



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA DE COMPUTACIÓN

**ANÁLISIS DEL CONSUMO DE ENERGÍA DEL PROCESADOR CUANDO
TRABAJA EN ESCENARIOS VIRTUALIZADOS**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Ciencias de la Computación

AUTOR: RAINER VLADIMIR VERGARA NAZARENO
TUTOR: JORGE ENRIQUE LÓPEZ LOGACHO

Quito - Ecuador
2025

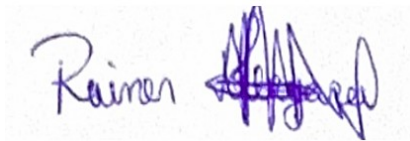
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Rainer Vladimir Vergara Nazareno con documento de identificación N° 0850105750 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total parcial el presente trabajo de titulación.

Quito, 28 de julio del 2025

Atentamente,



Rainer Vladimir Vergara Nazareno
0850105750

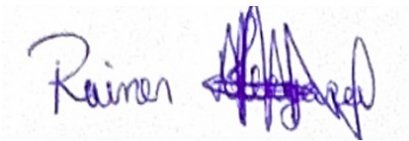
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Yo, Rainer Vladimir Vergara Nazareno con documento de identificación N° 0850105750, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Artículo Académico: “Análisis del consumo de energía del procesador cuando trabaja en escenarios virtualizados”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 28 de julio del 2025

Atentamente,



Rainer Vladimir Vergara Nazareno
0850105750

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Jorge Enrique López Logacho con documento de identificación N° 1712082484, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: ANÁLISIS DEL CONSUMO DE ENERGÍA DEL PROCESADOR CUANDO TRABAJA EN ESCENARIOS VIRTUALIZADOS, realizado por Rainer Vladimir Vergara Nazareno, con documento de identificación N.º 0850105750, obteniendo como resultado final el trabajo de titulación bajo la opción Artículo Académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 28 de julio del 2025

Atentamente,



Ing. Jorge Enrique López Logacho, MSc
1712082484

DEDICATORIA

Dedico este trabajo a todas las versiones de mí que alguna vez pensaron en rendirse. A ese niño de 15 años que sufrió en silencio el bullying, pero que encontró en el dolor la fuerza para soñar.

A mis padres, por su esfuerzo incansable, por cada sacrificio silencioso y por creer en mí incluso cuando yo dudaba.

A mi hermano mayor, que ya no está físicamente, pero vive en cada paso que doy. A mi abuela, por sus abrazos, sus palabras sabias y por enseñarme que los retos se enfrentan con el corazón en alto.

A mi fraternidad universitaria Santa María del Paraíso, por ser mi segunda familia, por acompañarme en cada etapa, por enseñarme el valor del compañerismo y brindarme un hogar lejos de casa.

Este logro es de ustedes, y por ustedes seguiré adelante.

Rainer Vladimir Vergara Nazareno

AGRADECIMIENTOS

Quiero agradecer a todas las personas que, de una u otra forma, me tendieron la mano en los momentos más difíciles. A quienes me ofrecieron una palabra de aliento cuando más lo necesité, a quienes supieron escucharme sin juzgarme, y a quienes creyeron en mí incluso cuando yo estaba al borde de rendirme.

Agradezco profundamente a mi universidad y a cada docente que aportó a mi formación. Mudarme a la capital no fue fácil; alejarme de mis raíces, mi gente y mi rutina fue un proceso doloroso, pero necesario. Viví noches de angustia, pérdidas familiares irreparables y momentos de soledad. Sin embargo, todo ese proceso me ayudó.

El 15 de mayo fue un día que marco mi vida para siempre: la felicidad de haber sido aceptado en una institución bancaria privada se mezcló con la profunda tristeza de perder a mi hermano. Aprendí que la vida puede cambiar en un segundo, pero también que el amor y el recuerdo pueden ser motores poderosos.

Gracias también a mi fraternidad universitaria, que se convirtió en mi segunda familia durante todos estos años. Por cada abrazo, cada palabra de aliento, por las risas compartidas y por enseñarme que la hermandad también se construye lejos del hogar.

Quiero expresar un agradecimiento muy especial a mi tutor, por su guía, paciencia y apoyo constante a lo largo de este proceso. Sus observaciones y enseñanzas fueron fundamentales para la culminación de este trabajo.

Gracias a mi familia, a mis amigos verdaderos, y a todos los que, sin saberlo, me dieron fuerzas para llegar hasta aquí.

