

Caracterización de prácticas sostenibles en el desarrollo de software y la sostenibilidad humana usando *Essence*

Characterization of Sustainable Practices in Software Development and Human Sustainability with Essence

Paola Noreña
Universidad EAFIT
Medellín, Colombia
panoreaac@eafit.edu.co

Elizabeth Suescún
Universidad EAFIT
Medellín, Colombia
esuescul1@eafit.edu.co

Liliana González
Universidad EAFIT
Medellín, Colombia
lgonzalez@eafit.edu.co

Resumen —El software desempeña un papel importante en la generación de eficiencia energética y sostenibilidad. La sostenibilidad del software se fundamenta en prácticas sostenibles que abarcan tres dimensiones: el software sostenible, la sostenibilidad humana y la sostenibilidad económica. Aunque la mayoría de los estudios se centran en la dimensión del software sostenible, a menudo se descuida su conexión con las dimensiones humana y económica. En este artículo se propone el uso de *Essence* para la caracterización de las prácticas sostenibles en el desarrollo de software, involucrando la sostenibilidad humana. Dicha caracterización busca dar a conocer y fomentar la aplicación de prácticas sostenibles del software y del factor humano.

Palabras Clave —Ciclo de vida del software; *Essence*; Prácticas sostenibles; Sostenibilidad del software; Sostenibilidad humana.

Abstract —Software plays an important role in generating energy efficiency and sustainability. Software sustainability is based on sustainable practices that encompass three dimensions: sustainable software, human sustainability, and economic sustainability. Most studies focus on the sustainable software dimension. However, its connection to the human and economic dimensions is often neglected. This paper proposes use to *Essence* in the characterization of sustainable practices in software development involving the human sustainability. Such characterization seeks for publicizing and promoting the application of sustainable practices and the human factor.

Keywords – *Essence*; Software life cycle; Sustainable practices; Software Sustainability; Human sustainability.

I. INTRODUCCIÓN

La sostenibilidad y el enfoque humano son dos temas importantes para el desarrollo de software en la industria 5.0. La sostenibilidad del software se basa en prácticas que permiten incorporar la sostenibilidad desde tres dimensiones [1]: software sostenible o *green software*—productos de

software que involucran procesos sostenibles en el ciclo de vida del software o en software que apoya la sostenibilidad ambiental en otras áreas [2]. En ese contexto, la sostenibilidad humana se refiere a los—aspectos sociológicos y psicológicos de la comunidad del software, es decir, usuarios, equipos y sociedad y sostenibilidad económica—aspectos financieros del proyecto.

El software es esencial para la implementación de soluciones, como un todo, debe contribuir a fomentar la eficiencia energética y la sostenibilidad en una variedad de sectores, por ejemplo, para el control y monitoreo de productos, monitorización ambiental, transporte sostenible, edificios inteligentes, entre otros, apoyando así a la mitigación de los impactos ambientales y al uso responsable de los recursos. El software puede influir significativamente en las características de ahorro de energía integradas a las plataformas y tecnologías que se usan en la infraestructura del desarrollo de software. Actualmente, el desarrollo de software es altamente influenciado por el uso de prácticas ágiles, la gestión de recursos y la minimización de desperdicios durante el desarrollo y su ciclo de vida. Enfoques como DevOps permiten la aplicación de un conjunto de prácticas en el desarrollo de software sostenible, que buscan mejorar la colaboración y la comunicación entre los equipos de desarrollo de software y operaciones (IT) [3], con el objetivo de acelerar el ciclo de vida de desarrollo, entrega y despliegue de los productos de software de forma temprana. El enfoque de DevOps busca superar las barreras tradicionales entre equipos de desarrollo y operaciones, fomentando una cultura de colaboración, automatización, mejora continua y generación de valor [4] [5].

La mayoría de los trabajos [1] [2] [4] [6] [7] [8] [9] que se centran en el software sostenible desde el ciclo de vida y su infraestructura, la sostenibilidad y la eficiencia energética sigue siendo un desafío para abordarse desde etapas tempranas del desarrollo [1]. Sumado a esto, faltan esfuerzos desde la sostenibilidad humana para impactar comunidades (equipos de

desarrollo y usuarios) que se involucren en el software sostenible [10] [11] [12] [13] [14] [15] y contribuyan en la sostenibilidad económica del software [16] [17].

