



*Asesorías y Tutorías para la Investigación Científica en la Educación Puig-Salabarría S.C.
José María Pino Suárez 400-2 esq a Lerdo de Tejada. Toluca, Estado de México. 7223898479*

RFC: ATI120618V12

Revista Dilemas Contemporáneos: Educación, Política y Valores.
<http://www.dilemascontemporaneoseducacionpoliticayvalores.com/>

ISSN: 2007 – 7890.

Año: V Número: 2 Artículo no.44 Período: Octubre, 2017 – Enero 2018.

TÍTULO: La sostenibilidad y el software.

AUTORA:

1. Dra. Anaisa Hernández González.

RESUMEN: Varias investigaciones se han realizado sobre la sostenibilidad vinculada a las tecnologías y los sistemas de información. Green IT, Green IS, el desarrollo de software sostenible, la gestión de proyectos sostenibles, las métricas verdes y la gestión de procesos de negocio verdes, entre otros, son temas objeto de estudio y en los que se han alcanzado algunos resultados en su aplicación práctica. Este trabajo presenta algunas de las revisiones sistemáticas a la literatura, que sobre el tema de sostenibilidad aplicado al campo del software se han realizado en los últimos años. Además, se identifican algunas brechas que existen en las investigaciones en la relación sostenibilidad-software.

PALABRAS CLAVES: Software verde, Software sostenible, Sostenibilidad.

TITLE: Sustainability and software.

AUTHOR:

1. Dra. Anaisa Hernández González.

ABSTRACT: Several researches have been carried out on sustainability linked to the technologies and the information systems. Green IT, Green IS, development of sustainable software, sustainable projects management, green metric and the business management of green processes among others, are topics of studies with some results reached in their practical application. This study presents systematic revisions of the literature developed within the sustainability topic applied to the field of software in the last years. Also, some gaps about the relationship sustainability-software are identified.

KEY WORDS: Green software, sustainable software, sustainability.

INTRODUCCIÓN.

Las presiones de la actividad humana sobre el medio ambiente afectan la calidad y cantidad de los recursos naturales disponibles; a lo que la sociedad responde con la definición de políticas económicas y ambientales que causan cambios en la conciencia y el comportamiento (Ou, Ley, & Chen, 2011).

Para disminuir el impacto al medio ambiente de la tecnología, la Agencia de Protección Ambiental de los EUA introduce en 1992 el programa Energy Star (Boyd, Dutrow, & Tunnessen, 2008) para promover los productos eléctricos con consumo eficiente de electricidad, reduciendo de esta forma la emisión de gas de efecto invernadero por parte de las centrales eléctricas. De esta manera, los productos e instalaciones que cumplen o superan determinados indicadores, son marcados para informar a empresas y consumidores que se está en presencia de una solución eficiente energéticamente.

Cuando en el año 2011 Google publicó que en el 2010 se había emitido 1,46 millones de toneladas de CO₂ como resultado del envío de e-mail por los usuarios (Google, 2011), el tema del impacto de las tecnologías de la información (TI) en la sostenibilidad ambiental recibió un importante impulso.

En aquella época, (Hilty & Lohmann, 2011), los temas más descuidados en la relación TI con el medio ambiente estaban vinculados con: la falta de transparencia de los costos energéticos en los servicios de TI (que provocaba problemas de correlación entre el precio del servicio y el costo de la energía que genera), la demanda de material para producir el hardware (debido a la cantidad de elementos químicos que se usan para construirlos, lo que hace difícil y menos eficiente el reciclaje, y a que otros productos como el indio que se usa en las pantallas de los monitores en concentraciones pequeñas, por lo que es difícil recuperarlo y tiene una alta demanda), la insuficiente comprensión del ciclo de vida de las TI (que consumen más energía durante la etapa de uso en relación con las etapas de producción y tratamiento al final de la vida útil), los efectos de rebote (que provocan que la respuesta habitual de los ingenieros de software a un aumento de la capacidad de potencia en procesamiento y almacenamiento disponible a un precio determinado, es captura más de lo mismo), y que las TI se han constituido como una infraestructura crítica porque cada vez la sociedad y los individuos dependen más de ella.

En (Costello, 2011), el autor presenta dos listas (tendencias de la tecnología y capacidades estratégicas) con diez elementos cada una, que en base a su experiencia y conocimiento de la empresa, deberán ser tomados en cuenta por las organizaciones para su implementación. Reconoce que pocas organizaciones toman en cuenta todos los elementos. El #8 de la 1ra lista es “Green IT (Information Technology) y la sostenibilidad” y plantea, que hasta ese momento, la mayoría de las empresas que optan por “Green” lo hacían para reducir costos de energía, papel, hardware y otros artículos que afectan la línea base; sin embargo, las regulaciones que ya existían y las que están por venir, requieren que las compañías operen de forma que se facilite el monitoreo, la información y el impacto medio ambiental.

En (Unkehar, 2011), el autor pronostica lo que debía ocurrir en los próximos cinco años asociados con la corriente y tendencias futuras para el creciente campo de las Green IT. Este trabajo es del 2011 y en un primer momento ocurría en ese primer año, y estaba orientado hacia la reducción de las emisiones de carbono vinculadas con las TI con medidas como el uso de monitores Green, el

uso de papel y tóner reciclados y la virtualización de los centros de datos. Entre el 1er y 3er año, las empresas que usan TI reducirían la emisión de CO₂ a través de la incorporación de Green BPM (Green Business Process Management), implementando software que mejore las emisiones de carbono y la adopción de la ISO 14001. El tercer momento estaría marcado por la colaboración verde a través de la adopción de una actitud holística verde, el emprendimiento de la colaboración entre los negocios, la selección de energía renovable, y el uso de la Biomimesis.

Durante estos años, varios trabajos han sido presentados en eventos y publicados en revistas especializadas en los que se aborda la investigación y experiencia en el campo de la sostenibilidad vinculada a las tecnologías y los sistemas de información. En este trabajo se utilizan como referencia estudios secundarios sobre los temas de Green IT, Green IS (Information Systems), las medidas del software verde, la sostenibilidad en la ingeniería del software, el desarrollo del software sostenible, la sostenibilidad en el negocio, y la gestión de proyectos sostenibles, para introducir a los especialistas en la bibliografía inicial que puede consultar para conocer qué se ha hecho sobre estas temáticas y sobre cuáles aspectos existen brechas para continuar investigando. No se procura realizar un estudio terciario (revisiones sistemáticas de revisiones sistemáticas), la pretensión es presentar el estado del arte, usando como base el estado de algunas revisiones sistemáticas a la literatura sobre los temas anteriormente mencionados.