Rainer Vladimir Vergara Nazareno

ANÁLISIS DEL CONSUMO DE ENERGÍA DEL PROCESADOR CUANDO TRABAJA EN ESCENARIOS VIRTUALIZADOS

ANALYSIS OF PROCESSOR POWER CONSUMPTION WHEN WORKING IN VIRTUALIZED SCENARIOS

Rainer Vergara - Nazareno ¹, Jorge López - Logacho ²

Resumen

La virtualización permite ejecutar múltiples máquinas virtuales sobre un único equipo físico, optimizando el uso del hardware. Sin embargo, plantea interrogantes sobre su impacto en el consumo energético. En el presente artículo se analiza el comportamiento energético de los procesadores al comparar entornos físicos y virtualizados, aplicando una metodología basada en pruebas de rendimiento con cargas computacionales paralelas, generadas mediante código en Python utilizando el módulo multiprocessing. Estas cargas se ejecutaron en entorno físico como en máquinas virtualizadas gestionadas por distintos hipervisores. Durante las pruebas se recolectaron métricas clave como el uso del CPU, frecuencia en MHz, memoria utilizada y consumo energético en W. Para ello se utilizaron herramientas de monitoreo de bajo nivel, capaces de acceder a sensores del sistema y registrar datos en tiempo real sobre el consumo energético, la carga computacional y la potencia utilizada por el procesador. En los entornos virtualizados, se emplearon scripts en Python con la biblioteca psutil, lo que permitió extraer, analizar y validar los resultados obtenidos de manera precisa. Se evaluaron los hipervisores VirtualBox, VMware, KVM e Hyper-V sobre los sistemas operativos Ubuntu 20.04 y Windows 10. Los resultados mostraron una reducción en el consumo energético en entornos virtualizados, alcanzando reducciones de hasta un 82 %. Esto demuestra que la virtualización es una estrategia eficiente para la gestión energética, ofreciendo beneficios importantes para la toma de decisiones en contextos académicos, empresariales y de infraestructura tecnológica.

Palabras clave: consumo energético, virtualización, hipervisores, multiprocessing, Python, eficiencia energética, psutil.

Abstract

Virtualization allows multiple virtual machines to run on a single physical computer, optimizing hardware utilization. However, it raises questions about its impact on energy consumption. This article analyzes the energy performance of processors by comparing physical and virtualized environments. It applies a methodology based on performance tests with parallel computational loads generated using Python code and the multiprocessing module. These loads were executed both in the physical environment and on virtualized machines managed by different hypervisors. During the tests, key metrics such as CPU utilization, frequency in MHz, memory used, and energy consumption in W were collected. Low-level monitoring tools were used to access system sensors and record real-time data on energy consumption, computational load, and processor power. In virtualized environments, Python scripts with the psutil library were used, allowing the results to be accurately extracted, analyzed, and validated. VirtualBox, VMware, KVM, and Hyper-V hypervisors were evaluated on the Ubuntu 20.04 and Windows 10 operating systems. The results showed a reduction in energy consumption in virtualized environments, reaching reductions of up to 82%. This demonstrates that virtualization is an efficient energy management strategy, offering significant benefits for decision-making in academic, business, and technological infrastructure contexts.

Keywords: energy consumption, virtualization, hypervisors, multiprocessing, Python, energy efficiency, psutil.

¹ Estudiante de la Carrera de Computación, Universidad Politécnica Salesiana

² Docente de la Carrera de Computación, Universidad Politécnica Salesiana

Autor para correspondencia: rainernazareno13@gmail.com

1. Introducción

El consumo energético en los centros de datos se ha convertido en una prioridad clave para reducir costos operativos y minimizar el impacto ambiental. En este contexto, la virtualización emerge como una tecnología esencial que permite ejecutar múltiples máquinas virtuales en un solo servidor físico, optimizando el uso de los recursos de hardware y reduciendo el consumo energético general.

1.1. Trabajos Relacionados

Lopez et al. [1] introduce que el procesador o CPU es uno de los componentes con mayor demanda energética, ya que ejecuta todas las instrucciones del sistema operativo y aplicaciones. Su consumo depende de factores como el número de núcleos, la frecuencia de reloj, la arquitectura y la carga de trabajo. Pessolani et al. [2], propone que la virtualización permite consolidar múltiples cargas de trabajo en menos servidores físicos mediante el uso de hipervisores. Esto reduce significativamente la cantidad de hardware necesario. No obstante, su impacto energético no es uniforme y puede variar según el tipo de hipervisor, la configuración del sistema y la naturaleza de las tareas ejecutadas.

El cómputo paralelo ha sido una de las soluciones más viables para abordar problemas de alta demanda computacional, especialmente en contextos donde el tiempo de respuesta es crítico. Según Tinetti y De Giusti et al. [3], aplicaciones como la inteligencia artificial, las redes neuronales, el modelado de fluidos, la visualización científica, el procesamiento de imágenes y el análisis de grandes volúmenes de datos requieren necesariamente de procesamiento paralelo para garantizar eficiencia y un rendimiento aceptable.

Para evaluar el impacto energético de la virtualización, es fundamental trabajar con datos medibles. Métricas como el consumo en vatios, la utilización del CPU y la temperatura del sistema permiten realizar comparaciones objetivas entre entornos físicos y virtualizados. En un sistema físico cada servidor opera de forma independiente, consumiendo energía constantemente, en cambio, los entornos virtuales permiten consolidar múltiples cargas de trabajo en un solo servidor físico, lo que puede generar un ahorro energético estimado hasta un 60%, dependiendo del tipo de carga.

Velez et al. [4], según los hipervisores desempeñan un papel central en la eficiencia energética de los entornos virtualizados. VirtualBox, presenta un rendimiento limitado. VMware se destaca en entornos empresariales por su optimización y capacidad de gestión. KVM, como hipervisor ofrece un alto rendimiento y eficiencia energética. Hyper-V, brinda una buena administración de recursos, siendo una opción sólida para entornos de Windows. Cada hipervisor [5] tiene características particulares que afectan el consumo energético, y su elección debe basarse en la compatibilidad con el sistema operativo, tipo de carga y necesidades del entorno.

