABACHE<sup>1</sup>

### RESUMEN

La eficiencia en la ejecución de tareas a través de diferentes sistemas operativos es un factor crítico que influye en la experiencia del usuario y la optimización de esta mismas. El presente estudio tiene como finalidad una revisión sistemática a la literatura con respecto a la comparación de diferentes sistemas operativos en las plataformas Windows, MacOS, Linux a través de un análisis exhaustivo de las fuentes y estudios empíricos, se utilizó el protocolo PRISMA para la filtración y selección de artículos científicos relevantes, asegurando la inclusión de estudios de alta calidad y relevancia. Se evaluaron métricas clave de rendimiento como la velocidad de procesamiento, utilización de recursos, estabilidad del sistema y satisfacción del usuario. Nuestros hallazgos revelan diferencias significativas en cómo estos sistemas operativos manejan diversas tareas, destacando fortalezas y debilidades específicas inherentes a cada plataforma. Llegando a la conclusión que Windows tiene un mejor rendimiento respecto a tareas de renderización de video por su alta compatibilidad de drivers directamente de los fabricantes y que Linux destaca más en el plano servidor debido a su destacable planificador de tareas.

**Palabras clave:** sistemas operativos, análisis comparativo, revisión de sistemas, eficiencia

### ABSTRACT

The efficiency in the execution of tasks through different operating systems is a critical factor that influences the user experience and the optimization of the same. This study aims to systematically review the literature regarding the comparison of different operating systems on Windows, MacOS, Linux platforms through an exhaustive analysis of sources

1 Universidad Nacional Federico Villarreal. Perú

Autor de correspondencia:  
2021018586@unfv.edu.pe

© Los autores. Este artículo es publicado por la Revista Campus de la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres. Este artículo se distribuye en los términos de la Licencia Creative Commons Atribución No-Comercial – Compartir-Igual 4.0 Internacional (<https://creativecommons.org/licenses/by/4.0/>), que permite el uso no comercial, distribución y reproducción en cualquier medio siempre que la obra original sea debidamente citada. Para uso comercial contactar a: [revistacampus@usmp.pe](mailto:revistacampus@usmp.pe).

<https://doi.org/10.24265/campus.2024.v29n38.06>

and empirical studies, using the PRISMA protocol for the filtering and selection of relevant scientific articles, ensuring the inclusion of high quality and relevant studies. Key performance metrics such as processing speed, resource utilization, system stability and user satisfaction were evaluated. Our findings reveal significant differences in how these operating systems handle various tasks, highlighting specific strengths and weaknesses inherent to each platform. We conclude that Windows performs better with respect to video rendering tasks due to its high driver support directly from the manufacturers and that Linux excels more on the server side due to its outstanding task scheduler.

**Keywords:** operating systems, comparative analysis, systems review, efficiency

## Introducción

Según (Yao et al., 2020), un sistema operativo es aquel intermediario crítico para la interacción productiva entre el hombre y la máquina. Asimismo, es un sistema de software que facilita el uso a una computadora de sobremesa o portátil en tareas que realizamos en la vida cotidiana como redactar documentos, navegar por internet y programar. En el mercado, existen muchas opciones desde Windows, Linux, Android, iOS, de las cuales cada uno de ellos con diferentes performance en el procesamiento de tareas, interfaces gráficas. Según (Tanenbaum & Bos, 2024), menciona que los distintos operativos pueden obtener lentitud a lo largo de su crecimiento en cuanto a funciones adicionales que no realmente sean necesarios y que obligan al hardware a ejecutar cada vez más tareas. En otras palabras, uno de los desafíos principales al buscar el mejor rendimiento de un sistema operativo es el propio crecimiento en nuevas características y funcionalidades de este

Además, (Leiserson et al., 2020) menciona que frente a las limitación que involucra una reducción en la velocidad de evolución del hardware que ha tenido en los último años debido a su creciente dificultad de fabricación, la única forma momentánea

que existe de confrontar dichas dificultades es el de desarrollar e implementar algoritmos más eficientes y buscar alternativas al crecimiento del rendimiento del hardware basado en la miniaturización de núcleos hechos de silicón.

Respecto a esta problemática del rendimiento, el siguiente artículo tendrá un enfoque de revisión sistemática en la comparación del rendimiento en sistemas operativos en distintas tareas, parámetros de medición en estos, su respectiva diferenciación de sus programadores de tareas y discusión acerca de los distintos enfoques que distintos sistemas operativos tienen para enfrentar tareas similares.

## Software

De acuerdo con (Pfeiffer, 2020), el software es un código distribuido con información que puede ser ejecutado en máquinas capaz de manejar esta información como computadoras. Son entidades no físicas pero que trabajan en las computadoras a nivel de código binario.

## Sistema Operativo

Según (Odun-Ayo et al., 2021), el sistema operativo es la interfaz principal

entre el hombre y la máquina para llevar a cabo las interacciones que desencadenan cálculos para cumplir con tareas propuestas por el hombre. Estos pueden ser:

### **Task Scheduler**

En el artículo (Soualhia *et al.*, 2020), definen a task scheduler o en español planificador de tareas como aquel apartado de todo sistema operativo que permite administrar y programar la ejecución de las tareas ejecutadas tanto programar el orden correspondiente de estos como la reprogramación de alguna de ellas en caso de fallos.

### **Benchmark**

De acuerdo con (Resch & Karpuzcu, 2019), benchmarking es la acción de realizar evaluaciones, en la actualidad a través del software, para medir el rendimiento de un hardware determinado. Esta acción cuenta con diferentes métricas que permiten la comparación del desempeño de hardware distinto en distintos aspectos como porcentaje del CPU utilizado, temperaturas alcanzadas, memoria RAM empleada, entre otros.

### **Flop**

El Flop (floating-point operation per second) según (San & Yakunin, 2019) es una magnitud escalar empleada en ciencias de la computación con el objetivo de medir la cantidad de operaciones aritméticas que puede realizar un microprocesador por segundo transcurrido. En general, esta magnitud es utilizada para dar una noción del rendimiento que un hardware determinado ofrece respecto a ciertas tareas.

### **Latencia**

De acuerdo con (Prasad *et al.*, 2014), la latencia dentro de la computación es una magnitud medida, generalmente, en milisegundos que compara el tiempo que tarda en ser enviado una unidad de información de un lado del sistema a otro. Por ejemplo, se puede hablar del tiempo que tarda la computadora en reconocer la pulsación y tipeo de un carácter en nuestro teclado.

### **Threads**

Sobre los threads (conocidos como hilos en español) en el área de la computación, según (Lin & Ding, 2021) son el segmento más pequeño de instrucciones que son manejados por un programa que permiten la concurrencia y paralelismo entre tareas dentro de una computadora. Además, gracias a la coordinación de estos hilos es posible dar un mejor aprovechamiento a los recursos disponibles de una computadora.

### **Modelo Cliente-Servidor**

Client-server model (en español modelo cliente-servidor) descrito por (Darbord *et al.*, 2023) es aquel modelo de arquitectura de computación en el cuál se realiza una conexión entre más de una computadora para realizar intercambios de servicios de ejecución e información a través de una red.

### **Client**

Dentro de la arquitectura cliente-servidor, (Qian *et al.*, 2020) define que el cliente es una entidad conformada por varias computadoras que son capaces de realizar requerimientos que serán recibidos por las computadoras que tienen el rol de

servidores. Estos requerimientos pueden ser la obtención de datos, guardado de datos, entre otros.

## Server

Ishlakhuddin (2021), manifiesta que *server* es un servidor en español y se refiere a aquellas computadoras que tienen como rol recibir los *requisitos* (*requests*) de los clientes. Estos pueden almacenar información, proporcionar servicios, cargar programas entre otros. Además, tienen la propiedad de que estos pueden ser accedidos por más de un cliente a la vez dentro de la red.