En este artículo se propone una caracterización de prácticas sostenibles en el desarrollo de software y prácticas que involucren la sostenibilidad humana. Para hacerlo se emplea *Essence*¹, un marco de trabajo estándar para la creación, uso y mejoramiento de prácticas en ingeniería de software. Con el uso de este estándar, las prácticas podrán ser adoptadas en las diferentes etapas del desarrollo de software en ambientes productivos del software. Por lo que, se espera que las prácticas sostenibles permitan generar eficiencia energética en procesos de desarrollo de software sostenible, promuevan el acceso a infraestructura en la nube, equidad social y laboral al incentivar habilidades de desarrollo de software en comunidades vulnerables y el uso responsable de productos de software sostenible. Es importante resaltar que se ha tomado como referente el mapeo de literatura de Guerrero et al. [3] en el cual se describen las principales prácticas de ingeniería de software que se utilizan en aquellos entornos de trabajo de DevOps, esto no solo nos permitió tener una visión unificada, también entender, analizar y contextualizar las prácticas de listadas en la sesión III, adicional a eso, algunos de los trabajos allí evidenciados fueron incluidos en el presente trabajo a través de la estrategia de *snowbaling* (se utilizan para encontrar más referencias relevantes a partir de las fuentes identificadas) descritas en [5].

Este artículo se estructura de la siguiente manera: En la Sección 2 se relacionan trabajos previos, en la Sección 3 se expone las prácticas sostenibles encontradas y en la Sección 4 se incluyen los resultados y discusión. Finalmente, se presentan las conclusiones y trabajos futuros en la Sección 5.

II. TRABAJOS PREVIOS

Según [1] para las tres dimensiones de sostenibilidad del software, la mayoría de los trabajos desde la literatura han sido en software sostenible. La distribución de literatura se puede observar en la Figura 1.

Algunos trabajos en software sostenible son: [6] proponen un ciclo de vida para el desarrollo de sistemas ciber-físicos, adoptando buenas prácticas y las dimensiones relacionadas a sostenibilidad. En [4] se propone un framework para el diseño arquitectural de sistemas ciber-físicos que contempla de manera transversal la sostenibilidad, en ese mismo sentido, Londoño [7] presenta un lenguaje de modelado de arquitectura para sistemas ciber-físicos adaptativos y [8] a *framework design for sustainable cyber-physical architectures*. [1] junto con otros autores del libro, presentan procesos de medida para el software sostenible, un *framework* para pruebas de eficiencia energética del software, impacto de la sostenibilidad del software en la organización, entre otros. Finalmente, en [9] presentan prácticas para el desarrollo sostenible desde un enfoque técnico basado en el patrón de arquitectura, modelo vista controlador (MVC) y computación en la nube, prácticas esenciales para garantizar la viabilidad a largo plazo de los sistemas de software.

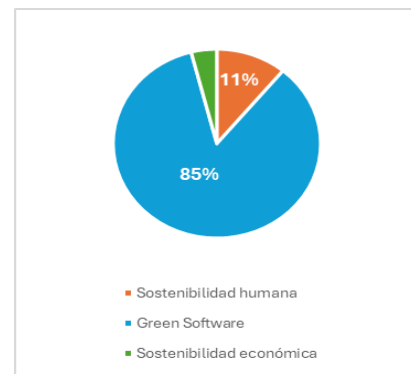


Figura 1. Distribución de literatura en trabajos de sostenibilidad del software [1]

También otros autores han trabajado en los siguientes enfoques desde sostenibilidad humana: [10] y [11] presentaron competencias sostenibles para ingeniería de software. [12] desarrollaron un laboratorio ágil y de co-creación entre los ingenieros de software y usuarios con extracción de aspectos humanos desde los requisitos. [13] y [14] definieron desafíos en esta dimensión: (i) capacitación de habilidades de TI, especialmente en desarrollo ágil, (ii) comercio transfronterizo en temas de empleabilidad y comunicación y (iii) barreras culturales. [15] realizaron un caso de estudio aplicado a Kenya en competencias de desarrollo permitiendo activar su economía. Finalmente, estos son enfoques desde la sostenibilidad económica: [16] proponen un *framework* de prácticas para el desarrollo sostenible en la dimensión económica y [17] presentaron un enfoque basado en esta dimensión para medir requisitos sostenibles.

Aunque, los trabajos propuestos presentan avances en las tres dimensiones, falta trabajar enfoques que permitan integrar el aspecto humano desde las diferentes prácticas sostenibles del proceso de desarrollo de software.