La virtualización, optimiza el uso de recursos mediante particiones lógicas y máquinas virtuales gestionadas

por hipervisores. En 1999, VMware implementó el primer hipervisor para arquitectura x86, facilitando la adopción masiva de esta tecnología. La evolución a arquitecturas x64 con soporte para multihilo ha potenciado el rendimiento y la escalabilidad de entornos virtualizados. Actualmente, la virtualización es clave para infraestructuras empresariales y la base del paradigma Cloud Computing, cuya automatización y escalabilidad redefinen la gestión de servicios TI [6].

Según Vayadande y Raut et al. [7], expone que la medición del consumo energético se realizó mediante herramientas de software libre y bibliotecas que permiten el monitoreo del uso del procesador, la memoria, la temperatura y otros sensores del sistema en tiempo real. En entornos físicos, se accedió a datos de sensores integrados en el sistema para registrar la energía utilizada durante las pruebas. En los entornos virtualizados, se utilizaron scripts desarrollados en Python con bibliotecas como psutil [8], que proporcionan información sobre el comportamiento del sistema operativo, permitiendo correlacionar la carga computacional con el uso de recursos. Este enfoque combinado facilitó una evaluación precisa del impacto energético en ambos entornos.

El procesamiento paralelo y el multiprocesamiento permiten dividir problemas numéricos complejos en tareas más pequeñas que pueden ejecutarse simultáneamente en arquitecturas multinúcleo, optimizando así el tiempo de cómputo. Lenguajes como Python ofrecen herramientas robustas para implementar este tipo de algoritmos, gracias a su ecosistema de bibliotecas especializadas que soportan la ejecución paralela. Aunque en el pasado Python enfrentaba limitaciones como el Global Interpreter Lock o GIL, actualmente dispone de soluciones eficientes para aplicaciones de alto rendimiento en áreas como multimedia, seguridad informática, computación científica y algoritmos [9].

Este artículo científico está estructurado en cuatro secciones. En la Sección 2, se detalla el diseño experimental, los entornos de prueba físicos y virtualizados, y las herramientas utilizadas para la recolección y estimación de consumo energético. En la Sección 3 se presentan y analizan los datos obtenidos en términos de consumo energético, destacando las diferencias entre hipervisores y sistemas operativos. La Sección 4 resume los principales hallazgos del estudio y sugiere líneas de investigación futura.

2. Materiales y Métodos

2.1. Tipos de Hipervisores y su Relación con el Consumo Energético del Procesador

Clasificación de Hipervisores: Los hipervisores se clasifican en dos tipos principales, según su interacción con el hardware.

- **Tipo 1:** Se ejecutan directamente sobre el hardware del host sin necesidad de un sistema operativo intermediario. Estos hipervisores ofrecen un mejor

rendimiento y eficiencia, siendo utilizados comúnmente en entornos de servidores y centros de datos. Ejemplos representativos incluyen *KVM*, *Hyper-V* y *VMware ESXi* [10, 11].

- **Tipo 2:** Funcionan sobre un sistema operativo ya instalado, lo que implica una capa adicional entre el hardware y las máquinas virtuales. Son frecuentemente utilizados en entornos de desarrollo o pruebas. Ejemplos típicos son *VirtualBox* y *VMware Workstation* [10, 11].

El proceso metodológico se desarrolló en tres etapas donde la primera fue la preparación de los entornos de prueba, luego la ejecución controlada de las cargas computacionales y recolección de datos, y por último el análisis comparativo del consumo energético entre entornos físicos y virtualizados. La Fig. 1 resume el proceso metodológico aplicado.

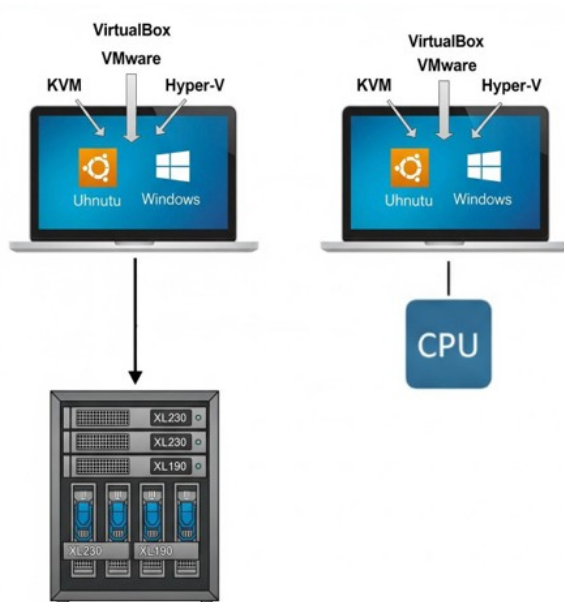


Figura 1. Metodología para el análisis comparativo de los entornos virtualizados y físicos.

