



JEEOS

JOURNAL OF ENERGY, ENGINEERING
OPTIMIZATION AND SUSTAINABILITY

2026

ENERGÍA Y SUSTENTABILIDAD



Vol. 10 Núm. 2 (2026)

Reserva de Derechos al Uso Exclusivo

No. 04-2026-050410175100-102. ISSN: 2448-8186



DIRECTORIO

L.D. GUILLERMO NARVÁEZ OSORIO

Rector

DR. WILFRIDO MIGUEL CONTRERAS SÁNCHEZ

Secretario de Investigación, Posgrado y Vinculación

DR. LUIS MANUEL HERNÁNDEZ GOVEA

Secretario de Servicios Académicos

LIC. ALEJANDRINO BASTAR CORDERO

Encargado del Despacho de la Secretaría de
Servicios Administrativos

Esta revista está citada en:

Sistema Regional de Información en Línea para Revistas Científicas de América Latina y el Caribe, España y Portugal LATINDEX. <http://www.1atindex.unam.mx/> JOURNAL OF ENERGY, ENGINEERING OPTIMIZATION AND SUSTAINABILITY, Vol 10, No. 2, Año 2026, es una publicación continúa editada por la Universidad Juárez Autónoma de Tabasco, Av. Universidad S/N, Zona de la Cultura, Col. Magisterial, C.P. 86040, Villahermosa, Tabasco, México. Tel. (+52) (933) 358 1500 Ext. 5040, <https://revistajeeos.ujat.mx/JEEOS/es/>, Email: jeeos@ujat.mx. Indizada en LATINDEX. Editora responsable: Laura Lorena Díaz Flores. Reserva de derechos al uso exclusivo No. 04-2026-050410175100-102, ISSN: 2448-8186, ambos otorgados por el Instituto Nacional del Derecho de Autor. Responsable de la última actualización de este número: Pauly González Mayo, Universidad Juárez Autónoma de Tabasco, Av. Universidad S/N, Zona de la Cultura, Col. Magisterial, Centro, Tabasco. C.P. 86040. Fecha de última actualización, 31 de agosto 2026.

This work is licensed under a Creative Commons Attribution- NonCommercialShareAlike 4.0 International License.

Dra. Laura Lorena Díaz Flores
EDITORA EN JEFE

M.A. Pauly González Mayo
GESTORÍA Y ASISTENCIA EDITORIAL

MIS Dalia Exaltación Medina Mandujano
MAEE. Mary Cruz Valenzuela Jiménez
ASISTENTE DE DISEÑO Y ESTILO

Dra. Sulma Guadalupe Gómez Jiménez
Dra. Margarita del Carmen Noguera Miceli
ASISTENCIA EN TRADUCCIÓN Y MAQUETADO

EDITORES ASOCIADOS INVITADOS

Dr. Darío Colorado Garrido. **Universidad Veracruzana**
Dra. Linda Viviana García Quiñonez. **Universidad Veracruzana**
Dr. Cristian Gómez. **Universidad Veracruzana**
Dra. Deysi Carmina Peña Ortiz. **Universidad Veracruzana**
Dr. Roberto Agustín Conde Gutiérrez. **Universidad Veracruzana**
Dr. Gerardo Alcalá Perea. **Universidad Veracruzana**
Dra. Zaira Ruth Zuviría López. **Universidad Autónoma de Guadalajara**
Dra. Karen Rodríguez Rosales. **Universidad Autónoma de Querétaro**
Dra. Laura López Díaz. **Universidad Juárez Autónoma de Tabasco**
Dr. Erik Ramírez Morales. **Universidad Juárez Autónoma de Tabasco**

COMITÉ EDITORIAL

Dr. Bassam Ali, **Facultad de Ingeniería, Universidad Autónoma de Yucatán**

Dr. Fabricio Nápoles Rivera, **Universidad Michoacana de San Nicolás de Hidalgo**

Dr. Francisco López Villareal, **Instituto Tecnológico de Villahermosa.**

Dr. Fernando Israel Gómez Castro, **Universidad de Guanajuato**

Dr. Juan Serrano Arellano, **Instituto Tecnológico de Pachuca**

Dra. Isabel María Valdivia Fernández, **Universidad de la Habana- Facultad de Geografía**

Dra. Ivett Zavala Guillén, **Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California**

