



Universidad de Cienfuegos
Facultad de Ingeniería
Carrera Ingeniería Informática

Propuesta de un Modelo de Procesos como base
para la formación de informáticos del territorio hacia
un desarrollo de software sostenible.

Trabajo de diploma para optar por el título de Ingeniero Informático

Autor:

Deisy González Bilbao

Tutores:

Ms.C Dailyn Sosa López

Ms.C Cinthya Rodríguez Hernández

Consultante:

Dr.C Raidell Avello Martínez

Cienfuegos, Cuba

Curso 2017 - 2018

Declaración de autoría

Declaro que soy el único autor (Declaramos que somos los únicos autores) de este trabajo y autorizo (autorizamos) al <Nombre del Departamento> de (de la, del) <Nombre de la Entidad> y al Departamento de Informática de la Facultad de Informática en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que hagan el uso que estimen pertinente con el trabajo de diploma.

Para que así conste firmo (firmamos) la presente a los ____ días del mes de ____ del ____.

(Si procede)

Nombre completo del primer autor

Nombre completo del segundo autor

(Si procede)

Nombre completo del primer tutor

Nombre completo del segundo tutor

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática señalada.

Firma Tutor Firma Tutor

Firma ICT

Firma Vicedecano

Opinión del usuario

El Trabajo de Diploma, titulado <Título del Trabajo de Diploma>, fue realizado en nuestra entidad <Nombre de la Entidad que utilizará el sistema>. Se considera que, en correspondencia con los objetivos trazados, el trabajo realizado nos satisface:

- ☐ Totalmente
- ☐ Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma le reportan a nuestra entidad los beneficios siguientes (cuantificar):

Como resultado de la implantación de este trabajo se reporta un efecto económico que asciende a <valor> MN y/o <valor> CUC. (Este valor debe ser REAL, no indica lo que se reportará, sino lo que reporta a la entidad. Puede desglosarse por conceptos, tales como: cuanto cuesta un software análogo en el mercado internacional, valor de los materiales que se ahorran por la existencia del software, valor anual del (de los) salario(s) equivalente al tiempo que se ahorra por la existencia del software).

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____.

Nombre del representante de la entidad	Cargo
--	-------

Firma	Cuño
-------	------

Opinión del tutor

Título: <Título del trabajo de diploma>

Autor(es): <Nombres y apellidos del autor o los autores>

El(Los) tutor(es) del presente Trabajo de Diploma considera(mos) que durante su ejecución el(los) estudiante(s) mostró(aron) las cualidades que a continuación se detallan.

<El tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes: Independencia, Originalidad, Creatividad, Laboriosidad y Responsabilidad>

<Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios)>.

Por todo lo anteriormente expresado considero que el estudiante está (no) apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de <2 – Desaprobado, 3 – Aprobado, 4 – Bien, 5 – Excelente>.

<Si considera que los resultados poseen valor para ser publicados, debe expresarlo también>

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año ____.

(Si procede)

Nombre completo del primer tutor

<Grado científico, Categoría docente
y/o investigativa>

Nombre completo del segundo tutor

<Grado científico, Categoría docente
y/o investigativa>

Fecha: _____

Agradecimientos

A mi mama, a mi hermano, y a Luisi.

A Jorge Adrián.

A Daniela.

A Jorgito.

A Annia.

A Chang.

A Osmay.

A Angarica.

A Yoel.

A Claudia, Sandor, Tatiana, Dayron, William, José Luis.

A mi amiga Cinthya.

A mis tutores.

A los profes que me ayudaron en todos los años de carrera.

Dedicatoria

A mi mamá por ser mi apoyo incondicional, a mi hermano por ser mi razón de ser, a Gato por ser mi alegría, a Luisi por ser el amor de mi vida y a todos los que creyeron que este momento nunca llegaría, pues toma ahí.

Resumen

El software sostenible es aquel cuyo desarrollo, despliegue y uso tiene un impacto mínimo, directo o indirecto, negativo o incluso positivo en la economía, sociedad, seres humanos y medio ambiente. Existen varios modelos de procesos planteados por diferentes autores de distintas partes del mundo, los cuales plantean varias vías para realizar una ingeniería de software sostenible, sin embargo, no se encuentra ningún modelo de procesos que se adecue a las características de la empresa cubana actual. Esta investigación tiene como objetivo: elaborar un modelo de proceso que tribute a la formación de los informáticos del territorio hacia un desarrollo de software sostenible en el contexto de la empresa cubana actual. Se realiza una caracterización del estado actual del desarrollo de software sostenible en empresas del territorio cienfueguero, que arroja como resultado que los informáticos no tienen conocimientos suficientes sobre la sostenibilidad del software, ni este aspecto representa un requisito en las empresas. Como resultado se ha elaborado una propuesta de Modelo de Procesos como base para la formación de informáticos del territorio hacia un desarrollo de software sostenible. Se definen fases, roles e indicadores para su posterior implementación y un sistema de conocimientos como base para la formación de los informáticos del territorio. La propuesta fue validada por criterio de expertos, que la calificaron de Muy Adecuada en la mayoría de los indicadores evaluados.

Palabras clave: software sostenible, modelo de procesos, indicadores, empresa cubana.

Summary

Sustainable software is one whose development, deployment and use has a minimal, direct or indirect, negative or even positive impact on the economy, society, human beings and the environment. There are several process models proposed by different authors from different parts of the world, which propose several ways to carry out a sustainable software engineering, however, there is no process model that fits the characteristics of the Cuban company. The general objective of the research: to elaborate a process model that pays tribute to the training of computer scientists in the territory towards a sustainable software development in the context of the current Cuban company. A characterization of the current state of the development of sustainable software in companies of the Cienfuegos territory is made, which shows that computer scientists do not have knowledge about the sustainability of the software nor does this represent a requirement in companies. Based on the Process Model for Agile and Sustainable Software Engineering proposed in 2015, a proposal is made that responds to the needs of the current Cuban company. Phases, roles and indicators are defined for their subsequent implementation and a knowledge system as a basis for the training of computer scientists in the territory. The proposal was validated by experts, who rated it as Very Adequate in most of the indicators evaluated.

Key words: sustainable software, process model, indicators, Cuban enterprise.

Índice

Introducción.....	1
1 – Fundamentos teóricos del desarrollo de software sostenible	8
1.1 – Introducción	8
1.2 – Principales conceptos.....	8
1.2.1 Concepto de sostenibilidad	8
1.2.2 Concepto de desarrollo sostenible	11
1.2.3 Concepto de software sostenible.....	12
1.2.4 Concepto de ingeniería de software sostenible	13
1.3 Estudios existentes sobre modelos de procesos de desarrollo de software sostenible.....	14
1.4 – Conclusiones.....	26
2 – Análisis y comparación de modelos de procesos de software sostenible.....	27
2.1 – Introducción	27
2.2- GREENSOFT Model	27
2.2.1- Ciclo de Vida del Producto de Software	28
2.2.2- Criterios y métricas de sostenibilidad.....	29
2.2.3- Modelo de Procedimiento.....	30
2.2.4- Recomendaciones y Herramientas.....	33
2.3- Modelo de Calidad para Software Sostenible de Naumann	34
2.4- Descripción de los modelos de procesos	38
2.4.1- Modelo Verde para Ingeniería de Software Sostenible	38
2.4.2- Proceso central de sostenibilidad de software	39
2.4.3- Modelo de Proceso para una Ingeniería de Software Ágil y Sostenible.....	41
2.4.4- GreenRM: Modelo de Referencia para el Desarrollo de Software Sostenible	43
2.5 – Comparación entre los modelos de procesos.....	45
2.6 – Conclusiones.....	47
3 – Propuesta de modelo de procesos de ingeniería de software sostenible para la empresa cubana	48
3.1- Introducción.....	48
3.2- Caracterización del uso del software sostenible en empresas del territorio.	48
3.3- Resultados de la encuesta aplicada.....	49
3.4- Modelo de Procesos.....	60
3.4.1- Modelo de Procesos adaptado a la Empresa Cubana Actual.....	60
3.5- Sistema de Conocimiento de Referencias	66
3.6 Validación del modelo propuesto	69
3.6.1 Método Delphi	69
3.6.2 Caracterización de los expertos	70
3.6.3 Resultados de los criterios dado por los expertos	71
3.7– Conclusiones	72
Conclusiones	73
Recomendaciones.....	74
Referencias bibliográficas	75
Anexos.....	79

Índice de tablas

Tabla 1: Propuesta para considerar la sostenibilidad en un proyecto de ingeniería/TFG.....	17
Tabla 2: Lista de factores de riesgo identificados.....	18
Tabla 3: Lista de riesgos identificados por continentes.....	19
Tabla 4: Respuesta Pregunta 4.....	49
Tabla 5: Respuesta Pregunta 5.....	50
Tabla 6: Respuesta Pregunta 6.....	51
Tabla 7: Respuesta Pregunta 7.....	51
Tabla 8: Respuesta Pregunta 8.....	52
Tabla 9: Respuesta Pregunta 9.....	53
Tabla 10: Respuesta Pregunta 10.....	53
Tabla 11: Respuesta Pregunta 12.....	54
Tabla 12: Respuesta Pregunta 13.....	54
Tabla 13: Respuesta Pregunta 14.....	55
Tabla 14: Respuesta Pregunta 15.....	55
Tabla 15: Respuesta Pregunta 16.....	56
Tabla 16: Respuesta Pregunta 17.....	57
Tabla 17: Respuesta Pregunta 18.....	57
Tabla 18: Respuesta Pregunta 19.....	58
Tabla 19: Respuesta Pregunta 20.....	58
Tabla 20: Respuesta Pregunta 21.....	59
Tabla 21: Respuesta Pregunta 21A.....	59

Índice de figuras

Figura 1 Resultados de las mediciones de los procesadores de texto WP1 y WP2	23
Figura 2: Resultados de las mediciones de CMS1 y CMS2.....	25
Figura 3 Modelo GREENSOFT. [34]	28
Figura 4: Ejemplo de un proceso de desarrollo de software que encaja en la categoría Desarrollo.....	31
Figura 5: Modelo de Calidad para Software Sostenible	35
Figura 6: Histograma Respuesta Pregunta 4	50
Figura 7: Histograma Respuesta Pregunta 5	50
Figura 8: Histograma Respuesta Pregunta 6	51
Figura 9: Histograma Respuesta Pregunta 7	52
Figura 10: Histograma Respuesta Pregunta 8	52
Figura 11: Histograma Respuesta Pregunta 9	53
Figura 12: Histograma Respuesta Pregunta 12	54
Figura 13: Histograma Respuesta Pregunta 13	54
Figura 14: Histograma Respuesta Pregunta 14	55
Figura 15: Histograma Respuesta Pregunta 15	56
Figura 16: Histograma Respuesta Pregunta 16	56
Figura 17: Histograma Respuesta Pregunta 17	57
Figura 18: Histograma Respuesta Pregunta 18	57
Figura 19: Histograma Respuesta Pregunta 19	58
Figura 20: Histograma Respuesta Pregunta 20	58
Figura 21: Histograma Respuesta Pregunta 21	59
Figura 22: Histograma Respuesta Pregunta 21A	60

Introducción

La Sostenibilidad aparece como "la idea central unificadora más necesaria en este momento de la historia de la humanidad" [1]. Se trata de un concepto que pretende movilizar la responsabilidad colectiva para hacer frente al conjunto de graves problemas y desafíos a los que se enfrenta la humanidad, apostando por la cooperación y la defensa del interés general. Tiene múltiples dimensiones (social, medioambiental, económica, técnica y humana), trasciende a múltiples disciplinas y requiere de la acción de múltiples niveles unida a un pensamiento a largo plazo.

La sostenibilidad es entendida como un replanteamiento de las relaciones de los grupos humanos entre sí y con el medio ambiente[2], una alternativa al concepto de desarrollo habitual o social, que pretende una homogeneidad y coherencia entre el crecimiento económico de la población en todos sus estratos, los recursos naturales y la sociedad, evitando comprometer la posibilidad de vida en el planeta, ni la calidad de vida de la especie humana tanto ahora, en los años venideros como en las generaciones futuras.

El informe de la Comisión Mundial del Medio Ambiente y del Desarrollo, conocido como Informe Brundtland [1], es un intento inicial de introducir el concepto de Sostenibilidad o Sustentabilidad: "El Desarrollo Sostenible es el desarrollo que satisface las necesidades de la generación presente sin comprometer la capacidad de las generaciones futuras para satisfacer sus propias necesidades".

La idea de un desarrollo sostenible es, como señala Novo en 2006 "situarse en otra óptica; contemplar las relaciones de la humanidad con la naturaleza desde enfoques distintos" [3]. Se trata de un concepto que parte de la suposición de que puede haber desarrollo, mejora cualitativa o despliegue de potencialidades, sin crecimiento, es decir, sin incremento cuantitativo de la escala física, sin incorporación de mayor cantidad de energía ni de materiales.

La filosofía de desarrollo de tecnología sostenible se ha convertido en un aspecto fundamental en las empresas, gracias al uso eficiente de los recursos tecnológicos con el objetivo de reducir el impacto ambiental. Las tecnologías de la información y las comunicaciones (TIC), y por tanto el software, deben incluirse en la evaluación del impacto

medioambiental de una empresa. El desarrollo de software sostenible permite afrontar los retos energéticos actuales, reduciendo la demanda de energía, servidores, almacenamiento y transferencia de datos en las soluciones geoespaciales y consiguiendo herramientas tecnológicas sostenibles con el medio ambiente.

El software sostenible es aquel cuyo desarrollo, despliegue y uso tiene un impacto mínimo, directo o indirecto, negativo o incluso positivo en la economía, sociedad, seres humanos y medio ambiente. Un pre-requisito para el software sostenible es un proceso de desarrollo sostenible, el cual considera el medio ambiente y otros impactos durante el ciclo de vida de este [4].

En base a esto puede definirse la Ingeniería del Software Sostenible como el arte de desarrollar software sostenible a través de procesos de ingeniería del software sostenible. Siendo así que el producto puede ser definido y desarrollado de una forma que el impacto positivo y negativo sea un resultado esperado o considerado durante el ciclo de desarrollo de software, siendo calificado, documentado, y aprovechado para la futura optimización del mismo[4].

El software sostenible debe cumplir 3 condiciones: [4]

1. Debe ser producido de forma que cumpla los objetivos de sostenibilidad.
2. Durante su uso debe tener un mínimo impacto negativo social y ambiental.
3. Sus funcionalidades deben reforzar el desarrollo sostenible o al menos no tener impactos negativos en la sociedad o el medio ambiente.