El presente estudio implementó procesamiento en paralelo mediante el módulo `multiprocessing` de Python, permitiendo ejecutar múltiples procesos simultáneamente en distintos núcleos del procesador. Esta técnica se alinea con el modelo de paralelismo de memoria compartida, aprovechando la arquitectura multinúcleo para generar una carga computacional intensiva controlada. De esta forma, se garantizó una utilización efectiva de los recursos del CPU durante las pruebas energéticas.

2.2. Configuración de entornos físicos y virtualizados

El entorno físico consistió en una estación de trabajo equipada con procesador Intel Core i7-7500U, 12 GB de SDRAM DDR4-2133 y con su respectivo almacenamiento.

En este sistema se ejecutaron cargas de trabajo directamente sin virtualización a través de hipervisores instalados localmente como: *VirtualBox*, *VMWare*, *KVM* e *Hyper-V*. En todos los casos, se ejecutó el mismo código de carga computacional paralela utilizando por Python y se recolectaron los datos energéticos mediante *Intel Power Gadget*, herramienta especializada para medir el consumo del procesador en entornos Windows.

Para los entornos virtualizados se utilizaron hipervisores como *VirtualBox*, *VMWare*, *KVM* e *Hyper-V*, instalados sobre los sistemas operativos *Ubuntu 20.04 LTS* y *Windows 10*. Cada hipervisor alojó una máquina virtual configurada con 2 núcleos lógicos y 12 GB de RAM. Las pruebas se realizaron de forma individual para garantizar el consumo de la carga.

La Tabla 1, garantiza una comparación objetiva entre los hipervisores, todas las máquinas virtuales fueron configuradas con los mismos parámetros:

Tabla 1. Configuración de las máquinas virtuales para la evaluación de hipervisores

Parámetro	Valor asignado
Arquitectura	x86_64 (64 bits)
Núcleos lógicos (vCPU)	2
Memoria RAM	4 GB
Sistemas operativos	Ubuntu 20.04 LTS / Windows 10

Esta configuración asegura que las diferencias en el consumo energético observadas se deban exclusivamente al funcionamiento del hipervisor y no a variaciones en los recursos asignados a las máquinas virtuales.

2.3. Herramientas y software empleados

Para la ejecución de las pruebas, monitoreo y análisis del consumo energético del procesador, se utilizaron los siguientes recursos tecnológicos:

- **Intel Power Gadget:** Herramienta empleada en sistemas físicos Windows para obtener métricas en tiempo real del consumo energético del procesador.
- **HWINFO:** Software de monitoreo avanzado que permitió registrar el uso energético del procesador desde el host durante la ejecución de máquinas virtuales.
- **psutil (Python):** Biblioteca utilizada para recolectar métricas dentro de las máquinas virtuales, incluyendo uso de CPU, frecuencia, tiempos de usuario y sistema, y uso de memoria.
- **Python 3.11:** Lenguaje base para el desarrollo de los scripts de carga y monitoreo.

- **Multiprocessing (Python):** Módulo de Python utilizado para generar carga paralela sobre múltiples núcleos del procesador.
- **Pandas y Matplotlib:** Librerías de análisis de datos y visualización gráfica utilizadas para el procesamiento de los datos recolectados y la elaboración de gráficas comparativas.
- **Visual Studio Code:** Entorno de desarrollo empleado para escribir, probar y ejecutar los scripts en Python.
- **Sistemas operativos Ubuntu 20.04 y Windows 10:** Plataformas sobre las cuales se configuraron tanto los entornos físicos como virtualizados para las pruebas.

2.4. Generación de carga computacional

2.4.1. Carga computacional paralela con Python y multiprocessing

En el Pseudocódigo 2 evalúa el impacto energético bajo condiciones de carga realistas, se desarrolló un script en Python utilizando el módulo multiprocessing con el objetivo de saturar los núcleos del procesador con tareas intensivas. Las tareas consistieron en la multiplicación continua de matrices aleatorias de gran tamaño, lo cual permite estresar el CPU y generar un comportamiento representativo del consumo en ambientes de alto rendimiento.

Tabla 2. Pseudocódigo para la carga computacional paralela con multiprocessing

Require: num_processes, matrix_size, duration_s

Ensure: Ejecución de procesos intensivos en CPU

- 1: Definir función `cpu_task()`:
- 2: **while** se esté ejecutando **do**
- 3: Crear matrices aleatorias A y B de dimensión `matrix_size`
- 4: Calcular $C = A \times B$
- 5: Verificar si se alcanzó el tiempo máximo; si es así, salir del ciclo
- 6: **end while**
- 7: Inicializar lista vacía `processes`
- 8: **for** $i = 1$ hasta `num_processes` **do**
- 9: Crear nuevo proceso que ejecute `cpu_task()`
- 10: Agregar proceso a la lista `processes`
- 11: Iniciar el proceso
- 12: **end for**
- 13: Esperar `duration_s` segundos
- 14: **for** cada proceso en `processes` **do**
- 15: Finalizar el proceso
- 16: **end for**

2.5. Recolección de métricas de rendimiento

Durante la ejecución de la carga, se utilizó un segundo script para recolectar datos de uso de CPU %, frecuencia del procesador en MHz, tiempos de usuario, sistema y uso

de memoria mediante la biblioteca psutil. Esta recolección se realizó desde el host de la máquina virtual con una frecuencia de 10 segundos, pseudocódigo 3.