Dr. Luis Alfonso García Cerda, **Centro de Investigación de Química Aplicada**

Dra. Nancy del Pilar Medina Herrera, **Universidad Autónoma de Nuevo León**

Dra. Ma. Guadalupe Garnica Romo, **Universidad Michoacana de San Nicolás de Hidalgo.**

Dra. María Guadalupe Alpuche Cruz, **Universidad de Sonora**

Dr. Pedro Cruz Alcantar, **Universidad Autónoma de San Luís Potosí**

Dr. Salvador Tututi Ávila, **Universidad Autónoma de Nuevo León**

Dra. Santa del Carmen Herrera Sánchez, **Universidad Autónoma del Carmen**

Índice Vol. 10 Núm. 2 (2026)

Carta editorial Zuviría-López Z.R.	i-ii
Integración de calor en la purificación de ciclohexano grado textil utilizando etilenglicol como solvente Cruz-Valdez J.A. ^{1*} , Álvarez-Alor B. ² , Montalvo-Salinas D. ³ , Garzón-Pérez A.S. ⁴ , Alvarado-Méndez P.E. ⁵	1-16
Fotosíntesis artificial efecto sinérgico de Ti^{3+}, Co^{2+}, Co^{3+} en los mecanismos de activación, acoplamiento c-c hacia la producción de isopropanol Pérez- de la Cruz A.Y. ¹ , Montalvo-Salinas D. ^{2*} , Cruz-Valdez J.A. ¹ , Pérez- Francisco J.M. ¹ , Jiménez- Castillo U. ¹ , Álvarez-Alor B. ³	17-30
Predicción de la potencia en una turbina vestas v52 mediante regresión polinomial y una red neuronal Olan-Barroso A.A. ^{1*} , Alaffita-Hernández F.A. ² , Escobedo-Trujillo B.A. ¹ , Colorado-Garrido D. ²	31-46
Estrategias de ingeniería de software verde para aplicaciones energéticamente eficientes Morales-Reyes E. ^{1*} , Pacheco-Reyes J.N. ¹ , Vázquez-Hernández R. ¹ , Gilbón-Aburto A. ¹ , Gómez-Manuel E. ¹	47-54
Estudio teórico y experimental de la eficiencia térmica de un colector cilindro parabólico en Coatzacoalcos, México Rodríguez-Flores E. ¹ , González-Martínez M.I. ² , Fuentes-Orozco S. ² , Garrido-Meléndez J. ³ , Márquez-Nolasco A. ⁴ , Conde-Gutiérrez R.A. ^{4*}	55-70
Determinación de la acumulación de metales en <i>eichhornia crassipes</i> para valorización de biomasa Benítez-Martínez X. S. ¹ , Domínguez-Villate W. A. ¹ , Núñez-Correa S. ¹ , Morales-Salas L. ^{2,3*}	71-80
Restricciones de sostenibilidad y riesgo de cola en portafolios accionarios de México Villada-Medina H.D. ^{1*} , Rendón-García J.F. ¹ , Alcalá-Perea G. ²	81-96
Análisis en lazo cerrado de un convertidor elevador en cascada con reducción del rizo de corriente de entrada para aplicaciones con celda de combustible de hidrógeno Hidalgo-Leal H.D. ¹ , Cruz-Gutiérrez F. ^{1*} , May-Burgos L.D. ¹ , Sarao-Cruz L.D.J. ¹ , Pérez-Aguilar A. ¹	97-114

Estrategias de ingeniería de software verde para aplicaciones energéticamente eficientes

Green software engineering strategies for energy-efficient applications

Morales-Reyes E.^{1*}, Pacheco-Reyes J.N.¹, Vázquez-Hernández R.¹, Gilbón-Aburto A.¹, Gómez-Manuel E.¹

¹Universidad Tecnológica del Sureste de Veracruz (UTSV), avenida Universidad Tecnológica Lote Grande No. 1,
CP 96360, Nanchital, Veracruz, México.

