Towards Energy Sustainability: A Literature Review of Green Software Development

Angelo Rodrigo Taco Jimenez
angelotacoj@gmail.com
https://orcid.org/0000-0001-9806-5379
Universidad de Lima, Perú

Received: July 11, 2023 / Accepted: September 12, 2023
doi: https://doi.org/10.26439/ciis2023.7079

ABSTRACT. In the organizational and business environment, the increasing consumption of electrical energy by IT equipment poses a challenge in terms of cost as well as environmental impact. To address this problem, a literature review is proposed to collect and examine the most recent developments in the field of green software and their impact on energy efficiency. To carry out this literature review, the PICo search strategy was adapted and implemented and a total of 23 relevant articles were selected. In relation to the problem addressed, the concept of green software, which seeks to create efficient and sustainable programs that optimize energy consumption, has been developed. The tools and practices of sustainable software design, techniques for the development of energy-efficient systems and approaches on how to address the problems of energy consumption in data centers and cloud computing are explored.

KEYWORDS: literature review / green software / energy consumption / energy efficiency / sustainability

HACIA LA SOSTENIBILIDAD ENERGÉTICA: UNA REVISIÓN DE LITERATURA DEL DESARROLLO DE *SOFTWARE* VERDE

RESUMEN. En el ámbito organizacional y empresarial, el creciente consumo de energía eléctrica por parte de los equipos informáticos representa un desafío en términos de costo, así como también de impacto ambiental. Para abordar esta problemática se propone una revisión de literatura para recopilar y examinar los avances más recientes en el campo del *software* verde y su impacto en la eficiencia del consumo de energía. Para llevar a cabo esta revisión de literatura, se adaptó y aplicó la estrategia de búsqueda PICo y se seleccionaron un total de 23 artículos relevantes. En relación a la problemática abordada, se ha desarrollado el concepto de *green software*, el cual busca crear programas eficientes y sostenibles que optimicen el consumo de energía. Se exploran las herramientas y prácticas de diseño de *software* sostenible, técnicas para el desarrollo de sistemas eficientes en mención de consumo energético y enfoques en cómo abordar los problemas del consumo de energía en centros de datos y los servicios basados en la nube.

PALABRAS CLAVE: revisión de literatura, software verde, consumo de energía, eficiencia energética, sostenibilidad

1. INTRODUCTION

In the business environment, efficiency in the use of information technology (IT) equipment is of vital importance to carry out operations effectively. However, it has been observed that this equipment consumes a considerable amount of electrical energy, which results in a significant expense for companies and a considerable impact on the environment (Masanet et al., 2020). Energy consumption is mainly attributed to equipment hardware, which is primarily responsible for energy consumption.

In order to reduce the environmental impact of technology, in 1992, the United States Environmental Protection Agency introduced the Energy Star Program, which seeks to promote the production and use of energy-efficient electrical products (Boyd et al., 2008). Despite the fact that IT equipment is essential for managing large volumes of information in the era of digitization, the energy consumption of data servers worldwide accounted for approximately 1 % in 2020, although this figure is expected to increase significantly over the next 10 years (Masanet et al., 2020).

Among the most prominent companies estimated to have high annual energy consumption are Google and Facebook. According to López et al. (2019), these companies have implemented measures to counteract their impact on energy consumption. One example is Google's server energy consumption optimization policy, which has resulted in the establishment of the world's most sustainable data center in terms of environmental metrics. These centers consume up to five times less electricity than other data centers (López et al., 2019). For its part, Facebook has called on the services of Power Assure, a company specialized in optimizing electricity consumption in data centers. This company offers a software to monitor the consumption of servers, which turns them off when they are inactive or turns them on when necessary. Thanks to this solution, it is possible to achieve energy savings of up to 80 % (López et al., 2019).

The adoption of green software practices complements the aforementioned initiatives of large companies to optimize energy consumption (Abdullah et al., 2015). Green software focuses on the design and development of programs that generate a positive impact on economic, social and environmental aspects, thus avoiding any direct or indirect negative impact on these areas. By implementing solutions and policies to optimize electricity consumption, companies can achieve significant cost savings and contribute to a more sustainable future for all (Kern et al., 2015).