Tabla 3. Pseudocódigo para monitoreo de métricas del sistema con psutil

Require: Intervalo de captura $t = 10$ segundos

Ensure: Registro de métricas en archivo CSV

- 1: Abrir archivo `metricas_virtualizacion.csv` en modo escritura
- 2: Escribir cabecera con columnas: `Timestamp`, `CPU_Usage_%`, `CPU_Freq_MHz`, `User_Time_s`, `System_Time_s`, `Memory_Usage_%`
- 3: **while** el programa esté en ejecución **do**
- 4: Capturar marca temporal actual
- 5: Obtener uso de CPU con `psutil.cpu_percent()`
- 6: Obtener frecuencia del procesador con `psutil.cpu_freq().current`
- 7: Obtener tiempos de CPU con `psutil.cpu_times()`
- 8: Obtener uso de memoria con `psutil.virtual_memory().percent`
- 9: Escribir fila con los valores recolectados en el archivo CSV
- 10: Esperar t segundos
- 11: **end while**

Este proceso permitió recolectar series temporales completas del comportamiento del procesador en cada entorno evaluado.

2.6. Recolección de datos y estimación del consumo

Se programó una recolección periódica cada 10 segundos, exportando los datos a archivos CSV. En el caso de entornos virtualizados, donde las herramientas no permiten medir directamente el consumo en Watts, se utilizó una fórmula estimativa basada en el uso de CPU y la frecuencia: Este valor fue comparado con los datos reales obtenidos desde HWINFO en el host.

$$\text{Consumo estimado (W)} = \left(\frac{\text{Uso CPU (\%)}}{100} \right) \times \text{Frecuencia (MHz)} \times 0.005 \quad (1)$$

La extracción del consumo energético se centró en el procesador, sin considerar otros componentes como GPU, el disco duro y la memoria RAM. Las pruebas se realizaron bajo condiciones controladas de laboratorio para garantizar la consistencia y validez de los resultados obtenidos.

2.7. Consumo Energético del Procesador en Entornos Virtualizados

El consumo energético de un procesador (CPU) está determinado por varios factores, entre los que destacan la carga de trabajo, la frecuencia de operación y las estrategias de gestión de energía del sistema.

En entornos virtualizados, el hipervisor desempeña un papel clave en la eficiencia energética. Su capacidad para distribuir tareas de forma balanceada, gestionar los estados de bajo consumo *C-states* y ajustar dinámicamente las frecuencias de operación *P-states* incide directamente en el consumo energético global del sistema [12].

Diversos estudios han analizado cómo los distintos tipos de hipervisores afectan este consumo, siendo los de tipo 1 generalmente más eficientes en entornos de carga intensiva, debido a su cercanía al hardware [12].

3. Resultados y Discusión

En la Tabla 4 se presentan los promedios de consumo energético del procesador en entornos físicos y virtualizados, evaluando cuatro hipervisores: *VirtualBox*, *VMWare*, *KVM* e *Hyper-V*, ejecutados sobre sistemas operativos *Ubuntu 20.4* y *Windows 10*.

Tabla 4. Promedio de consumo energético por hipervisor y sistemas operativos

Sistemas Operativos	Tipo	Promedio (W)
VirtualBox - Ubuntu 20.4	Local - Física	14.93
VirtualBox - Windows 10	Local - Física	14.94
VMWare - Ubuntu 20.4	Local - Física	14.40
VMWare - Windows 10	Local - Física	14.35
KVM - Ubuntu 20.4	Local - Física	13.93
KVM - Windows 10	Local - Física	13.65
Hyper-V - Ubuntu 20.4	Local - Física	14.09
Hyper-V - Windows 10	Local - Física	14.93
VirtualBox - Ubuntu 20.4	Virtualización	3.73
VirtualBox - Windows 10	Virtualización	3.63
VMWare - Ubuntu 20.4	Virtualización	2.50
VMWare - Windows 10	Virtualización	2.98
KVM - Ubuntu 20.4	Virtualización	3.15
KVM - Windows 10	Virtualización	3.65
Hyper-V - Ubuntu 20.4	Virtualización	3.77
Hyper-V - Windows 10	Virtualización	3.80

El gráfico de la Figura 2 presenta la comparativa del consumo energético promedio del procesador medido en Watts, diferenciando los valores obtenidos en *entornos físicos barras azules* y *virtualizados barras rojas*. Se incluyeron las plataformas *VirtualBox*, *VMWare*, *KVM* e *Hyper-V* ejecutadas sobre *Ubuntu 20.4* y *Windows 10*.

Los resultados muestran de forma clara que en todos los casos evaluados, el consumo energético del procesador es considerablemente menor en entornos virtualizados. El hipervisor *VMWare*, cuando trabaja en *Ubuntu*, destaca como el más eficiente energéticamente, con un consumo de solo 2.5W frente a los 14.4W del entorno físico, lo que representa una reducción cercana al 82.6%.

Por el contrario, *VirtualBox* mostró una menor diferencia entre los dos entornos, aunque también mantuvo una tendencia a favor del entorno virtualizado. *Hyper-V* presentó resultados consistentes en ambos sistemas operativos, superando a *KVM* y *VirtualBox* en eficiencia energética. En todos los casos, se evidencia una reducción de al menos un 70% al utilizar virtualización.