DESARROLLO.

Conceptos asociados con la sostenibilidad.

La Comisión Mundial sobre el Medio Ambiente en el año 1987 (Brundtland G. H. et. al., 1987) definió al desarrollo sostenible como el desarrollo que satisface las necesidades del presente sin comprometer la capacidad de las generaciones futuras para satisfacer sus necesidades. La sostenibilidad se puede ver en tres dimensiones: ambiental (pone el acento en preservar la biodiversidad sin tener que renunciar al progreso económico y social), económica (se encarga de

que las actividades que buscan la sostenibilidad ambiental y social sean rentables) y social (busca la cohesión de la población y una estabilidad de la misma).

En el año 2015 sale a la luz un manifiesto que recoge un conjunto de principios y compromisos vinculados con el diseño sostenible (Becker, 2015). Sus suscriptores se refieren a que: la sostenibilidad tiene múltiples dimensiones (social, medioambiental, económica, técnica y humana) y que todas hay que incluirlas en el análisis; la sostenibilidad es sistémica y nunca es una propiedad aislada, la sostenibilidad trasciende a múltiples disciplinas por lo que las personas de diferentes disciplinas deben trabajar juntas. La sostenibilidad se aplica al sistema y al contexto más amplio al que forma parte, la visibilidad del sistema es una condición previa necesaria y disponible para el plan de sostenibilidad; la sostenibilidad requiere de la acción de múltiples niveles. La sostenibilidad requiere de pensamiento a largo plazo y es posible satisfacer las necesidades de generaciones futuras sin sacrificar a la generación actual.

El término de computación verde (Abdullah, Abdullah, Din, & Tee, 2015), se refiere a un modelo que incluye el estudio y la práctica de las tecnologías de la información relacionadas con la arquitectura hardware, software, redes, sistemas y procesos, con el fin de gobernar la huella de carbono del medio ambiente mediante el uso de los recursos informáticos de manera eficaz y eficiente con un mínimo de efectos negativos para el medio ambiente y sin comprometer la productividad económica y la responsabilidad social a largo plazo. En la actualidad se utilizan dos términos diferentes para abarcar la multiplicidad de aristas asociadas con la computación y la sostenibilidad: Green IT (Green Information Technologies o Tecnologías de la Información Verdes) y Green IS (Green Information Systems o Sistemas de Información Verdes).

La variedad de definiciones de Green IT y Green IS (Chan & Johansson, 2014) puede deberse a que se está en las etapas iniciales de la investigación en este campo. En este sentido, los autores resumen que cuando se habla de Green IT se refiere al estudio de la producción, aplicación, operación y disposición de las TI y productos basados en las tecnologías durante todo su ciclo de vida con un mínimo o ningún impacto ambiental. En este trabajo se concluye, a partir de las

publicaciones revisadas, que las tecnologías de la información son fuentes de problemas del medio ambiente y que solo su uso eficiente y efectivo, y la producción de tecnologías de la información con un mínimo impacto ambiental, puede minimizar este impacto; lo que es objeto de estudio de Green IT.

Según (Hedwing, Malkowski, & Neumann, 2009), el término Green IT se refiere a todas las actividades y esfuerzos que cuenten con tecnologías y procesos ecológicamente amigables en todo el ciclo de vida de la tecnología de la información y la comunicación. Por su parte, (Tuskeentushi, Sedera, & Recker, 2013) lo define como iniciativas que se centran en la remodelación de las tecnologías de la información en formas respetuosas del medio ambiente y promueve el diseño, la fabricación y la gestión de equipos y servicios que consumen un mínimo de energía a lo largo de todo su ciclo de vida.

Cuando se escribe sobre Green IS se refiere a los procesos de la organización para la mejora del comportamiento medio ambiental; es decir, lo consideran una solución para reducir las degradaciones ambientales que provocan las tecnologías de la información. De acuerdo a (Watson, Boudreau, & Chen, 2010), contrasta con Green IT en que no se limita a la eficiencia energética y la utilización de los equipos de TI, porque se centra en el diseño e implementación de sistemas de información que contribuyen a la implementación de procesos de negocio sostenibles.

Dentro del término de Green IT se reconocen varias subdivisiones. ***Green IN IT*** se utiliza cuando el verde se refiere a las tecnologías de información (software o hardware), y dentro de ella se identifica a ***Green IN software engineering*** (proceso, producto) y ***Other Green software aspect*** (procesos de negocio, gobernanza). En cambio, ***Green BY IT*** se usa cuando las tecnologías de la información son utilizadas como una herramienta para soportar metas verdes, es decir, uso de aplicaciones y métodos con la intención de producir pocos residuos para el medio ambiente y reducir el consumo de energía tanto como sea posible a través de las tecnologías de la información. Según (Calero, Moraga, & Bertoa, 2013), el impacto más directo de los productos de software al medio ambiente está relacionado con el consumo de energía, pero plantean que otros recursos

también pueden tener impacto negativo en la sostenibilidad del software, y mencionan el uso de la CPU, el incremento de las necesidades de memoria y almacenamiento en disco duro, y la utilización de la red y el ancho de banda. Estos autores proponen una definición de software sostenible tomando como base la propuesta de (Brundtland G. H. et. al., 1987) y lo definen como: “a un modo de desarrollo de software en el que el uso de recursos tiene como meta satisfacer las necesidades del producto de software mientras se garantiza la sostenibilidad del sistema natural y el medio ambiental”.

Hay otros productos como el papel, la tinta y el tóner que se utilizan como resultado del uso del software producido, que también son relevantes cuando se trata de la sostenibilidad del producto de software.

El término de software verde o sostenible según (Abdullah, Abdullah, Din, & Tee, 2015) y (Karn, Dick, & Naumam, 2015) se refiere al diseño y producción de software que tiene efecto negativo directo o indirecto en la economía del país, la gente, la sociedad y el medio ambiente que son el resultado de antes, durante y después de la ejecución de las fases de desarrollo del software de forma negligente o que tienen un efecto positivo en la producción de software sostenible con el medio ambiente; con el objetivo de producir pocos residuos y consumir eficientemente la energía.

En (Lami & Buglione, 2012) se adiciona a las características del producto propuestas en la taxonomía del estándar ISO/IEC 25010:2011, la sostenibilidad definiéndola como la “capacidad del producto de software para satisfacer las necesidades actuales de las funcionalidades requeridas sin comprometer la capacidad de satisfacer las necesidades futuras”; por lo que estos autores valoran esta nueva característica desde la infraestructura, los recursos humanos, los procesos y el producto; es decir, que no solo se vea desde el punto de vista medioambiental, pues el ahorro de recursos debe ser de todo tipo, incluyendo la correcta gestión del personal y su trabajo intelectual y la planificación del proyecto (el proyecto es sostenible si se libera en tiempo, con calidad y en presupuesto).