Se considera que los aspectos de la sostenibilidad en la ingeniería de software son: el proceso de desarrollo de software con el uso responsable de recursos ecológicos, humanos y financieros, el proceso de mantenimiento del software hasta que se reemplace por uno nuevo (abarcado la sostenibilidad durante la producción del software y su uso), la producción del sistema (sostenibilidad del sistema y del producto con respecto al uso de los recursos para la producción) y el uso del sistema (el impacto y su responsabilidad con el ambiente y el uso de procesos de negocio verdes). [5]

En su publicación Sustainable Software Engineering: Process and Quality Models, Life Cycle and Social Aspects [4], Stephen Naumann propone un modelo para el software sostenible y

su ingeniería, con un enfoque para definir las características de sostenibilidad para productos de software a lo largo de su ciclo de vida.

Existen varios modelos de procesos planteados por diferentes autores de distintas partes del mundo, los cuales plantean varias vías para realizar una ingeniería de software sostenible, o incluso, mantener la sostenibilidad del software hasta el fin de su ciclo de vida. Los principales autores internacionales que tratan el tema son:

Lami en 2012 [6] define un grupo de procesos de alto nivel que añaden el concepto de sostenibilidad a los procesos de referencia de software definidos en ISO/IEC 12207. El proceso de referencia estándar define todas las actividades directa e indirectamente relacionadas con el desarrollo, mantenimiento y operación del software.

Mahmoud y Ahmad [7] presentan un modelo de dos niveles. En un primer nivel un proceso de ingeniería de software sostenible y en un segundo nivel, una colección de herramientas de software que promueven la computación sostenible.

Bozelli [8] describe y clasifica las métricas de consumo de energía del software y analiza la evolución de esas métricas en términos de método, contexto y evaluación. Rashid y Khan en 2015 [9] identifican 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.

Naumann [4] proporciona 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. De las 4 dimensiones de la sostenibilidad (social, económica, ambiental y humana) concluye que la menos abordada es la humana.

Otegui, Aguilar, Cruz y Fuentes en 2015 [10] identifican las tendencias en la introducción de la sostenibilidad en la gestión de proyectos y concluyen que evolucionar la gestión de proyectos tradicional a la gestión de proyectos sostenible implica transformar la línea base a

un triple objetivo (medir el rendimiento económico, social y ambiental) y que se necesita incorporar a cada proceso y área de conocimiento las tres dimensiones de la sostenibilidad.

En Cuba, en la Facultad de Informática Matemática de la Universidad de Holguín se realizó una investigación en el año 2012 [11], con el objetivo de elaborar un procedimiento para la valoración de sostenibilidad de un producto informático soportado en una aplicación web a través de la docencia en la formación profesional del ingeniero. Concluyen que la valoración de sostenibilidad de un producto informático favorece prevenir y mitigar riesgos de impacto de tipo económico-administrativo, ambiental, socio humanista y tecnológico del producto informático y potenciará que se mantengan en el tiempo proporcionando que las empresas asuman la sostenibilidad como un factor de calidad de su producto.

Anaisa Hernández González [12], presenta algunas de las revisiones sistemáticas a la literatura sobre las prácticas verde/sostenibles en el proceso de desarrollo del software, lo que permite a las personas que se inician en la temática poseer un análisis de varios de los estudios secundarios que se han realizado en los últimos años.

Situación problemática

El desarrollo sostenible es un tema emergente en Cuba, además de formar parte de los objetivos de la agenda de 2030, convirtiéndose por tanto en prioridad para el país. El desarrollo de software sostenible hace referencia 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. El termino sostenibilidad puede ser abordado desde varios puntos de vista, en esta investigación desde el punto de vista del tratamiento de las TIC. En la empresa cubana actual uno de los profesionales de mayor impacto para el desarrollo y competitividad de la empresa es el Ingeniero Informático, debido al uso e implementación de TIC, las cuales le permiten la automatización de los diversos procesos, para obtener de manera oportuna el recurso más importante que es la información. Como parte de un estudio para hacer una caracterización de las empresas del territorio teniendo en cuenta los términos sostenibilidad, desarrollo sostenible y software sostenible se aplica una encuesta como instrumento informal donde se constató que el 90% de los desarrolladores y usuarios de software no conocen en profundidad estos términos, y por tanto no lo aplican. Además, en las empresas no están

creados los mecanismos o procesos que aborden estos temas, y tampoco la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, como unidad docente de superación de los graduados de informática y carreras afines, que ejercen como desarrolladores, les brinda los modelos de procesos necesarios como base para la formación de estos hacia un desarrollo de software sostenible.

Teniendo en cuenta lo planteado anteriormente se identifica como **problema a resolver**: ¿Cómo contribuir a la formación de informáticos del territorio en el desarrollo de software sostenible?

Se presenta como **objeto de estudio**: el desarrollo software sostenible y como **campo de acción**: modelos de procesos para el desarrollo del software sostenible.

Se plantea como **idea a defender** que: la propuesta de un modelo de procesos adaptado a la empresa cubana, contribuirá a la formación de los informáticos del territorio en el desarrollo de software sostenible.

Se define como **objetivo general** de la investigación: elaborar un modelo de proceso que tribute a la formación de los informáticos del territorio hacia un desarrollo de software sostenible en el contexto de la empresa cubana actual.

Partiendo del objetivo general se plantean los siguientes **objetivos específicos**:

- Analizar la bibliografía sobre el desarrollo de software sostenible en Cuba y el mundo.
- Caracterizar el estado actual de desarrollo de software sostenible en empresas del territorio.
- Proponer un modelo de procesos para el desarrollo de software sostenible adaptado al contexto de la empresa cubana actual.
- Elaborar un sistema de conocimientos de referencia para la formación de los informáticos del territorio.
- Validar por criterio de expertos.

La **justificación de esta investigación** está dada por la necesidad de formar los informáticos de las empresas del territorio hacia el desarrollo del software sostenible en

consonancia con el desarrollo mundial de la tecnología y dar respuesta a las exigencias del desarrollo sostenible.

Como métodos de investigación se emplearon fundamentalmente los siguientes:

De los Métodos Teóricos se utilizaron:

Método Histórico Lógico: Se utiliza con el objetivo de profundizar en los antecedentes de las teorías correspondientes al desarrollo del software sostenible, considerando los elementos más significativos de este proceso y revelando las características y tendencias.

Método Análisis- Síntesis: Se utilizará para captar y resumir varios documentos y procedimientos por los cuales se rige el desarrollo del software sostenible, de ellos se extraerán las ideas fundamentales y al mismo tiempo se detallará la información necesaria para la propuesta del modelo adaptado al contexto cubano.

Métodos Empíricos:

Encuesta: A los usuarios de las empresas para conocer el uso actual del software en la empresa cubana, su desarrollo e implementación.

Criterio de expertos: permite obtener valoraciones respecto a la validez del modelo propuesto, además de sugerencias para su mejora.

La tesis está estructurada, en Resumen, Introducción, 3 capítulos, Conclusiones, Recomendaciones, Bibliografía y Anexos.

Capítulo 1: Fundamentos teóricos del desarrollo de software sostenible.

Se hace un análisis de la literatura científica en Cuba y el mundo sobre los principales conceptos asociados al desarrollo sostenible, profundizando en el desarrollo del software sostenible. Una revisión de los principales teóricos del tema, tendencias y metodologías actuales para la evaluación del software sostenible.

Capítulo 2: Análisis y comparación de modelos de procesos de software sostenible.

Se realiza un análisis de modelos de procesos planteados por diferentes autores. Se comparan los modelos considerando diferentes criterios como roles, actividades, capacitación, entre otros, para la selección del más adecuado para el contexto cubano.

Capítulo 3: Propuesta de modelo de procesos como base para la formación de informáticos del territorio hacia un desarrollo de software sostenible.

Se realiza un estudio a través de encuestas a los principales usuarios y desarrolladores de sistemas informáticos sobre el desarrollo y uso del software en la empresa cubana, características, impacto y necesidades del mismo. Se propone un modelo de procesos adaptado a las necesidades de las empresas del territorio y un sistema de conocimientos de referencia como base para la formación de los informáticos del territorio hacia un desarrollo de software sostenible.

1 – Fundamentos teóricos del desarrollo de software sostenible

1.1 – Introducción

En el siguiente capítulo se hace un análisis de la literatura científica en Cuba y el mundo sobre los principales conceptos asociados al desarrollo sostenible, profundizando en el desarrollo de software sostenible. También se realiza una revisión de los principales teóricos del tema y tendencias, metodologías y tecnologías actuales para la evaluación de software sostenible.

1.2 – Principales conceptos

1.2.1 Concepto de sostenibilidad

El adjetivo sostenible se refiere a algo que está en condiciones de conservarse o reproducirse por sus propias características, sin necesidad de intervención o apoyo externo. El término puede aplicarse sobre diversas cuestiones: métodos productivos, procesos económicos, etc. [13]

El origen del concepto sostenibilidad se sitúa a principios de la década de los años 80, a partir de perspectivas científicas sobre la relación entre el medioambiente y la sociedad y la publicación de varios documentos relevantes, principalmente la Estrategia Mundial para la Conservación (World Conservation Strategy, UICN, 1980, Primera estrategia global de Desarrollo Sostenible) y el conocido como Informe Brundtland (Nuestro futuro común, CMMAD, 1988) [1].

“Un futuro amenazado” es el título del primer capítulo del Informe Brundtland, a la que se debe un primer intento de introducir el concepto de Sostenibilidad o Sustentabilidad: "El Desarrollo Sostenible es el desarrollo que satisface las necesidades de la generación presente sin comprometer la capacidad de las generaciones futuras para satisfacer sus propias necesidades".

Habla de una etapa geológica nueva, el antropoceno, término propuesto por el premio Nobel Paul Crutzen, para destacar la responsabilidad de la especie humana en los profundos cambios que está sufriendo el planeta [14], vinculados a lo que Folke [15] califica como “la Gran Aceleración de la actividad humana”, especialmente a partir de la década de 1950, que amenaza con sobrepasar los límites del planeta.

Surge como resultado de los análisis de la situación del mundo, que puede describirse como una “emergencia planetaria” [2] y de larga duración como una situación insostenible, fruto de las actividades humanas, que amenaza gravemente el futuro (y ya el presente) de la humanidad. Según Bybee [2] se trata de “la idea central unificadora más necesaria en este momento de la historia de la humanidad”.

A partir de la publicación del informe Brundtland se alcanza un consenso sobre el desarrollo de una consciencia sobre los límites del planeta, las injusticias sociales que se cometen, y la importancia de trabajar de una manera sostenible. Sin embargo el concepto sostenibilidad ha recibido muchas críticas, la mayoría de ellas relacionadas con la vaguedad del término, al ser aparentemente confuso y estar lleno de contradicciones, con la falta de operatividad, con la indefinición de lo que se debe o no sostener, o con las lista de necesidades que se han de satisfacer. [16]

Norton [17] plantea dos tipos de sostenibilidad: sostenibilidad débil y sostenibilidad fuerte. La primera la ubica dentro del paradigma de la economía estándar, mientras que la segunda esta vincula a la termodinámica y a la ecología.

Sostenibilidad débil: La sostenibilidad débil es un concepto genérico que puede definirse como «la viabilidad de un sistema socioeconómico en el tiempo». Esta viabilidad se consigue manteniendo el capital global (las capacidades en términos del informe Brundtland), generación tras generación, siendo este capital global el resultado de otros dos: el capital natural y el capital de formación humana. El capital de formación humana hace referencia a la disponibilidad de capital monetario, tecnología, personal formado, etc.

Sostenibilidad fuerte: Es definida como la viabilidad de la relación que mantiene un sistema socioeconómico con un ecosistema [18]. En esta definición, el énfasis se pone en la

interacción entre estos dos sistemas dinámicos, teniendo en cuenta que el sistema socioeconómico es dependiente del ecosistema en el sentido de que éste podría funcionar autónomamente. La interacción consiste en una permanente co-adaptación.

Mientras las sociedades se abastecen de recursos y expulsan sus desechos, los ecosistemas sufren cambios y se reajustan; a menudo tales cambios provocan modificaciones tecnológicas, económicas y sociales. El atributo sostenible es propio de esta interacción establecida entre ambos sistemas pero no lo es de cualquiera de los dos por separado [19].

En 2015 se publica un manifiesto que recoge un conjunto de principios y compromisos vinculados con el diseño sostenible que plantea que: [20]

- la sostenibilidad tiene múltiples dimensiones (social, medioambiental, económica, técnica y humana) y que todas deben incluirse 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.

1.2.2 Concepto de desarrollo sostenible

El desarrollo sostenible es un término asociado a la posibilidad de lograr que una región crezca a partir de la explotación de sus recursos, sin que dicha explotación ponga en riesgo la existencia futura de los recursos en cuestión. El desarrollo sostenible también contempla que el crecimiento se consiga sin injerencia del exterior [16]. Si un país basa su desarrollo en la sobreexplotación de sus recursos naturales no renovables: tarde o temprano, el crecimiento será interrumpido. Por eso el desarrollo sostenible es aquel que se logra sin perjudicar al medio ambiente y sin poner en riesgo las posibilidades de desarrollo de las generaciones futuras.

El modelo de desarrollo sostenible alude al desarrollo socioeconómico, y su definición se plantea por primera vez en el Informe Brundtland. Es un concepto desarrollado como alternativa al concepto de desarrollo habitual o social, que pretende una homogeneidad y coherencia entre el crecimiento económico de la población en todos sus estratos, los recursos naturales y la sociedad.

Afirmaba que el desarrollo sostenible podría lograrse con un marco normativo integrado que abarcara esos tres pilares. Sin embargo, el desarrollo sostenible se mantiene como un concepto de aceptación generalizada, pero todavía no como una realidad cotidiana y práctica. Según la ONU ha faltado voluntad política e incorporar este concepto en el debate nacional e internacional dominante sobre las políticas económicas. La mayoría de los encargados de adoptar decisiones económicas siguen pensando que el desarrollo sostenible es algo ajeno a sus responsabilidades, mas relacionadas con la gestión macroeconómica y la política económica.

El concepto general de desarrollo sostenible comienza en los años 70, incluso con otras acepciones como eco-desarrollo, desarrollo sustentable, crecimiento sostenible, entre otros, consiste en un modelo de crecimiento, que afecta al sistema económico, social y ecológico y que está vinculado a la economía, el consumo, la obsolescencia programada, la edificación, el crecimiento de las ciudades, la educación, la política, el uso de combustibles fósiles como el petróleo, el aprovechamiento de las fuentes de energías renovables como la eólica, la

fotovoltaica o solar térmica, las emisiones de CO₂ a la atmósfera, el calentamiento global, la generación de residuos, etc. [21]

Este concepto supone el poder mantener niveles de crecimiento y desarrollo que beneficien a las diferentes regiones del globo pero que no impliquen continuar con ese abuso medioambiental sino que, por el contrario, se basen en el uso de energías renovables, en el reciclado de materiales que puedan ser reutilizables, etc.