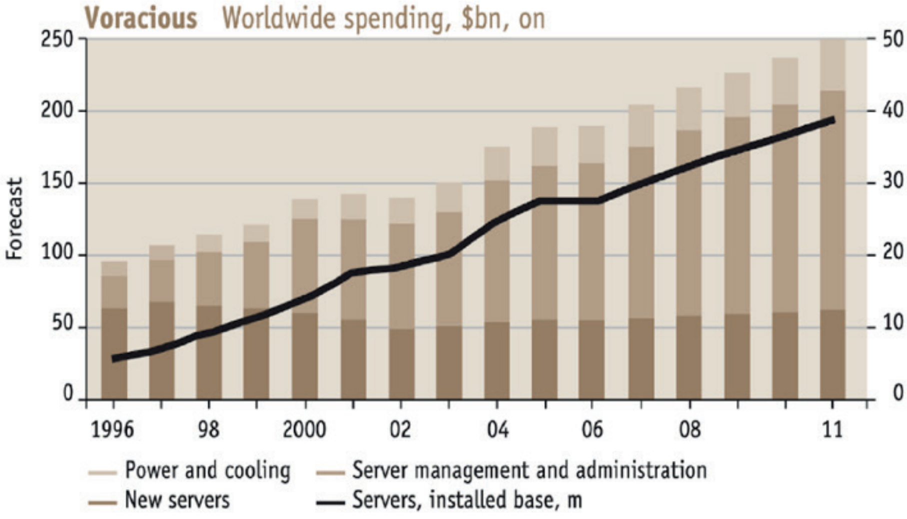
Recent research has shown that the use of optimized software can generate savings of more than 140 % compared to non-optimized software (Capra et al., 2012). This highlights the importance of considering energy consumption in sustainable software design and development. To achieve this, work is being done to implement the green software methodology, which involves the creation of software that follows specific energy efficiency metrics and promotes the proper use of computer hardware, thus avoiding overloads and reducing energy consumption during task execution (Beghoura et al., 2017).

According to Bustamante et al. (2014), in 2006, data centers in the United States used US$ 4,5 billion worth of electricity. Furthermore, it is estimated that most enterprise data centers will invest a similar amount of money in energy (power and cooling) and hardware infrastructure over the next five years, as mentioned by analyst Gartner (Kumar, 2007). This situation poses a significant challenge in terms of cost and sustainability.

To illustrate the importance of implementing sustainable measures, we will rely on Figure 1 (Where the cloud meets the ground, 2008), which shows an estimate of energy consumption.

Figure 1

Energy consumption from 1996 to 2011



Note. From *Where the cloud meets the ground* (2008).

It is interesting to note how the increase in the number of servers in data processing centers has led to an increase in the energy consumption required for cooling and maintenance. This trend poses significant challenges in terms of energy efficiency and sustainability at the enterprise level.

In summary, the adoption of green software practices emerges as a key solution to address energy consumption in the business environment (Munoz et al., 2017b). The implementation of green software methodology allows maximizing the energy efficiency of equipment through the development of optimized programs in terms of energy consumption. Therefore, it is crucial to conduct a thorough review of the current state of energy consumption and evaluate possible sustainable software alternatives.

2. BACKGROUND

2.1 Green information technology

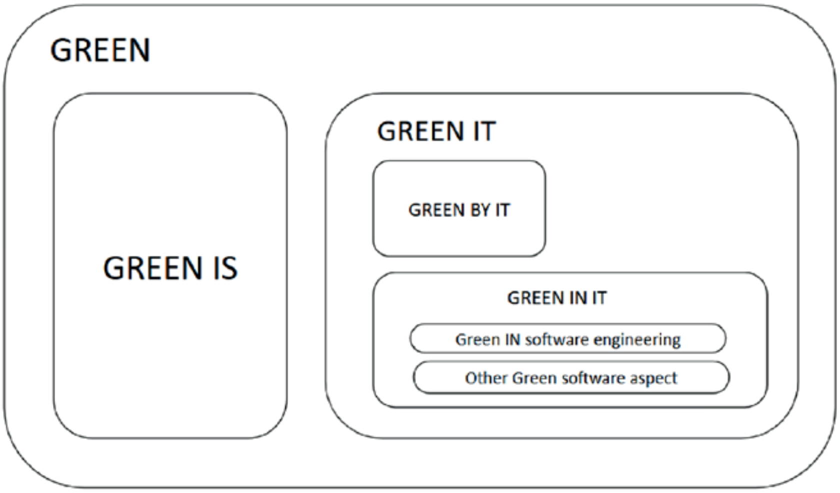
According to Hernández (2016), two different concepts that describe aspects related to information technology and its impact on sustainability are put forward: green information technologies (Green IT) and green information aystems (Green IS).