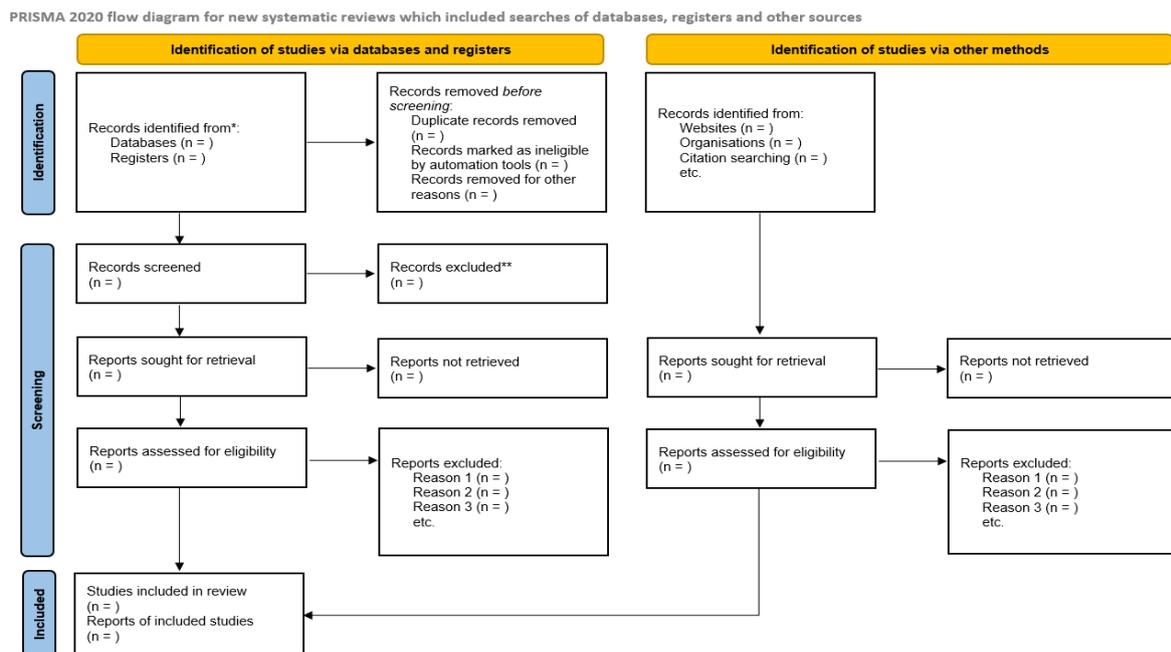
## Método

Este artículo es una revisión sistemática que se guía bajo la metodología o protocolo PRISMA. Según Page *et al.* (2021), documento oficial de Elsevier, PRISMA es el conjunto de lineamientos universales

que facilitan la redacción y ejecución de una revisión sistemática. Es decir, son las pautas y pasos que nos permiten filtrar un gran grupo de artículos empleando criterios de exclusión con el fin de obtener artículos relevantes al tema de investigación propuesto.

Respecto a estos criterios de exclusión se proponen preguntas de investigación relevantes al tema de investigación para realizar un análisis de las posturas de los autores respecto a los temas que dichas preguntas proponen para así llevar dichas posturas a una discusión y obtener conclusiones que sean útiles para futuras investigaciones en el campo de la computación e informática. PRISMA propone un total de 27 lineamientos que tienen el objetivo de justificar, sintetizar y fortalecer el proceso de investigación mediante la ilustración de proceso de clasificación de artículos tal y como se puede apreciar en la Figura 1.

**Figura 1**  
*Fases de la revisión sistemática recomendado por Elsevier*



\*Consider, if feasible to do so, reporting the number of records identified from each database or register searched (rather than the total number across all databases/registers).  
\*\*If automation tools were used, indicate how many records were excluded by a human and how many were excluded by automation tools.

From: Page MJ, McKenzie JE, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, et al. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ* 2021;372:n71. doi: 10.1136/bmj.n71. For more information, visit: <http://www.prisma-statement.org/>

## Preguntas de investigación

El objetivo principal de este artículo es el de comparar la evidencia experimental de otros papers empleando la revisión sistemática respecto al rol que tienen los sistemas operativos en el rendimiento que ofrecen en distintas tareas. Las preguntas de investigación identificadas para este estudio se muestran a continuación:

**RQ1:** ¿Cómo el planificador de tareas afecta el rendimiento en cada sistema operativo?

**RQ2:** ¿Qué tipo de pruebas de rendimiento utilizan los estudios recolectados?

**RQ3:** ¿Qué sistemas operativos son los más eficientes para la renderización de videos?

**RQ4:** ¿Qué sistema operativo tiene mejor rendimiento para sistemas de servidores en red?

A su vez, se da conocimiento de los objetivos de cada una de las preguntas de investigación en la Tabla 1.

**Tabla 1**

*Objetivos de investigación*

ID	Objetivo
RQ1:	Identificar los distintos planificadores de tareas entre los distintos sistemas operativos y reconocer su impacto en la ejecución de tareas.
RQ2:	Reconocer a través de comparaciones estadísticas hechos a través de benchmarks, las fortalezas de cada sistema en cálculos básicos.
RQ3:	Discernir el sistema operativo con mayor rendimiento de renderización de videos.
RQ4:	Reconocer los sistemas operativos con mejor desempeño en cuanto a trabajo como servidores en red.

## Fuentes y estrategias de búsqueda

Las bases de datos empleados en esta investigación fueron escogidas mayormente por su afinidad a las ciencias de la computación y afines para obtener

artículos científicos relevantes con el tema central a tratar de los sistemas operativos. El listado de las base de datos que fueron empleados para la presente revisión puede ser observado en la Tabla 2.

**Tabla 2**

*Principales bases de datos*

Nº	Fuente
1	IEEE
2	Arxiv
3	Dialnet
4	Scielo
5	Scopus
6	Google Académico

En cuanto a la estrategia de búsqueda aplicada en este trabajo de investigación, se consideraron los operadores booleanos para dar más precisión en las búsquedas de artículos

relevantes a los objetivos propuestos anteriormente. Se emplearon tanto los operadores AND (y) e OR (o) para combinar los términos de búsquedas listados en la Tabla 3.

**Tabla 3**

*Términos utilizados para búsqueda de artículos*

Índice	Término
T1	Evaluación OR Benchmark
T2	Rendimiento
T3	Sistema Operativo
T4	Performance
T5	Eficiencia

Los resultados que logramos obtener son los siguientes denotados en la Tabla 4.

**Tabla 4**

*Resultados iniciales sin procesar*

Índice	Término
IEEE	4,389
Arxiv	1,051
Dialnet	1,411
SciELO	923
Scopus	24,366
Google Scholar	266,000
Total	298,140

### **Criterios de selección**

Se definieron criterios de exclusión para evaluar con precisión la calidad de los estudios disponibles. Los estudios fueron revisados y discutidos por los autores para su exclusión. Los criterios de exclusión dados para la revisión de los estudios fueron los siguientes.

CE1: Los artículos encontrados no son artículos científicos o similares.

CE2: Los artículos tienen menos de seis años de su publicación.

CE3: Los artículos están escritos en un idioma distinto al inglés o al español.

CE4: Los títulos y las palabras clave de los artículos son inadecuados.

CE5: Los resúmenes de los artículos son irrelevantes.

CE6: Acceso incompleto al artículo

CE7: Los artículos están duplicados.