El ámbito del desarrollo sustentable puede dividirse conceptualmente en tres partes: ecológico, económico y social. Se considera el aspecto social por la relación entre el bienestar social con el medio ambiente y la bonanza económica. El triple resultado es un conjunto de indicadores de desempeño de una organización en las tres áreas. [22]

1.2.3 Concepto de software sostenible

El incremento de las demandas de energía y el impacto ambiental negativo debido al uso creciente de los servicios TIC ha provocado que sean incluidas en el análisis del desarrollo sostenible. El uso de TIC sostenibles es aplicable a un rango de dominios de alta tecnología, incluyendo a los datacenters, computación móvil, y sistemas embebidos. Las emisiones globales de dióxido de carbono anuales han alcanzado recientemente los 9.100 millones de toneladas, donde al menos el 2% de las mismas pueden ser atribuidas a los sistemas de TIC. [23]

La mayoría de los estudios y controles regulatorios se enfocan en las mediciones, análisis y control del consumo de energía asociado con el hardware, sin considerar el software. Aunque los sistemas de software no consumen energía directamente, afectan la utilización del hardware, lo cual conduce a un consumo de energía indirecto. El dominio de investigación en ingeniería de software ha comenzado a prestar atención a la sustentabilidad, tal como lo demuestra el creciente número de publicaciones y estudios empíricos. [4] [5] [11]

De la misma forma que las TIC sostenible, el software sostenible busca reducir el impacto ambiental del software en sí mismo. El software sostenible es una característica emergente de calidad de software, las empresas de software han comenzado a confrontar el conflicto

entre ser todo lo amigables posible con el ambiente contra las presiones de los clientes por nuevos requerimientos funcionales y alta calidad. De esta forma los sistemas de software también pueden jugar un rol proactivo en el ahorro de energía al proporcionar realimentación sobre las formas en que los recursos consumen energía.

El término de software sostenible según Abdullah [24] y Naumam [25] 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.

Algunos autores argumentan que el software libre es equivalente a software sostenible, planteando que es el futuro del desarrollo de aplicaciones, teniendo como base el principal argumento que el software libre otorga permisos para analizar, modificar, actualizar y distribuir el código fuente de una aplicación, permitiendo su reutilización por otros programadores. [16]

Sin embargo Mendez (2015) plantea que el software sostenible debe ser capaz además de adaptarse a las condiciones cambiantes del sistema de información al cual pretende automatizar. Es decir aquel programa portable, que permite al usuario cambiar de hardware, sistema operativo o que permite migrar los datos a otro servidor web al que se desee cambiar. Un software sostenible debe ser flexible a los cambios venideros, debe permitir aplicar las actualizaciones necesarias de manera fácil y sin exponer el funcionamiento global del sistema. [26]

1.2.4 Concepto de ingeniería de software sostenible

En 2014 Ahmad, Barahom y Hussain [27] definen la ingeniería de software sostenible como el 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.

En 2015 Rashid y Khan plantean el concepto de ingeniería de software sostenible como un paradigma emergente que tiene como objetivo desarrollar software con características sostenibles para reducir el impacto negativo en el medio ambiente. Por su parte, Naumann en ese mismo año lo plantea como el arte de definir y desarrollar productos de software de forma que los impactos positivos y negativos resulten los esperados por el ciclo de vida y sean continuamente evaluados, documentados y optimizados. [5]

1.3 Estudios existentes sobre modelos de procesos de desarrollo de software sostenible

1. Durante los años 2004 a 2006 se llevó a cabo una investigación en la carrera de Ingeniería Informática de la Universidad de Holguín Oscar Lucero Moya (UHOLM) que abordaba la sostenibilidad en la formación de ingenieros informáticos. Como resultado de la misma se diseñó un modelo de educación para la sostenibilidad que relaciona la interacción dialéctica entre problemas de informatización empresarial y social en contraposición a la gestión de proyectos informáticos sostenibles. La investigación propone un procedimiento para valorar la sostenibilidad de un producto informático. [11]

El procedimiento valora la sostenibilidad desde la etapa de diseño del proyecto, evaluando factores de tipo económico, ambiental, social y tecnológico del producto informático que favorece su autorregulación, para la satisfacción de la necesidad que resuelve, con un uso racional de recursos y la toma de decisiones adecuadas a las condiciones del contexto. En este trabajo se mantiene que la educación ambiental universitaria se sustente en criterios del desarrollo sostenible integrándolo al modo de actuación profesional. [28]

Se establecen las dimensiones administrativa, socio humanista, ambiental y tecnológica, lo que permite hacer una valoración de impactos para un proyecto sostenible. Las dimensiones socio humanista, administrativo, ambiental, que coinciden con los tres objetivos del desarrollo sostenible: social, ambiental, económico, y se añaden la dimensión tecnológica, que se refiere al conjunto de conocimientos, habilidades, actitudes, valores, para producir, distribuir y utilizar bienes y servicios requeridos para el desempeño profesional. Estas cuatro

dimensiones favorecen la valoración integrada de la sostenibilidad de un producto informático y se caracterizan como: [28]

- Dimensión administrativa: Factores económicos de ahorro, gasto, precio de venta, calidad de la producción y los servicios, administración de recursos, toma de decisiones administrativas.
- Dimensión socio humanista: Modo de vida, desarrollo de un grupo social, satisfacción de necesidades sociales, formación ético humanista, la ciencia y la tecnología como procesos sociales.
- Dimensión ambiental: Condiciones favorables o no a las personas o cosas, minimizar daños e impactos negativos al medio ambiente.
- Dimensión tecnológica: Uso de tecnología adecuada y asimilable con el contexto y el usuario.

El procedimiento se estructura en dimensiones, factores y preguntas orientadoras. Las cuatro dimensiones, 26 factores que organizan las dimensiones y preguntas asociadas a cada factor con el objetivo de orientar la valoración personal del sujeto. El procedimiento se validó mediante varias rondas del método Criterio de expertos y se estableció a través de una aplicación web, constituyendo un eje transversal articulador de la diversidad de contenidos del plan de estudios en la unidad de la educación para la sostenibilidad.

2. En 2010, Markus Dick y Stefan Naumann, en su publicación “Mejorar los procesos de Ingeniería de Software para el diseño de un producto de software sostenible” proponen una extensión genérica para procesos de desarrollo de software cuya meta es una integración entre cuestiones de sustentabilidad en procesos de desarrollo de software arbitrarios y guiar a un mejor producto de software sostenible. Las herramientas, listas de chequeo y líneas guías de asistencia profesional, con la aplicación de su modelo, se enfocan en el pilar medioambiental de la sostenibilidad y el consumo de energía. [5]

3. En 2012 Lami define un grupo de procesos de alto nivel que añaden los conceptos de sostenibilidad a los procesos de referencia de software definidos en ISO/IEC

12207. Los procesos de referencia estándar describen todas actividades directamente e indirectamente relacionadas con el desarrollo, mantenimiento y operación del software. Además señala que los procesos respecto a la sostenibilidad están faltantes en la ISO/IEC12207, por lo tanto, define un grupo de tres procesos de sostenibilidad complementarios: el Proceso de Dirección Sostenible, el Proceso de Ingeniería Sostenible y el Proceso de Capacitación Sostenible. [6]

4. En 2013 la Universidad Politécnica de Cataluña, presenta un estudio realizado para incluir los conceptos de sostenibilidad en un proyecto de ingeniería, en la guía del Trabajo de Final de Grado (TFG) de Ingeniería Informática de la Facultad de Informática de Barcelona, donde siguen la idea de establecer una serie de preguntas que inviten al estudiante a reflexionar sobre la forma de afrontar su trabajo. Sin embargo, las ideas y métodos discutidos sirven en general para cualquier proyecto TIC y, por extensión y con ciertas modificaciones, para cualquier ingeniería. [16]

Inspirándose en las ideas de Felber desarrollan una matriz que contiene una lista de preguntas que el ingeniero debe hacerse a la hora de plantear y desarrollar su proyecto. La respuesta a estas preguntas permitirá confeccionar el informe de sostenibilidad del proyecto. Descartan la posibilidad de usar las etapas del proyecto en los ejes de la matriz porque el enfoque resultaba altamente complejo, además de poco útil, ya que estas reflexiones deben realizarse de forma holística integrada con la ética personal y profesional del ingeniero. También descartaron usar en los ejes las dimensiones de la sostenibilidad (ambiental, económica y social), porque para que un proyecto sea sostenible debe considerar simultáneamente las tres dimensiones. Hay preguntas en las que dos o las tres dimensiones están presentes, y su evaluación por separado sería redundante.

En la matriz se propone que en las filas haya una lista de preguntas relacionadas directamente con la elaboración del proyecto (producto/proyecto, equipo, proveedores/financiación/materiales, manufactura/distribución y vida útil), mientras que en las columnas sitúan las preguntas relativas a consideraciones para la puesta en producción, al impacto de los resultados y a los posibles riesgos. Las celdas de la matriz contienen

preguntas asociadas a las tres dimensiones de la sostenibilidad y relacionadas con la economía del bien común.

Las preguntas no están formuladas de modo detallado y preciso, sino de forma deliberadamente simplificada para que sea fácil adaptarlas a cualquier tipo de proyecto. La Tabla 1 muestra la propuesta. La matriz de preguntas se ha planteado para que pudiera servir en cualquier proyecto de ingeniería relacionado con las TIC, pero siendo conscientes de que algunas de las preguntas sobrepasan los conocimientos que debería tener un estudiante en su TFG. Por eso, se señala en verde aquellas cuestiones que sí deberían plantearse y en negro aquellas preguntas que van más allá de lo que debería plantearse un proyectista en su TFG.

Sostenible?	Consideraciones para la puesta en producción	Resultados	Riesgos
Producto / Proyecto	¿Económicamente viable? ¿Satisface alguna necesidad básica de los usuarios?	¿Mejora de calidad de vida?	¿Cambios de comportamiento de los usuarios? ¿Dependencias creadas? ¿Riesgos de uso? ¿Cambios medioambientales?
Equipo de trabajo	¿Criterios de contratación? ¿Transparencia en la información? ¿Consenso en las decisiones? ¿Condiciones de trabajo dignas/valores éticos?	¿Puestos de trabajo que se destruyen/crean?	¿Grado de implicación y satisfacción de los trabajadores?
Proveedores Materiales Financiación	¿Condiciones de trabajo dignas/valores éticos de los proveedores? ¿Criterios de selección de materiales? ¿Fuentes de energía sostenibles? ¿Financiación de banca ética?		¿Aumento de la huella ecológica? ¿Trazabilidad de la materia prima?
Manufactura Distribución	¿Reutilización de recursos de otros proyectos? ¿Optimización de consumos? ¿Transparencia en los procesos? ¿Transporte de materias primas / producto manufacturado / trabajadores?	¿Beneficios/contribución a otros proyectos?	¿Residuos fabricación/distribución? ¿Impacto ambiental?
Vida útil	¿Duración prevista y por qué? (Obsolescencia) ¿Diseño/fabricación pensado en el reciclado?	¿Necesidad de mantenimiento o modificación?	¿Reúso/reciclado/residuos?

Tabla 1: Propuesta para considerar la sostenibilidad en un proyecto de ingeniería/TFG.

Las preguntas no deben plantearse en un punto concreto del proyecto, sino que deben permanecer activas en la conciencia del proyectista o ingeniero durante todo el ciclo de vida

del mismo, desde la licitación de los requisitos y la definición del problema hasta su desmantelamiento final. [16]

5. En 2015 Stefan Naumann, Eva Kern, Markus Dick y Timo Johann, en su artículo “Ingeniería de software sostenible: procesos y modelos de calidad, ciclo de vida y aspectos sociales” definen los conceptos de software sostenible y de ingeniería de software sostenible. Realizan un estudio de investigaciones previas relacionadas con el tema, y basándose en estas presentan un modelo propio de consulta para evaluar la sostenibilidad del software y su ingeniería. Este modelo muestra los aspectos cualitativos de la ingeniería de software y su objetivo principal es hacer que el software por sí mismo sea tan sostenible como el proceso de ingeniería. [4]

6. En 2015 el Grupo de Investigación de Ingeniería de Software del departamento de Ciencias de la Computación y Tecnología de la Información, de la Universidad de Malakand, Pakistan, presentan un estudio en el cual exponen un enfoque diferente para evaluar el uso de metodologías ágiles para el desarrollo de software sostenible usando revisión de literatura y encontrar los riesgos de este camino. Esta investigación encontró 8 factores de riesgo que se enumeran en la Tabla 2, de una muestra de 42 documentos, de los cuales se han declarado algunos riesgos como críticos debido a su alta frecuencia. [29]

S. No	Risks Factors	Frequency N=42	Percentage
1	Insufficient system documentation	26	62
2	Limited support for real-time systems and large systems	19	45
3	Management overhead	25	60
4	Lack of customer's presence	26	62
5	Lack of formal communication	18	43
6	Limited support for reusability	07	17
7	Insufficient knowledge of the customer	15	36
8	Lack of long term planning	20	48

Tabla 2: Lista de factores de riesgo identificados.

Este estudio también presenta un análisis de los riesgos identificados por continentes, como se muestra en la Tabla 3. [29]

Risk Factors	Sample Size N=42								Chi-Square test (Linear- by -Linear Association) α=0.5		
	Asia N=12		N. America N=09		Europe N=18		Mixed N=03				
	Frequency	%	Frequency	%	Frequency	%	Frequency	%	X ²	df	p
Insufficient system documentation	09	75	06	67	10	56	03	100	2.105	1	0.147
Limited support for real-time systems and large systems	06	50	03	33	10	56	0	0	.208	1	.648
Management overhead	10	83	02	22	10	56	03	100	.137	1	.711
Lack of customer’s presence	09	75	06	67	10	56	01	33	2.105	1	.147
Lack of formal communication	04	33	04	44	08	44	02	67	.844	1	.358
Limited support for reusability	01	8	02	22	04	22	0	0	.128	1	.669
Insufficient knowledge of the customer	06	50	04	44	04	22	01	33	2.025	1	.155
Lack of long term planning	08	67	04	44	07	39	01	33	2.25	1	1.133

Tabla 3: Lista de riesgos identificados por continentes.

Este trabajo logra identificar los factores de éxito y los riesgos que puede tener un efecto directo e indirecto en el desarrollo de software sostenible utilizado métodos ágiles con el objetivo de obtener un modelo que ayude a los proveedores de desarrollo global a medir su madurez ágil para el diseño y la producción de software sostenible.