The term *green IT* is used to refer to any action or initiative that is carried out using technologies and processes that are environmentally friendly in all stages of the life of information and communication systems and technologies. On the other hand, the concept of *green IS* refers to the procedures adopted by an organization with the aim of improving its environmental performance. It is considered a way to address and reduce the negative effects that information technologies can have on the environment (Hedwig et al., 2009; Hernández, 2016).

Figure 2, presented by Hernández (2016), provides a visual support that illustrates the terminology related to green technologies.

Figure 2

Terms associated with green technology



Note. From ¿Por dónde empezar para aplicar prácticas verde/sostenibles en el proceso de desarrollo del software y obtener un producto verde/sostenible? [Where to begin to implement practice green/sustainable in the development process of software and get a green/sustainable product?], by A. Hernández, 2016, 18 *Convención Científica de Ingeniería y Arquitectura, IX Simposio Universitario Iberoamericano sobre Medioambiente* (SUIMA), p. 3

Hernández (2016) provides a detailed description of the subdivisions found within the term *green IT*. One of these subdivisions is *green IN IT*, which refers to information technologies, either software or hardware. In turn, two distinct approaches can be identified within *green in IT*. The first is *green in software engineering*, which focuses on the process and product of green software development. The second approach is *other green software aspect*, which addresses the business processes and governance related to green software implementation.

On the other hand, the term *green by IT* is used when information technologies are used as tools to achieve green objectives. This involves minimizing waste production and reducing energy consumption through the implementation of specific applications and methods. It is crucial to consider these subdivisions within the scope of *green IT* in order to develop appropriate and effective solutions to sustainability challenges in the field of technology (Hernández, 2016).

2.2 Sustainable software engineering

The importance of software maintainability as a fundamental quality requirement is increasingly evident today. As Venters et al. (2017) highlight, the ability to maintain and upgrade software is a key factor in ensuring the durability and effectiveness of a product. In other words, software sustainability plays a critical role in the long-term success of any software development project. However, sustainability is not limited to software durability alone, but also encompasses its impact on the environment.

Although software itself does not consume energy, it has a significant impact on the energy consumption of the hardware on which it runs. Data centers, mobile devices and smart devices are just a few examples of devices that are affected by software energy consumption.

It is in this context where green technology becomes relevant, as it seeks to use hardware resources in a sustainable way in order to reduce environmental impact. To achieve this, it is necessary to review software development methodologies and promote the creation of more sustainable software. Software engineers play a key role in this process, as they must consider the impact on energy consumption of each design and implementation decision in the final system. Microsoft Learn (2023), through a course, presents two key philosophies for sustainable software engineering:

- **Everyone must be involved in solving the climate problem:**

  Sustainable software engineering promotes the involvement of everyone in solving the climate problem, and small changes can have a big impact. Environmentalists work across all engineering disciplines, and standardizing the discussion of sustainability in technical meetings can lead to major changes in any organization.

- **Sustainability alone is sufficient to justify our work:**

  As sustainable software engineers, we know that creating sustainable applications is beneficial because of their lower cost, higher performance and resilience. But the main reason is sustainability.

The development of software that is easy to maintain is a goal highlighted by González (2018), who states that this allows necessary changes to be made efficiently and cost-effectively to adapt to changes in the business, rather than replacing the software entirely. In this way, it is possible to extend the useful life of the software and make it more sustainable. In addition, Akinli (2013) mentions that various green metrics can be used to assess the environmental impact of software, such as the percentage of CPU usage, storage medium, work throughput and system energy consumption in kilowatt hours (kWh). The application of these metrics is essential to monitor the influence of information technologies and systems on the environment, as well as to evaluate the performance of companies that choose to implement sustainable approaches to software development.