Aunque se coincide con los autores de este trabajo, en la práctica a estos últimos se le ha prestado mayor atención que a la sostenibilidad medioambiental tomando en cuenta para considerar el éxito o fracaso de un proyecto.

En (Rashid & Khan, 2015) se define como ingeniería de software verde a un paradigma emergente que tiene como objetivo desarrollar software con características verdes para reducir el impacto negativo en el medio ambiente. Por su parte, en (Ahmad, Barahom, & Hussain, 2014) se define a la ingeniería de software sostenible como desarrollo de productos de software para sistemas de vida largos que puedan satisfacer las necesidades del presente y de las futuras generaciones con la integración de los tres pilares de la sostenibilidad (medio ambiente, económico y social) para cumplir con requisitos básicos.

La gestión de procesos de negocio soporta el ciclo de vida de los procesos de negocio mediante la participación de las diferentes partes que actúan de manera cooperativa y tratará de alcanzar los objetivos de negocio comunes. Las actividades incluyen la definición, ejecución, seguimiento, control, análisis y mejora de procesos de negocio (Dos Santos & Fantinato, 2013) y ha surgido el término de Green BPM.

Desde la perspectiva de un investigador en ingeniería de software, en (Morioka & Monteiro de Carvalho, 2016) se define a Green BPM como la suma de todo: ingeniería de software-actividades de apoyo a la gestión, que ayudan a monitorear y reducir el impacto medioambiental de los procesos de negocio en su diseño, mejora, implementación u operación, así como llevar a cabo un cambio cultural dentro del ciclo de vida del proceso. El marco de investigación, según estos autores, se puede ver en tres dimensiones: actividades de gestión (planificación, ejecución, comprobación y actuación), etapas del ciclo de vida del proceso (diseño, medición/monitoreo, mejora/cambio y operación) y objetivos/metas del Green BPM (reducir el impacto ambiental y ayudar al cambio cultural dentro de la cadena del proceso).

Para (Hoesch-Klohe, Ghose, & Lam-Son, 2010), Green BPM describe a una nueva tecnología que extiende BPM (los procesos de diseño, análisis, ejecución y monitoreo) vinculados con la huella

de carbono: por lo tanto, cuando se habla de desarrollo de software sostenible, se está haciendo referencia, según (Calero, Bertoa, & Moraga, Sustainability and quality: icing on the cake, 2013), a un modo de desarrollo de software en el que el uso de recursos para satisfacer las necesidades del producto de software garantiza al mismo tiempo la sostenibilidad del sistema natural y del medio ambiente.

Estado del arte.

Cuando se habla de investigación se está haciendo referencia a la acción de investigar, de intentar descubrir o conocer algo estudiando, examinando atentamente cualquier indicio o realizando las diligencias para averiguar o aclarar un hecho. Diferentes autores han documentado diversas clasificaciones para los tipos de métodos de investigación que pueden aplicarse.

En función del nivel de evidencia que proporciona, en (Genero, Cruz, & Piattini, 2014) se clasifican en métodos primarios, secundarios y terciarios. Los primeros permiten *“obtener evidencia empírica sobre un tema de interés”* y los segundos permiten *“recopilar de manera sistemática y rigurosa los estudios primarios relacionados con una investigación específica con el objetivo de sintetizar la evidencia disponible para responder a dicha pregunta”*. Los estudios terciarios, por su parte, revisan estudios secundarios relacionados con la misma pregunta de investigación.

Las revisiones sistemáticas a la literatura (SLR- Systematic Literature Review) son un tipo de estudio secundario que sintetiza la evidencia empírica existente sobre un tema de interés (Genero, Cruz, & Piattini, 2014). Según (Kitchenham & Charters, 2007), son un medio para identificar, evaluar e interpretar toda la investigación disponible para responder a preguntas de investigación específicas. En (Fink, 2010) se define como un *“método sistemático, explícito y reproducible para identificar, evaluar y sintetizar”*.

Según (Genero, Cruz, & Piattini, 2014) y (Zhang & Ali Babar, 2013), las razones que justifican la decisión de usar las SLR como punto de partida para el estudio se centran en: permiten obtener nuevos hallazgos y la proposición de las ideas innovadoras para futuras investigaciones; la planificación y ejecución sistemática y metódica de la revisión de estudios primarios proporciona un mayor nivel de validez de las conclusiones de la SLR, de lo que se podría concluir que en cualquiera de los estudios individuales analizados se logra, sobre un tema específico, recopilar la evidencia empírica publicada en un periodo dado.

SLR revisadas.

Varias revisiones sistemáticas a la literatura han sido publicadas en los últimos años sobre la sostenibilidad asociada a las tecnologías y los sistemas de información. Los temas de Green IT, Green IS y la sostenibilidad en la ingeniería de software han recibido mayor atención por sobre otros como las medidas del software verde, el desarrollo del software sostenible, la sostenibilidad en el negocio y la gestión de proyectos sostenible (Hernández, 2016). En la tabla 1 se presentan algunos de estos estudios, que a juicio de la autora, pueden servir como referentes para comprender los caminos que se han seguido en este campo (Hernández, 2016).

Tabla 1 Resumen de las SLR revisadas.

SLR.	Finalidad.
Sustainability in software engineering; systematic literature review (Pezenstandler, Calero, & Franch, 2013)	Proporcionar una visión general de diferentes aspectos de la sostenibilidad en la investigación de Ingeniería de software profundizado en los trabajos publicados durante 1991-2011, los temas abordados, las limitaciones de la investigación actual, cómo está apoyada la sostenibilidad, los métodos usados, los casos de estudio disponibles y los dominios considerados.
A systematic literature review a green software metrics (Bozzelli, Gu, & Lago, 2013)	Describir y clasificar las métricas de consumo de energía del software y analizar la evolución de esas métricas en términos de método, contexto y evaluación.