### **Selección de estudios**

Los criterios de selección se aplicaron para la búsqueda de los artículos científicos que fueron empleados para

esta revisión sistemática. La cantidad de artículos excluidos por el proceso de revisión sistemática puede ser observada

en la tabla ubicada a continuación en la Tabla 5.

**Tabla 5**  
*Ejecución de criterios de exclusión*

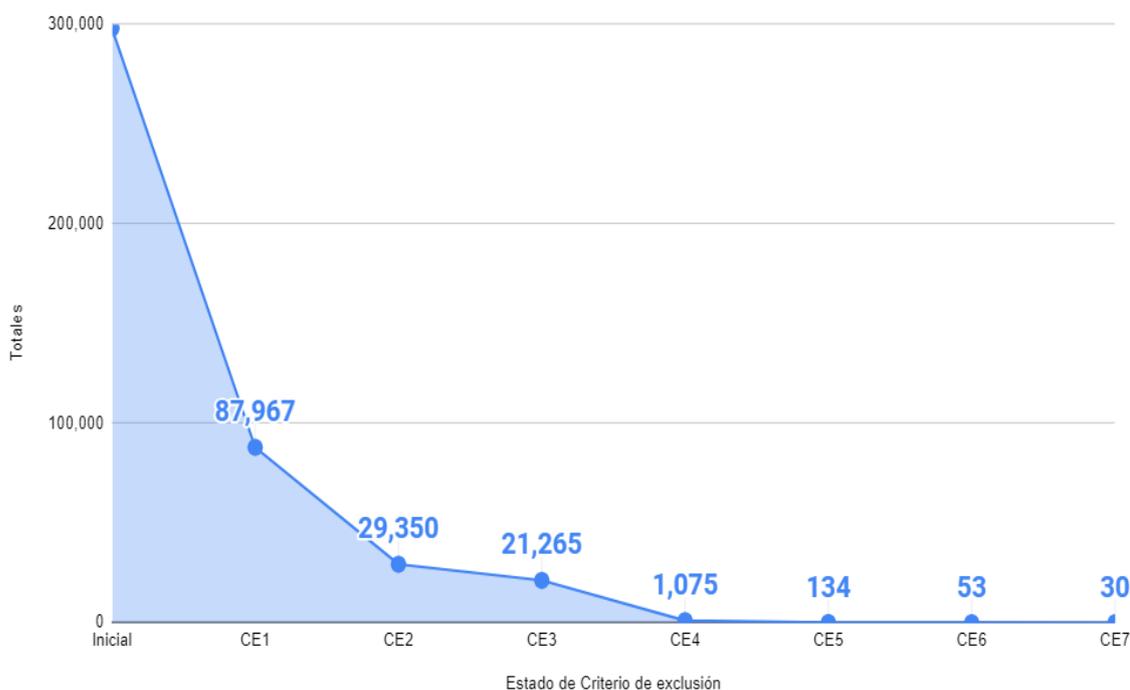
Fuente	Cant. Inicial	CE1	CE2	CE3	CE4	CE5	CE6	CE7
IEEE	4,389	4,367	2155	1287	110	12	6	6
Arxiv	1,051	1,051	887	768	675	3	3	2
Dialnet	1,411	389	300	300	129	49	5	2
Google Académico	266	71.400	15,6	15,4	15	13	11	11
Scielo	923	711	301	243	116	43	23	5
Scopus	24,366	10,049	10,107	3,267	30	17	5	4
<b>TOTAL</b>	<b>298,140</b>	<b>85,945</b>	<b>28,496</b>	<b>21,265</b>	<b>1,075</b>	<b>134</b>	<b>53</b>	<b>30</b>

Adicionalmente, adjuntamos un gráfico que indica la evolución del filtro de criterios de exclusión que se encuentra en la Figura 2. Observamos que en

comparación a la cantidad inicial tan solo el 9.97% de los papers en total que se obtuvo con la búsqueda inicial empleando los términos clave pasaron los filtros.

**Figura 2**  
*Gráfico evolutivo del filtro de papers*

Total de artículos frente a criterios de exclusión



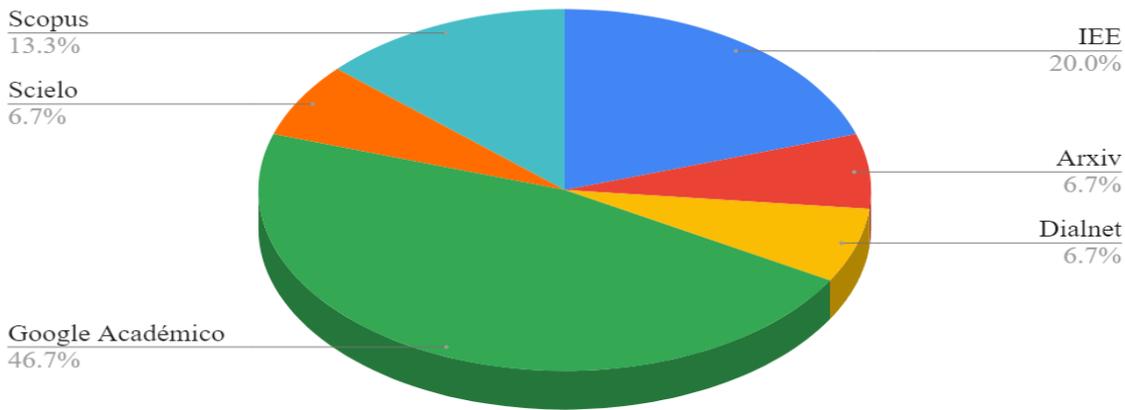
Además, luego de aplicar los criterios de exclusión, quedaron un total de 31 artículos los cuales se emplearon durante la presente revisión sistemática. Estos están clasificados en subgrupos

que conforman las fuentes de los cuales se obtuvieron en un principio como IEE, SciELO, Scopus, entre otros. Evidenciamos la conformación de dichos artículos en la siguiente Figura 3.

### Figura 3

Conformación de artículos restantes luego de CE7

Porcentaje de artículos restantes después de CE7



### Evaluación de calidad

Luego de aplicar los criterios de exclusión (CE), obtuvimos un total de 31 artículos restantes. Luego de ello, cada uno de los participantes del presente artículo los evaluó con respecto a los siguientes criterios de aseguramiento de calidad, o inglés *Quality Assurance*:

- QA1: ¿El paper presenta sus objetivos de investigación de forma clara?
- QA2: ¿Las condiciones para la evaluación de rendimiento fueron realizadas de forma justa y aceptable?
- QA3: ¿El paper emplea parámetros matemáticos para una fácil comparación?
- QA4: ¿El paper está bien organizado?
- QA5: ¿Los métodos empleados al analizar sus resultados fueron clara y correctamente ejecutados?

- QA6: ¿El campo de estudio está bien identificado?
- QA7: ¿Los resultados de su experimentación fueron presentados de forma clara?

### Extracción de información

Para esta etapa, cada uno de los 27 estudios incluidos en esta revisión sistemática proporcionó los datos para esta investigación utilizando el formulario de extracción en la herramienta de aplicación Excel.

Las propiedades que se extrajeron se especificaron como título, URL, fuente, año, país, número de páginas, e idiomas, tipo de publicación, nombre de la publicación, metodología de investigación, autores, afiliación, número de citas, resumen y palabras clave, Detalle

1 y Detalle 2, si es necesario; Detalle 3 y tamaño de la muestra.

Además, se seleccionaron las páginas que contenían información relevante para responder a las preguntas de investigación planteadas en este estudio sobre la comparación del rendimiento de tareas en diferentes sistemas operativos.