III. PRÁCTICAS SOSTENIBLES EN EL DESARROLLO DE SOFTWARE Y LA SOSTENIBILIDAD HUMANA USANDO *ESSENCE*

Para llevar a cabo la contribución de este trabajo, se inicia con la identificación de características sostenibles del software a través de un mapeo de artículos relacionados con el desarrollo de software sostenible, los cuales se encuentran referenciados en [3] y [18]. Estas características proporcionan perspectivas valiosas para la aplicación y búsqueda de prácticas sostenibles en el desarrollo de software. Por lo que posteriormente, se utilizan como palabras clave para realizar la búsqueda y caracterización de prácticas en software sostenible y sostenibilidad humana. Con base en los hallazgos obtenidos, se procede a especificar un compendio de estas prácticas utilizando el estándar de *Essence*, con el propósito de fomentar un enfoque más respetuoso con el medio ambiente y socialmente responsable desde el desarrollo de software. La caracterización y especificación de estas prácticas se lleva a cabo en colaboración con expertos en sostenibilidad y desarrollo de software, asegurando un enfoque integral desde ambas perspectivas.

¹<https://www.omg.org/hot-topics/essence.htm>

A. Identificación de características sostenibles del software

De las diferentes características sostenibles, se hace relevante evaluar y tratar el consumo energético en el proceso de desarrollo de software, el producto y el hardware. Las características sostenibles relacionadas directamente con la calidad del software son: modificabilidad, modularidad, usabilidad, accesibilidad, soportabilidad, eficiencia, desempeño, portabilidad, adaptabilidad, escalabilidad, flexibilidad, mantenibilidad, seguridad, eficiencia energética, consumo energético, efecto e impacto ambiental, vida útil del hardware- residuos y obsolescencia (recursos computacionales y de infraestructura), capacidad sin conexión, factibilidad, impacto y responsabilidad social, legalidad y educación [1] [9] [19] [20] [21], el detalle de estas características se encuentra en el siguiente [link](#)². Estas características buscan ser utilizadas y medibles en prácticas sostenibles en el proceso de desarrollo de software.

B. Caracterización y especificación de prácticas sostenibles en el desarrollo de software usando Essence

Se utiliza *Essence* (Esencia de SEMAT, un marco de trabajo estándar para la creación, uso y mejoramiento de prácticas en ingeniería de software) para la representación y especificación de prácticas sostenibles en software. *Essence* permite describir elementos fundamentales de métodos y prácticas existentes en ingeniería de software. De acuerdo con su notación, las prácticas se representan mediante un rombo del color de las áreas de interés: solución (producción del software, amarillo), esfuerzo (involucra al equipo y la forma en como el equipo hace su trabajo, azul) y cliente (usuarios e interesados, verde), las cuales se pueden presentar en su composición como un panel de abejas y mediante enlaces de composición ($\leftarrow\triangleright$) para detallar una jerarquía [22]. Se toma como referencia las dimensiones prácticas definidas por [1] para la sostenibilidad del software. Por lo que se representa en la Figura 2, la dimensión práctica general “sostenibilidad del software” y sus dimensiones: (i) “*green software* o software sostenible” en color amarillo, ya que se utiliza en el ciclo de vida del software o en software que apoya la sostenibilidad ambiental en otras áreas; (ii) “sostenibilidad humana” y “sostenibilidad económica” en color azul, ya que involucran el esfuerzo, forma de trabajo, el equipo y comunidad del software y costos del proyecto). Este trabajo se concentra en especificar las dimensiones prácticas “*green software* o software sostenible” y “sostenibilidad humana.”

Correspondientemente, a partir de la práctica de “*green software* o software sostenible” se especifican dos subprácticas “sostenibilidad en el ciclo de vida del software” y “sostenibilidad técnica” las cuales se representan en la Figura 3. La especificación de estas prácticas surge a partir de los hallazgos [1] [2] [3] [9] [18] [20] [22] [23] [24] y se describen en el siguiente [link](#)³.

Se puede notar que la “sostenibilidad en el ciclo de vida del software” se compone de prácticas en “gestión de

requisitos”, “entrega continua” y “control del impacto ambiental” mientras que la “sostenibilidad técnica” se compone de prácticas en “gestión técnica o de arquitectura” y “control del impacto ambiental”, algunas de ellas se definen desde prácticas ágiles DevOps por su impacto en la evolución continua del software sostenible al reducir el desperdicio de recursos, mejorar la eficiencia operativa y fomentar una cultura de mejora continua [20].