7. En 2015 Marcello Thiry y Liliane Frez [30] presentan un modelo de referencia para el desarrollo de software sostenible. En su artículo se enfocan en la definición de la estructura del modelo y los beneficios identificados con su implementación. También presentan una comparación entre los procesos del modelo y atributos con los requerimientos medioambientales de la ISO/IEC 14001 para demostrar su compatibilidad.

Además presenta una evaluación del modelo en tres organizaciones de software brasileñas para evaluar el modelo técnico y la viabilidad financiera. Este artículo presenta el GreenRM

modelo de referencia para apoyar a las organizaciones de software en el esfuerzo para lograr la mejora del proceso, pero considerando la calidad ambiental con la introducción de la gestión ambiental. [30]

8. En 2018 Achim Guldner, Marcel Garling, Marlies Morgen, Stefan Naumann, Eva Kern y Lorenz M. Hilty [31] publican un artículo en el cual presentan un método para medir el consumo de energía que el software induce al hardware, aplicándolo a dos grupos de software: los procesadores de texto (WP) y sistemas de gestión de contenido (CMS). Aunque los dos grupos son muy diferentes en relación a sus requisitos, fueron exitosos en la creación de un ambiente de medición que soporta la producción de un resultado confiable y verificable, admitiendo la comparación del consumo de energía producido por sistemas de software con una funcionalidad similar. El método muestra los resultados viables para sistemas de escritorio y servidores, preparando el camino para configuraciones adicionales como por ejemplo dispositivos movibles e implantados.

El artículo se concentra en la medición de la utilización de equipo físico y presenta un método de medición para el consumo de electricidad. Para asegurar la equivalencia entre los diferentes tipos de productos de software, WP y CMS, se define un guión de uso estándar para cada categoría. [31]

Hay algunos desafíos involucrados en la creación de tal guion. Primero se analiza las tareas que los usuarios generalmente realizan con el software y se consideran las funciones del software y módulos que son de uso frecuente. También se estudian los potenciales de optimización dentro del software. Por lo tanto se tiene cuidado con las funcionalidades que pueden causar mayor consumo de energía en el hardware. [31]

Finalmente, se debe conseguir un guion que sea tan realista como posible, considerando el comportamiento del usuario. Para esto graban a varios usuarios llevando a cabo un escenario, por ejemplo para un CMS, usando otros medios como una herramienta generadora de carga que llama los sitios web generados por el CMS muchas veces por minuto, simulando muchos usuarios. [31] Se llevan las mediciones verdaderas del consumo de energía y la utilización del hardware a un test que se crearon después que la ISO/IEC 14756.

En este, el producto de software es instalado sobre un sistema bajo prueba (SUT), que consiste en el hardware (computadora de escritorio, servidor, smartphone, etcétera), sistema operativo y si fuera necesario, promover ambientes de tiempo de ejecución. El SUT no sólo opera el producto de software sino también graba su utilización durante las mediciones. Adicionalmente, se usa un medidor de vatios industrial con una frecuencia de muestreo de 20 kHz por fase y resoluciones de 0.01 V y 1 mA, respectivamente, para medir el consumo de energía del SUT. Los datos recolectados son promediados por segundo y luego guardados y analizados en un agregador y evaluador central de datos. [32]

Otro factor a considerar es la sobrecarga sobre el SUT, resultando de grabar la utilización del hardware. En las mediciones, se encontró que la sobrecarga de las herramientas de grabación de datos es insignificante. Sin embargo, si las mediciones más precisas son necesarias, podría ser aconsejable medir el software de grabación primero y luego restar este punto de partida de la medición verdadera. Cuando la configuración está completa, usamos un generador de carga para ejecutar los guiones automática y repetidamente sobre el SUT, para asegurar unos resultados estadísticamente robustos.

A continuación, se describe el método para analizar la medición de los resultados de la evaluación del consumo de energía medio y la utilización media del hardware de los productos de software mientras efectuamos el guión de uso estándar.

El consumo de energía es calculado como un valor aritmético medio sobre cada medición. La energía para cada muestra de medición es calculada como la suma sobre el vataje medio por segundo (P_i) que fue medido para el SUT. Se divide el resultado entre 3600 s/h para recibir el resultado por el indicador en watt-horas (Wh). Por lo tanto la fórmula para calcular el indicador para el consumo de energía es: [31]

$$E = \frac{1}{3600 \cdot n} \sum_{k=1}^n \sum_{i=1}^m P_i,$$

Donde la duración de k-th es m segundos y ahí está un total de n mediciones.

La utilización del procesador es calculada como el valor de media aritmética sobre cada muestra de medición en por ciento (%) del rendimiento de procesador máximo del SUT. Esto

quiere decir que si hay un total de n mediciones, calculamos el indicador para la utilización de procesador como:

$$U_P = \frac{1}{n} \sum_{k=1}^n \bar{x}_k,$$

Donde $\bar{x}_k$ es la media aritmética de la muestra de k - th en %. Se calculan los indicadores para otra calidad de hardware como la utilización de RAM en una manera análoga.

Para el grupo de procesadores de texto, se seleccionaron dos procesadores de texto generalizados para compararlos, WP1 y WP2. La decisión de usar estos productos ha sido tomada por el hecho de que son ampliamente usados y uno de ellos es un producto de software comercial mientras que otro es de código abierto. Se implementaron las acciones como editar el contenido y el diseño del documento, además de salvarlo, varias veces durante el guión. Se usó el documento "Anatomía quirúrgica" de José Maclise, el cual contiene más de 300 páginas. Al final se crearon dos documentos pdf idénticos, uno con cada procesador de texto.

Para tomar una muestra representativa los dos productos fueron instalados secuencialmente en el SUT e iniciados en la configuración estándar. Adicionalmente, prestamos atención a una visualización similar de la aplicación que comprende, por ejemplo, abrir el software en la pantalla completa, ajustar el nivel de desplazamiento de foco y activar los caracteres de control.

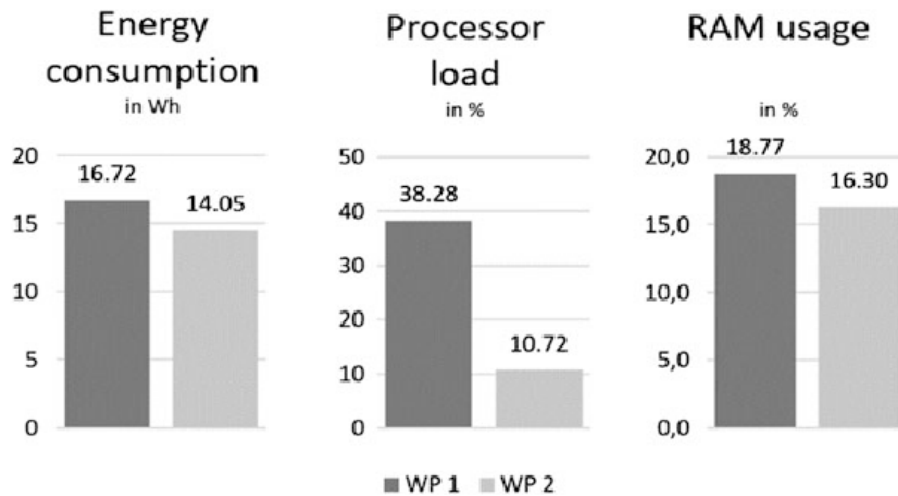


Figura 1 Resultados de las mediciones de los procesadores de texto WP1 y WP2

Al comparar WP1 con WP2 se hallan diferencias significativas en cuanto a consumo de energía, carga del procesamiento y uso de la RAM (Fig. 1). Ambas aplicaciones tenían un máximo apogeo en el principio mientras ponían en marcha el software, abrían el documento y cambiaban el diseño del contenido. [31]

Además de eso, las mediciones de WP2 indican los máximos apogeos mientras guardaba el documento. Las otras acciones tienen solamente influencias pequeñas sobre el nivel general del consumo de energía. Sin embargo, WP1 presentaba los giros mucho más grandes mientras llevar a cabo las acciones de todas clases. Eso es parte de la razón por la cual también se analiza la transición al modo de reposo para ambos productos de software. [31]

Para este guión, se abre un nuevo documento sin ningún contenido y se da una mirada en la reacción de software. WP1 espera 26 s hasta que el cursor deja de destellar. Después, el consumo de energía y utilización de equipo físico disminuyen a un nivel insustancial de la lectura. WP2 se queda en modo activo por 18 s sin la entrada del usuario antes de que cambie al modo de reposo y la utilización del hardware y el consumo de energía disminuyen a un mínimo. [31]

Los resultados indican que hay potencial de optimización para WP1. Especialmente para el cursor destellando. Además, en WP2 la mayor parte del editar, como añadir el contenido o cambiar el diseño del documento, tenía que un inferior impacto en el consumo de energía en WP1.

[31]Para los grupos de sistemas de contenidos, se escogieron dos CMS (CMS1 y CMS2) debido a su larga participación en el mercado. Para poder llevar a cabo las mediciones necesarias con el CMS, se crea el contenido en forma de un sitio web. Para desempeñar la equivalencia de los resultados, este contenido tiene que ser igual para todos productos. La demanda para la igualdad cubre el contenido y la estructura, tanto como el diseño del sitio web. Se cumplen estos requisitos desarrollando una plantilla básica, implementarlo para cada CMS y luego compilar la misma estructura y el contenido en cada uno. En relación con el tamaño, el sitio web representa un sitio de información pequeño, comprimible en texto e imágenes, como esas a menudo operadas por pequeñas empresas. [31]

El guión de uso sobre la base del sitio web cubre las típicas interacciones usuario de un editor que trabaja en el panel de administrador. Después de que el usuario entra al sistema, algunos comentarios son dirigidos, entonces una nueva página es creada, editada, y publicada. El usuario carga archivos de PDF y los vincula. Después, son retirados otra vez con la herramienta de revisión incorporada. Definitivamente, todos cambios son revocados, permitiendo que la próxima corrida de medición empiece con las condiciones idénticas. [31]

Se crea este guión para correr sobre un cliente, que sirve de generador de carga, mientras que el CMS fue instalado en el servidor (SUT). Se miden ambos sistemas simultáneamente durante las corridas de prueba. El procedimiento es aplicado repetidamente para medir el impacto de diferentes niveles de tensión: en la primera configuración, solamente el usuario autómatas del guión de uso ejecuta las interacciones con el sistema. En una segunda configuración sin embargo, se usa la herramienta de generador de carga mencionada anteriormente para llevar a cabo 14,000 solicitudes del sitio web adicionales por minuto, simulando algunos cliente de computadoras automáticamente. [31]

Es extraordinario que ejecutar el guión que usa CMS1, el servidor envía por promedio más datos que cuando corre CMS2 (CMS1: 0.09 megabytes/ s; CMS2: 0.03 megabytes/ s). Aunque la actividad de la red en el lado del cliente es más alta para CMS1, CMS2 es el que muestra la actividad de escribir en el disco duro (CMS1: 0.08 megabytes/ s; CMS2: 0.13 megabytes/ s).

Los resultados de la segunda configuración (una con la herramienta de generador de carga en uso) brinda una nueva percepción mucho más diferenciada. El consumo de energía y la utilización de hardware son ahora elevadas evidentemente. Comparando los dos sistemas bajo tensión, CMS2 resulta ser más absorbente que CMS1 en muchos aspectos. Tiene un consumo de energía más alto, un uso de CPU más alto y ahora también una actividad de transferencia de datos más alta. (Fig. 2)

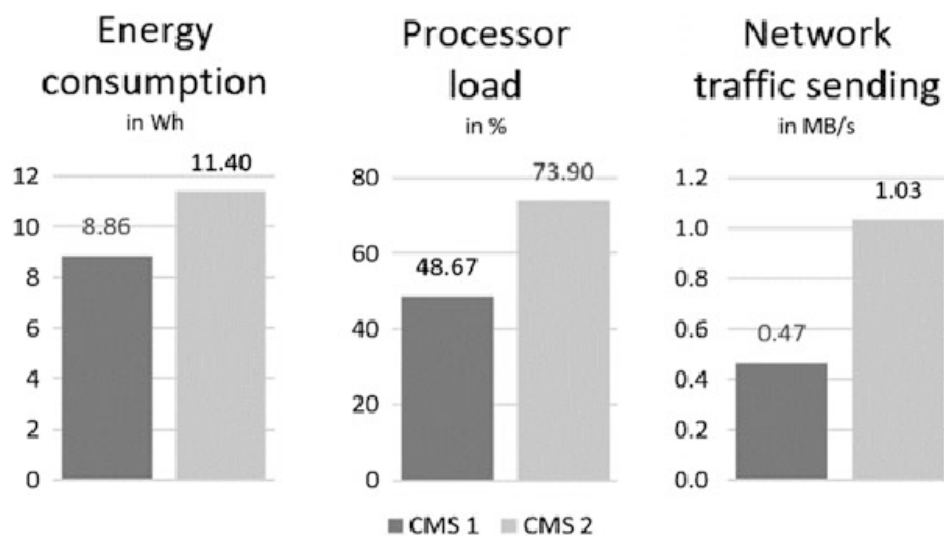


Figura 2: Resultados de las mediciones de CMS1 y CMS2

Sobre la base de estos resultados, los desarrolladores pueden mejorar su código, y los compradores de software pueden tomar las decisiones, sobre la base de la sostenibilidad de los productos. [31]

9. En este año 2018, se publica un modelo que describe las cadenas causales de productos de software a sus impactos sobre recursos naturales, incluyendo fuentes de energía, desde una perspectiva de ciclo de vida. [33] Se concentra en: 1-las demandas del software para las capacidades del hardware (local, lejano, y en la red conectada) y la demanda de energía resultante del hardware, 2- las expectativas de usuarios que prestan atención a tales demandas y cómo éstos afectan la vida operativa del hardware, y 3- la autonomía de los usuarios en dirigir el uso de su software con respecto a la eficiencia de recurso. Se propone un grupo de criterios e indicadores para valorar estos impactos. [33]

Se hace una demostración de la aplicación de este grupo de criterios, incluyendo la definición de los guiones de uso estándar para las categorías elegidas de productos de software. Se habla de la viabilidad de este tipo de valoración, su aceptabilidad para algunos stakeholders y las consecuencias potenciales para la eco - etiqueta de productos de software y diseño de software sostenible. [33]

El objetivo del proyecto es desarrollar un método para valorar los impactos ambientales de productos de software y suministrar recomendaciones a ingenieros de software para desarrollar el software con el impacto ambiental bajo. El método de evaluación está pensado para apoyar tanto la adquisición de los productos de software con la consideración de los criterios ambientales como el desarrollo del software eficiente. [33]

El método permite una comparación de dos productos de software en particular con la funcionalidad similar, donde la comparación se concentra en los impactos de su uso sobre recursos naturales. Sobre la base de la formulación de los estándares mínimos ambiciosos, el método ayudará a definir los criterios para otorgar una etiqueta ambiental o de calidad para productos de software sostenibles. [33]

1.4 – Conclusiones

Se definen los conceptos asociados al campo de acción y objeto de estudio, según los principales autores nacionales e internacionales. Se realiza un estudio del estado del arte teniendo en cuenta investigaciones internacionales y nacionales, y los modelos de procesos de desarrollo de software que han sido propuestos. Se concluye que a pesar de que este es un tema emergente en el país se han realizado pocas investigaciones, encontrándose solo dos, sin embargo a nivel mundial existen varios autores publicados sobre este tema.