### Síntesis de la información

Después de extraer la información de cada estudio, se realizó un análisis exhaustivo de los datos para responder a cada pregunta de investigación. Para QA1, se evaluó si el paper presenta los objetivos de investigación de forma clara. Para QA2, se verificó si las condiciones para la evaluación del rendimiento fueron realizadas de forma justa y aceptable. Para QA3, se analizó si el paper emplea parámetros matemáticos para facilitar la comparación. Para QA4, se examinó si el paper estaba bien organizado. Para QA5, se determinó si los métodos empleados al analizar los resultados fueron clara y correctamente ejecutados. Para QA6, se identificó si el campo de estudio está bien definido. Finalmente, para QA7, se evaluó si los resultados de la experimentación fueron presentados de forma clara.

La información extraída para estas preguntas de evaluación se ha tabulado y presentado como datos cuantitativos, que se utilizaron para desarrollar una comparación estadística entre los distintos resultados de cada pregunta. Las estadísticas desarrolladas ayudaron a descubrir ciertos patrones de investigación, así como las direcciones de investigación llevadas a cabo en el intervalo de 2018 a 2024.

### Respuestas a preguntas de investigación

En la presente sección del artículo, se dará respuesta a las preguntas de investigación que fueron presentadas anteriormente 3.1. A través del análisis de los datos recopilados y la revisión de la literatura relevante, se abordarán los interrogantes planteados con el objetivo de proporcionar una comprensión profunda y detallada sobre el tema de estudio. Estas respuestas permitirán contextualizar los hallazgos obtenidos y evaluar las implicaciones de los resultados en los distintos campos.

### Visualización de búsqueda

Los mapas de redes son generalmente empleados para facilitar la visualización de algún dato presente en los resultados de la búsqueda de una determinada base de datos que nos permite comprender cómo estos trabajos de investigación se encuentran conectados entre sí. Es posible realizar múltiples visualizaciones ya sea en función a los autores, coautores, términos mencionados, citas entre artículos y otros.

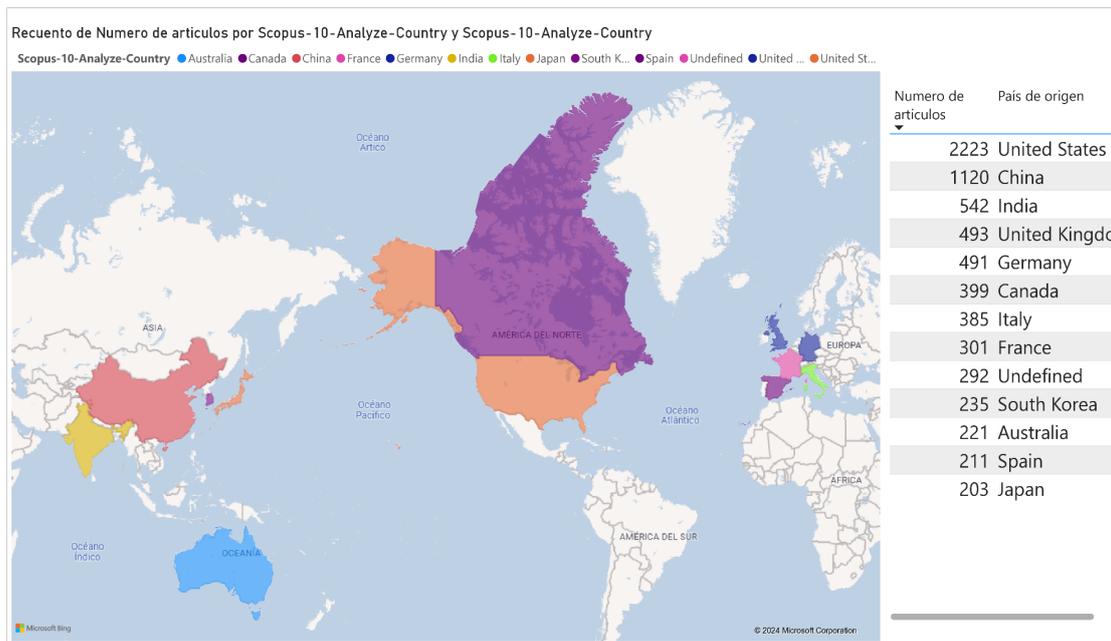
Respecto a la presente investigación sobre rendimiento de los sistemas operativos, se empleó el software VOSviewer para realizar el mapa de redes que determina los términos claves más utilizados dentro de los papers de investigación de los temas relevantes. Primero, en la Figura 4 se observan los resultados observados de la búsqueda dentro de la plataforma de IEE empleando términos clave de esta investigación. Siendo los términos más destacados *thread* (hilo en español), *schedules* (planificador u organizador) y *virtualización* (virtualización).



Adicionalmente, a través de la base de datos Scopus, se analizó el CSV proveniente de su motor de búsqueda sobre los artículos relevantes al tema de investigación de la presente revisión sistemática y su respectivo país de origen. Se empleó Power BI para realizar la

visualización de dicha data. Entre los países más destacados empleando los términos clave, están los Estados Unidos con más de 2000 artículos, China con más de 1000 artículos y la India con 542 artículos. Todo lo mencionado se puede evidenciar en la Figura 6.

**Figura 6**  
*Recuento de artículos respecto a su país de origen*



**RQ1: ¿Cómo el planificador de tareas afecta el rendimiento en cada sistema operativo?**

De acuerdo a los artículos

científicos empleados, existe información sobre el planificador de tareas dentro del sistema operativo. Esto se evidencia en la Tabla 6.

**Tabla 6**  
*RQ1 Respuestas*

ID	Base de datos	Fuentes	Cantidad (%)
1	IEEE	(Fan et al., 2020) (Singh et al., 2022) (Silva et al., 2019) (Boras et al., 2020) (Alfonso & Almeida, 2020)	56%
2	Arxiv	(Durbhakula, 2018)	11%
3	Dialnet	(Moreno Rivera, 2020)	11%
4	Google Académico	(Ungurean, 2020)	11%
5	SciELO	-	0%
6	Scopus	(Ramegowda & Lin, 2022)	11%

**RQ2: ¿Qué tipo de pruebas de rendimiento utilizan los estudios recolectados?**

Dentro del mundo de la informática y la computación, existen distintos métodos de medición del rendimiento del software en determinados hardwares. Generalmente, se emplean seis tipos de pruebas de rendimiento dentro de softwares empleados para realizar los denominados benchmarks.

El objetivo de las pruebas de rendimiento es el de medir la cantidad de

operaciones que, en este caso particular, un sistema operativo puede realizar en función a un tipo en concreto de tareas como podría ser procesamiento de información, cálculos, renderización de videos, entre otros. Adicionalmente, cada uno de estos campos son medidos con magnitudes de tiempo (segundos), latencia (milisegundos), fotogramas por segundo (fps), entre otros.

Los tipos de mediciones empleados en cada uno de los estudios recolectados en el presente artículo se ubican en la Tabla 7 a continuación.

**Tabla 7**  
*RQ2 Respuestas*

Nº	Base de datos	Fuentes	Artículos relevantes al tema (%)
1	Pruebas de carga	(Bastidas García et al., 2023) (Ortiz et al., 2024) (Singh et al., 2022) (Boras et al., 2020) (Alfonso & Almeida, 2020)(Malallah et al., 2021a) (Silva et al., 2019) (Ungurean, 2020)	64%
2	Pruebas de resistencia	(Malallah et al., 2021a)	7%
3	Pruebas de estrés	(Ungurean, 2020) (Fan et al., 2020) (Giatsintov et al., 2024)(Durbhakula, 2018) (Boras et al., 2020) (Arshad et al., 2021) (Chen & Yu, 2020)	50%
4	Pruebas de picos	(Giatsintov et al., 2024) (Reddi et al., 2022)	14%
5	Pruebas de volumen	(Reddi et al., 2022) (Alfonso & Almeida, 2020) (Silva et al., 2019)	21%
6	Pruebas de escalabilidad	(Durbhakula, 2018) (Silva et al., 2019)	14%

**RQ3: ¿Qué sistemas operativos son los más eficientes para la renderización de videos?**

Esta es un área principal dentro del estudio propuesto en el presente artículo. Generalmente, se utilizan magnitudes

de latencia, tiempo y fotogramas por segundo para calcular cómo un sistema operativo aprovecha cierto hardware dedicado a la renderización de video para lograr la visualización de imágenes bidimensionales y tridimensionales.