Este gráfico respalda cuantitativamente el impacto positivo de la virtualización en la gestión energética del procesador, constituyéndose en un criterio relevante al seleccionar tecnologías para entornos de centros de datos o despliegues en la nube.

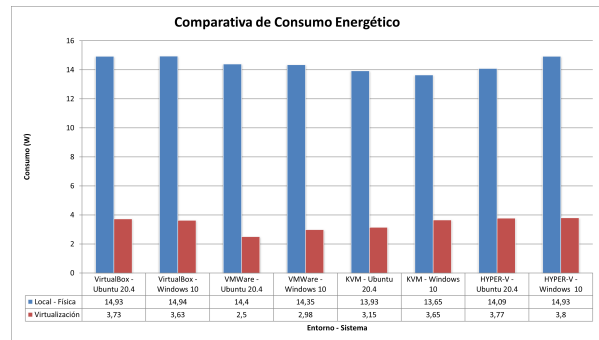


Figura 2. Comparativa de Consumo Energético en entornos Virtualizados y Físicos

3.1. Resultados con VirtualBox

VirtualBox es un hipervisor de tipo 2, lo que significa que requiere un sistema operativo anfitrión para funcionar. Se utiliza comúnmente en pruebas de desarrollo y entornos educativos debido a su interfaz amigable y compatibilidad multiplataforma.

Durante las pruebas, se ejecutó un script desarrollado en Python con multiprocessing, simulando una carga computacional intensiva durante un periodo prolongado. Esta carga aprovechó múltiples núcleos del procesador, generando una actividad constante.

El consumo energético promedio del procesador fue de 14.93W *Ubuntu* y 14.94W *Windows* en físico. En virtualización, los valores fueron 3.73W y 3.63W respectivamente, con un ahorro estimado de 75.1%.

Se observó que la frecuencia promedio del CPU en la VM era más alta que en otros hipervisores, lo que sugiere que *VirtualBox* aplica una gestión básica del procesador virtualizado, aunque no tan optimizada como otros hipervisores.

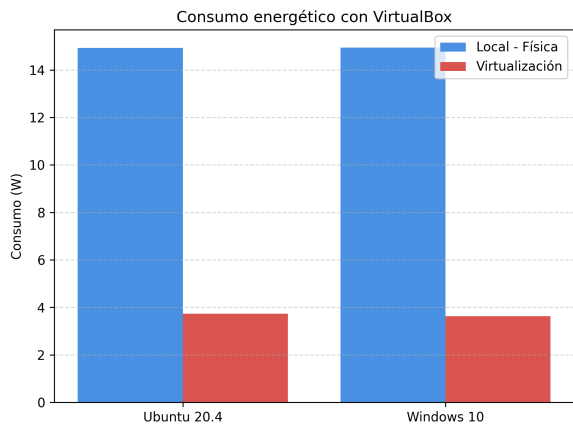


Figura 3. Comparación de consumo energético en físico vs virtualizado con VirtualBox

Este resultado se debe a que VirtualBox no implementa mecanismos avanzados de optimización energética como otros hipervisores. Su gestión del procesador es más básica, lo que hace que la frecuencia del CPU se mantenga alta durante toda la ejecución, incluso cuando no es necesario. Esto explica el valor promedio más elevado observado en las pruebas.

3.2. Resultados con VMWare

VMWare se posicionó como el hipervisor más eficiente en términos energéticos durante las pruebas. En Ubuntu 20.4, el consumo fue de 14.40W en físico y 2.50W en virtualizado. En Windows 10, pasó de 14.35W en físico a solo 2.98W en virtualizado. Estas cifras representan una reducción de hasta 82%, la más alta de todos los casos estudiados.

Este comportamiento podría explicarse por la avanzada integración que tiene VMWare con los recursos del sistema anfitrión, así como por su capacidad para optimizar el uso del procesador dentro de la máquina virtual. Esto se justifica que VMWare permite asignar recursos de manera más eficiente. Además, su motor de virtualización incorpora técnicas avanzadas de gestión dinámica del procesador, ajustando la frecuencia y el uso del CPU según la carga real de trabajo, lo que mejora tanto el rendimiento como el consumo energético.

Este hipervisor mantuvo una frecuencia del CPU más baja dentro de la VM, lo que indica un mejor manejo de los estados de reposo y escalado dinámico. VMWare fue el que logró el menor consumo en todos los escenarios. El escalado dinámico adapta la potencia del CPU según los procesos, mientras que el reposo minimiza el consumo cuando el sistema está inactivo.