2 – Análisis y comparación de modelos de procesos de software sostenible

2.1 – Introducción

En este capítulo se realiza un análisis de modelos de procesos planteados por Stefan Naumann, Eva Kern, Markus Dick y Timo Johann, Lami, y Mahmoud y Ahmad, Marcello Thiry, Liliane Frez y Alessandra Zoucas. Se comparan los modelos considerando diferentes aspectos para la selección del más adecuado para el contexto cubano.

2.2- GREENSOFT Model

El modelo GREENSOFT [34] es un modelo de referencia conceptual para el software sostenible, que tiene el objetivo de respaldar a desarrolladores de software administradores y usuarios para crear, mantener y usar el software de manera sostenible. El modelo (Fig. 1) comprende un modelo de ciclo de vida holístico para productos de software, criterios de sostenibilidad y métricas para productos de software, modelos de procedimiento para stakeholders diferentes, y recomendaciones para la acción, tanto como herramientas que soportan stakeholders en desarrollar, comprar, proporcionar, y usar el software en una manera sostenible.

Este modelo contiene un “Ciclo de Vida de Productos de Software”. Esto es, en contraste con los ciclos de vida tradicionales, conocido como Idea de Ciclo de Vida (Life Cycle Thinking (LCT)). Este tiene el objetivo de tasar la compatibilidad ecológica, social, humana y económica de un producto durante su ciclo de vida. Empieza con las etapas tempranas del desarrollo de producto y termina con el traspaso y el reciclaje del producto. [34]

La segunda parte del modelo GREENSOFT es llamada “Criterios y métricas de sostenibilidad”. Esta cubre las métricas comunes y los criterios para la medición de la calidad de software y da una clasificación de los criterios y las métricas para valorar la sostenibilidad de un producto de software. [34]

El último componente del modelo contiene las “Recomendaciones y Herramientas”. Éstos respaldan a stakeholders con niveles de destreza profesionales diferentes en aplicar las técnicas sostenibles en general, cuando desarrollan, compran, administran, o usan productos

de software. Los papeles posibles son desarrolladores de software, adquirentes del software, administradores, tanto como usuarios profesionales y usuarios privados.

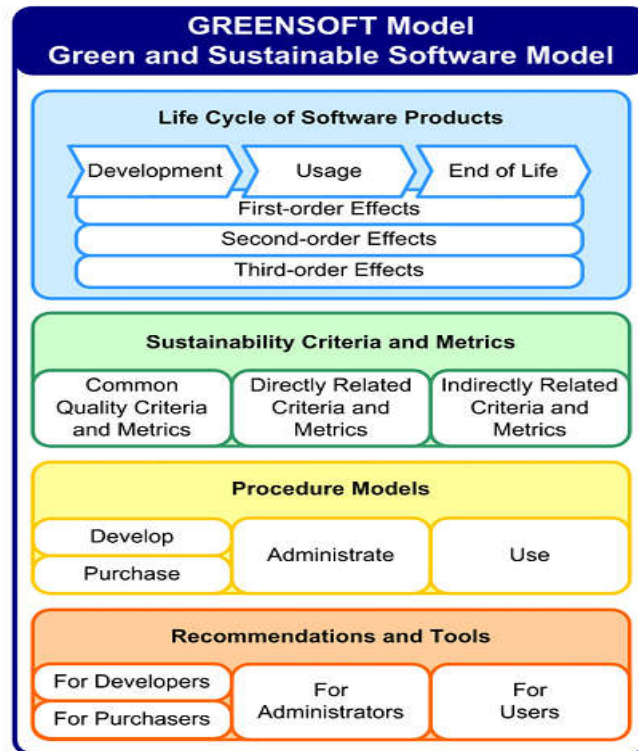


Figura 3 Modelo GREENSOFT. [34]

2.2.1- Ciclo de Vida del Producto de Software

El “Ciclo de Vida del producto de software” es un LCT. Su objetivo es permitir a los stakeholders que midan los impactos sobre el desarrollo sostenible de acuerdo con los tres niveles de impactos diferentes. [34]

- Impactos de Primer Orden: son los efectos ambientales que resultan de la producción y uso de las TIC, uso de recursos y contaminación por la producción de hardware, consumo de energía durante el uso y los desperdicios de equipos electrónicos.
- Impactos de Segundo Orden: Son los efectos secundarios del uso de las TIC, como la conservación de recursos y energía mediante procesos de optimización, o la conservación de recursos debido a la sustitución de productos materiales por su contraparte inmaterial.

- Impactos de Tercer Orden: son los efectos indirectos a largo plazo en el medio ambiente que resultan del uso de las TIC, como cambios de estilos de vida que promueven un rápido crecimiento de la economía.

Cuando se trabajan con estos niveles, los stakeholders deben ser conscientes holísticamente considerando todos los niveles, porque es más fácil concentrarse en los impactos de primer orden cuando son inmediatos y obvios. Sin embargo, los impactos de primer orden podrían ser los más amenazadores, aunque los de segundo y tercer orden también tienen potenciales inmensos para impulsar el desarrollo sostenible. El “Ciclo de Vida del producto de software” consta de tres fases: Fase de Desarrollo, Fase de Uso y Fin de la Vida: [34]

Fase de Desarrollo: esta fase explica los impactos sobre el desarrollo sostenible que resultan directamente de las actividades involucradas en el desarrollo de software, así como las actividades indirectamente implicadas.

La Fase de Distribución, que pertenece a la Fase de Desarrollo, explica los impactos sobre el desarrollo sostenible que resultan de distribuir el producto de software.

La Fase de Traspaso, que pertenece a la fase de Fin de la Vida, explica los impactos sobre el desarrollo sostenible que resultan del traspaso y el reciclaje de los subproductos materiales mencionados antes.

Fase de Uso: Esta fase considera los impactos que resultan de desplegar, usar, y mantener actualizado el producto de software.

Fase Fin de la Vida: Si un producto de software es sacado del servicio, es necesario cambiar los datos disponibles por un formato que puede ser procesado por el producto de software siguiente, o hacerlo accesible de alguna otra manera. Si los datos no pueden ser convertidos fácilmente, por ejemplo porque es guardado en un formato de datos reservado, esto podría tener un impacto sobre la sostenibilidad económica de una organización.

2.2.2- Criterios y métricas de sostenibilidad

El modelo tiene la capacidad de representar tres categorías de criterios y métricas sostenibles para productos de software: criterios de calidad comunes y métricas, los criterios directamente relacionados y métricas, y los criterios indirectamente relacionados y métricas. El primero se relaciona con los criterios de calidad comunes para el software. [35] Los segundos comprenden los criterios y métricas que se relacionan con los efectos de primer

orden. Los últimos se relacionan con los efectos de segundo y tercer orden. Todos los criterios deben ser clasificados de acuerdo con las fases del ciclo de vida del producto de software propuesto. También es necesario clasificar las métricas y los criterios según el producto de software.

2.2.3- Modelo de Procedimiento

Green and Sustainable Software Engineering produce Software Verde y Sostenible de una manera ecológica y sostenible. En consecuencia, durante el proceso de ingeniería, debe tenerse en cuenta todo el ciclo de vida del producto de software diseñado, así como las circunstancias bajo las cuales se producirá. [34]

Modelo de sub-procedimiento Desarrollo

Se presenta un ejemplo de un proceso de desarrollo de software que encaja en la categoría Desarrollo del modelo de procedimiento como parte del Modelo GREENSOFT. [34]

Este ejemplo (Fig. 3) no implementa un proceso completo de desarrollo de software. En cambio, propone varias mejoras para procesos de desarrollo de software arbitrarios que permiten a las partes interesadas reconocer los impactos, que resultan de "producir" el producto de software, y los impactos, que resultan del uso del producto de software. Las mejoras propuestas son: Reseñas y revisiones de sostenibilidad, Evaluación de procesos, Diario de sostenibilidad y la llamada Retrospectiva de sostenibilidad. [5]

En principio, las mejoras forman un ciclo de mejora continua que se ocupa de cuestiones de sostenibilidad. La evaluación de procesos ayuda a optimizar la sostenibilidad del proceso de "producción", mientras que las revisiones y pre visualizaciones de sostenibilidad ayudan a optimizar la sostenibilidad del producto de software en evolución. [34] Ambos esfuerzos se combinan en la Retrospectiva de sostenibilidad, por lo que finalmente se cubren los impactos en todo el ciclo de vida del producto de software.

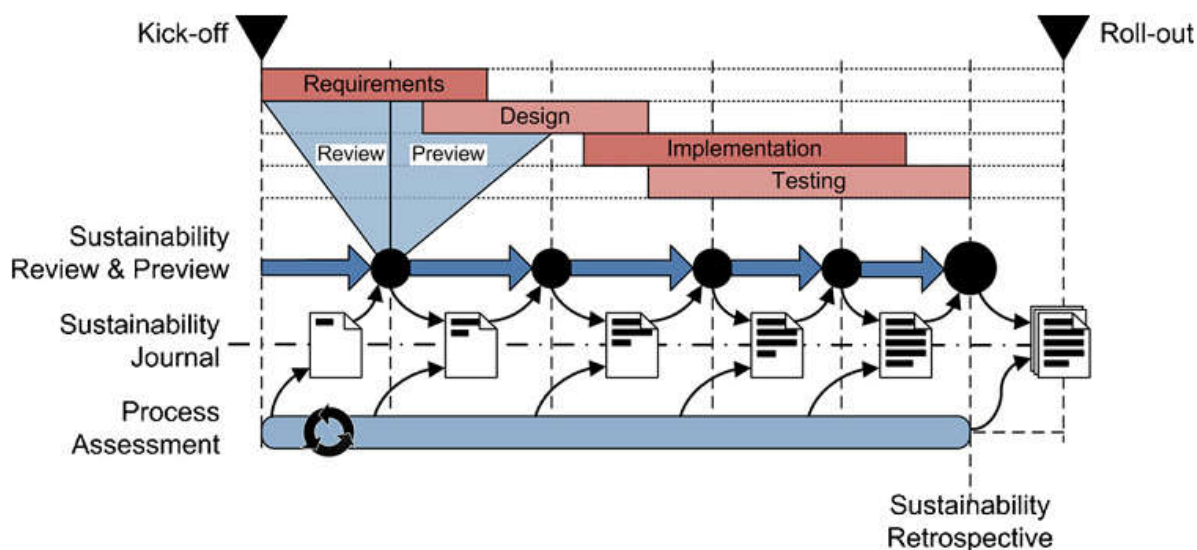


Figura 4: Ejemplo de un proceso de desarrollo de software que encaja en la categoría Desarrollo

Las revisiones y pre visualizaciones de sostenibilidad analizan el trabajo realizado, evalúan los resultados de acuerdo con los problemas de sostenibilidad y desarrollan medidas, que se realizan hasta la próxima Revisión y vista previa de la sostenibilidad con el fin de optimizar la sostenibilidad del producto de software en desarrollo. [5]

La actividad de Evaluación de procesos supervisa continuamente el proceso de desarrollo. Por lo tanto, se recopilan diferentes datos del proceso de desarrollo para evaluar sus impactos en SD e identificar los factores que deben optimizarse para mejorar el proceso. Estos datos también se pueden usar como base para realizar un ACV del producto de software. [5] La Retrospectiva de Sostenibilidad resume los datos recopilados por las Evaluaciones y revisiones de sostenibilidad y Evaluación de procesos, evalúa el impacto general en SD del producto de software y busca formas de mejorar la sostenibilidad de los próximos proyectos de software y sus productos de software. [34]

Este modelo de procedimiento establece solo un marco organizacional que ayuda a administrar la sostenibilidad de un producto de software. [34] Por lo tanto, no hay garantía de que los productos de software resultantes sean más sostenibles que lo que hubiera sido sin aplicar este proceso. Además de reflejar el ciclo de vida propuesto de un producto de software, existen otros métodos que ayudan a los arquitectos, diseñadores y desarrolladores de software a producir aplicaciones de software ecológicas y sostenibles.

Como se muestra anteriormente, estas métricas y sus propiedades de calidad correspondientes también se pueden interpretar con respecto a la sostenibilidad. Por otro lado, producir software ecológicamente sano, eficiente en recursos y energía también es un problema hoy en día. [36]

En un proyecto de software, cuando los primeros artefactos de software desplegados están disponibles, es fácilmente posible medir su consumo de energía, ya sea midiendo el consumo de energía directamente o usando contadores de monitor de rendimiento de procesadores modernos como indicadores. [38] Estas mediciones complementan las primeras estimaciones y pueden proporcionar indicaciones adicionales sobre componentes de software que inducen un alto consumo de energía y, por lo tanto, deben optimizarse con prioridad. Con base en estas arquitecturas de software y medidas centradas en el desarrollo de software, las organizaciones de desarrollo de software también deben estimar la demanda total de energía y recursos que se espera de acuerdo con su número proyectado de instalaciones o cifras de ventas de un producto específico y los escenarios de uso estimados asociados. [34]

Estas estimaciones no solo deberían incluir los impactos de primer orden de la fase de uso, sino también los impactos de primer orden de las otras fases, especialmente de la fase de distribución. Como un paso más y dependiendo del tipo de software, también es posible estimar los impactos secundarios y de tercer orden, respectivamente.