SLR.	Finalidad.
A systematic literature review on sustainability studies in software engineering (Ahmad, Barahom, & Hussain, 2014)	Aportar a lo publicado sobre ingeniería de software sostenible centrándose en el enfoque y métodos utilizados y averiguar las limitaciones de las obras existentes.
A systematic literature review of Green software development in collaborative knowledge management (Abdullah, Abdullah, Din, & Tee, 2015).	Cubrir la brecha de la implementación de técnicas de gestión de conocimiento en el Desarrollo de Software Verde (DSV) para lo cual se proponen elaborar una SLR sobre DSV y evaluar cómo la gestión de conocimiento puede ayudar en el DSV.
Green IT segment analysis an academic literature review (Tuskeentushi, Sedera, & Recker, 2013)	Evaluar la situación de la literatura sobre Green IT.
Green IS- a systematic literature review (Chan & Johansson, 2014).	Obtener una visión más profunda y clara del estado de la tecnología de IS para la investigación de sostenibilidad del medio ambiente con el objetivo de aclarar qué es Green IS y qué es Green IT porque es importante tener claras estas definiciones para avanzar en la transformación de la sociedad en una dirección de sostenibilidad.
The status quo and the prospect of Green IT and Green IS: a systematic literature review (Esfahani, Rahma, & Zakania, 2015)	Definir el estado de la investigación en sostenibilidad ambiental en la comunidad de los Sistemas de Información verdes y las Tecnologías de Información verdes e identificar los aspectos en los que se deben centrar las investigaciones futuras.
Valving business sustainability: a systematic review (Peloza & Yachnin, 2008)	Mostrar que, a pesar de las tres décadas de trabajo en este campo, no existen métricas consistentes para medir la sostenibilidad y que se presta poco atención a las métricas que relacionan la sostenibilidad y el rendimiento financiero de la empresa.
Green Business Process Management- A definition and research framework (Optiz, Krup, & Kolbe, 2014)	Analizar el estado de la investigación en Green IS, BPM y Green BPM y definir Green BPM y un marco multidimensional para investigaciones futuras.

SLR.	Finalidad.
A systematic review towards a conceptual framework for integrating sustainability performance into business (Morioka & Monteiro de Carvalho, 2016)	Examinar cómo en la literatura son abordadas las temáticas de medición, gestión y presentación de informes a partir de la hipótesis de que se analizan por separado y que deben integrarse para un desempeño de sostenible del negocio y proponer un marco para la integración del desempeño de la sostenibilidad para mejorar la contribución de la empresa al desarrollo sostenible.
Green agility for global software development vendor: a systematic literature review protocol (Rashid & Khan, 2015)	Identificar los factores de éxito y los riesgos que puede tener un efecto directo e indirecto en el desarrollo de software verde y sostenible utilizando métodos ágiles con el objetivo de obtener un modelo que ayude a los proveedores de desarrollo global a medir su madurez verde-ágil para el diseño y la producción de software verde y sostenible.
Towards sustainable Project management. A literature review (Otegi, Aguiar, Aguilar, Cruz, & Fuentes, 2015)	Identificar las tendencias en la introducción de la sostenibilidad en la gestión de proyectos.
Integrating science and business models of sustainability for environmentally-challenging industries such as secondary lead smelters: A systematic review and analysis of findings (Genaidy, y otros, 2010)	Revisar y evaluar críticamente los modelos de sistemas sostenibles con especial énfasis en la industria de fundición de plomo secundario en Estados Unidos.

El Software y la Sostenibilidad.

La reducción en el consumo de energía durante las etapas de desarrollo del software y el hardware, la utilización del producto de software y el hardware informático, y el fin de los desechos han cobrado cada vez más relevancia en las investigaciones que se realizan en el mundo; lo que se

evidencia en el creciente número de estudios primarios que sobre el tema se encuentran y que han propiciado que se realicen estudios secundarios sobre los mismos.

El proceso de desarrollo de software para ser sostenible tiene que hacer un uso responsable de los recursos que consumen energía y que usan materias primas no renovables o que son difíciles de reciclar.

El software que se elabore tiene que construirse de forma tal que sea fácil de mantener; es decir, que los cambios que se produzcan en el negocio que lo impacten, puedan realizarse con un costo mínimo de manera que se alargue su vida útil ya que es más eficiente adecuarlo a las necesidades que reemplazarlo por otro software.

Existen varias métricas verdes que permiten evaluar la energía, el rendimiento, la utilización, el consumo y el impacto económico; por lo que no existe justificación para su no aplicación como medida de control de la influencia de las tecnologías y los sistemas de información en el medio ambiente y del rendimiento de las empresas que optan por la implantación de métodos verdes en el desarrollo de software.

Es posible identificar un problema científico vinculado con *la relación que existe entre el uso de un framework de desarrollo para la obtención de aplicaciones web que informatizan los procesos de una organización y el desarrollo sostenible, en particular, en su dimensión medioambiental*; para lo cual se requiere explorar varias medidas de sostenibilidad en los framework más utilizados en el contexto de las aplicaciones reales que elaboraran los desarrolladores de software.

El término “framework” o marco de trabajo identifica a un conjunto estandarizado de conceptos, prácticas y criterios que se utilizan para enfocar en un tipo de problemática particular en la que sirve como referencia para enfrentar y resolver nuevos problemas o situaciones de índole similar. Para los desarrolladores proporciona una base para crear programas usando un determinado lenguaje de programación y una arquitectura de software, simplificando el proceso de desarrollo ya que proporciona elementos para procesar entradas, gestionar los dispositivos de hardware e

interactuar con el software del sistema, y por lo general, dan solución a varios de los problemas frecuentes del diseño en los sistemas de mediana y alta complejidad.

En (Riehle, 2000) se plantea que “representa una arquitectura de software que modela las relaciones generales de las entidades del dominio y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio; por lo tanto, cuando se habla de framework en el desarrollo de software, se refiere a un esquema para el desarrollo y/o implementación de una aplicación. De aquí se infiere que proporcionan una estructura global para la aplicación, por lo que el uso de un framework implica un costo inicial de aprendizaje que se revertirá en un incremento en la productividad tanto en el desarrollo como en el mantenimiento de un producto de software.

Se revisó la literatura científica buscando investigaciones que evaluaran el impacto del uso de framework en la generación de código sostenible, no identificándose estudios particulares del tema, pero sí algunos trabajos en el campo de la evaluación de la calidad.