Los artículos que midieron las competencias de distintos sistemas operativos para la renderización de video

se encuentran en la Tabla 8 ubicada a continuación.

**Tabla 8**  
*RQ3 Respuestas*

SO / Fuente	IEEE	Arxiv	Dialnet	Google Académico	SciELO	Scopus
Windows	(Fan et al., 2020)			(Adekotujo et al., 2020) (Malallah et al., 2021b) (Kopel & Bozek, 2023)		
Linux	(Singh et al., 2022)			(Bastidas García et al., 2023) (Ortiz et al., 2024) (Ward, 2018)		
iOS	(Chivu, 2020)					
Android	(Arshad et al., 2021)	(Reddi et al., 2022) (Alfonso & Almeida, 2020)				
<b>TOTAL</b>	<b>25%</b>	<b>17%</b>	<b>8%</b>	<b>50%</b>	<b>0%</b>	<b>0%</b>

**RQ4: ¿Qué sistema operativo tiene mejor rendimiento para sistemas de servidores en red?**

La pregunta va referida a los sistemas que se emplean en los denominados servidores. Estas son computadoras que se encargan de almacenar y distribuir información a distancia a través de una red que puede

ser local o internet y son empleadas en distintos tipos de servicios como aplicaciones web, base de datos, entre otros.

Luego de la revisión de los artículos que fueron incluidos, se pueden observar las opiniones del mejor sistema operativo comúnmente empleado para servidores entre windows server, linux server y otros. Esto se evidencia en la Tabla 9.

**Tabla 9**  
*RQ4: Respuestas*

SO / Fuente	IEEE	Arxiv	Dialnet	Google Académico	SciELO	Scopus
Windows Server	-	-	-	(Jakić, 2021)	-	-
Linux Server	-	-	-	(Goyal et al., 2018) (Ward, 2018) (Ortiz et al., 2024) (Hamdani et al., 2022)	(Calle et al., 2018) (Vdovjak et al., 2020) (Darsh & Rahul, 2021)}	-
Otros	-	-	-	(Larrea & Barbalace, 2020)	-	-

## Resultados

Luego de extrapolar los resultados hallados dentro de los artículos seleccionados dentro del presente paper. Podemos extraer resultados de acuerdo a la información brindada por dichos papers en las preguntas de investigación propuestas en secciones anteriores.

### Resultados de RQ1

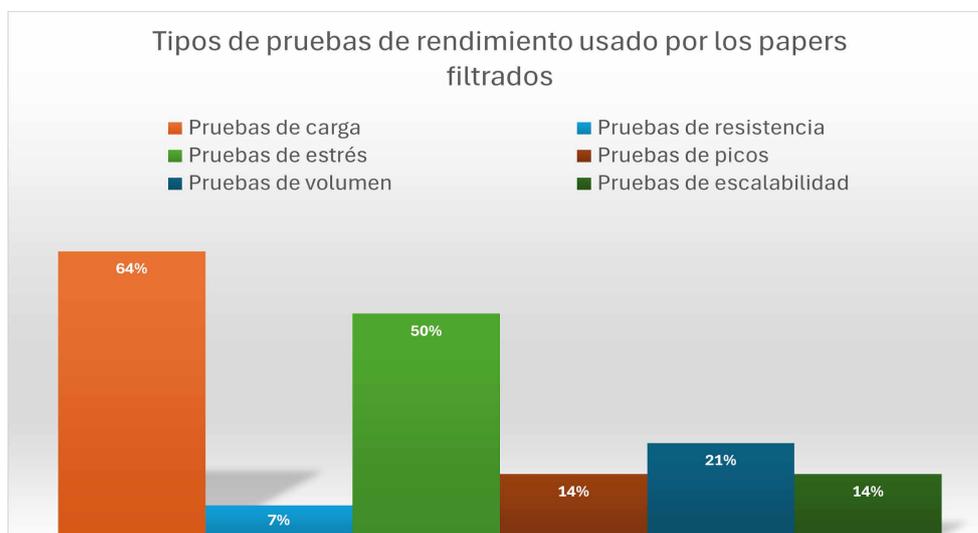
Según los artículos recolectados en la Tabla 6, los sistemas operativos se ven influenciados por el planificador de tareas de los siguientes modos:

- Optimización del tiempo
- Elección de prioridades
- Rapidez de ejecución de tareas
- Designación de hilos

En general, sostienen que el planificador de tareas es fundamental y hasta determinante para el rendimiento de tareas básicas en sistemas operativos de diversos tipos. Por ejemplo, en (Fan *et al.* 2020) habla de diferencias claras en cuanto a la latencia favoreciendo a Windows.

### Figura 7

*Tipos de pruebas de rendimiento usadas por cada paper filtrado*



Sin embargo, en cuanto a otras tareas cómo la gestión de redes para servidores, (Goyal, Ranawat, and Nayak 2018) Linux tiene una amplia ventaja respecto al de Windows con una mejor gestión de recursos de paquetes de red.

### Resultados de RQ2

Sobre RQ2 ¿Que tipo de pruebas de rendimiento utilizan los estudios recolectados?, podemos observar que en la distribución de los tipos de pruebas de rendimiento se distribuyen tal y como lo ilustra la Figura 5

RQ2 fue planteado con el objetivo de obtener las características de las pruebas empleadas dentro de los artículos y sus comparativas de sistemas operativos. Siendo las pruebas de carga y de estrés mayoritarias dentro de las pruebas de comparación de rendimiento.

Estos resultados son visibles dentro de la Figura 7. En el cuál se observa que los métodos más usados para las pruebas de rendimiento son las pruebas de carga con un 64% de los artículos y las pruebas de estrés con un 50%.

## Resultados de RQ3

Conocemos que la renderización de video es el proceso en que una computadora logra transformar datos en imágenes ya sea bidimensionales o tridimensionales que se proyectan dentro de un monitor.

En la siguiente tabla se enumeran las conclusiones de los artículos que se refieren a este tema y los argumentos que presentan respecto al resultado que los sistemas operativos listados muestran en la Tabla 10.