Sub-procedimiento modelo Administrar

En este modelo, "administrar" significa hacer que el software esté disponible instalándolo, configurándolo y manteniéndolo. Esto también cubre educar y capacitar a los usuarios, que trabajan con el software en una organización. La configuración de los productos de software cubre solo la configuración desde un punto de vista de la organización, aquí llamado "macroconfiguración". [34]

Configuraciones que pueden realizar los usuarios sin la necesidad de permisos especiales del sistema, la instalación de complementos para navegadores, la configuración de un procesador de textos a través de su diálogo de preferencias, etc., están cubiertos como "microconfiguración" por el modelo de subprocedimiento Uso. Por lo tanto, la macroconfiguración cubre la configuración y el mantenimiento del sistema de PC de

escritorio, clientes ligeros, así como servidores y centros de datos virtualizados. Especialmente, las diversas publicaciones sobre eficiencia energética del centro de datos encajan en esta parte del modelo. [34]

Un modelo de procedimiento minimalista debe implementar un ciclo de mejora continua, que sigue el paradigma planea-ha-chequea-actúa. Aquí, es decir, la eficiencia energética, el consumo de energía y el consumo de recursos deberían controlarse regularmente para mejorarlos con medidas apropiadas. Esto no se limita a las operaciones del centro de datos, sino que incluye infraestructura de red, computadoras de escritorio, software instalado y usuarios, atendidos por una división de servicios de TI de una organización [40] .

Modelo de sub-procedimiento Uso

Los procedimientos para los usuarios, ya sean profesionales o usuarios domésticos, relacionados con software verde y sostenible, no se pueden poner en un flujo de trabajo estático o modelo de procedimiento, porque estos son principalmente procedimientos ad-hoc. Sin embargo, de forma similar al componente de modelo de subprocesos Administrar, un ciclo de mejora continua se puede aplicar intuitivamente. [34] En este contexto, los usuarios deben reflexionar sobre cada una de sus acciones que puedan tener un impacto negativo en SD para buscar una orientación adecuada que ayude a mitigar estos impactos negativos.

Como se mencionó en la sección "Administrar" del modelo de subprocesos, los usuarios realizan algún tipo de "microconfiguración" como instalar complementos en navegadores web, configurar el procesador de textos, etc. Y usan productos de software. Sin embargo, debido a la ausencia de un proceso más formalizado y el carácter ad-hoc de las acciones, los procedimientos están más o menos representados directamente como recomendaciones para la acción, como directrices o listas de verificación. [34]

2.2.4- Recomendaciones y Herramientas

Las "Recomendaciones y Herramientas" se dirigen a stakeholders con roles diferentes. [34] Los roles considerados por el modelo GREENSOFT son: Desarrolladores, Compradores, Administradores y Usuarios. Sin embargo pueden haber otros roles especializados: Ingenieros de Requerimientos, Arquitectos de Software, administrador Web o Desplegadores de la Aplicación. En principio estos pueden ser asumidos por los roles antes mencionados.

Las “Recomendaciones y Herramientas” apoyan a stakeholders con niveles de destreza diferentes en aplicar las técnicas verdes y sostenibles cuando desarrollan, administran o usan productos de software. Las recomendaciones pueden ser las pautas, las listas de verificación, los ejemplos de buenas prácticas, reportes de implementación, etcétera. Las herramientas pueden ser herramientas de software, pero también cualquier otra herramienta.

2.3- Modelo de Calidad para Software Sostenible de Naumann

Para determinar si el software es sostenible o no, son requeridos los criterios apropiados. Un primer enfoque para desarrollar éstos es brindado por el Modelo de Calidad para el Software Sostenible (Fig. 2). En general, los diferentes aspectos pueden ser agrupados en los criterios comunes, los criterios directamente relacionados, y los criterios indirectamente relacionados y métricas, como se describe en el modelo. [4]

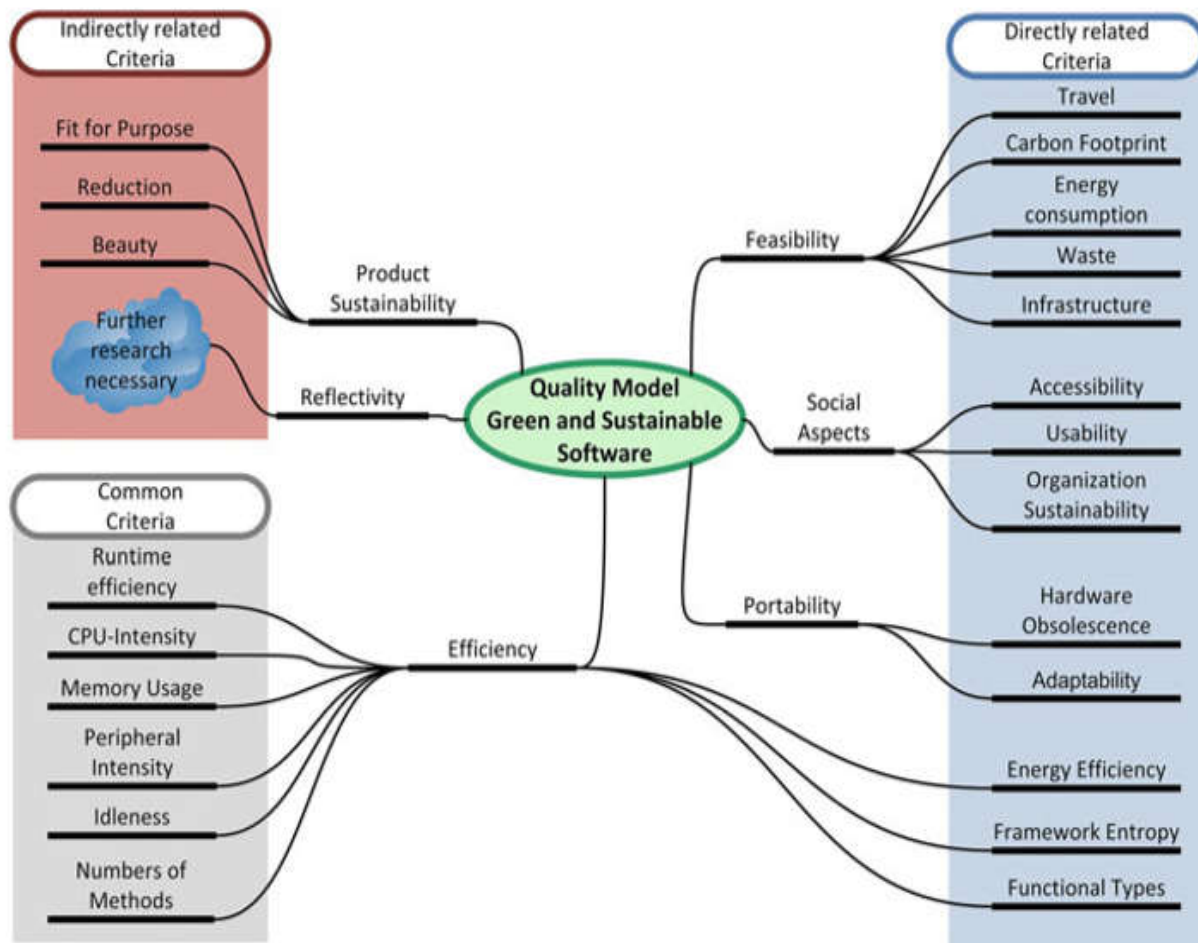


Figura 5: Modelo de Calidad para Software Sostenible

Criterios comunes de calidad (Common Quality Criteria): Los criterios comunes surgen de los conocidos y estandarizados aspectos de calidad de software, emitidos por la Organización Nacional de Estandarización. El modelo toma aspectos como: Eficiencia, Reusabilidad, Escalabilidad y Usabilidad. Los siguientes indicadores pertenecen al criterio Eficiencia: [4]

- Eficiencia en el tiempo de corrida (Runtime Efficiency): se considera el tiempo necesario para terminar la ejecución del software dependiendo de su implementación.
- Intensidad de la CPU (CPU Intensity), Uso de la Memoria (Memory Usage) e Intensidad de los periféricos (Peripheral Intensity): Comprenden los recursos utilizados por la ejecución del software.

- Inactividad (Idleness): Describe cuan a menudo el sistema está inactivo (usualmente utilizado en los servidores).
- Número de métodos (Number of Methods): Expresa el tamaño de la aplicación.

Criterios Directamente Relacionados (Directly Related Criteria): En contraste con los criterios comunes de eficiencia, la eficiencia energética del sistema en si puede ser evaluado en un sistema de referencia. Esto significa que su meta es optimizar el consumo de energía en relación con los servicios del sistema. Por tanto el indicador Eficiencia Energética (Energy Efficiency), que pertenece al criterio Eficiencia, es asignado a los Criterios Directamente Relacionados, junto con la Entropía del Framework (Framework Entropy) y Tipos Funcionales (Functional Types). [4]

- Entropía del Framework (Framework Entropy): representa el grado de uso de librerías externas y peticiones de alto nivel.
- Tipos Funcionales (Functional Types): Indican los diferentes tipos de aplicaciones con sus respectivos tipos de niveles de eficiencia energética.

Obsolescencia del Hardware: considera la cantidad de hardware que necesita ser reemplazado antes de que se termine la sustentabilidad de la vida útil, considerando los altos requerimientos del sistema. Este indicador, junto con Adaptabilidad, pertenece al criterio Portabilidad. Adaptabilidad describe las posibilidades del software de adaptarse a las necesidades específicas del consumidor.

El criterio de calidad Factibilidad (Feasibility) evalúa los efectos del desarrollo del producto, o más específicamente el proceso de ingeniería de software en el desarrollo sostenible. En este contexto se tiene: [4]

- Viajes (Travel): Incluye viajes de negocios que corresponden al desarrollo del software o incluso los viajes al centro de trabajo diariamente.
- Huella de carbón (Carbon footprint): especifica la cantidad de CO2 emitido a causa del desarrollo del software en el curso del ciclo de vida de este.
- Consumo de energía (Energy Consumption): indica la cantidad de energía consumida a causa del desarrollo del software.

- Desperdicios (Waste): Cuenta los recursos consumidos durante el desarrollo del software por actividades que no tienen valor adicional para consumidores o usuarios finales.
- Infraestructura (Infrastructure): condiciones necesarias para que el software pueda ser utilizado.

Desde la perspectiva de los Aspectos Sociales del desarrollo sostenible se tiene los siguientes indicadores: [4]

- Accesibilidad (Accessibility): Indica si el software puede ser usado por la mayor cantidad de personas posible sin restricciones.
- Usabilidad (Usability): Cubre la entendibilidad, operabilidad y facilidad de aprendizaje del software por parte de los usuarios.
- Sostenibilidad de la organización (Organization Sustainability): Cubre la situación social de la compañía.

Criterios Indirectamente Relacionados (Indirectly Related Criteria): Describe como por el uso del software, el uso de recursos y energía pueden ser reducido en otros sectores. Por eso, los criterios indirectamente relacionados con el producto de software deben ser también considerados.

El criterio Sostenibilidad del Producto (Product Sustainability) hace un sumario de los efectos del software en otros productos o servicios. [4]

- Ajusta al propósito (Fit for Purpose): considera que como se ajusta el software al propósito para el que fue desarrollado.
- Reducción (Reduction): Es la contribución del software dentro de su dominio de aplicación para reducir emisiones, desperdicios.
- Belleza (Beauty): Contribución del software al desarrollo sostenible.
- Reflectividad (Reflectivity): se refiere a la calidad como un aspecto de eficiencia. Especifica la manera en la que el uso del software influye en los recursos que son usados indirectamente por otras personas o sistemas. Este aspecto no pertenece a los Efectos de Primer Orden en el sentido propio del término pero cuenta como un

efecto del uso e indica un estándar de calidad de los Criterios Indirectamente Relacionados. Sin embargo se requiere más investigación en este campo con el objetivo de considerar el comportamiento del usuario y los escenarios de uso.

2.4- Descripción de los modelos de procesos

2.4.1- Modelo Verde para Ingeniería de Software Sostenible

El Modelo Verde para Ingeniería de Software Sostenible (Green Model for Sustainable Software Engineering (GSEP)) está basado en el modelo GREENSOFT. Es un modelo dos niveles, que tiene un proceso de ingeniería de software sostenible completo en el primer nivel y una colección de herramientas de software ejemplares, promoviendo la computación verde en el segundo nivel. [4]

El proceso consta de nueve fases sucesivas: [4] Requisitos, Diseño, Pruebas De Unidad, Implementación, Pruebas Del Sistema, Análisis Verde, Uso, Mantenimiento Y Traspaso. Como se puede apreciar, las fases tradicionales de procesos de software secuenciales son extendidas con el Análisis Verde, el Uso, y el Traspaso. Un acontecimiento incremental puede ser puesto entre la Implementación y las Pruebas de Sistema. El acontecimiento de lanzamiento del sistema ocurre entre el Análisis Verde y el Uso. La circulación del proceso hace que el modelo permita retornar a los Requisitos desde las Pruebas de Unidad, las Pruebas del Sistema y el Mantenimiento.

La fase de Requisitos es la fase más crítica e importante para el resultado sostenible del producto de software porque los requisitos definen las características básicas y afectan las siguientes fases. Por lo tanto, es importante que los requisitos consideren los asuntos de sostenibilidad y sus objetivos. Esta fase incluye una actividad de valoración de riesgo en relación con la eficiencia de energía tanto como una actividad de prueba de requisitos. Si cualquier asunto es descubierto durante esta actividad, la fase de Requisito debe ser empezada otra vez. [4]

La fase de Diseño desarrolla la arquitectura del sistema sobre la base de los Requisitos. Tres pautas de decisión de diseño son proveídas para apoyar a los arquitectos: [4]

- Dar una idea general de un diseño compacto de las estructuras de datos y los algoritmos eficientes.
- Diseñar funcionalidades inteligentes y eficientes.
- Determinar cuidadosamente qué marcos y las bibliotecas externas son necesarias.

Los componentes deben ser reusados si posible. Si reusar componentes no cubre los requisitos de eficiencia de energía deben ser reorganizados para ajustarse a estos requisitos. La fase Implementación es descrita junto con la fase de Diseño, y no viene con ninguna descripción de procesos adicionales ya que depende de esta.

La fase de Prueba de Unidad se antepone a la Puesta en Práctica, porque si la no conformidad con los requisitos es encontrada con aquí, menos energía es consumida si el proceso regresa a Requerimiento tanto como si las Pruebas de Unidad vienen después de Implementación. Sin embargo, esto no quiere decir que las Pruebas de Unidad durante la fase de Implementación no son admitidas. Después de la Puesta en Práctica y las Pruebas de Sistema, la fase de Análisis Verde es llevada a cabo. Esta fase condiciona la sostenibilidad de cada incremento aplicando las métricas específicas a la clave desarrollada.