En el estudio realizado, hasta el momento se encontró lo siguiente, que puede ser un marco de referencia para el trabajo (Parra & Hernández González, 2017). La comunidad de desarrolladores que mantienen un activo intercambio por la red sobre las ventajas y desventajas y la comparación entre framework, usan diferentes criterios de comparación, siendo los más frecuentes: la **Madurez** (edad, estabilidad, historial de Bugs y errores disponibles en el sitio de cada framework), la **Adopción** (popularidad, referencias, contribución de la comunidad y publicaciones), el **Liderazgo de desarrollo** (líder del grupo y estilo de dirección), la **Actividad** (número de desarrolladores identificados, actividad en los errores (bugs), actividad en las funcionalidades y características y actividad de las versiones), la **Independencia de los desarrollos**, los **Servicios** (formación y/o entrenamiento, soporte y consejos), la **Documentación** (páginas de ayuda, guías, foros en donde consultar sobre cualquier inconveniente, etc.), la **Operatividad** (facilidad de uso, administración/supervisión), la **Modularidad**, los **Subproductos** (facilidad de modificación del código existente y facilidad de extensión del código), la **Licencia**, los **Sponsor**, **Quién utilizará/modificará la**

aplicación (en cantidad y conocimientos), la **Arquitectura de aplicaciones** (incorporación del patrón Modelo Vista Controlador, soporte a las operaciones CRUD (Create, Retrieve, Update y Delete), independiente del manejador de base de datos, soporte a **múltiples bases de datos** sin necesidad de cambiar nada, inclusión de otros objetos de bases de datos (por ejemplo, TableGateway), la **Implementación de código HTML** (uso de plantillas e implementación de ayudantes de plantillas), la **Seguridad** (manejo propio de sesiones por usuarios, manejo de privilegios de acceso a secciones de la aplicación y verificación de la salida generada en HTML por procesamiento de peticiones), la **Usabilidad y acceso rápido** (almacenamiento en caché de las vistas y almacenamiento en caché de configuración de las aplicaciones), **Herramientas de programación** (generación de código PHP, herramientas de prueba y depuración, interfaz de línea de comandos para la creación y mantenimiento de aplicaciones y almacenamiento de logs de funcionamiento del framework) y la **Extensibilidad y opciones adicionales** (integración con otras herramientas a través de plugins, soporte para Web Services, soporte para envío de correo electrónico, generación de archivos PDF y soporte para internacionalización y localización de contenidos). Como se aprecia, son disímiles los criterios que se usan para comparar y seleccionar un framework de desarrollo de software sobre otro, pero explícitamente ninguno de estos estudios evalúa el impacto de su uso y del producto de software generan en el medio ambiente, así como del producto de software.

Medidas de sostenibilidad.

En (Seacord, y otros, 2003) se define la sostenibilidad como una característica, y como tal, se propone cómo medirla a partir del esfuerzo requerido para adaptar el software, el tiempo que toma hacer la adaptación y la cantidad de adaptaciones requeridas. Como se aprecia, esta forma de medir está en consonancia con lo propuesto por (Lami & Buglione, 2012), pero directamente no se refiere al impacto medioambiental, una arista imprescindible a abordar en las investigaciones.

Para (Mfon-Abasi, 2014), la sostenibilidad está asociada con la resistencia y los procesos, y que el principio de organización para la sostenibilidad es el desarrollo sostenible que incluye cuatro dominios interconectados: ecología, economía, política y cultura. Valoran que en la definición de software sostenible hay varias definiciones, que agrupa en tres enfoques. Aquellos que se refieren a la capacidad para hacer frente a los cambios por lo que se asocia a las cualidades como la fiabilidad, la adaptabilidad y la capacidad de mantenimiento. Otra vertiente se refiere a su impacto directo en el medio ambiente debido al consumo de energía y los residuos electrónicos de computadoras que se hacen obsoletas debido a actualizaciones del software, y el tercero, a los efectos indirectos en el medio ambiente por el comportamiento humano sostenible que puede producir su utilización diaria.

En (Zhang, 2012) se justifica la necesidad de crear aplicaciones eficientes para reducir el consumo de energía con el fin de aumentar el tiempo de duración de la batería en un móvil. En tal sentido, este trabajo se centra en el impacto del consumo de energía de la CPU a partir de los estados de consumo de energía de los procesadores modernos (estados activos y estados de suspensión) y cómo el consumo de energía de la CPU se ve afectada por algunos usos de WinRT API. Aunque esta no fue la finalidad de este trabajo, si evidencia que medir el consumo de energía de la CPU es un indicador cuando se quiere evaluar el impacto medioambiental del uso de las TI.

En (Calero, Moraga, & Bertoa, 2013) se propone adicionar, al modelo de calidad del producto de software propuesto por la ISO/IEC 25010, la característica de sostenibilidad con tres subcaracterísticas: el consumo de energía, la optimización de recursos y la perdurabilidad. Los resultados de este estudio constituirán un importante criterio, desde la perspectiva de la sostenibilidad medioambiental, a tomar en consideración en el momento donde se defina la arquitectura del software a construir al ser un indicador que permita inclinar o no la balanza ante la decisión del framework de desarrollo a usar en la construcción de una aplicación web.

En (Akinli, 2013) se mencionan como métricas verdes para evaluar la eficiencia energética del sistema a: el % de uso de la CPU, de los dispositivos de entrada/salida y el medio de

almacenamiento, el rendimiento del trabajo (consumo de energía/trabajo completado en vatios/transacciones por segundo) y el uso de la energía del sistema (Kwh).

En (Bourdon, Noureddine, Rouvoy, & Seintuner, 2012) se indica que es necesario medir los recursos que usan los servidores, centrándose en el uso de la CPU, de la memoria, de la actividad de entrada/salida y del tráfico de la red.

Se buscaron publicaciones que analizaran métricas de sostenibilidad en el software a partir de aplicar estudio de casos. No se encontró nada que comparara framework con esta óptica, pero resultó interesante el trabajo de (Hinde, 2012). Esta investigación prueba más de 500 versiones de Firefox 3.6 para demostrar que el cambio de software puede inducir cambios en el consumo de energía. Evalúa tres variables independientes (uso del espacio de CPU, acceso a disco, y uso de la cache) y concluye que no hay relación entre el LOC y el consumo de energía, pero que sí es relevante el uso de la memoria y el CPU. Los cambios impactan no solo en el consumo que se produce para implementarlos, sino además, en el funcionamiento de la aplicación con respecto a la versión anterior.

Métodos tradicionales y ágiles.

Un método de desarrollo de software proporciona la experiencia técnica para elaborar software. Incluyen un conjunto amplio de tareas como la comunicación, el análisis de los requerimientos, la modelación del diseño, la construcción del programa, las pruebas y el apoyo. Se basan en un conjunto de principios fundamentales que gobiernan cada área de la tecnología e incluyen actividades de modelación y otras técnicas descriptivas.

Como parte del estudio que se ha realizado, se revisó la literatura científica para identificar la existencia de investigaciones que abordaran la comparación de los métodos tradicionales y ágiles en cuanto a la sostenibilidad, encontrándose que existen referencias a la sostenibilidad de las metodologías ágiles. Las publicaciones de los últimos 10 años se concentran en presentar los

beneficios del uso del enfoque ágil sobre el enfoque tradicional, o del ágil con algún método tradicional en particular (cascada, espiral u otro).