*eunice.morales@utsv.edu.mx

Artículo Científico

Publicado: 31 de agosto 2026

RESUMEN

El crecimiento sostenido de las tecnologías de la información ha incrementado el consumo energético asociado al software, generando interés en enfoques orientados a la sostenibilidad computacional. En este contexto, la ingeniería de software verde propone integrar criterios de eficiencia energética dentro del desarrollo de sistemas digitales. El presente trabajo tiene como objetivo analizar las implicaciones de sostenibilidad energética derivadas del comportamiento de distintos lenguajes de programación mediante un análisis secundario de resultados experimentales previamente publicados en la literatura científica. Para ello, se examinan métricas comparativas de consumo energético, tiempo de ejecución y uso de memoria reportadas en estudios empíricos basados en benchmarks computacionales equivalentes. A partir de la reinterpretación de dichos resultados desde una perspectiva de sostenibilidad del software, se identifican patrones consistentes entre el tipo de ejecución del lenguaje y su eficiencia energética relativa. El análisis evidencia que los lenguajes compilados presentan menores niveles de consumo energético en comparación con lenguajes interpretados o ejecutados sobre máquinas virtuales, aunque la eficiencia final depende también del diseño algorítmico y de las prácticas de desarrollo. Como principal aportación, el estudio propone integrar la eficiencia energética como criterio de decisión dentro de la ingeniería de software, contribuyendo a la construcción de sistemas más sostenibles sin necesidad de realizar nuevas mediciones experimentales. Estos elementos muestran que el análisis de evidencia empírica existente constituye un enfoque válido para orientar prácticas de desarrollo alineadas con la sostenibilidad tecnológica.

Palabras clave: eficiencia energética; sostenibilidad del software; lenguajes de programación; consumo energético; ingeniería de software verde.

ABSTRACT

The sustained growth of information technologies has increased the energy consumption associated with software, generating growing interest in approaches focused on computational sustainability. In this context, green software engineering proposes integrating energy efficiency criteria into the development of digital systems. The objective of this study is to analyze the sustainability implications related to the behavior of different programming languages through a secondary

analysis of experimental results previously reported in the scientific literature. To this end, comparative metrics of energy consumption, execution time, and memory usage reported in empirical studies based on equivalent computational benchmarks are examined. By reinterpreting these results from a software sustainability perspective, consistent patterns are identified between the execution model of programming languages and their relative energy efficiency. The analysis shows that compiled languages tend to exhibit lower levels of energy consumption compared to interpreted languages or those running on virtual machines, although overall efficiency also depends on algorithm design and development practices. As a main contribution, this study proposes the integration of energy efficiency as a decision-making criterion within software engineering, contributing to the development of more sustainable systems without requiring new experimental measurements. These findings suggest that analyzing existing empirical evidence is a valid approach for guiding development practices aligned with technological sustainability.

Keywords: energy efficiency; software sustainability; programming languages; energy consumption; green software engineering.

INTRODUCCIÓN

El crecimiento sostenido de las tecnologías de la información y comunicación ha transformado profundamente los procesos sociales, económicos e industriales; sin embargo, este avance tecnológico ha incrementado significativamente el consumo energético asociado a la infraestructura digital y al desarrollo de software. Los sistemas informáticos y servicios digitales representan una fracción creciente del consumo energético global, lo que ha generado preocupación respecto al impacto ambiental derivado de las operaciones computacionales y la emisión indirecta de gases de efecto invernadero [1]. En consecuencia, la sostenibilidad tecnológica ha emergido como un área prioritaria de investigación orientada a reducir la huella ambiental de los sistemas digitales sin comprometer su funcionalidad o rendimiento.

Dentro de este contexto surge la ingeniería de software verde, disciplina que propone integrar criterios de eficiencia energética a lo largo del ciclo de vida del software, desde el diseño hasta la operación y mantenimiento de los sistemas [2]. Diversos estudios han demostrado que decisiones técnicas como la selección del lenguaje de programación, el diseño algorítmico y el modelo de ejecución influyen directamente en el consumo energético de las aplicaciones [2,3,4].

Investigaciones empíricas han evidenciado diferencias significativas en eficiencia energética entre lenguajes de programación al ejecutar tareas computacionales equivalentes [5]. Estos resultados indican que el comportamiento energético del software se encuentra influenciado por características propias del lenguaje y su entorno de ejecución, así como por la forma en que se gestionan los recursos durante la ejecución.

A diferencia de estudios orientados a la medición experimental directa, el presente trabajo adopta un enfoque de análisis secundario basado en la reinterpretación de resultados empíricos previamente reportados en la literatura científica. Este enfoque permite analizar implicaciones

relacionadas con la sostenibilidad energética del software desde una perspectiva integradora, vinculando evidencia cuantitativa existente con principios de ingeniería de software sostenible.

En este contexto, el objetivo del presente estudio es analizar la relación entre los modelos de ejecución de los lenguajes de programación y la eficiencia energética del software mediante la reinterpretación de resultados experimentales reportados en estudios previos, con el fin de identificar implicaciones relevantes para el desarrollo de software sostenible.