Como un ejemplo de indicadores de rendimiento sostenible que caracterizan la sostenibilidad de cada escenario en el proceso de ingeniería de software Mahmoud [7] sugiere la interpretación o el uso de CPU. Los usos de energía identificados, deben ser mapeados en el código fuente para optimizar los algoritmos afectados. Después es determinado si cualquier cambio en los requisitos es necesario para optimizar el uso de energía. Esto quiere decir que el proceso puede volver a Requerimientos para añadir los requerimientos faltantes, corregir los errores y de este modo optimizar el producto. Esto solo es aceptable si resulta en errores menos graves. Si solo son necesarios cambios en el código fuente, se implementan en la fase actual sin regresar atrás. [4]

2.4.2- Proceso central de sostenibilidad de software

Lami [6] define un grupo de procesos de alto nivel que añaden los conceptos de sostenibilidad a los procesos de referencia de software definidos en ISO/IEC 12207. Los procesos de referencia estándar describen todas actividades directamente e indirectamente relacionadas con el desarrollo, mantenimiento y operación del software. Además señala que

los procesos respecto a la sostenibilidad están faltantes en la ISO/IEC12207, por lo tanto, define un grupo de tres procesos de sostenibilidad complementarios: el Proceso de Dirección Sostenible, el Proceso de Ingeniería Sostenible y el Proceso de Capacitación Sostenible.

Estos procesos de referencia adicionales tienen por objetivo mejorar constantemente la sostenibilidad tanto del producto de software como de los procesos del software. De acuerdo con su definición de procesos de software sostenibles, un proceso es sostenible si sus impactos sobre la sostenibilidad son pequeños. [6]

Proceso de dirección Sostenible: El proceso de dirección incluye una fase preliminar, una fase de planificación, una fase de observación, y un control de sostenibilidad del proveedor. En la fase preliminar, los principios y los criterios para la sostenibilidad son establecidos. En la fase de planificación, las actividades de sostenibilidad del proceso de desarrollo son demostradas, y los requisitos correspondientes tanto como recursos necesarios son planificados. Después, la sostenibilidad de las actividades desplegadas y su conformidad con los requisitos son monitoreadas. La última parte del proceso de dirección trata sobre políticas de sostenibilidad y requisitos para productos y servicios proporcionados. Aquí, un acuerdo tiene que ser encontrado y la sostenibilidad del proveedor tiene que ser monitoreada. [6]

Proceso de Ingeniería Sostenible: El proceso de ingeniería está ocupado por las herramientas apropiadas y métodos para permitir y respaldar un proceso de desarrollo sostenible. En este contexto los asuntos y principios de sostenibilidad para el desarrollo son definidos, aplicados, y analizados. La energía y el consumo de recurso son los factores que afectan la sostenibilidad y deben ser identificados en el principio del proceso de ingeniería. En el próximo paso, los impactos de estos efectos deben ser analizados para poner los objetivos de sostenibilidad para el proceso de desarrollo posteriormente. Además de los impactos del proceso mismo, los impactos de pedidos de cambio sobre la sostenibilidad deben ser determinados. [6]

Proceso de Capacitación Sostenible: El proceso de capacitación es aplicable a recursos externos como son ingeniería y las herramientas de soporte de dirección. Apuntado a productos sostenibles, estos recursos externos también tienen que ser sostenibles. Para asegurar su calidad, una estrategia de capacitación, un plan de puesta en práctica para la

estrategia, la documentación de los resultados de la capacitación, y un informe de capacitación son requeridos. [6]

2.4.3- Modelo de Proceso para una Ingeniería de Software Ágil y Sostenible.

El Modelo de Proceso para una Ingeniería de Software Ágil y Sostenible (The Software Process Enhancement for Agile and Sustainable Software Engineering (GSSE)) [4] es presentado como un proceso ejemplar que forma parte del modelo de procedimiento del GREENSOFT Model. No es un proceso de desarrollo de software completo, pero afirma ser adaptable a cualquier proceso sin considerar si trabaja con las repeticiones, las fases o ambos. El único requisito es la posibilidad de inyectar sus actividades repetidamente en el proceso.

El incremento es ligero. Hay solamente dos actividades que constituyen el proceso: la Valoración del Proceso y las Evaluaciones de Sostenibilidad y Pre visualizaciones. El enfoque de la Valoración de Proceso es la sostenibilidad del proceso de desarrollo de software mismo. La sostenibilidad del producto de software es dirigida con Evaluaciones y Pre visualizaciones. [4] Ambas actividades dependen de un modelo de ciclo de vida esquemático, motivado por la valoración del ciclo de vida que denota los diferentes impactos y los efectos del software sobre el desarrollo sostenible que debe ser dirigido.

El proceso distingue entre tres roles: [4] el Representante del Cliente, el Equipo de Desarrollo, y el Ejecutivo de Sostenibilidad. El Representante del Cliente representa la organización o los usuarios que financian y usan el software. El Equipo de Desarrollo es responsable de desarrollar el software, y de mejorar el software con respecto a los requisitos no funcionales relacionados con la sostenibilidad. El Ejecutivo de Sostenibilidad es responsable de la continua Valoración del Proceso, organizando las Evaluaciones y las Pre visualizaciones, y haciendo un informe final con las cifras claves y conclusiones de las Evaluaciones, Pre visualizaciones y Valoraciones del Proceso.

La Valoraciones del Proceso es una actividad ininterrumpida junto al proceso de desarrollo de software. Esta trata de recolectar y editar datos desde el proceso que pueden ser usados para calcular la huella de carbón o incluso para la valoración del ciclo de vida. Los siguientes

datos pueden ser monitoreados y recolectados: Datos que tratan la energía usada en las oficinas de los desarrolladores, la infraestructura del IT, monitorear los medios de transporte para los viajes de negocios y para desplazarse diariamente al lugar de trabajo, tanto como la cantidad de artículos de papelería usados, por ejemplo papel para informes. Durante el proceso de desarrollo de software, la Valoración del Proceso debe presentar un pequeño reporte a Evaluaciones y Pre visualizaciones para mantener informados a los desarrolladores. [4]

Las Evaluaciones de sostenibilidad y los Pre visualizaciones se concentran en el software bajo desarrollo para optimizarlo en relación con sus impactos y efectos sobre el desarrollo sostenible. [4] Durante el desarrollo, varias evaluaciones son realizadas, por ejemplo después de dos tercios de una iteración, lo cual establece un ciclo continuo de mejoras. En las revisiones, se examina el trabajo reciente en busca de problemas de sostenibilidad y luego se preparan soluciones junto con la tasa de mejora esperada, que es la parte de pre visualización. En la siguiente sesión de revisión y pre visualización, estas soluciones se evalúan para determinar si los problemas se han abordado o no. Por lo tanto, los desarrolladores necesitan métodos y cifras para explorar estos problemas.

Es propuesto un método para medir la eficiencia energética del software el cual está basado en Pruebas de Unidad. [4] Otros métodos también pueden ser apropiados, porque desde la perspectiva del ciclo de vida, el consumo de energía es una sola cuestión. Otros aspectos son los recursos y requerimientos del hardware, accesibilidad, o incluso efectos del uso, como la sustitución, desmaterialización, y los efectos de rebote. Por lo tanto, las Evaluaciones y Pre visualizaciones deben empezar en etapas tempranas del proceso cuando el producto de software es definido. El aporte de la Evaluación del Proceso debe ser considerado si hay problemas que pueden ser resueltos por los propios asistentes.

Al final del proyecto de desarrollo de software, el informe incluye los resultados y las claves encontradas por la Valoración del Proceso (tal vez los cálculos de la huella de carbón) tanto como las evaluaciones y Pre visualizaciones deben ser entregados a los Representantes del Cliente. Además, los resultados sobresalientes, las lecciones aprendidas y las mejores prácticas deben ser preservados en una base de conocimiento para estar disponibles para

otros proyectos. [4]

2.4.4- GreenRM: Modelo de Referencia para el Desarrollo de Software Sostenible

GreenRM es un modelo de referencia basado en los conceptos de madurez y capacidad y fue desarrollado sobre la base de dos pilares, el marco del método de PRO2PI-MFMOD y el estándar medioambiental ISO/IEC 14001. El PRO2PI-MFMOD fue usado para establecer un método de construcción del modelo de referencia para el desarrollo del GreenRM. El ISO/IEC 14001 fue usado como un soporte para identificar procesos relevantes y practicas ya consolidadas. [30]

Los niveles verdes desarrollados en GreenRM son los niveles de madurez del modelo. Cada nivel de madurez tiene un grupo de procesos requeridos. Además, los procesos están también clasificados en tres categorías: procesos principales, procesos organizativos y procesos de mantenimiento. [30]

El primer nivel verde de madurez (G1-La sostenibilidad medioambiental es dirigida) define cuatro procesos: Dirección de la Solución Verde (GSM) (principal), Dirección del Centro de Datos (DCM) (principal), Dirección de Desechos Electrónicos (EWM) (principal) y Dirección de Riesgos Medioambientales (ERM) (soporte). El segundo nivel de madurez (G2-La sostenibilidad medioambiental es institucionalizada) define cuatro procesos: Adquisición Verde (GAQ) (organizativo), Mejora y Evaluación de la solución verde; (GEI) (soporte), Desarrollo del Software Verde (GSD) (principal), Instalación y Retiro del software verde (GIR) (principal). El tercer nivel verde de madurez (G3-La Sostenibilidad Medioambiental es constantemente mejorado) define tres procesos: Medición de la Solución Verde (GMS) (organizativo), Capacitación Organizacional para la Solución Verde (OTG) (organizativo), y Dirección de Conocimientos Medioambientales (EKM) (organizativo). [30]

El nivel G1 asegura una dirección ordenada de iniciativas medioambientales llevadas a cabo por la organización. Estas iniciativas son identificadas como soluciones verdes y son tratadas en el primer proceso del nivel. Para garantizar la gestión medioambiental de las soluciones específicas para el centro de procesamiento de datos y para el desperdicio electrónico, el segundo y tercer proceso del nivel son responsables de estos asuntos. Para ayudar a

identificar estas soluciones verdes, el último proceso del nivel permite el análisis de riesgos e impactos medioambientales. [30]

El nivel G2 asegura a los procesos medioambientales ser incorporados a los procesos organizativos. En este nivel, el proceso de auditoría es establecido para garantizar que las soluciones verdes sean llevadas a cabo en orden de satisfacer los objetivos medioambientales de la organización. Centrándose en el cumplimiento de los requisitos medioambientales relacionado con el ciclo de vida del software, son definidos dos procesos. Estos dos procesos se enfocan en las prácticas directas medioambientales relacionadas con las fases de desarrollo, instalación y retiro del software. [30] Todavía en este nivel, un proceso para adquisición verde es definido, exigiendo que la selección y evaluación de los proveedores estén basadas en los criterios medioambientales.

El nivel G3 permite analizar y medir los beneficios obtenidos de la aplicación de las soluciones verdes, a través de la medición de los conocimientos adquiridos con la implementación de las soluciones verdes.

Niveles de Capacitación: Para la dimensión de capacitación de GreenRM (la evaluación del modelo), se presentan tres niveles de capacitación y once atributos de procesos (PA) definidos por los niveles del uno al tres. La ISO/IEC 15504 y la ISO/IEC 14001 colaboraron en el desarrollo de estos atributos de proceso significativamente. La mayoría de estos atributos están relacionados con las necesidades de la dirección ecologista especificadas en la ISO / IEC14001. [30]

El nivel 0 no está incluido en ningún tipo de indicador por representa un proceso no implementado o uno defectuoso a la hora de lograr los resultados. El nivel de capacitación 1 representa que el proceso es dirigido: PA1- los logros del proceso son definidos en los resultados, PA2- es establecida la documentación formal del proceso, y se comunica a todos los involucrados, PA3- el proceso es planificado y monitoreado, PA4- los roles requeridos y las responsabilidades para llevar a cabo el proceso son identificadas, PA5- el personal que lleva a cabo el proceso es competente en base a una educación apropiada, entrenamiento y experiencia y PA6- el proceso es especificado en la política medioambiental de la organización. [30]

El nivel de capacitación 2 es definido como el proceso institucionalizado. Los atributos del proceso son evaluados entonces: PA7- un grupo de mejora del proceso es formalmente establecido para discutir y revisar el proceso continuamente, y PA8- los resultados del proceso son revisados en diferentes niveles jerárquicos de la organización, incluyendo el más alto nivel de dirección. Y finalmente, el nivel de capacitación 3 representa que el proceso es medido y valorado constantemente. Para valorar este nivel, los atributos de proceso son demostrados: PA9- los resultados conseguidos por el desempeño del proceso permiten demostrar los beneficios medioambientales, PA10- la medición del proceso es recolectada y analizada para respaldar la decisión a tomar sobre las soluciones medioambientales, y PA11- mejores oportunidades derivadas de innovaciones tecnológicas y soluciones medioambientales son identificadas, evaluadas y seleccionadas para sostener los objetivos de la empresa. [30]

Los atributos de procesos PA2, PA4, PA6, PA8 y PA10 fueron identificados del análisis de los requerimientos medioambientales exigidos por el estándar ISO/IEC 14001. Logrando estos atributos es posible garantizar la existencia de la política ambiental y la formalización de procesos en esta política, definiendo los papeles y las responsabilidades involucradas, el análisis de la administración en la toma de decisiones, también como las mediciones para identificar las acciones correctivas y preventivas. [30]

El GreenRM puede ser usado como guía para la certificación ambiental y para la solicitud de las prácticas de TIC sostenible. El objetivo principal no es solamente las prácticas de soporte relacionadas con la infraestructura si no también mejorar la sostenibilidad ambiental en el proceso de desarrollo de software. [30]

2.5 – Comparación entre los modelos de procesos

A continuación, se hace una comparación de los modelos anteriores considerando sus diferentes fases, iteraciones, roles definidos y actividades correspondientes. Se comparan los factores que afectan la sostenibilidad explícitamente dentro de los requisitos de la ingeniería.

Tanto GSSE como GSEP implementan solamente la fase de Planificación del Proceso de Dirección de Sostenibilidad. GSEP define explícitamente diferentes fases secuenciales, incluyendo la fase especializada Análisis Verde, también como la posibilidad de tener varias

iteraciones. Es necesario planear el número de iteraciones y programar las diferentes actividades relacionadas con la fase de Análisis Verde. Lo mismo sucede con GSSE, donde las Evaluaciones y Vistas Previas deben ser programadas después de dos tercios de una iteración o fase. Más específicamente, el Ejecutivo de Sostenibilidad es responsable de organizar las actividades sostenibilidad central, que obviamente incluyen la planificación de tiempo tanto como la asignación de recursos. Para otras fases del proceso de referencia, la Fase Preliminar, Fase de Monitoreo y Control de sostenibilidad del proveedor, ningún equivalente puede ser detectado ni en GSSE ni en GSEP. Muy diferente sucede con el modelo GreenRM, el cual no tiene fases definidas, si no varios niveles de madurez, los cuales deben seguir un orden lógico guiándose por los diferentes procesos de cada nivel.