On the other hand, Padilla et al. (2023) point out that the software programming process involves electrical energy consumption, and as the amount of code increases, so does the energy expenditure, which has an impact on the carbon footprint and the environment. Danny van Kooten, a programmer, has proposed the reduction of lines of code as a way to contribute to the fight against climate change within the technology industry (Padilla et al., 2023). According to calculations made by Ramos, this modification could have a significant impact on the reduction of CO_2 emissions, equivalent to stopping driving a car for 421 000 kilometers or reducing CO_2 emissions by 59 tons per month. It is important to note that reducing lines of code not only has the potential to reduce energy consumption and CO_2 emissions but also can simplify and improve software efficiency, reduce maintenance costs and improve software durability (Padilla et al., 2023).

3. METHODOLOGY

The main purpose of the study was to analyze the most recent advances in energy saving through green software. To conduct this research, a review process was chosen based on the methodology proposed by Nina et al. (2021) but adapting it to the objective of this research, instead of performing a systematic mapping as the aforementioned authors did; i.e., a focused and specific review of the relevant literature was carried out.

First, planning was carried out to identify the interest, population and current context of the review. Subsequently, questions applicable to the review were formulated and a search for articles was conducted using keywords and constructing search strings. Finally, articles related to the research were filtered and selected. These steps were carried out in detail to ensure a thorough and rigorous review of the existing literature on the topic.

3.1 Study planning

The first step began by conducting a study planning using the PICo technique:

- Population: Articles selected to address the questions posed.
- Interest: Identification of progress in the field of study.
- Context: Application of green software for the reduction of energy consumption.
- The following four questions were identified:
  - What are the current tools and frameworks that promote energy sustainable software design?
  - What techniques and principles are recommended to optimize energy efficiency in software development?
  - What strategies are employed in resource planning to manage energy consumption in data centers and cloud services?
  - How can virtual machine allocation algorithms be optimized to reduce energy consumption in data centers and cloud computing platforms?

3.2 Article search

As a second step, a search for articles was carried out in renowned databases such as Scopus, IEEE and ACM, which are the most frequently used for studies related to computer systems, information technologies, among other specialties. These databases were chosen not only because of their renown but also because of the precision they offer when performing detailed searches using keywords, which speeds up and improves the selection of relevant publications. The search for articles was focused on the topic of software development aimed at reducing energy consumption. As a result, 92 articles were obtained. Each of these were reviewed and evaluated based on their abstract and where they were published, either in a journal or a conference.

This paper mainly focuses on the analysis of trends in the field of software engineering, green computing, cloud services, green software, energy saving, data centers, intelligent software, software development. An analysis of articles published between 2016 and 2022 is presented. The filtering process included reviewing titles and abstracts to select those related to the study.

To be included in the research, the articles had to be from databases or journals ranked Q1 or Q2, be directly related to the main topic and be available in English or Spanish In addition, the articles presented at recognized international symposia in the field of computer engineering and informatics were included.

On the other hand, the inclusion of articles from journals ranked Q3 or lower was rejected, as well as those whose focus was far from the main topic. Likewise, articles derived from international symposia that were not within the specific field of computer engineering and informatics were discarded.

Finally, after filtering the articles, a total of 23 papers were selected, including journal articles and papers presented at conferences, which met the acceptance criteria established above. These papers were subject to a detailed analysis to identify trends and approaches related to software development aimed at reducing energy consumption, with the purpose of answering the questions posed in section 3.1. This selection and analysis process ensured the inclusion of only the most important and highest quality papers in the research.

4. RESULTS

4.1 What are the current tools and frameworks that promote energy sustainable software design?

According to the research conducted by Georgiou et al. (2020), two main categories of literature that focus on promoting sustainable and energy-efficient software development can be identified. On the one hand, there are those centered on the generation and application of tools and frameworks that facilitate the task and monitoring of sustainable software development. On the other hand, there are papers that explore and evaluate various design lines and development techniques that can be applied in a wide range of projects.

Munoz et al. (2017a) took a significant step along the way by developing a plugin that monitors the performance and energy efficiency of security interfaces in mobile operating systems. This line of research and development is complemented by other efforts, such as that of Dorn et al. (2019), who introduced the Producing Green Applications Using Genetic Exploration algorithm. This tool has the ability to identify variants of a program based on specific code transformation criteria, enabling optimal energy consumption without compromising software functionality. These advances are testimony to the growth and diversity in energy sustainability-oriented strategies in software development.