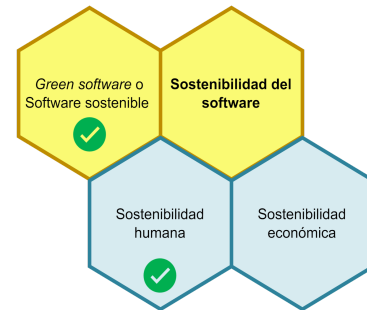


Figura 2. Dimensiones prácticas de la sostenibilidad del software. Los autores basados en [1]

C. Caracterización y especificación de prácticas sostenibles en la sostenibilidad humana del software usando Essence

De las prácticas predominantes en la sostenibilidad humana se pueden apreciar la “responsabilidad social”, la “gestión Lean” y la “gestión cultural” como se observan en la Figura 4 y se describen en la Tabla I a partir de los hallazgos en los trabajos [1] [3] [10] [11] [13] [14] [19] [22] [23] [24] [25] [26]. La “gestión Lean” y la “gestión cultural” también hacen parte de las prácticas DevOps y se han constituido como pilares fundamentales en la cultura y ambiente laboral de las empresas para generar bienestar en los empleados [3] [22]. Se incluye la “responsabilidad social” como una práctica requerida para los procesos de software, en relación con sus equipos de desarrollo y su entorno, donde se busca promover “entornos inclusivos” en los miembros del equipo en cualquier lugar del mundo y equidad con relación a la “accesibilidad a los recursos y las herramientas” para un mejor desempeño de su labor, incentivando la “generación de empleo” para comunidades vulnerables. También las prácticas de “integración de ética social”, “gestión del talento” y “educación en prácticas sostenibles” se perciben como prácticas que promueven la sostenibilidad humana en el desarrollo de software [1][12][13][17] (véase la Figura 4).

TABLA I. PRÁCTICAS EN LA SOSTENIBILIDAD HUMANA DEL SOFTWARE

Prácticas sostenibles	Descripción	Referencias
Responsabilidad social	Contribuye al beneficio del equipo, los usuarios y la sociedad en general alrededor del software.	[19]
Integración de ética social	Aspectos de legalidad, transparencia en la recopilación y uso de datos, derechos humanos y derechos de los usuarios.	[19]
Entornos inclusivos	Asegurar la representación equitativa de diversidad y cultura en los equipos y el involucramiento de la comunidad.	[13] [14]
Accesibilidad a los recursos y herramientas	Recursos de conectividad y energía y herramientas de software que sean accesibles a los desarrolladores.	[13] [14]

²<https://github.com/panorenac/PracticasSostenibles/blob/fa0d320861328596a4659d7f6e868a56235fa40e/Caracter%C3%ADsticas%20en%20sostenibilidad%20del%20Software.pdf>

³<https://github.com/panorenac/PracticasSostenibles/blob/fa0d320861328596a4659d7f6e868a56235fa40e/Pr%C3%A1cticas%20sostenibles%20del%20software.pdf>

Prácticas sostenibles	Descripción	Referencias
Gestión del talento	Implementar políticas y procesos que reconozcan y recompensen el rendimiento excepcional, así como también el bienestar y la salud mental de los empleados.	[13] [14]
Generación de empleo	Fomento en el desarrollo profesional y económico de los miembros del equipo.	[1] [13] [14]
Experiencia usuario	cómo se siente una persona al interactuar con un sistema,. Implica aspectos como la usabilidad, la satisfacción y la impresión general que tiene el usuario durante su interacción.	[25][26]
Necesidades en contexto	Diseño de soluciones basadas en las necesidades actuales de la comunidad.	[26]
Educación en prácticas sostenibles	Capacitaciones al equipo sobre prácticas sostenibles fomentando la conciencia sobre el impacto ambiental de las decisiones de desarrollo de software.	[1] [10] [11]
Gestión Lean	Crear valor para los clientes, el aseguramiento de la calidad, la creación del flujo, el liderazgo con humildad y respeto por cada individuo.	[22][23] [24]
Aprobación de cambios	Garantizar que los cambios propuestos en el software sean evaluados y aprobados antes de su implementación, lo que requiere comunicación efectiva por parte del equipo.	[22][23] [24]
Notificación proactiva de fallas	Mantener de forma permanente la socialización y/o documentación de aprendizajes del equipo de trabajo.	[22][23] [24]
Visualización del trabajo	Mantener de forma permanente la socialización y/o documentación de aprendizajes del equipo de trabajo.	[22][23] [24]
Límite de trabajo en proceso	Asegurar que los equipos no sean sobrecargados y expone los obstáculos del flujo de trabajo.	[22][23] [24]
Monitoreo y observabilidad continua	Observar y comprender el estado de sus sistemas mediante métricas o logs y depurar activamente su sistema explorando propiedades que se requieran.	[3][22][23] [24]