Los resultados publicados recalcan los beneficios siguientes de los ágiles sobre los tradicionales (Kumar & Kumar, 2014), (Papadopoulos, 2015) y (da Silva & Di Felippo, 2016): mejora la calidad del producto, mejor satisfacción del cliente, mayor moral del equipo, aumenta la colaboración equipo-cliente y entre los miembros del equipo, estructuras personalizadas del equipo, más métricas relevantes y precisas para estimar el tiempo y el costo, mejora la visibilidad del rendimiento, mayor control del proyecto, mejora la previsibilidad del proyecto, y reducción del riesgo eliminando la posibilidad de fracaso del proyecto.

Los criterios que se utilizan para comparar los enfoques son (Chan & Thong, 2009), (Ahmed, Ahmad, Ehsam, Mirza, & Sarwar, 2010), (Kumar & Kumar, 2014) y (Garousi, Coskuncay, Demiröos, & Yacizi, 2016): documentación, participación de los clientes, tamaño del equipo, definición de la arquitectura, énfasis, requisitos, contrato, roles que participan, cultura organizacional, momento en que entrega el producto, flexibilidad, enfoque, medición del éxito, ciclos, plan del proyecto y retorno de la inversión.

Como se observa, explícitamente no se declara la sostenibilidad como un criterio de comparación, aunque algunas de las medidas vinculadas con la sostenibilidad (según Calero, Moraga y Bertoa 2013) podrían potenciarse si se aceptan como válidos los beneficios del enfoque.

Las investigaciones que se presentan en (Rashid & Khan, 2015) evalúan los factores de éxito para la adaptación de técnicas ágiles que puedan ayudar a los ingenieros de software en las organizaciones de Desarrollo Global de Software (DGS) para el diseño de software verde y sostenible, siendo el diseño simple, el desarrollo interactivo y la codificación estándar los más frecuentes en los trabajos revisados.

Los cambios en los requisitos en cualquier momento, el desarrollo de software con eficiencia energética, y el desarrollo de componentes de software de pequeño tamaño, prácticamente no son considerados. Los factores son: eficiente uso de los recursos tiempo e informáticos,

documentación mínima, retrabajo / reingeniería mínimas, costo reducido, mejora de la calidad, mejora de la gestión del ciclo de vida del producto, reducción al mínimo de los desechos electrónicos, diseño simple, refactorización, integración continua, codificación estándar, prevención temprana de efectos, optimización de los procesos, cambios de requisitos en cualquier momento, la planificación ágil y requisitos ágiles, entrega rápida, desarrollo interactivo, desarrollo temprano, la prueba / validación continua temprana, revisión y previsión, desarrollo de componentes de software de pequeño tamaño, colaboraciones eficientes entre los desarrolladores y clientes, energía para el Desarrollo de software eficiente, y la coordinación eficaz entre los miembros de los equipos ágiles.

Resultaría interesante comparar estos métodos en cuanto el consumo de energía y la perdurabilidad, midiendo el tiempo de desarrollo de la aplicación y el requerido para modificar algo sin introducir defectos.

Gestión de procesos de negocios verdes.

En la SLR, que se describe en (Optiz, Krup, & Kolbe, 2014), los 108 trabajos que revisan, refieren que solo 11 abordan Green BPM. Plantean que las investigaciones se enfocan en describir el estado actual dentro de las empresas o el diseño de métodos, herramientas u otros artefactos para llevar a realidad las metas de Green BPM. Proponen tres dimensiones en el marco de investigación de Green BPM: **actividades de gestión** (planear, hacer, comprobar y actuar), **etapas del ciclo de vida del proceso** (diseño, medición/monitoreo, mejora/cambio y operación) y **objetivos/metas de Green BPM** (reducir el impacto ambiental y ayudar al cambio del ciclo de vida del proceso).

En (Morioka & Monteiro de Carvalho, 2016), los autores valoran que en la literatura se abordan por separado los aspectos relacionados con las dimensiones económica, ambiental y social, y que la mayoría de los trabajos hace énfasis en la dimensión ambiental, en específico, en el impacto de los aspectos ambientales en el desempeño financiero de la empresa.

El marco para la integración del desempeño de la sostenibilidad en el negocio que proponen incluye: principios de sostenibilidad corporativa (promoción de los valores compartidos para impulsar la toma de decisiones), los factores que influyen en los elementos de negocio sostenibles (factores de contexto interno y externo) y los elementos de negocio sostenibles (procesos y prácticas, capacidades, ofertas y contribuciones). En la propuesta incluyen los tres enfoques de desempeño de la sostenibilidad que evalúan en los trabajos publicados (medición, gestión y presentación de informes).

En (Nuhoff-Isakhanyan & Wubben, 2016) se comentan los beneficios ambientales, económicos y sociales de la colaboración entre las organizaciones en las empresas con base biológica, vinculadas con la reducción de emisiones de carbono y gases de efecto invernadero, la reducción del uso de fertilizantes y la reducción de la eliminación de residuos. Sustentan que solo se puede ser sostenible en este tipo de empresas a través de la colaboración, ya que aporta ventajas competitivas y oportunidades de sinergia a lo largo de las cadenas de procesamiento con respecto a la corriente y el intercambio de recursos o al uso compartido de recursos como materia prima, edificios, personal, infraestructura e información; sin embargo, no hacen alusión a cómo se gestionan los procesos de negocios para lograr estos beneficios.

En (Jakobi, Castelli, N., & Stevens, 2016), los autores sostienen que faltan enfoques integrados en cuanto al formato de la sostenibilidad en las organizaciones y que deriven en estrategias para la reducción de la brecha entre la estrategia de planificación y el trabajo diario, con la finalidad de gestionar estrategias de sostenibilidad de manera más eficaz y eficiente. Los autores desarrollan un proyecto con el fin de obtener una comprensión más sofisticada de cómo conducir la gestión de procesos de negocio verdes de colaboración con la naturaleza.

CONCLUSIONES.

La reducción en el consumo de energía durante las etapas de desarrollo del software y el hardware, la utilización del producto de software y el hardware informático y el fin de los desechos ha cobrado cada vez más relevancia en las investigaciones que se realizan en el mundo; lo que se evidencia en el creciente número de estudios primarios que sobre el tema se encuentran que han propiciado que se realicen estudios secundarios sobre los mismos.

El proceso de desarrollo de software para ser sostenible tiene que hacer un uso responsable de los recursos que consumen energía y que usan materias primas no renovables o que son difíciles de reciclar.