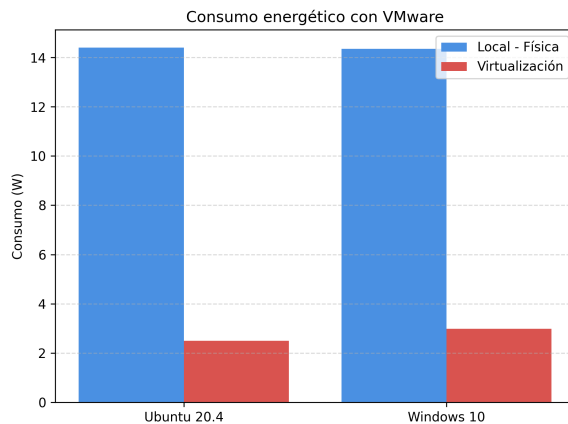


Figura 4. Comparación de consumo energético en físico vs virtualizado con VMWare

3.3. Resultados con KVM

KVM, un hipervisor de tipo 1 integrado en el kernel de Linux, mostró un comportamiento estable pero no tan eficiente como VMWare. En Ubuntu 20.4, se registró un consumo de 13.93W en físico y 3.15W en virtualizado. En Windows 10, los valores fueron 13.65W y 3.65W respectivamente.

Si bien se observa una mejora significativa, el ahorro del 74-76% es menor en comparación con otros hipervisores. Esto podría explicarse porque el rendimiento depende de las configuraciones avanzadas del kernel de Linux, las cuales son para optimizar el manejo dinámico de la frecuencia del procesador. Sin una correcta configuración, el control del consumo energético puede verse limitado, afectando el potencial de ahorro en entornos virtualizados.

KVM se comportó de forma estable, con un uso de CPU más lineal y consistente. Aunque no alcanzó los niveles de eficiencia de VMWare, sí superó a VirtualBox en todos los casos.

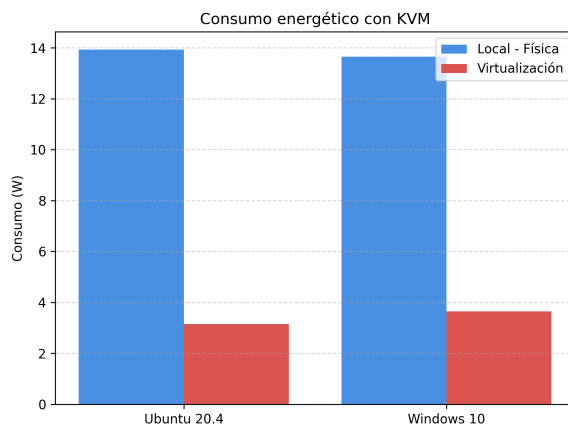


Figura 5. Comparación de consumo energético en físico vs virtualizado con KVM

3.4. Resultados con Hyper-V

Hyper-V es un hipervisor de tipo 1 desarrollado por Microsoft. Se obtuvieron valores consistentes en físico 14.09W en Ubuntu y 14.93W en Windows y en virtualización 3.77W y 3.80W respectivamente. El porcentaje de ahorro fue cercano al 74%, lo que lo posiciona como un hipervisor eficiente y equilibrado.

Hyper-V mantuvo una carga estable y su consumo de energía presentó mínimas variaciones. Aunque su eficiencia no superó a VMWare, fue superior a VirtualBox, especialmente en plataformas Windows.

Aunque no alcanzó los niveles de optimización de VMWare, Hyper-V mostró buen rendimiento, especialmente en entornos Windows, donde su integración es más profunda.

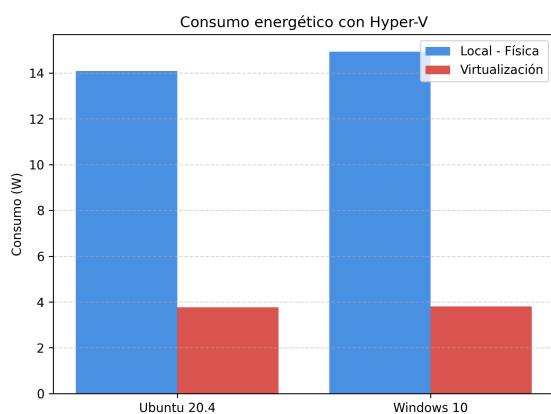


Figura 6. Comparación de consumo energético en físico vs virtualizado con Hyper-V

3.5. Resumen general comparativo

En todos los casos, el uso de entornos virtualizados redujo significativamente el consumo energético del procesador. La magnitud de esta reducción varió según el hipervisor y el sistema operativo. VMWare fue el más eficiente, seguido por Hyper-V, KVM y VirtualBox.

Tabla 5. Comparación de consumo y ahorro energético por hipervisor

Hipervisor	Consumo Local (W)	Consumo VM (W)	Ahorro (%)
VirtualBox (Win/Ub)	14.94 / 14.93	3.63 / 3.73	75.6 / 75.0
VMWare (Win/Ub)	14.35 / 14.40	2.98 / 2.50	79.2 / 82.6
KVM (Win/Ub)	13.65 / 13.93	3.65 / 3.15	73.2 / 74.9
Hyper-V (Win/Ub)	14.93 / 14.09	3.80 / 3.77	74.5 / 73.2

3.6. Análisis del Ahorro Energético

En la Tabla 5 se puede observar el porcentaje de ahorro energético obtenido al comparar los entornos físicos con sus respectivas versiones virtualizadas. Este valor se calculó aplicando la siguiente fórmula:

$$\text{Ahorro (\%)} = \left(\frac{\text{Consumo físico} - \text{Consumo virtual}}{\text{Consumo físico}} \right) \times 100 \quad (2)$$

Los resultados muestran una reducción significativa en el consumo energético al ejecutar las cargas computacionales en entornos virtualizados. El mayor ahorro se alcanzó utilizando VMWare sobre Ubuntu 20.4, con una reducción del 82.64%. También se obtuvieron ahorros superiores al 70% en las pruebas con VirtualBox, KVM e Hyper-V.