Yeganeh et al. (2019) presented the development of a dynamically priced capacity planning tool, which was designed specifically for green data centers in mobile networks seeking to efficiently balance energy consumption and operational costs. The main focus was on determining the optimal number of servers and balancing the cost of operation with service waiting times. This tool provides a significant advance in the search for solutions for a more sustainable software design.

Finally, Cruz and Abreu (2017) developed a tool focused on improving energy consumption in mobile devices. In a similar but more technical approach, Mancebo et al. (2018) proposed a framework that uses physical and digital modules, such as Energy Efficiency Tester and Software Energy Assessment, to accurately measure the energy consumption of a program during its execution.

4.2 What techniques and principles are recommended to optimize energy efficiency in software development?

In energy efficiency-oriented software design, there are principles and techniques that must be considered to ensure an effective implementation. In line with this, it is imperative to mention the energy focused virtual machine selection policy (Mandal et al., 2020). This technique focuses on the dynamic consolidation of virtual machines (VMs). By considering crucial aspects such as host server overload conditions and the current state of the VMs, it seeks to achieve two main goals: a significant reduction of energy consumption and minimization of service level agreement (SLA) violations.

Pereira et al. (2017) introduced the SPectrum-based energy leak localization. This technique is based on statistical analysis and execution tracing to identify critical points in the source code where less efficient energy consumption occurs. The proposal by Conoci et al. (2018) show us a strategy that seeks to efficiently leverage multiple processing threads on modern processors by setting precise energy limits.

Finally, in the context of the development of the Internet of Things (IoT) and fog computing, Lenka et al. (2019) stress the importance of optimizing energy efficiency through specialized routing protocols for IoT sensing infrastructure. They propose an approach that involves the creation of a rendezvous region in the center of the network area and adopts clustering and multipathing techniques. These strategies have been shown to effectively reduce power consumption and extend the lifetime of IoT network infrastructure.

It is clear that to effectively address the challenges of optimization and energy efficiency in software development, it is critical to take an interdisciplinary approach. Collaboration among diverse specialists offers a more holistic and effective approach to sustainability in this domain.

4.3 What strategies are employed in resource planning to manage energy Consumption in data centers and cloud services?

Currently, a number of approaches focused on resource planning have been recognized with the purpose of addressing the challenges of energy consumption in data centers and cloud services. Within this line, several studies have emphasized the implementation of advanced algorithms that directly correlate with energy optimization. Garg et al. (2019) not only analyzed this relationship but also highlighted the importance of reliability and stability of running applications.

On the one hand, Haddad et al. (2021) proposed a biphasic methodology focused on efficiency and sustainability. This methodology emphasizes, in its first phase, the optimal allocation of computing resources through a binary search algorithm, ensuring adequate attention to the demands of the IT area. And in its second phase, priority is given to the integration of sustainable electrical resources, such as photovoltaic panels and wind turbines, complemented with energy storage systems to ensure continuity.

In addition, the use of machine learning techniques has been explored to find efficient solutions to this problem. Sahoo et al. (2018) developed a framework that uses machine learning to optimize task scheduling. In doing so, they managed to obtain promising results in terms of efficiency and resource optimization.

Hu et al. (2021) provide another perspective by introducing a fine-grained heterogeneous power distribution model for geographically dispersed data centers. This model, together with a two-phase online algorithm, seeks to minimize energy costs and the derived distribution gap. Simulated results confirm its effectiveness.

4.4 How can virtual machine allocation algorithms be optimized to reduce energy consumption in data centers and cloud computing platforms?

Several techniques have been proposed in the literature with the objective of improving the allocation of virtual machines and reducing their energy consumption (Gupta et al., 2018). One such technique is virtual machine migration, which uses specific algorithms to optimize available resources (Hu et al., 2021).

Another approach is workload balancing, which seeks to improve the efficiency of existing algorithms (Liu et al., 2016) or predict the load that will be presented on different VM resources to perform efficient migrations (Wu et al., 2017; Li et al., 2018).

In addition, it is taken into consideration that the growth in demand for cloud-based computing services has led to the need to increase the energy efficiency of data centers. To address this challenge, an energy-efficient hybrid framework has been proposed by Alarifi et al. (2020). This framework combines request scheduling and server consolidation, allowing virtual machines to be allocated more efficiently. Client requests are ranked according to their energy and time requirements, and specialized algorithms are used to manage resource migration and consolidation. Results indicate that an energy-efficient hybrid is superior to other approaches, offering improved energy efficiency, performance and cost savings.