**Tabla 10**  
*RQ3: Resultados*

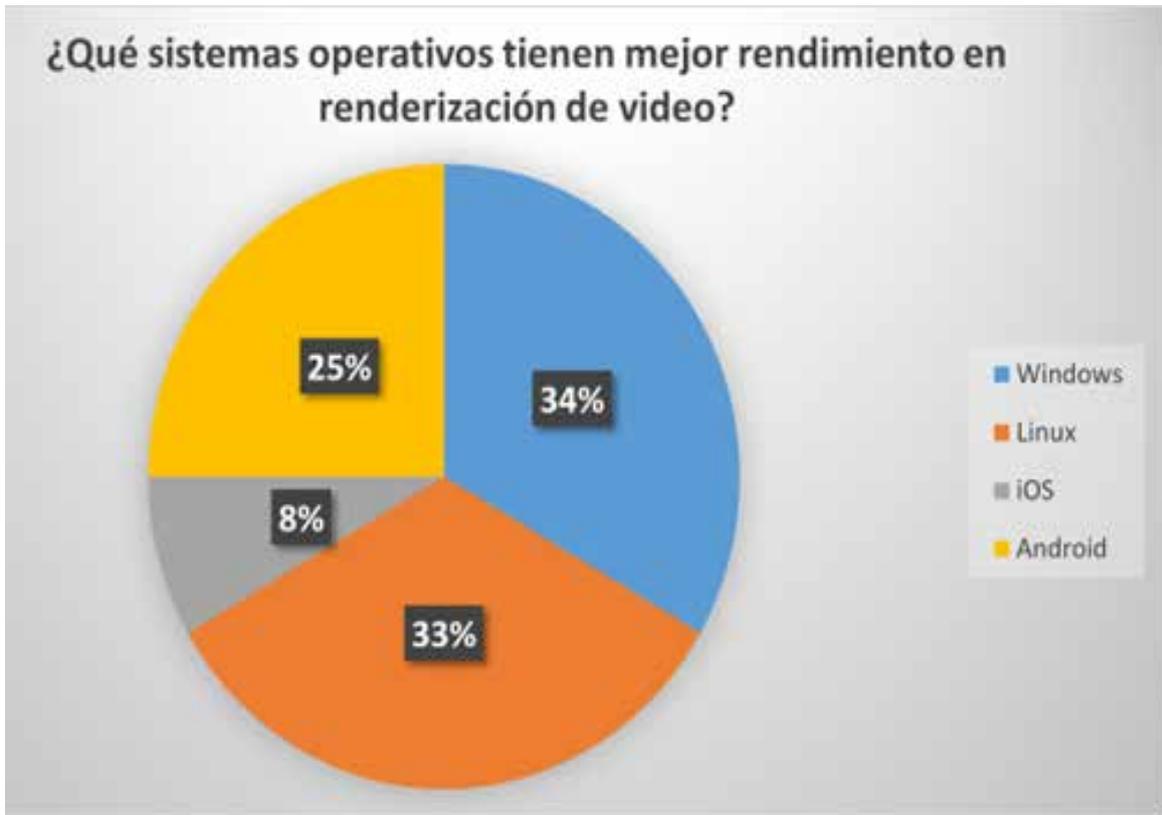
Sistema operativo	Artículos que lo respaldan	Puntos fuertes	Puntos débiles	Porcentaje
Windows	(Fan <i>et al.</i> , 2020) (Adekotujo <i>et al.</i> , 2020) (Malallah <i>et al.</i> , 2021b) (Kopel & Bozek, 2023)	<ul style="list-style-type: none"> <li>• Mayor compatibilidad con drivers</li> <li>• Robustez en su retrocompatibilidad</li> </ul>	<ul style="list-style-type: none"> <li>• Muchos procesos en segundo plano</li> <li>• Poco amigable con el programador</li> </ul>	34%
Linux	(Singh <i>et al.</i> , 2022) (Bastidas García <i>et al.</i> , 2023) (Ortiz <i>et al.</i> , 2024) (Ward, 2018)	<ul style="list-style-type: none"> <li>• Menor cantidad de tareas en segundo plano</li> <li>• Código abierto y colaboración</li> </ul>	<ul style="list-style-type: none"> <li>• Falta de soporte oficial en drivers</li> <li>• Falta de uniformidad entre distribuciones</li> </ul>	33%
iOS	(Chivu, 2020)	<ul style="list-style-type: none"> <li>• Gran nivel de uniformidad del sistema.</li> <li>• Mejor coordinación hardware</li> </ul>	<ul style="list-style-type: none"> <li>• Código muy cerrado al programador</li> <li>• Altísima dificultad al realizar pruebas</li> </ul>	8%
Android	(Arshad <i>et al.</i> 2021) (Reddi <i>et al.</i> 2022) (Alfonso & Almeida, 2020)	<ul style="list-style-type: none"> <li>• Empleo de arquitectura ARM eficiente en energía.</li> <li>• Facilidad para la realización de pruebas</li> </ul>	<ul style="list-style-type: none"> <li>• Muy limitada retrocompatibilidad</li> <li>• Hardware muy distinto entre sí</li> </ul>	25%

Adicionalmente, el siguiente gráfico estadístico de la Figura 8, expresa en la tabla anterior, de forma más clara, la

cantidad de artículos que cumplen RQ3 señalando la preferencia en los resultados de cada uno de los seleccionados.

### Figura 8

Resultado de los artículos que avalan a sistemas operativos.



### Resultados de RQ4

El objetivo de RQ4 es el de verificar el rendimiento de sistemas operativos enfocados a las redes. En suma, sistemas operativos que se emplean a nivel de servidor con los cuáles las computadoras domésticas y de oficina se comunican alrededor del mundo a través de la red ya sea para almacenar, solicitar o manejar información en distintas páginas y aplicaciones web.

Se puede observar en la Tabla 11 que existe una mayor predilección sobre Linux para servidores web gracias a un buen planificador de tareas. Debido a ello varios de los artículos presentados en la tabla mencionan el uso masivo de Linux server en una gran cantidad de servidores a nivel mundial. Una profundización de los resultados se puede observar a continuación.

**Tabla 11**  
*RQ4: Resultados*

SO	Artículos que lo respaldan	Puntos fuertes	Puntos débiles	Porcentaje
Windows Server	(Vdovjak <i>et al.</i> , 2020)	<ul style="list-style-type: none"> <li>• Rapidez en tareas básicas de descarga</li> <li>• Similitudes con Windows de escritorio</li> </ul>	<ul style="list-style-type: none"> <li>• Mala gestión de tareas</li> <li>• Alto consumo de recursos</li> </ul>	11%
Linux server	(Goyal <i>et al.</i> , 2018) (Ward, 2018) (Calle <i>et al.</i> , 2018) (Ortiz <i>et al.</i> , 2024) (Hamdani <i>et al.</i> , 2022) (Vdovjak <i>et al.</i> , 2020) (Darsh & Rahul, 2021)	<ul style="list-style-type: none"> <li>• Excelente planificador de tareas</li> <li>• Acceso abierto a código</li> </ul>	<ul style="list-style-type: none"> <li>• Muy alta variedad de alternativas dentro de Linux</li> </ul>	78%
Otros	(Larrea & Barbalace, 2020)	<ul style="list-style-type: none"> <li>• Excelente ahorro energético</li> </ul>	<ul style="list-style-type: none"> <li>• Poca documentación y distribución</li> </ul>	11%

En la Figura 9 se aprecia una mejor visualización del porcentaje de artículos que da mayor respaldo a Linux como mejor sistema operativo en servidores.

**Figura 9**  
*Conclusiones de artículos respecto a RQ4*



## Discusión

Según Fan *et al.* (2020), la razón del mejor rendimiento de Linux en comparación a Windows es el su mejor planificador de tareas para la mayoría de escenarios por lo que el planificador de tareas tiene un innegable rol para determinar el desempeño con los sistemas operativos. Sin embargo, (Boras, Balen, and Vdovjak 2020) menciona que también hay una cierta importancia en la compatibilidad que este sistema operativo tiene con el hardware.

Un punto clave es que Linux demostró ser significativamente mejor en la gestión de redes de servidores en comparación con Windows, lo cual es compatible con conclusiones anteriores. Estos resultados llevarían a la consciencia de las ventajas de usar Linux en las redes (referencias). Sin embargo, Windows fue líder en cuanto a la compatibilidad con drivers y la retrocompatibilidad, la cual es una característica indispensable en los ambientes en el que se usan diferentes aplicaciones y hardware.