METODOLOGÍA

Enfoque metodológico. El presente estudio se desarrolló mediante un enfoque de análisis secundario basado en la revisión e interpretación de resultados experimentales previamente publicados en la literatura científica sobre eficiencia energética de lenguajes de programación. No se realizaron mediciones experimentales propias; en su lugar, se analizaron datos derivados de estudios empíricos con el objetivo de identificar implicaciones relacionadas con la sostenibilidad del software.

El análisis se fundamentó principalmente en los resultados reportados por Pereira et al. [5], quienes evaluaron el comportamiento energético de múltiples lenguajes de programación mediante benchmarks computacionales equivalentes. De manera complementaria, se consideraron revisiones sistemáticas de literatura sobre métricas de software verde con el fin de contextualizar los resultados dentro del marco teórico de la sostenibilidad en ingeniería de software [3,4].

Variables de análisis. Las variables consideradas en el estudio incluyeron:

- Consumo de energía, expresado en joules (J)
- Tiempo de ejecución, medido en milisegundos (ms)
- Uso de memoria, expresado en megabytes (MB)

Estos indicadores permiten evaluar de manera integral la eficiencia energética del software, considerando tanto el desempeño temporal como la utilización de recursos computacionales.

Los valores analizados corresponden a métricas normalizadas reportadas en el estudio base, lo que permite realizar comparaciones relativas entre lenguajes bajo condiciones equivalentes de ejecución.

Fundamentación de medición energética. El cálculo del consumo energético en el estudio de referencia se basa en la relación física:

$$E = T \times P$$

donde E representa la energía consumida, T el tiempo de ejecución y P la potencia promedio utilizada durante la ejecución del programa [5].

Para la obtención de estas mediciones, el estudio original empleó la interfaz Running Average Power Limit (RAPL), tecnología integrada en procesadores Intel que permite estimar el consumo energético del CPU y subsistemas asociados mediante registros de hardware. Este enfoque es ampliamente utilizado en investigaciones sobre eficiencia energética debido a su capacidad para proporcionar mediciones reproducibles bajo condiciones controladas.

Criterios de análisis. El análisis se centró en identificar patrones de comportamiento energético asociados a los diferentes modelos de ejecución de los lenguajes de programación, clasificándolos en lenguajes compilados, interpretados y ejecutados sobre máquinas virtuales. Esta clasificación permitió relacionar las características técnicas de cada lenguaje con su desempeño energético relativo, proporcionando una base para la interpretación de los resultados desde la perspectiva de la ingeniería de software sostenible.

RESULTADOS

El análisis de los datos permite identificar patrones consistentes en la eficiencia energética de los lenguajes de programación en función de su modelo de ejecución. El comportamiento observado indica que la eficiencia energética del software se encuentra estrechamente relacionada con el nivel de abstracción y el tipo de ejecución del lenguaje, incluso cuando se ejecutan tareas computacionales equivalentes.

Los resultados comparativos normalizados se presentan en la **Tabla 1**. Los datos analizados corresponden a resultados experimentales previamente reportados por Pereira et al. [5], utilizados en este trabajo como base empírica para su reinterpretación desde la perspectiva de la sostenibilidad del software.

Los resultados muestran que los lenguajes compilados presentan menores niveles de consumo energético relativo y tiempos de ejecución más reducidos en comparación con los lenguajes interpretados o aquellos que se ejecutan sobre máquinas virtuales. En particular, C, Rust y C++ destacan como las alternativas con mejor desempeño energético, lo cual se relaciona con la compilación directa a código máquina y con un mayor control en la gestión de los recursos del sistema.

Por otra parte, los lenguajes basados en máquinas virtuales, como Java y C#, presentan un comportamiento intermedio, reflejado en un aumento moderado tanto en el consumo energético como en el uso de memoria. Este comportamiento puede vincularse con la sobrecarga adicional que implica la ejecución en entornos gestionados.

Tabla 1. Resultados globales normalizados de eficiencia energética entre lenguajes de programación (Pereira et al., 2017).