In addition, approaches that combine container and virtual machine migration have been proposed with the goal of reducing energy consumption (Gholipour et al., 2020). Algorithms that take into account the target energy load and available resources have also been developed to determine the optimal allocation of virtual machines and thus reduce the final energy consumption (Mohammadhosseini et al., 2019).

Finally, idle virtual machines, consuming up to 70 % of the total server energy, present a significant challenge. An innovative algorithm called Active & Idle Virtual Machine Migration (Hossain et al., 2020), inspired by Ant Colony Optimization, has emerged to address this issue. This tool relocates idle virtual machines, promoting significant energy savings and optimizing efficiency in data centers. These advances in green software point to future innovations that further refine the management of virtual machines in ecological terms.

5. CONCLUSIONS

The technology sector is undergoing a transition to more sustainable and energy-efficient practices, especially in software development and data center management. This research focuses on understanding the current state of these initiatives and identifying areas of opportunity.

Despite the existence of multiple tools and techniques aimed at fostering sustainability in programming, a significant gap is evident: the lack of integrated and structured methodologies for projects specifically dedicated to green software. While existing tools offer solutions to improve energy efficiency at the code level, the absence of a coherent methodological approach could slow down the actual implementation of these practices in technology projects.

In the realm of data centers and cloud-based services, there is a tendency to rely on simulations rather than testing in real scenarios. While simulations are valuable for conceptualizing and evaluating initial proposals, their limitation lies in not being able to capture all the uncertainties and variabilities that arise in real-world situations. Strategies that prove successful in a controlled environment may not be as successful in a practical context.

It is imperative, therefore, to conduct further research in real-world scenarios in order to validate the proposed solutions for energy management in data centers. Furthermore, it would be beneficial for the technology industry to advocate for the development and adoption of comprehensive methodologies, specifically targeted at sustainable software, thus facilitating its implementation throughout the development lifecycle.

REFERENCES

Abdullah, R., Abdullah, S., Din, J., & Tee, M. (2015). A systematic literature review of green software development in collaborative knowledge management environment. *International Journal of Advanced Computer Technology (IJACT)*, *4*(1), 63-80. https://www.ijact.org/ijactold/volume4issue1/IJ0410057.pdf

Akinli, S. (2013). Green software development and design for environmental sustainability. *11th International Doctoral Symposium on Empirical Software Engineering (IDoESE 2013), ID183.* https://www2.umbc.edu/eseiw2013/idoese/pdf/eseiw2013_IDoESE_183.pdf

Alarifi, A., Dubey, K., Amoon, M., Altameem, T., El-Samie, F. E. A., Altameem, A., Sharma, S. C., & Nasr, A. A. (2020). Energy-efficient hybrid framework for green cloud computing. *IEEE Access: practical innovations, open solutions*, *8*, 115356–115369.

Beghoura, M. A., Boubetra, A., & Boukerram, A. (2017). Green software requirements and measurement: random decision forests-based software energy consumption profiling. *Requirements Engineering*, *22*(1), 27-40. https://doi.org/10.1007/s00766-015-0234-2

Boyd, G., Dutrow, E., & Tunnessen, W. (2008). The evolution of the ENERGY STAR® energy performance indicator for benchmarking industrial plant manufacturing energy use. *Journal of Cleaner Production*, *16*(6), 709-715. https://doi.org/10.1016/j.jclepro.2007.02.024

Bustamante, F. P., Guzman, C. A. P., & Vargas, J. D. L. (2014). Análisis de la aplicación del Green IT en las organizaciones. *Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology.* https://www.laccei.org/LACCEI2014-Guayaquil/RefereedPapers/RP235.pdf

Capra, E., Francalanci, C., & Slaughter, S. A. (2012). Is software "green"? Application development environments and energy efficiency in open source applications. *Information and Software Technology*, *54*(1), 60–71.