Este análisis evidencia que la virtualización no solo optimiza el uso de recursos físicos, sino que también representa una estrategia efectiva para mejorar la eficiencia energética, especialmente en contextos como centros de datos o entornos de alta demanda computacional.

4. Conclusiones

Este estudio permitió analizar de forma comparativa el comportamiento del consumo energético del procesador entre entornos físicos y virtualizados, utilizando distintos hipervisores sobre sistemas operativos Ubuntu 20.04 y Windows 10. A través de la ejecución de cargas computacionales intensivas mediante scripts de Python con procesamiento paralelo *multiprocessing* y herramientas de monitoreo como *Intel Power Gadget*, *HWINFO* y *psutil*, se obtuvieron métricas confiables para realizar el análisis.

Los resultados evidencian que la virtualización no solo mejora la utilización de recursos físicos, sino que también puede representar una ventaja desde el punto de vista energético. En todos los escenarios evaluados, el consumo energético fue significativamente menor en entornos virtualizados, con reducciones de hasta un 82% en algunos casos. VMWare fue el hipervisor que mostró la mayor eficiencia, seguido por Hyper-V, KVM y VirtualBox.

Además, se comprobó que es posible realizar una estimación razonable del consumo energético desde dentro de la máquina virtual mediante una fórmula basada en el uso de CPU y la frecuencia, lo que resulta útil cuando no se dispone de herramientas de monitoreo externas o hardware especializado.

Esta investigación demuestra que los hipervisores no se comportan de forma homogénea en términos de eficiencia energética, por lo que su elección debe considerarse cuidadosamente en entornos de centros de datos o computación distribuida donde la optimización energética es crítica.

Se recomienda que futuros trabajos incorporen el análisis de otros componentes del sistema como GPU, memoria RAM y almacenamiento, así como el uso de cargas reales de software empresarial para evaluar el impacto energético en condiciones de producción.

Referencias

- [1] A. López Medina, “Análisis de la virtualización de sistemas operativos,” https://www.maia.ub.es/~sergio/linked/an_lisis_de_la_virtualizaci_n_de_sistemas_operativos.pdf, Universitat de Barcelona, Departament de Matemàtica Aplicada i Anàlisi, Barcelona, Espanya, Treball final de carrera, Jul. 15 2010.
- [2] G. S. M. T. F. G. C. T. Pessolani, Pablo Andrés, “Sistema de virtualización con recursos distribuidos,” <https://sedici.unlp.edu.ar/handle/10915/18375>, 2012.
- [3] C. Guzmán and S. Nesmachnow, “Evaluación de herramientas para procesamiento distribuido de grandes volúmenes de datos,” *Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 17, no. 1, pp. 1–10, 2022. [Online]. Available: <https://sedici.unlp.edu.ar/handle/10915/156319>
- [4] H. I. Vélez Álava, “Análisis comparativo de vmware, hyper-v y virtualbox en la virtualización de servidores de las pymes en el sector de distribución alimentaria del ecuador,” <https://dspace.utb.edu.ec/handle/49000/17959>, Universidad Técnica de Babahoyo, Examen Complexivo - Sistemas de Información, 2025.
- [5] R. M. Espinosa Tigre, “Análisis comparativo de un hipervisor nativo propietario y libre como alternativa de solución para el proceso de almacenamiento de datos,” <https://dspace.esPOCH.edu.ec/items/672ecb8e-6506-4064-90b7-cd4975e48d8c>, Escuela Superior Politécnica de Chimborazo, Trabajo de grado / Informe técnico, 2021.
- [6] R. E. Caballero, “Estudio de técnicas de programación paralela en python,” Trabajo Fin de Máster, Universidad Politécnica de Madrid, 2016. [Online]. Available: <https://oa.upm.es/47422/>
- [7] R. B. V. P. A. P. S. P. Kuldeep Vayadande, Ankur Raut, “Efficient system for cpu metric visualization,” *Universia de la Rioja*, vol. 11, pp. 239–250, 2022. [Online]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=8788101>
- [8] G. Rodòla and BillieJoeX, “psutil: Python system and process utilities,” Library documentation, version 7.0.0, 2025. [Online]. Available: <https://psutil.readthedocs.io/en/latest/>
- [9] J. Palach, “Python parallel processing and multiprocessing,” <https://dlwqtxs1xzle7.cloudfront.net/70703701/618-libre.pdf>, 2018.
- [10] U. Israel, “Análisis de virtualización y consumo energético de cpu en entornos cloud,” 2021. [Online]. Available: <http://repositorio.uisrael.edu.ec/handle/47000/3612>
- [11] R. J. Fernández, “Comparación de performance de hipervisores,” 2015. [Online]. Available: <https://repo.iaa.edu.ar/handle/123456789/629>
- [12] M. L. Sagebiel and J. C. Samaras, “Power and performance measurements in virtual environments,” United States Department of Energy, Tech. Rep., 1999. [Online]. Available: <https://www.osti.gov/etdeweb/biblio/331645>