Lenguaje	Tipo de ejecución	Energía relativa	Tiempo relativo	Memoria relativa
----------	-------------------	------------------	-----------------	------------------

C	Compilado	1.00	1.00	1.17
Rust	Compilado	1.03	1.04	1.54
C++	Compilado	1.34	1.56	1.34
Ada	Compilado	1.70	1.85	1.47
Java	Máquina virtual (JVM)	1.98	1.89	6.01
Pascal	Compilado	2.14	3.02	1.00
Go	Compilado	3.23	2.83	1.05
C#	Máquina virtual (.NET)	3.14	3.14	2.85
JavaScript	Interpretado/JIT	4.45	6.52	4.59
TypeScript	Transpilado/JIT	21.50	46.20	4.69
PHP	Interpretado	29.30	27.64	2.57
Python	Interpretado	75.88	71.90	2.80
Ruby	Interpretado	59.34	69.91	3.97
Lua	Interpretado	45.98	82.91	6.72
Perl	Interpretado	65.79	79.58	6.62

Nota: Valores correspondientes a medias geométricas normalizadas reportadas por Pereira et al. (2017).

En cambio, los lenguajes interpretados registran los valores más altos de consumo energético relativo y de tiempo de ejecución. Este comportamiento se asocia con los procesos de interpretación dinámica y con mayores niveles de abstracción durante la ejecución de los programas.

De igual manera, se observa que no existe una relación directa entre el uso de memoria y el consumo energético total. Esto indica que la eficiencia energética depende en mayor medida de la forma en que se administran los recursos durante la ejecución, más que de la cantidad de memoria utilizada.

A partir de estos resultados, se puede identificar que la elección del lenguaje de programación influye de manera importante en el consumo energético del software, lo cual debe considerarse dentro del diseño de sistemas orientados a la sostenibilidad.

DISCUSIÓN: EL DILEMA ENTRE DESEMPEÑO Y EFICIENCIA ENERGÉTICA

Los resultados obtenidos permiten cuestionar la idea de que un programa más rápido es necesariamente más eficiente desde el punto de vista energético. Si bien existe una relación estrecha entre el tiempo de ejecución y el consumo energético, el análisis muestra que esta relación no es completamente proporcional. En algunos casos, lenguajes con menor tiempo de ejecución presentan un consumo energético relativamente mayor, lo que sugiere que la potencia utilizada durante la ejecución varía en función del modelo de ejecución y del uso de recursos del sistema [5].

Lo observado coincide con lo reportado en estudios recientes sobre eficiencia energética del software, donde se señala que el consumo energético no depende únicamente del tiempo de ejecución, sino también de factores como la gestión de memoria, la optimización del compilador y

la interacción con el hardware [6]. En este sentido, la eficiencia energética debe entenderse como una propiedad multifactorial que no puede reducirse a una sola métrica de desempeño.

En relación con el uso de memoria, los resultados analizados muestran que no existe una correlación directa entre el consumo máximo de memoria y el consumo energético total. Este hallazgo sugiere que la eficiencia energética depende en mayor medida de la forma en que la memoria es gestionada a lo largo del tiempo que del volumen máximo utilizado. Este comportamiento ha sido identificado en investigaciones previas, donde se destaca la importancia de analizar el comportamiento dinámico de los recursos computacionales en lugar de métricas estáticas aisladas [5].

Por otra parte, el análisis por tipo de ejecución permite observar diferencias claras entre lenguajes compilados, interpretados y ejecutados sobre máquinas virtuales. Los lenguajes compilados presentan ventajas en términos de eficiencia energética debido a su cercanía al hardware y a la ausencia de capas adicionales de abstracción. En contraste, los lenguajes interpretados y aquellos basados en máquinas virtuales introducen una sobrecarga computacional que impacta tanto en el tiempo de ejecución como en el consumo energético. Este fenómeno ha sido ampliamente discutido en la literatura, donde se reconoce que los niveles de abstracción influyen directamente en la eficiencia energética del software [6].

Desde una perspectiva más amplia, estos resultados pueden vincularse con enfoques como Green Business Process Management (Green BPM), el cual propone integrar criterios de sostenibilidad dentro de los procesos organizacionales y tecnológicos [7]. Bajo este enfoque, la eficiencia energética del software no se limita a una optimización técnica, sino que forma parte de una estrategia más amplia orientada a reducir el impacto ambiental de los sistemas de información.

De manera complementaria, el Manifiesto de Karlskrona establece que la sostenibilidad en el desarrollo de software debe abordarse considerando múltiples dimensiones interrelacionadas, incluyendo aspectos técnicos, ambientales, económicos y sociales [8]. En este sentido, la eficiencia energética del software no debe analizarse de forma aislada, sino como parte de un sistema complejo donde las decisiones tecnológicas tienen implicaciones a largo plazo.