Conoci, S., Di Sanzo, P., Ciciani, B., & Quaglia, F. (2018). Adaptive performance optimization under power constraint in multi-thread applications with diverse scalability. *Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering (ICPE '18),* 16-27. https://doi.org/10.1145/3184407.3184419

Cruz, L., & Abreu, R. (2017). Performance-based guidelines for energy efficient mobile applications. *2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft),* 46-57. https://doi.org/10.1109/MOBILESoft.2017.19

Dorn, J., Lacomis, J., Weimer, W., & Forrest, S. (2019). Automatically exploring tradeoffs between software output fidelity and energy costs. *IEEE Transactions on Software Engineering*, *45*(3), 219-236. https://doi.org/10.1109/tse.2017.2775634

Garg, R., Mittal, M., & Son, L. H. (2019). Reliability and energy efficient workflow scheduling in cloud environment. *Cluster Computing*, *22*(4), 1283-1297. https://doi.org/10.1007/s10586-019-02911-7

Georgiou, S., Rizou, S., & Spinellis, D. (2020). Software development lifecycle for energy efficiency: Techniques and tools. *ACM Computing Surveys*, *52*(4), 1-33. https://doi.org/10.1145/3337773

Gholipour, N., Arianyan, E., & Buyya, R. (2020). A novel energy-aware resource management technique using joint VM and container consolidation approach for green computing in cloud data centers. *Simulation Modelling Practice and Theory*, *104*. https://doi.org/10.1016/j.simpat.2020.102127

González, A. H. (2018). La sostenibilidad y el software. *Dilemas contemporáneos: Educación, Política y Valores, 5*(2), artículo 44.

Gupta, G., Singh Khichar, G., Rathi, R., & Singh, R. (2018). Energy-aware and cost effective VM allocation strategy based on Best Fit MultiValued Binpaking algorithm. *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence),* 647–652.

Haddad, M., Da Costa, G., Nicod, J.-M., Péra, M.-C., Pierson, J.-M., Rehn-Sonigo, V., Stolf, P., & Varnier, C. (2021). Combined IT and power supply infrastructure sizing for standalone green data centers. *Sustainable Computing: Informatics and Systems*, *30*, 100505. https://doi.org/10.1016/j.suscom.2020.100505

Hedwig, M., Malkowski, S., & Neumann, D. (2009). Taming energy costs of large enterprise systems through adaptive provisioning. *ICIS 2009 Proceedings*, *140*. http://aisel.aisnet.org/icis2009/140

Hernández, A. (2016). ¿Por dónde empezar para aplicar prácticas verdes/sostenibles en el proceso de desarrollo del software y obtener un producto verde/sostenible? [Where to begin to implement practice green/sustainable in the development process of software and get a green/sustainable product?] *18 Convención Científica de Ingeniería y Arquitectura, IX Simposio Universitario Iberoamericano sobre Medioambiente (SUIMA)*. https://www.researchgate.net/publication/312538106

Hossain, M. K., Rahman, M., Hossain, A., Rahman, S. Y., & Islam, M. M. (2020). Active & idle virtual machine migration algorithm- a new ant colony optimization approach to consolidate virtual machines and ensure green cloud computing. *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE),* 1-6.

Hu, X., Li, P., & Sun, Y. (2021). Minimizing energy cost for green data center by exploring heterogeneous energy resource. *Journal of Modern Power Systems and Clean Energy*, *9*(1), 148–159. https://doi.org/10.35833/MPCE.2019.000052

Kern, E., Dick, M., Naumann, S., & Hiller, T. (2015). Impacts of software and its engineering on the carbon footprint of ICT. *Environmental empact essessment Review*, *52*, 53-61. https://doi.org/10.1016/j.eiar.2014.07.003

Kumar, R. (2007). *Eight critical forces shape enterprise data center strategies*. *Gartner.*

Lenka, R. K., Rath, A. K., & Sharma, S. (2019). Building reliable routing infrastructure for green IoT network. *IEEE Access: practical innovations, open solutions*, *7*, 129892–129909.

Li, Z., Yan, C., Yu, L., & Yu, X. (2018). Energy-aware and multi-resource overload probability constraint-based virtual machine dynamic consolidation method. *Future Generation Computer Systems*, *80*, 139-156. https://doi.org/10.1016/j.future.2017.09.075

Liu, Y., Shu, W., & Zhang, C. (2016). A parallel task scheduling optimization algorithm based on clonal operator in green cloud computing. *Journal of Communications, 11*(2), 85–191.