En el caso de la renderización de video, (Fan *et al.* 2020) menciona que Windows proporciona una mejor compatibilidad tanto con hardware nuevo y antiguo. Adicionalmente, (Bastidas García, Vargas Moreno, and Osuna Cerecer 2023) respalda lo anteriormente mencionado debido a que se puede observar en su comparativas que en la mayoría de software ejecutado en las computadoras Windows posee una ventaja mayor incluso con una latencia menor por 0.5 milisegundos y un mejor uso de la GPU. Las excepciones a esta tendencia solo se dieron cuando el

software evaluado presenta una versión nativa específica para la versión de Linux empleado dentro de las pruebas de estrés.

Dejando a un lado Windows y Linux en sistemas de escritorio, tanto (Arshad *et al.* 2021), (Reddi *et al.* 2022) como (Afonso and Almeida 2020) sostienen que en cuanto a sistemas basados en la arquitectura de procesadores ARM, sostienen que Android es el mejor en cuanto a renderización de video en esta categoría de dispositivos móviles, aunque con un énfasis en la mayor facilidad de acceder al código de este sistema ya que es más abierto y amigable con el programador comparado con iOS. Sin embargo, (Chivu 2020) sostiene que iOS tendría mejor rendimiento en este campo debido a que, contrario a ser un obstáculo, la poca disposición de iOS a más tipo de smartphones significa por el contrario una ventaja mayor debido a que existe un control total sobre el hardware sobre el que el sistema operativo se manejaría.

Finalmente, (Goyal, Ranawat, and Nayak 2018), (Ward 2018), (Ortiz *et al.* 2024) (Hamdani *et al.* 2022), (Calle *et al.* 2018) (Vdovjak, Balen, and Nenadić 2020), (Darsh and Rahul 2020) dan su veredicto en favor a Linux server debido a que la mayor ventaja que tiene sobre Windows es también la misma ventaja que tiene Linux para escritorio que es su planificador de tareas que, en el caso particular de los requerimientos para un buen servidor, logra administrar mejor los request o requerimientos de múltiples clientes simultáneos aprovechando mejor los hilos en dicha tarea dando como resultado una mayoritaria preferencia a este sistema operativo en este campo.

## Conclusiones

Respecto a la presente investigación, la metodología PRISMA empleada para llevar a cabo la revisión sistemática fue útil para obtener los mejores artículos disponibles en la actualidad sobre el rendimiento de sistemas operativos en los puntos tratados anteriormente.

Podemos concluir, inicialmente, el alto grado de magnitud que influye un planificador de tareas respecto a los resultados finales que podemos obtener en sistemas operativos. Sin embargo, necesita de una gran afinidad con el hardware a través de drivers optimizados que permitan obtener el mayor rendimiento posible de esto.

También, respecto al tipo de pruebas usadas para medición de rendimiento, las pruebas de estrés son las más usadas por la mayoría de estudios. Sin embargo, debemos considerar que estas pruebas, generalmente, se remiten a los peores casos de mayor carga de tareas, situaciones que no necesariamente se dan en nuestras computadoras de escritorio o dispositivos móviles. Por lo que las pruebas realizadas por los otros autores fueron tomadas de forma referencial. Además que no debemos perder de vista factores externos que puedan afectar el hardware como tiempo de uso, temperatura externa, subprocesos en ejecución, humedad del ambiente, hardware disponible, entre otros. Todo esto se menciona con el fin de evitar sesgos al momento de revisar estos estudios.

Respecto a dos tareas específicas uno por parte del cliente y otro por parte del servidor dan cuenta que un factor

crucial que afecta el rendimiento de un sistema operativo son el cómo este escala con el tiempo y sus tareas en segundo que deben ser manejados de la mejor forma por el planificador de tareas. No existe como tal un sistema operativo “universal” que abarque a la perfección todo tipo de tareas. Sino que tal y cómo se abordó en la discusión, Windows destaca mucho por el lado de renderización de video mientras que Linux tiene una robusta arquitectura que se alinea con los requerimientos de un servidor. Además tanto Android como iOS tiene su fuerte en su arquitectura ARM siendo este capaz de realizar tareas optimizando mucho más el uso de energía que en sistemas de escritorio además de la portabilidad de los dispositivos en los que estos se encuentran instalados en su mayoría.

Este estudio aportó tanto una vista general de cómo estos estudios de evaluación de rendimiento en sistemas operativos son realizados, como el de sus conclusiones. Se recomienda entonces para los futuros estudios que empleen este mismo estado del arte evitar factores externos que afecten al hardware y como tal de posibles riesgos a sesgos en la realización de *benchmarks*.

En cuanto al uso de sistemas operativos, tanto Windows como Android y iOS tienen fortalezas por parte del cliente mientras que Linux, aunque cada vez más cerca de Windows en tareas comunes de escritorio, tiene su mayor fuerte en la gestión de requerimientos dentro de un servidor que puede ser implementado por la gran mayoría de empresas gracias a que Linux tiene un código abierto que en resumen es más amigable al acceso del programador.

## Referencias

- Afonso, S., & Almeida, F. (2020). RANCID: Reliable benchmarking on Android platforms. *IEEE Access*, 8, 143342–143358. <https://doi.org/10.1109/ACCESS.2020.3014533>
- Arshad, M. R., Hussain, M., Tahir, H., Qadir, S., Memon, F. I. A., & Javed, Y. (2021). Forensic analysis of Tor Browser on Windows 10 and Android 10 operating systems. *IEEE Access*, 9. <https://doi.org/10.1109/ACCESS.2021.3119724>
- Bastidas García, J. M., Vargas Moreno, L. F., & Osuna Cerecer, E. R. (2023). Análisis de rendimiento entre Linux y Windows en la ejecución de videojuegos utilizando diferentes niveles de hardware. *ReDTIS*, 7(1), 9–14. <https://doi.org/10.61530/redtis.vol7.n1.2023.168.9-14>
- Boras, M., Balen, J., & Vdovjak, K. (2020). Performance evaluation of Linux operating systems. En *2020 International Conference on Smart Systems and Technologies (SST)* (pp. 115–120). IEEE. <https://doi.org/10.1109/SST49455.2020.9264055>
- Calle, M. A., Tovar, J. D., Castaño-Pino, Y. J., & Cuéllar, J. C. (2018). Comparación de parámetros para una selección apropiada de herramientas de simulación de redes. *CIT Inform. Technol.*, 29(6), 253–266. <https://doi.org/10.4067/S0718-07642018000600253>
- Chen, K.-S., & Yu, C.-M. (2020). Fuzzy test model for performance evaluation matrix of service operating systems. *Computers & Industrial Engineering*, 140(106240). <https://doi.org/10.1016/j.cie.2019.106240>
- Chivu, D. C. (2020). Evolución y comparación de los sistemas operativos en dispositivos inteligentes. *MoleQla: Revista de Ciencias de la Universidad Pablo de Olavide*, 39, 61–633. <https://dialnet.unirioja.es/servlet/articulo?codigo=7618124>
- Darbord, G., Verhaeghe, B., Etien, A., Anquetil, N., Shatnawi, A., Seriai, A., Derras, M., Pautasso, C., & Zimmermann, O. (2023). Migrating the communication protocol of client-server applications. *IEEE Software*, 40, 11–18. <https://doi.org/10.1109/MS.2023.3263019>
- Darsh, P., & Rahul, R. (2020). Performance analysis of network anomaly detection systems in consumer networks. En *Advances in Intelligent Systems and Computing*. [https://doi.org/10.1007/978-981-15-4218-3\\_21](https://doi.org/10.1007/978-981-15-4218-3_21)
- Durbhakula, M. (2018). OS scheduling algorithms for improving the performance of multithreaded