Prácticas sostenibles	Descripción	Referencias
Retroalimentación y mejora continua	Mantener de forma permanente la socialización y/o documentación de aprendizajes del equipo de trabajo.	[22][23] [24][22]
Trabajo por lotes pequeños	Permite probar rápidamente hipótesis sobre una mejora en particular para que tenga el efecto que desea.	[22][23] [24]
Experimentación en equipo	Los equipos experimentan creando prototipos y probando ideas rápidamente a la medida.	[22][23] [24]
Gestión cultural	Cooperación entre los miembros de una organización, con normas para el trabajo, comunicación, dirección de problemas y calidad de soluciones.	[22][23] [24]
Cultura organizativa generativa	Fomenta la creatividad, la colaboración y la capacidad de adaptación del equipo experimentando con nuevas ideas y enfoques.	[22][23] [24]
Aprendizaje continuo	Es la capacitación constante que tienen los equipos para actualizar sus conocimientos en habilidades técnicas y blandas.	[22][23] [24]
Colaboración entre equipos	Forma de trabajo en la que los equipos del proyecto trabajan en conjunto.	[22][23] [24]
Medición del rendimiento	Se busca medir el rendimiento del proceso, el software (todas las características de la tabla I) y el equipo (la productividad y bienestar)	[22][23] [24]
Satisfacción laboral	Generar bienestar en los empleados en el desarrollo profesional, en el ambiente laboral y en su salud física, emocional y mental.	[3][22][23] [24]
Liderazgo transformacional	Inspirar y motivar al equipo para alcanzar niveles más altos de desempeño y lograr cambios significativos en la organización.	[22][23][24]