Asimismo, investigaciones realizadas recientemente destacan la necesidad de incorporar métricas específicas que permitan evaluar la sostenibilidad del software, lo que refuerza que la eficiencia energética debe integrarse como un criterio dentro del proceso de desarrollo [9]. Estas propuestas coinciden en que la sostenibilidad debe integrarse desde la fase de análisis y diseño del sistema, donde se definen los requerimientos y la arquitectura, en lugar de tratarse como un aspecto posterior. Desde esta perspectiva, la eficiencia energética no depende únicamente del desempeño computacional, sino de una interacción entre el modelo de ejecución, la gestión de recursos y las decisiones de diseño, lo que implica que el desarrollo de software sostenible demanda un enfoque que articule criterios técnicos considerando el uso de materiales reciclables, así como la reducción de residuos electrónicos.

CONCLUSIONES

El análisis realizado muestra que los lenguajes compilados presentan, en general, mayores niveles de eficiencia energética en comparación con lenguajes interpretados o ejecutados sobre máquinas virtuales. No obstante, la eficiencia energética del software no depende exclusivamente del lenguaje utilizado, sino también de factores como el diseño algorítmico, la gestión de recursos y el modelo de ejecución.

También se observa que la relación entre desempeño y consumo energético no es completamente directa, lo que indica que optimizar el tiempo de ejecución no garantiza necesariamente una reducción proporcional en el consumo energético. Este hallazgo refuerza la necesidad de considerar la eficiencia energética como un criterio independiente dentro del proceso de desarrollo de software.

Asimismo, se identifica la necesidad de fortalecer la investigación en torno a la integración de prácticas de sostenibilidad dentro de la ingeniería de software, particularmente en lo relacionado con el uso de herramientas, métricas y marcos de referencia que permitan evaluar el impacto energético del software de manera sistemática.

Desde esta perspectiva, se plantea la importancia de incorporar criterios de sostenibilidad desde las etapas tempranas del desarrollo, de modo que las decisiones técnicas contribuyan a la construcción de sistemas más eficientes y con menor impacto ambiental. Este enfoque resulta especialmente relevante en el contexto actual, donde el crecimiento de los sistemas digitales implica desafíos significativos en términos de consumo energético y sostenibilidad.

REFERENCIAS

- [1] Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Penzenstadler, B., Seyff, N., & Venters, C. (2015). Sustainability design and software: The Karlskrona manifesto. In: Proceedings of the 37th International Conference on Software Engineering (ICSE), 467–476. <https://doi.org/10.1109/ICSE.2015.179>
- [2] Belkhir, A., & Elmeligi, A. (2018). Assessing ICT global emissions footprint: Trends to 2040 & recommendations. *Journal of Cleaner Production*, 177, 448–463. <https://doi.org/10.1016/j.jclepro.2017.12.239>
- [3] Calero, C., Moraga, M. Á., & Bertoa, M. F. (2013). Towards a software product sustainability model. In: Proceedings of the 2nd International Workshop on Green and Sustainable Software (GREENS), 30–36. <https://doi.org/10.1109/GREENS.2013.6606421>
- [4] Chan, H., & Johansson, M. (2014). Green IS: a systematic literature review. University of Borås, Sweden. Disponible en: <https://www.diva-portal.org/smash/get/diva2:1309721/FULLTEXT01.pdf>. Fecha de consulta: marzo de 2026.

- [5] Hilty, L., & Lohmann, W. (2011). The five most neglected issues in Green IT. *UPGRADE: The European Journal for the Informatics Professional*, 12(4), 12–15.
- [6] Lago, P., Gu, Q., & Bozzelli, P. (2014). A systematic literature review of green software metrics. VU University Amsterdam. Disponible en: <https://research.vu.nl/en/publications/a-systematic-literature-review-of-green-software-metrics/>. Fecha de consulta: marzo de 2026.
- [7] Opitz, N., Krup, H., & Kolbe, L. (2014). Green Business Process Management: A definition and research framework. In: *Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS)*, 3808–3817. <https://doi.org/10.1109/HICSS.2014.475>
- [8] Pereira, R., Couto, M., Ribeiro, F., Rua, R., Cunha, J., Fernandes, J., & Saraiva, J. (2017). Energy efficiency across programming languages. In: *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering (SLE)*, 256–267. <https://doi.org/10.1145/3136014.3136031>
- [9] Pinto, G., Castor, F., & Liu, Y. (2014). Understanding energy behaviors of thread management constructs. In: *Proceedings of the ACM International Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, 345–360. <https://doi.org/10.1145/2660193.2660201>