López, M., Huedo, E., & Garbajosa, J. (2019). *Green IT: tecnologías para la eficiencia energética en los sistemas TI* [Green IT: energy efficient technologies for systems]. Fundación Madrid+d para el Conocimiento.

Mancebo, J., Arriaga, H. O., García, F., Moraga, M. Á., de Guzmán, I. G.-R., & Calero, C. (2018). EET: a device to support the measurement of software consumption. *GREENS '18: Proceedings of the 6th International Workshop on Green and Sustainable Software,* 16-22.

Mandal, R., Mondal, M. K., Banerjee, S., & Biswas, U. (2020). An approach toward design and development of an energy-aware VM selection policy with improved SLA violation in the domain of green cloud computing. *The Journal of Supercomputing*, *76*(9), 7374-7393.

Masanet, E., Shehabi, A., Lei, N., Smith, S., & Koomey, J. (2020). Recalibrating global data center energy-use estimates. *Science*, *367*(6481), 984-986. https://doi.org/10.1126/science.aba3758

Microsoft Learn. (2023). *Introducción a la ingeniería de software sostenible* [An introduction to sustainable software engineering] [Video]. Microsoft. https://learn.microsoft.com/es-es/training/modules/sustainable-software-engineering-overview/2-overview

Mohammadhosseini, M., Toroghi Haghighat, A., & Mahdipour, E. (2019). An efficient energy-aware method for virtual machine placement in cloud data centers using the cultural algorithm. *The Journal of Supercomputing*, *75*, 6904-6933. https://doi.org/10.1007/s11227-019-02909-3

Munoz, D.-J., Montenegro, J. A., Pinto, M., & Fuentes, L. (2017a). Green security plugin for pervasive computing using the HADAS toolkit. *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech),* 796-803. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.136

Munoz, D.-J., Pinto, M., & Fuentes, L. (2017b). Green software development and research with the HADAS toolkit. *ECSA '17: Proceedings of the 11th European Conference on Software Architecture: Companion Proceedings,* 205-211. https://doi.org/10.1145/3129790.3129818

Nina, H., Pow-Sang, J. A., & Villavicencio, M. (2021). Systematic mapping of the literature on secure software development. *IEEE Access: Practical Innovations, Open Solutions*, *9*, 36852-36867. https://doi.org/10.1109/access.2021.3062388

Padilla, A. F. A., Ávila, E. S. B., & Solórzano, R. A. T. (2023). Programación verde en el software [Green software programming]. *Revista UNESUM–Ciencias*, *7*(1), 187-196. https://revistas.unesum.edu.ec/index.php/unesumciencias/article/view/419

Pereira, R., Carção, T., Couto, M., Cunha, J., Fernandes, J. P., & Saraiva, J. (2017). Helping programmers improve the energy efficiency of source code. *ICSE-C '17: 2017 Proceedings of the 39th International Conference on Software Engineering Companion,* 238-240. https://doi.org/10.1109/ICSE-C.2017.80

Sahoo, S., Sahoo, B., & Turuk, A. K. (2018). An energy-efficient scheduling framework for cloud using learning automata. *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT),* 1-5. https://doi.org/10.1109/ICCCNT.2018.8493692

Venters, C. C., Seyff, N., Becker, C., Betz, S., Chitchyan, R., Duboc, L., Mcintyre, D., & Penzenstadler, B. (2017). Characterising sustainability requirements: A new species, red herring, or just an odd fish? *ICSE '17: 39th International Conference on Software Engineering: Software Engineering in Society Track,* 3-12. https://hdl.handle.net/1807/75783

Where the cloud meets the ground. (2008, October 25). *The Economist*. https://www.economist.com/special-report/2008/10/25/where-the-cloud-meets-the-ground

Wu, Y., Tornatore, M., Ferdousi, S., & Mukherjee, B. (2017). Green data center placement in optical cloud networks. *IEEE Transactions on Green Communications and Networking*, *1*(3), 347-357. https://doi.org/10.1109/tgcn.2017.2709327

Yeganeh, H., Salahi, A., & Pourmina, M. A. (2019). A novel cost optimization method for mobile cloud computing by capacity planning of green data center with dynamic pricing. *Canadian Journal of Electrical and Computer Engineering*, *42*(1), 41-51.