- workloads. En *Advances in Intelligent Systems and Computing* (pp. 194–208). [https://doi.org/10.1007/978-3-030-22871-2\\_15](https://doi.org/10.1007/978-3-030-22871-2_15)
- Fan, W.-C., Wong, C.-S., Lee, W.-K., & Hwang, S.-O. (2020). Comparison of interactivity performance of Linux CFS and Windows 10 CPU schedulers. En *2020 International Conference on Green and Human Information Technology (ICGHIT)* (pp. 31–34). IEEE. <https://doi.org/10.1109/ICGHIT49602.2020.00014>
- Goyal, K., Ranawat, K. R. S., & Nayak, N. (2018). Operational distinctions between Linux and Windows. *International Journal of Scientific Research in Science and Technology*. <https://doi.org/10.32628/ijrst18401152>
- Giatsintov, A., Mamrosenko, K., & Bazhenov, P. (2023). Architecture of graphics system with 3D acceleration support for embedded operating systems. *Tsinghua Science and Technology*. <https://doi.org/10.26599/TST.2023.9010045>
- Hamdani, S. W. A., Abbas, H., Janjua, A. R., Shahid, W. B., Amjad, M. F., Malik, J., Murtaza, M. H., Atiquzzaman, M., & Khan, A. W. (2022). Cybersecurity standards in the context of operating systems: Practical aspects, analysis, and comparisons. *ACM Computing Surveys*, 54(3), 1–36. <https://dl.acm.org/doi/10.1145/3442480>
- Ishlakhuddin, F., & Sn, A. (2021). Ontology-based chatbot to support monitoring of server performance and security by rule-base. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*. <https://doi.org/10.22146/IJCCS.58588>
- Jakić, E. P. (2021). Comparative analysis of the impact of server operating systems on website performance. *Sinteza 2021: International Scientific Conference on Information Technology and Data Related Research*, 180–186. <https://doi.org/10.15308/Sinteza-2021-180-186>
- Kopel, M., & Michal, B. (2023). Is Proton good enough? A performance comparison between gaming on Windows and Linux. *Springer*. <https://doi.org/10.1007/9>
- Larrea, J., & Barbalace, A. (2020). The server kernel operating system. *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, 13–18. <https://doi.org/10.1145/3378679.3394537>
- Leiserson, C., Thompson, N. C., Emer, J., Kuszmaul, B. C., Lamson, B., Sanchez, D. S., & Schardl, T. (2020). There's plenty of room at the top: What will drive computer performance after Moore's law? *Science*, 368. <https://doi.org/10.1126/science.aam9744>
- Lin, Z.-G., & Ding, H. (2021). Research on multithread programming method based on Java programming. *2021*

- 3rd International Conference on Artificial Intelligence and Advanced Manufacture.* <https://doi.org/10.1145/3495018.3501112>
- Malallah, H., Zeebaree, S. R. M., Zebari, R. R., Sadeeq, M. A. M., Ageed, Z. S., Ibrahim, I. M., Yasin, H. M., & Merceedi, K. J. (2021). A comprehensive study of kernel (issues and concepts) in different operating systems. *Asian Journal of Research in Computer Science*, 16–31. <https://doi.org/10.9734/ajrcos/2021/v8i330201>
- Moreno Rivera, E. A. (2020). Percepción del uso y seguridad en los sistemas operativos informáticos. *RevTECHNO*, 8(2), 87–96. <https://doi.org/10.37467/gka-revtechno.v8.2273>
- Odun-Ayo, I., Okokpujie, K., Oputa, K., Ogbu, H., Emmanuel, F., Shofadekan, A., & Okuazun, G. (2021). Comparative study of operating system quality attributes. *IOP Conference Series: Materials Science and Engineering*, 1107. <https://doi.org/10.1088/1757-899X/1107/1/012061>
- Ortiz, J. R. M., García, J. A., Ochoa Rivera, C. A., Rojas Luna, A. Y., & Cruz Landa, A. J. (2024). Análisis comparativo del rendimiento de servicios de red en plataformas Windows y Linux. *TERC*, 11(1), 27–34. <https://www.terc.mx/index.php/terc/article/view/369>
- Pal, J. S., Yadav, S., Chauhan, V. K., Bhatia, J. K., & Singh, P. K. (2022). Experimental analysis of performance paradigms for real-time operating systems (RTOS). *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*. <https://doi.org/10.1109/SMART55829.2022.10046685>
- Page, M., McKenzie, J., Bossuyt, P., Boutron, I., Hoffmann, T., Mulrow, C., & Shamseer, L. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *Journal of Clinical Epidemiology*. <https://doi.org/10.1016/j.jclinepi.2021.03.001>
- Pfeiffer, R. (2020). What constitutes software? An empirical, descriptive study of artifacts. *2020 IEEE/ACM 17th International Conference on Mining Software Repositories (MSR)*, 481–491. <https://doi.org/10.1145/3379597.3387442>
- Prasad, C., Veena, G. S., Agrawal, C., & Srivastava, R. (2014). Achieving low latency networks through high performance computing. *International Journal of Advanced Research in Computer and Communication Engineering*, 8753–8756. <https://doi.org/10.17148/IJARCCCE.2014.31220>
- Qian, Z., Kavvos, G. A., & Birkedal, L. (2020). Client-server sessions in linear logic. *Proceedings of the ACM on Programming Languages*, 5, 1–31. <https://doi.org/10.1145/3473567>

- Ramegowda, D., & Lin, M. (2022). Energy efficient mixed task handling on real-time embedded systems using FreeRTOS. *Journal of Systems Architecture*, 131(102708), 102708. <https://doi.org/10.1016/j.sysarc.2022.102708>
- Reddi, V. J., Kanter, D., Mattson, P., Duke, J., Nguyen, T., Chukka, R., Shiring, K., *et al.* (2022). MLPerf mobile inference benchmark. *arXiv preprint*. <https://arxiv.org/abs/2012.02328>
- Resch, S., & Karpuzcu, U. R. (2019). Benchmarking quantum computers and the impact of quantum noise. *ACM Computing Surveys (CSUR)*, 54, 1–35. <https://doi.org/10.1145/3464420>
- San, A. M., & Yakunin, A. (2019). Hardware implementation of floating-point operating devices by using IEEE-754 binary arithmetic standard. *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus)*, 1624–1630. <https://doi.org/10.1109/EICONRUS.2019.8656775>
- Silva, M., Cerdeira, D., Pinto, S., & Gomes, T. (2019). Operating systems for internet of things low-end devices: Analysis and benchmarking. *IEEE Internet of Things Journal*, 6(6), 10375–10383. <https://doi.org/10.1109/jiot.2019.2939008>
- Soualhia, M., Khomh, F., & Tahar, S. (2020). A dynamic and failure-aware task scheduling framework for Hadoop. *IEEE Transactions on Cloud Computing*, 8, 553–569. <https://doi.org/10.1109/TCC.2018.2805812>
- Tanenbaum, A. S., & Bos, H. (2024). *Modern operating systems, global edition* (5th ed.). Pearson Education.
- Ungurean, I. (2020). Timing comparison of the real-time operating systems for small microcontrollers. *Symmetry*, 12(4), 592. <https://doi.org/10.3390/sym12040592>
- Vdovjak, K., Balen, J., & Nenadić, K. (2020). Experimental evaluation of desktop operating systems networking performance. *International Journal of Electrical and Computer Engineering*. <https://doi.org/10.32985/IJECES.11.2.2>
- Ward, B. (2018). Why SQL Server on Linux. *Springer Link*, 1–22. [https://doi.org/10.1007/978-1-4842-4128-8\\_1](https://doi.org/10.1007/978-1-4842-4128-8_1)
- Yao, A., Sun, P., Yang, S., & Li, D. (2020). Evolution of function-call network reliability in Android operating system. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(4), 1264–1275. <https://doi.org/10.1109/TCSI.2020.2972995>