El software que se elabore tiene que construirse de forma tal que sea fácil de mantener; es decir, que los cambios que se produzcan en el negocio que lo impacten, puedan realizarse con un costo mínimo de manera que se alargue su vida útil, ya que es más eficiente adecuarlo a las necesidades que reemplazarlo por otro software.

Existen varias métricas verdes que permiten evaluar la energía, el rendimiento, la utilización, el consumo y el impacto económico; por lo que no existe justificación para su no aplicación como medida de control de la influencia de las tecnologías y los sistemas de información en el medio ambiente y del rendimiento de las empresas que optan por la implantación de métodos verdes en el desarrollo de software.

De acuerdo a la literatura consultada, existe una brecha en las publicaciones de este tipo asociadas con el concepto de Green BPM, la generación de código sostenible usando framework y la comparación de los métodos ágiles y los tradicionales en cuanto a la sostenibilidad.

Involucrar a los equipos de desarrollo de software en estas medicaciones, la autora considera que tendrá un impacto directo en la: **preservación de los activos ambientales**, por cuanto las medidas e indicadores evaluarán cuán sostenible es el proceso de construcción del software y su posterior uso y en la **participación ciudadana**, pues hará más consciente a los desarrolladores de software en el impacto de las decisiones que toman cuando construyen una aplicación informática y los

hará partícipes del desarrollo de software que reducen el impacto medio ambiental del uso de las TIC en la informatización de los procesos y las organizaciones.

Reconocimientos.

Agradecemos a la Universidad de Castilla La Mancha y la Convocatoria de Ayudas para estancias de investigadores invitados para el año 2016 y 2017 cofinanciadas por Programa Operativo Feder de Castilla – La Mancha 2014-20, y en particular a la Dra. Coral Calero y al Grupo Alarcos por las facilidades ofrecidas para la obtención de la información.

REFERENCIAS BIBLIOGRÁFICAS:

1. Abdullah, A., Abdullah, S., Din, J., & Tee, M. (2015). A systematic literature review of Green software development in collaborative knowledge management. *International Journal of Advanced Computer Technology*, 3(1), 63-80.
2. Ahmad, R., Barahom, F., & Hussain, A. (2014). A systematic review on sustainability studies in software engineering. *Knowledge Management International Conference (KMICe)*. Obtenido de <http://www.kmice,csms.net.my>
3. Ahmed, A., Ahmad, S., Ehsam, N., Mirza, E., & Sarwar, S. (2010). Agile software development: impact on productivity and quality. *ICMIT2010 IEEE International Conference on Management of Innovation and Technology*, (págs. 287-291).
4. Akinli, S. (2013). Green software development and design for environmental sustainability. *IDOESE2013: 11th International Doctoral Symposium an Empirical Software Engineering*, 9.
5. Becker, C. e. (2015). The Karlskrona manifesto for sustainability design. Version 1.0. Obtenido de arxiv.org/pdf/140.6968.pdf
6. Bourdon, A., Noureddine, A., Rouvoy, R., & Seintuner, L. (2012). A preliminary study of the impact of software engineering on the Green IT. *GREEN'12/ICSE'12: 1st International Workshop on Green and Sustainable Software*, (págs. 21-27). Obtenido de we.wmn.fr/x-info/eco_conception_logicrels/upload/SLIDE/13-inriian_bourdon.pdf

7. Boyd, G., Dutrow, E., & Tunnessen, W. (2008). The evolution of the ENERGY STAR® energy performance indicator for benchmarking industrial plan manufacturing energy use. *Journal of Cleaner Production*, 16, 709-715.
8. Bozzelli, P., Gu, Q., & Lago, P. (2013). A systematic literature review a green software metrics. Technician report, VU University Amsterdam. Obtenido de <http://www.s2group.es.vu.nl/cup-content/uploads/2013/01/LS.PaoloBozzelli.GreenMetrics.pdf>
9. Brundtland G. H. et. al. (1987). Report on the World Commission on Environment and Development: Our Common Future. United Nations Conference on Environment and Development. Obtenido de <http://www.un.org/es/comun/docs/?symbol=A/42/427>
10. Calero, C., Bertoa, M., & Moraga, M. (2013). Sustainability and quality: icing on the cake. RE4SuSy@ RE Second International Workshop on Requirements Engineering for Sustainable Systems, 995. Obtenido de:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.415.6075&rep=rep1&type=pdf>
11. Calero, C., Moraga, M., & Bertoa, M. (2013). Toward a software product sustainability model. WSSPE2013: Working towards Sustainable for Science: Practice and Experiences. Obtenido de <http://arxiv.org/BS/1309.1640>
12. Chan, F., & Thong, J. (2009). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems*, 46, 803-814.
13. Chan, H., & Johansson, M. (2014). Green IS- a systematic literature review. (U. o. IT, Ed.) Obtenido de bada.hb.se/bitstream/2320/13797/1/2014KANI07.pdf
14. Costello, T. (2011). IT Teach and strategy trends. *IT Professional*, 13(1), 61-63.
15. Da Silva, S., & Di Felippo, A. (2016). The agility construct on project management theory. *International Journal of Project Management*, 34, 660-674.
16. Dos Santos, R., & Fantinato, M. (August de 2013). The use of software product lines for business process management: a systematic literature review. *Information and software technology*, 55(8), 1355-1377.

17. Esfahani, M., Rahma, A., & Zakania, N. (2015). The status quo and the prospect of Green IT and Green IS: a systematic literature review. *Journal of soft computing and decision support systems*, 2(1), 18-34.
18. Fink, A. (2010). *Conducting research literature review: from paper to internet*. (Third Edition ed.). Sage publications.
19. Garousi, V., Coskuncay, A., Demirörs, O., & Yacizi, A. (2016). Cross-factor lysis of software engeneering practce versus practitioner demographics: an exploratory study in Turkey. *The Journal of systems*, III, 49-73.
20. Genaidy, A., Sequeira, R., Tolaymat, T., Khohler, J., Wallace, S., & Rinder, M. (2010). Integrating science and business models of sustainability for environmentally-challenging industries such as secondary lead smelters: A systematic review and analysis of findings. *Journal of Environmental Management*, 92, 1872-1882.
21. Genero, M., Cruz, J. A., & Piattini, M. G. (2014). *Métodos de investigación en ingeniería de software*. España: RA-MA.
22. Google. (19 de September de 2011). Google Green y el costo energético de enviar un e-mail . Obtenido de <http://noticias.masverdedigital.com/google-green-y-el-costo.energetico-de-enviar-un-email>
23. Hedwing, M., Malkowski, S., & Neumann, D. (2009). Taming energy cost of large Enterprise systems through adaptive provisioning. *International Conference on Information Systems (ICIS 2009)*, Association for Information System. Obtenido de: <http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1065&context=icis2009>
24. Hernández, A. (2016). ¿Por dónde empezar para aplicar prácticas verde/sostenibles en el proceso de desarrollo del software y obtener productos verde/sostenibles? 18 Convención Científica de Ingeniería y Arquitectura, IX Simposio Universitario Iberoamericano sobre Medioambiente (SUIMA).