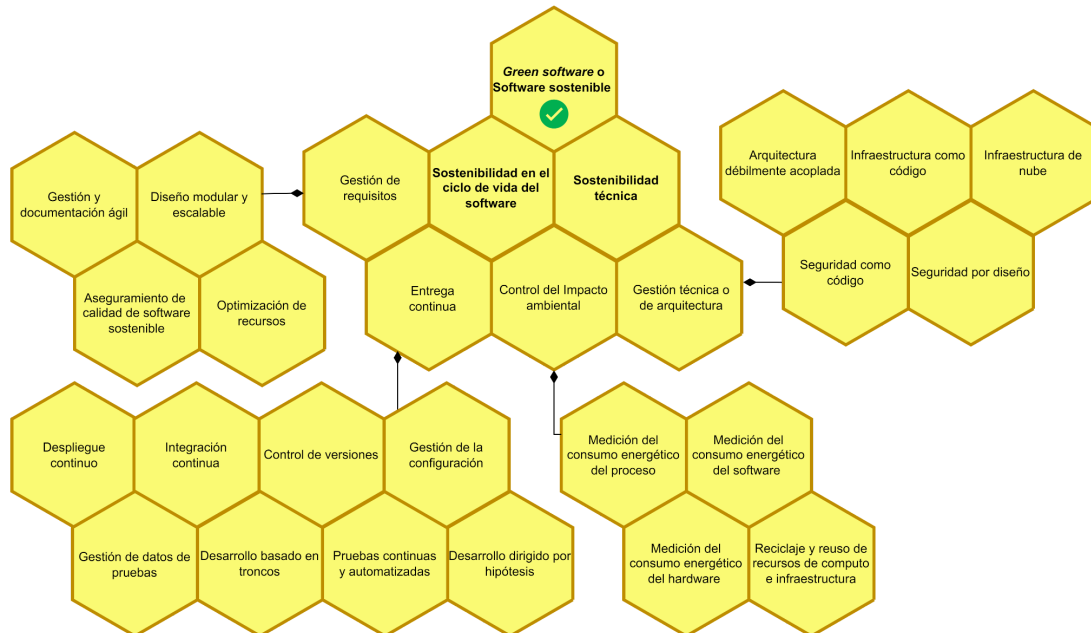


Figura 3. Prácticas sostenibles en el desarrollo de software. Los autores

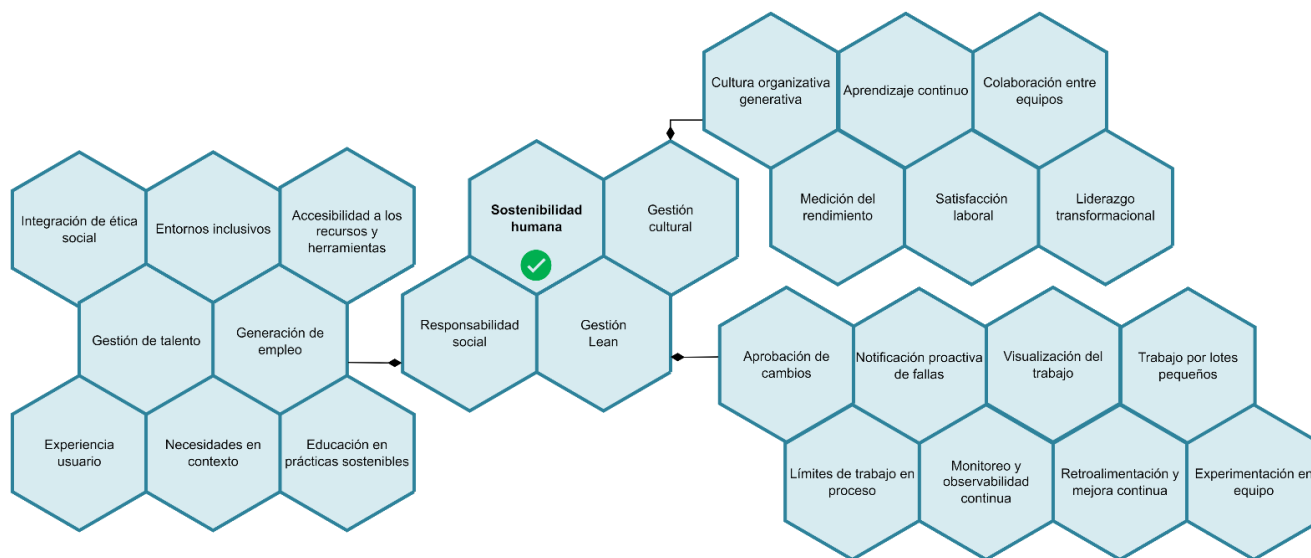


Figura 4. Prácticas sostenibles en la sostenibilidad humana del software. Los autores

IV. RESULTADOS Y DISCUSIÓN

Abordar la sostenibilidad en el desarrollo de software desde la perspectiva del software sostenible y la sostenibilidad humana implica adoptar prácticas que no solo sean amigables con el medio ambiente, sino también socialmente responsables. Estas prácticas se deben considerar durante el ciclo de vida completo en el desarrollo de software. Esto implica no solo la eficiencia durante la ejecución, sino también la minimización de residuos y el uso sostenible de recursos en todas las etapas.

El acercamiento a los objetivos de desarrollo sostenible (ODS) desde ambas perspectivas el software sostenible y la sostenibilidad humana se enfocan principalmente en los siguientes: 7. energía asequible y no contaminante, 8. Trabajo decente y crecimiento económico, 9. Industria, innovación y crecimiento, 10. Reducción de las desigualdades, 12. Producción y consumo responsables, 13. Acción por el clima y 17. Alianza para lograr los objetivos⁴. Desde la sostenibilidad humana específicamente los ODS 8, 10 y 12 y desde el software sostenible los ODS 7, 9, 12, 13 y 17 [1].

Algunos de los aspectos más relevantes para ser tratados en el desarrollo de software son: la priorización, el diseño y la implementación de software eficiente en términos energéticos, minimizando el consumo de recursos computacionales. La optimización de código y la eficiencia en el uso de hardware contribuyen a la sostenibilidad ambiental. Además, se debe fomentar la colaboración y la comunicación efectiva dentro de los equipos de desarrollo. Por lo que, la sostenibilidad humana se promueve al crear entornos laborales inclusivos, donde la diversidad es valorada y se fomenta la participación de todos los miembros del equipo en cualquier lugar del mundo, con accesibilidad a los recursos y herramientas para un mejor desempeño de su labor. Conforme a lo anterior, se debe garantizar que el software sea accesible para todas las personas, independientemente de sus capacidades o

limitaciones. Esto no solo promueve la equidad, sino que también amplía el alcance del producto, beneficiando a una audiencia más amplia.

Por otro lado, la importancia del equilibrio entre la vida laboral y personal es un punto crucial para promover la salud mental y física de los miembros del equipo con horarios razonables, flexibilidad laboral y apoyo emocional. En este punto, la inversión en el desarrollo profesional y personal de los miembros del equipo mejora la eficiencia y la calidad del trabajo y contribuye a la satisfacción, retención del personal y la integración de prácticas éticas en el desarrollo de software, como la transparencia en la recopilación y uso de datos, para asegurar que el software respete la privacidad y los derechos de los usuarios. También, se deben implementar métricas para evaluar el impacto ambiental y social del software desarrollado. Estas métricas pueden guiar la mejora continua y permitir a los equipos ajustar sus prácticas para maximizar la sostenibilidad del software.

Finalmente, generar conciencia entre los desarrolladores sobre la importancia de la sostenibilidad, tanto en términos ambientales como humanos es de suma importancia por lo que se requiere una educación o capacitación continua sobre prácticas sostenibles que puedan inspirar un cambio cultural positivo en la comunidad de desarrollo.

V. CONCLUSIONES Y TRABAJOS FUTUROS

La caracterización de prácticas sostenibles en el desarrollo de software presentada en este artículo ofrece ideas valiosas para fomentar un enfoque más respetuoso con el medio ambiente y socialmente responsable en la ingeniería de software. Al integrar estas prácticas en el ciclo de vida del desarrollo de software, las organizaciones pueden no solo lograr una mayor eficiencia energética, sino también contribuir a la equidad social al capacitar a comunidades marginadas con habilidades en desarrollo de software. El énfasis en factores humanos favorece a los equipos y comunidades, mejorando la calidad de los ambientes laborales inclusivos y el acceso de

⁴ <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>

recursos para aquellos que trabajan desde lugares vulnerables, lo redunda en la posibilidad de generar empleabilidad, impactando positivamente la Economía del país. Además, la promoción de la accesibilidad a la infraestructura en la nube subraya la importancia de aprovechar los avances tecnológicos para crear entornos de desarrollo más inclusivos y sostenibles.

Por otro lado, el énfasis en el uso responsable de productos de software sostenible destaca la necesidad de prácticas de consumo consciente dentro de la industria del software. Al fomentar la adopción de estas prácticas, los interesados pueden trabajar hacia la mitigación del impacto ambiental del desarrollo de software, al mismo tiempo que abordan las desigualdades sociales. Se encontraron características relacionadas con el software sostenible, que pueden ser medidas como características de calidad en beneficio de la eficiencia energética. También se relacionaron prácticas en desarrollo de software sostenible desde el ciclo de vida y la sostenibilidad técnica, las cuales permiten ayudar a los ambientes productivos y contribuyen a minimizar el impacto ambiental, social y económico. Además, se caracterizaron prácticas sostenibles en la sostenibilidad humana del software.

Al abordar la sostenibilidad desde ambas perspectivas, el software sostenible y la sostenibilidad humana, se puede construir un entorno de desarrollo de software que sea consciente de su impacto en el mundo, tanto desde el punto de vista ambiental como humano. Esto no solo beneficia a la empresa y al equipo de desarrollo, sino que también crea productos más éticos y socialmente responsables, contribuyendo al desarrollo social de las comunidades.

En resumen, la adopción de prácticas sostenibles en el desarrollo de software representa un paso crucial hacia la construcción de un futuro más sostenible, donde la tecnología sirva como un catalizador para un cambio social y ambiental positivo.

Como trabajos futuros se espera integrar estas prácticas a un modelo de sostenibilidad humana en el desarrollo de software y puede ser validado posteriormente con expertos y con casos de estudio de acuerdo con [18]. También en [18] se usa un modelo de métricas que puede ser aplicado para la evaluación de las prácticas propuestas en el presente trabajo, el valor del grado de cumplimiento se define de acuerdo con la escala establecida en la norma ISO/IEC 15504. Este modelo de valoración usa el paradigma Goal, Question, Metric – GQM [27], el cual propone: (i) un nivel conceptual (Goal), (ii) un nivel operacional (Question) y (iii) un nivel cuantitativo (Metric). Además, se pueden generar aplicaciones en contextos donde la infraestructura tecnológica es limitada frente a recursos energéticos y de conectividad a internet con comunidades vulnerables. También, pueden surgir capacitaciones de desarrollo que involucren educación en prácticas sostenibles en desarrollo de software.

AGRADECIMIENTOS

Este artículo surge como resultado del proyecto “Modelo de sostenibilidad humana en productos de desarrollo de software con infraestructura tecnológica limitada” en el programa de Mujeres en la ciencia 2023 patrocinado por

Minciencias, Icetex y L’oreal-Unesco y la institución ejecutora, Universidad EAFIT.