25. Hilty, L., & Lohmann, W. (2011). The five most neglected issues in Green IT. *Revista Novática*, XXXVII(213), 15-19.
26. Hinde, A. (2012). Green minig: investigations power consumption across version. En M. Glinz, G. Murphy, & M. Pezzé (Ed.), *ICSE'12 Proceedings of the 34th International Conference on Software Engineering* IEEE Press Piscataway, (págs. 1301-1304). NJ.
27. Hoesch-Klohe, K., Ghose, A., & Lam-Son, L. (2010). Towards Green Business Process Management. *IEEE International Conference on Services Computing*. Obtenido de: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5557256&queryText=.QT.Green%20BPM.QT.&newsearch=true>
28. Jakobi, T., Castelli, N., N., S., & Stevens, G. (2016). Chapter 15 Towards Collaboration Green Business Process Management as a conceptual framework. En J. Marx Gómez, U. Vogel, B. Rapp, M. Somescheing, A. Winter, & N. Giesen, *Select and extended contributions from the 28th International Conference on Informatic for Environmental Protection. Advances and new trend in enviromental and energy informatics* (págs. 275-296). Springer International Publishing Switzerland.
29. Karn, E., Dick, M., & Naumam, M. (2015). Impact of software and its engineering on the carbon footprint of ICT. *Environment impact assessment review*, 52, 53-61.
30. Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literatures reviews in software engineering. *EBSE Technical Report: EBSE-2007-01*, Keele University.
31. Kumar, G., & Kumar, P. (2014). Comparative analysis of software engineering models from traditional to modern methodologies. *ourth International Conference on Advanced Computig & Communication Technologies*, (págs. 186-196).
32. Lami, G., & Buglione, L. (2012). Measuring software sustainability from a process-centric perspective. *Join Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement*, (págs. 53-59).

33. Mfon-Abasi, R. (2014). Measuring sustainability impact of software. *International Journal of Computer Trends and Technology*, 6(1), 5-7.
34. Morioka, S., & Monteiro de Carvalho, M. (2016). A systematic review towards a conceptual framework for integrating sustainability performance into business. *Journal of Cleaner Production*, XXX, 1-13. Obtenido de <http://dx.doi.org/10.1016/j.jcleopro.2016.01.104>
35. Nuhoff-Isakhanyan, G., & Wubben, E. (2016). Sustainability benefits and challenges of inter-organizational collaboration in Bio-based business: a systematic literature review. *Sustainability*, 8(4).
36. Optiz, N., Krup, H., & Kolbe, L. (2014). Green Business Process Management- A definition and research framework. 47th Hawaii International Conference on System Science, IEEE Computer Society, (págs. 3808-3817).
37. Otegi, J., Aguiar, Aguilar, M., Cruz, C., & Fuentes, L. (2015). Towards sustainable Project management. A literature review". 19th International Congress on Project Management and Engineering, (págs. 43-56).
38. Ou, J., Ley, J., & Chen, C. (2011). Initial development of a metrics framework for green competitiveness. First International Technology Management Conference, (págs. 706-710).
39. Papadopoulos, G. (2015). Moving from traditional to agile software development methodologies also large, distributed projects. *Procedia-Social and Behavioral Science*, 175, 455-463.
40. Parra, D., & Hernández González, A. (2017). Diseño de un caso de estudio para evaluar framework de desarrollo en cuanto a la sostenibilidad. Trabajo de diploma, La Habana.
41. Pelozo, J., & Yachnin, R. (2008). Valving business sustainability: a systematic review. Research network for business sustainability. Obtenido de <http://nbs.net/uploads/content/uploads/NBS-Systematic-Review-Valving.pdf>

42. Pezenstandler, B., Calero, C., & Franch, X. (2013). Sustainability in software engineering; systematic literature review. 2nd International Workshop on Green and Sustainable Software, IEEE Press Piscataway USA, (págs. 46-53).
43. Rashid, N., & Khan, S. (2015). Green agility for global software development vendor: a systematic literature review protocol. *Pakistan Academy of Science*, 52(4), 301-313.
44. Riehle, D. (2000). Framework design: A role modeling approach. Ph. D Thesis #13509, Zurich, Switzerland, ETH Zurich.
45. Seacord, R., Elm, J., Goethert, W., Lewis, G., Plaskosh, D., Robert, J., & Lindvall, M. (2003). Measuring software sustainability. *ICSMs'03: 19th IEEE International Conference on Software Maintenance*, (págs. 450-461).
46. Tuskeentushi, B., Sedera, D., & Recker, J. (2013). Green IT segment analysis an academic literature review. *Twentieth Americas Conference on Information Systems (AMCIS 2014)*, Association for Information Systems, Savannah, Georgia, USA, (págs. 1-15).
47. Unkehar, B. (2011). The new five years. *IT Professional*, 13(2), 56-69.
48. Watson, R., Boudreau, M., & Chen, A. (2010). Information systems and environmentally sustainable development: energy informatics and new directions for the IS community. *MIS quarterly*, 34(1), 23-38.
49. Zhang, H., & Ali Babar, M. (2013). Systematic reviews in software engineering: an empirical investigation. *Information and Software Technology*, 55(7), 1341-1354.
50. Zhang, S. (12 de October de 2012). Writing energy-efficient windows. Store applications for mobile device-impact of CPU utilization on plataform power. Obtenido de: <https://software.intel.com/en-us/articles/writing-energy-efficient-windows-store-applications-for-mobile-device>

DATOS DE LA AUTORA:

1. Anaisa Hernández González. Ingeniera en Sistemas Automatizados de Dirección, Máster en Informática Aplicada a la Ingeniería y la Arquitectura, y Doctora en Ciencias Técnicas. Profesora Titular de la Facultad de Ingeniería Informática en la Universidad Tecnológica de La Habana “José Antonio Echeverría”, CUJAE, La Habana, Cuba. Correo electrónico: anaisa@ceis.cujae.edu.cu

RECIBIDO: 8 de septiembre del 2017.

APROBADO: 26 de septiembre del 2017.