REFERENCIAS

- [1] C. Calero, M. Moraga, and M. Piattini, *Introduction to Software Sustainability*. Software Sustainability. Springer, 2021.
- [2] Martínez, A., Suescun, E., Noreña, P. A., González, L., and Pardo, C., “Implementación de prácticas DevOps en un sistema de Mainframe Legado,” *Investigación e Innovación en Ingenierías*, vol. 10, no. 2, pp. 129-146, 2022.
- [3] J. Guerrero, K. Zúñiga, C. Certuche, and C. Pardo, “A systematic mapping study about DevOps,” *Journal de Ciencia e Ingeniería*, vol. 12, no. 1, pp. 48-62, 2020.
- [4] L. Restrepo, P. Bernal, E. Suescun, J. Aguilar, and C. Calvache, “Toward a conceptual framework for designing sustainable cyber-physical system architectures: A systematic mapping study,” *Heritage and Sustainable Development*, vol. 5, no. 2, pp. 253-279, 2023.
- [5] C. Wohlin, “Guidelines for snowballing in systematic literature studies and a replication in software engineering,” in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14*, 2014, pp. 1-10, doi: 10.1145/2601248.2601268.
- [6] L. Restrepo, J. Aguilar, M. Toro, and E. Suescun, “A sustainable-development approach for self-adaptive cyber-physical system’s life cycle: A systematic mapping study,” *Journal of Systems and Software*, vol. 180, 111010, 2021.
- [7] Londoño L, *Lenguaje de modelado de arquitectura para sistemas ciber-físicos adaptativos*, Tesis de Maestría, Universidad EAFIT, 2020.
- [8] P. Bernal, *A Framework design for sustainable cyber-physical architectures*, Tesis de Maestría, Universidad EAFIT, 2021.
- [9] D. E. Soto, A. X. Reyes, J. C. Giraldo, A. E. Villamizar, and F. A. Vidal, *Buenas prácticas para el desarrollo de software sostenible*, RISTI, vol. 49, 449-460, 2022.
- [10] S. Semerikov, A. Striuk, L. Striuk, M. Striuk, and H. Shalatska “Sustainability in Software Engineering Education: a case of general professional competencies”. *E3S Web of Conferences* vol. 166, 10036. <https://doi.org/10.1051/e3sconf/202016610036>, 2020.
- [11] R. Heldal, N. T. Nguyen, A. Moreira, P. Lago, L. Duboc, S. Betz, C. Coromã, B. Penzenstadler, J. Porras, R. Capilla, I. Brooks, S. Oyedele, and C. C. Venters, “Sustainability Competencies and Skills in Software Engineering: An Industry Perspective”. *arXiv preprint arXiv:2305.00436*, 2023.
- [12] J. Grundy, H. Khalajzadeh, J. McIntosh, T. Kanij, and I. Mueller, “Humanise: Approaches to achieve more human-centric software engineering. International Conference on Evaluation of Novel Approaches to Software Engineering”, pp. 444-468. Springer International Publishing, 2020.
- [13] V. Amrutha and S. Geetha “A systematic review on green human resource management: Implications for social sustainability”, *Journal of Cleaner Production* <https://doi.org/10.1016/j.jclepro.2019.119131>, 2019.
- [14] V. Ramautar, S. Overbeek, and S. España, “Human Sustainability in Software Development”, pp. 329-348. *Software Sustainability*. Springer, 2021.
- [15] P. Were and J. Madeley, “MunsellFair M. Trade Software: empowering people”, *enabling economies*, *Journal of Fair Trade*, vol. 2, pp. 1-9, 2020.
- [16] D. Peláez y C. Corrales, *Identificación y caracterización de buenas prácticas orientadas al desarrollo de software sostenible desde el punto de vista económico*. Trabajo de grado, Tecnológico de Antioquia, 2018.
- [17] B. Ojameruaye and R. Bahsoon “Sustainability ArchDebts: AnEconomics-Driven Approach for Evaluating Sustainable Requirements”, pp. 369-398. *Software Sustainability*. Springer, 2021.
- [18] C. Pardo, J. Guerrero, and E. Suescun, “DevOps Model in Practice: Applying a Novel Reference Model to Support and Encourage the Adoption of DevOps in a Software Development Company as Case Study,” *Periodicals of Engineering and Natural Sciences (PEN)* vol. 10 no. 3, pp. 221-35, 2022.
- [19] C. Arce, L. Restrepo, E. Suescun, and J. Aguilar, “NFR-based Framework For The Analysis Of Sustainability In Cyber-Physical Systems (CPS)”, *Unpublished*, 2024.
- [20] P.A. Noreña, E. Suescun y J. Mejia, “Desarrollo dirigido por hipótesis en productos de software sostenible: Un mapeo sistemático”, *Unpublished*, 2024.
- [21] S. Kreten, A. Guldner, and S. Naumann, “An analysis of the energy consumption behavior of scaled, containerized web apps”, *Sustainability* vol. 10, no. 8, 2018.
- [22] P.A. Noreña, C. Durango y E. Suescun “Prácticas ágiles en el ciclo de vida DevOps: Una mirada desde *Essence*”, *Unpublished*, 2024.
- [23] N. Forsgren, J. Humble, and G. Kim, *Accelerate. The science of Lean Software and DevOps. Bulding and Scaling High Performing Technology Organizations*. IT Revolution, 2018.
- [24] N. Forsgren, G. Kim, J. Humble, P. Debois, and J. Willis, *The DevOps HandBook. How to create world-class, Agility & security in technology organizations*. 2nd Edition, IT Revolution, 2021.
- [25] D. Norman, D. The design of everyday things: Revised and expanded edition. Basic books, 2013.
- [26] K. Holtzblatt, *Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design (Interactive Technologies)*. Elsevier Science, 2005.
- [27] V. R. Basili, G. Caldiera, and H. D. Rombach, “Software modeling and measurement: The Goal, metric, and question paradigm,” *NASA/GSFC AND AFOSR*, 1992.