El Proceso de Ingeniería Sostenible es implementado por GSEP, GSSE y GreenRM, y todos proporcionan los resultados definidos. GSEP identifica los factores que afectan la sostenibilidad explícitamente dentro de los requisitos de la ingeniería, GSSE referencia un modelo de Ciclo de Vida del producto de software que tiene el único propósito de compaginar efectos ejemplares e impactos en el desarrollo sostenible con las fases del Ciclo de Vida desde el Análisis del Ciclo de Vida (la fabricación, la distribución, el uso, final de la vida)y GreenRM asegura una dirección ordenada de iniciativas medioambientales llevadas a cabo por la organización desde el nivel 1. Todos efectúan el análisis de sostenibilidad para determinar el impacto de los factores de sostenibilidad. GSEP tiene los Requisitos y la fase de Análisis Verde, GSSE tiene las Evaluaciones y las Vistas Previas tanto como la Valoración de Proceso y GreenRM tiene la Medición de la Solución Verde (GMS).

GSEP explícitamente define los objetivos de sostenibilidad en Requisitos, mientras que GSSE menciona los objetivos de sostenibilidad implícitamente junto con el Equipo de Desarrollo y GreenRM asegura a los procesos medioambientales ser incorporados a los procesos organizativos en el nivel 2. Respecto a las técnicas y los métodos para la sostenibilidad, GSEP define un grupo grande de métricas y herramientas en su segundo nivel. GSSE menciona un cálculo de huella de carbón y especialmente los métodos y las métricas para medir y tasar la eficiencia de energía del software. GreenRM define un plan de capacitación exhaustivo, que incluye desde definir los papeles y las responsabilidades

involucradas, hasta el análisis de la administración en la toma de decisiones, como las mediciones para identificar las acciones correctivas y preventivas.

El proceso de referencia requiere un análisis de la sostenibilidad de las demandas de cambio, que no es implementado por GSEP o GSSE específicamente. Sin embargo, ambos procesos afirman ser ágiles y abiertos a los cambios. GSEP puede analizar pedidos de cambio en Requisitos o el Análisis Verde; GSSE puede manejar pedidos de cambio en evaluaciones y pre visualizaciones. GSSE define un grupo de roles los cuales juegan un papel predeterminado en el proceso, lo cual le brinda un nivel superior de organización a este. Esto no sucede en los otros modelos, ya que estos se refieren a los roles como si ya estuvieran definidos.

Tanto GSEP como GSSE se centran en la implementación de un Proceso de Ingeniería Sostenible. El Proceso de Capacitación de sostenibilidad no existe en ninguno de los modelos anteriores, existiendo en el GreenRM. En cuanto al Proceso de Dirección Sostenible, solo la fase de Planificación puede ser implementada rudimentariamente. GSSE no fue diseñado con el proceso de Lami en mente, GSEP menciona a Lami y podía haber sido diseñado para reconocer más aspectos del proceso de referencia. En particular, GSSE debe describir cómo son obtenidos los objetivos de sostenibilidad o que rol debe determinarlos. Resumiendo, GSEP y GSSE cubren la mayoría de los aspectos del Proceso de Ingeniería Sostenible, mientras que los aspectos de otros procesos están casi totalmente faltantes.

2.6 – Conclusiones

Se seleccionan los cuatro modelos más referenciados en la literatura científica internacional, para realizar un análisis de los mismos en cuanto a su funcionamiento, sus niveles o fases, los roles, actividades que realizan, entre otros. Se realiza una comparación entre los modelos considerando los factores que afectan la sostenibilidad, específicamente dentro de los requisitos de la ingeniería del software.

3 – Propuesta de modelo de procesos de ingeniería de software sostenible para la empresa cubana

3.1- Introducción

En este capítulo se realiza una caracterización del uso del software sostenible en las empresas del territorio a partir de las encuestas aplicadas a sus trabajadores y se propone un modelo de procesos adaptado a las necesidades de las empresas cubanas basándonos en el Modelo de Proceso para una Ingeniería de Software Ágil y Sostenible presentado por Stefan Naumann, Eva Kern, Markus Dick y Timo Johann.

3.2- Caracterización del uso del software sostenible en empresas del territorio.

Una encuesta es un procedimiento dentro de los diseños de una investigación descriptiva en el que el investigador busca recopilar datos por medio de un cuestionario previamente diseñado, sin modificar el entorno ni el fenómeno donde se recoge la información ya sea para entregarlo en forma de tríptico, gráfica o tabla. Los datos se obtienen realizando un conjunto de preguntas normalizadas dirigidas a una muestra representativa o al conjunto total de la población estadística en estudio, integrada a menudo por personas, empresas o antes institucionales, con el fin de conocer estados de opinión, ideas, características o hechos específicos. [41]

Para seleccionar el cuestionario a aplicar se revisaron trabajos e investigaciones que se habían centrado en la problemática del desarrollo sostenible y el desarrollo de software sostenible, y propuestas de instrumentos elaborados por diferentes autores e instituciones. Para diagnosticar el conocimiento de los trabajadores de las empresas de la localidad sobre la sostenibilidad del software, y la situación real del software sostenible en dichas empresas se elaboró una encuesta basada en los criterios y métricas planteados por Stefan Naumann, Eva Kern, Markus Dick y Timo Johann en su Modelo de Calidad para Software Sostenible. (Anexo 1)

En el caso de la población de estudio se corresponde con trabajadores de las diferentes empresas seleccionadas (ETECSA, DESOFT, Refinería de petróleo Camilo Cienfuegos y la Universidad de Cienfuegos). La muestra queda conformada por 39 trabajadores

distribuidos en las diferentes empresas. Entre los universitarios se encuentran graduados de Ingeniería Informática, Mecánica, Automática y Cibernética. Todos los trabajadores tienen acceso directo y diario a diferentes tipos de software.

Luego de aplicadas las encuestas a la muestra seleccionada, se realiza su procesamiento a través del paquete estadístico SPSS en su versión 22 [42], los resultados se presentan en tablas y gráficos con la finalidad de analizar, interpretar e inferir sobre la sostenibilidad del software en la empresa cubana actual.

3.3- Resultados de la encuesta aplicada

Las preguntas iniciales son de tipo abierta y se enfocan fundamentalmente en los conocimientos generales sobre sostenibilidad, desarrollo sostenible y software sostenible. En estas preguntas el 90% de los encuestados de las empresas demuestran no conocer los términos sostenibilidad y desarrollo sostenible, solo el 10% dio respuestas relativamente acertadas sobre estos términos.

La pregunta 4 está dirigida a conocer la posición de la empresa ante los principios de sostenibilidad. El 43.6% de los trabajadores creen que la empresa respeta los principios de sostenibilidad, mientras que un 48.7% cree que no y solo un 7.7% no conoce del tema. Los encuestados que contestaron afirmativamente consideran que la sostenibilidad no representa actualmente un requisito en la empresa ni se posiciona como un tema de prioridad, pero que sí lo será en un futuro cercano.

	Frecuencia	Porcentaje
Válido Si	17	43,6
No	19	48,7
No se	3	7,7
Total	39	100,0

Tabla 4: Respuesta Pregunta 4

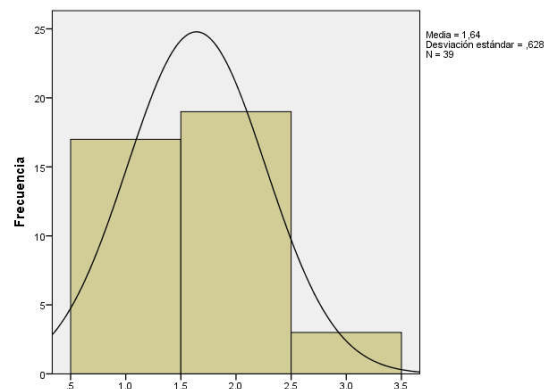


Figura 6: Histograma Respuesta Pregunta 4

En la pregunta 5 el 84.6% de los trabajadores consideran que el software utilizado en la empresa es adecuado a sus propósitos de operación específicos, mientras que un 2.6% cree que no y solo un 12.8% no conoce. De las respuestas negativas, el 100% de los encuestados explica que actualmente la empresa está migrando a software que se adecuen a sus funciones, pero que este proceso se desarrolla lentamente.

		Frecuencia	Porcentaje
Válido	Si	33	84,6
	No	1	2,6
	No se	5	12,8
	Total	39	100,0

Tabla 5: Respuesta Pregunta 5

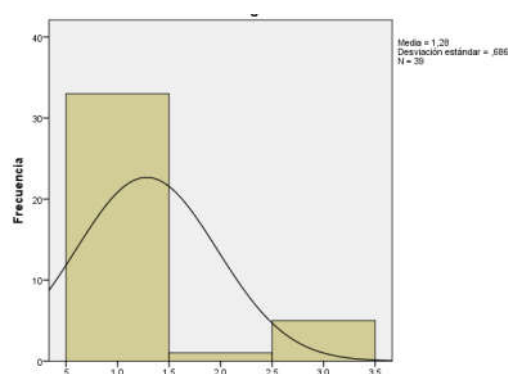


Figura 7: Histograma Respuesta Pregunta 5

En la pregunta 6, el 12.8% de los trabajadores consideran que el software utilizado produce algún desperdicio dentro de su dominio de aplicación, considerando como

desperdicio el espacio que ocupa el sistema con informaciones obsoletas o los documentos impresos como afectación al medio ambiente por el consumo de papel. Un 66.7% cree que no produce desperdicio alguno y solo un 20.5% no sabe, argumentando además no comprender como un software puede ocasionar algún desperdicio.

	Frecuencia	Porcentaje
Válido Si	5	12,8
No	26	66,7
No se	8	20,5
Total	39	100,0

Tabla 6: Respuesta Pregunta 6

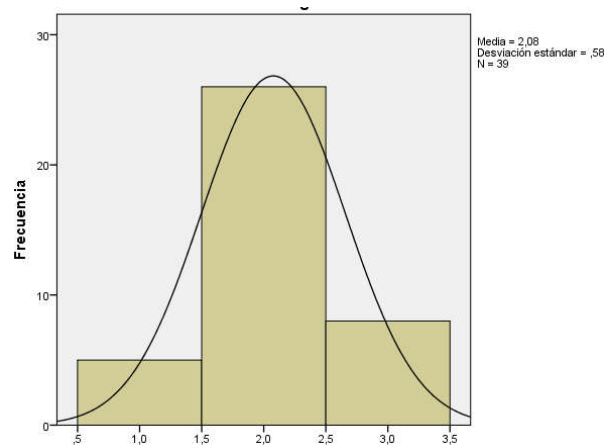


Figura 8: Histograma Respuesta Pregunta 6

En la pregunta 7, el 41% de los trabajadores consideran que el software utilizado en la empresa cumple con los principios de software sostenible, un 17,9% cree que no y un 42% no sabe. Esta respuesta coincide con la dada en el pregunta 3, en la que solo el 42% de los encuestados afirma conocer que es un software sostenible.

	Frecuencia	Porcentaje
Válido Si	16	41.02
No	7	17,9
No se	16	41.02
Total	39	100,0

Tabla 7: Respuesta Pregunta 7

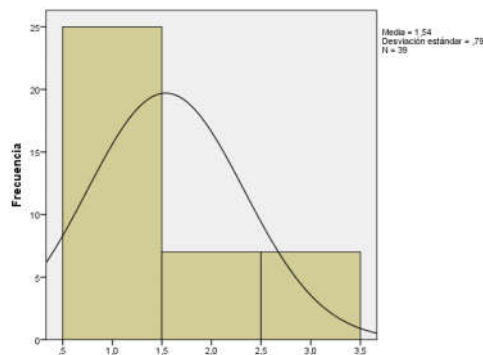


Figura 9: Histograma Respuesta Pregunta 7

En la pregunta 8, el 71.8% de los trabajadores consideran que el software funciona rápido, o sea se ejecuta en un tiempo prudente, un 12.8% cree que no porque las computadoras utilizadas son muy lentas y un 15.4% no sabe. Los que contestaron positivamente se basan en los tiempos de respuesta de los servidores y las tecnologías utilizadas en la empresa.

	Frecuencia	Porcentaje
Válido Si	28	71,8
No	5	12,8
No se	6	15,4
Total	39	100,0

Tabla 8: Respuesta Pregunta 8

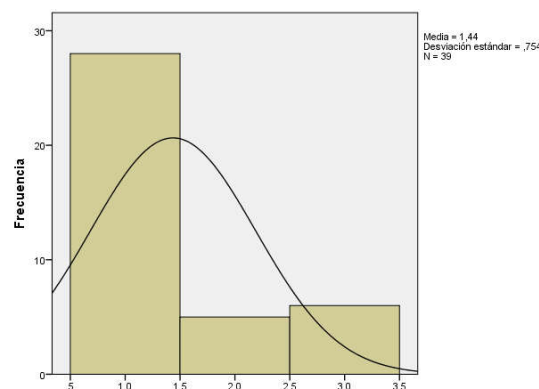


Figura 10: Histograma Respuesta Pregunta 8

En la pregunta 9, el 79.5% de los trabajadores consideran que el hardware utilizado permite la correcta ejecución del software y un 20.5% cree que no, planteando que las computadoras disponibles para trabajar son de bajos recursos.

	Frecuencia	Porcentaje
Válido Si	31	79,5
No	8	20,5
Total	39	100,0

Tabla 9: Respuesta Pregunta 9

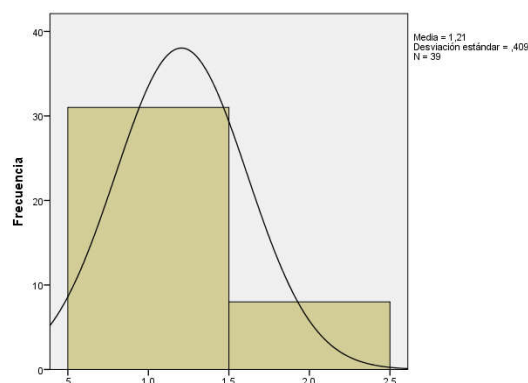


Figura 11: Histograma Respuesta Pregunta 9

En la pregunta 10, los trabajadores utilizan el software entre 4 y 8 horas aproximadamente. Sin embargo los encuestados que son administradores de red plantean que en los servidores el software funciona 24 horas.

N	Válido	39
	Perdidos	0
Media		7,05

Tabla 10: Respuesta Pregunta 10

La pregunta 11 está enfocada en el tamaño del software considerando el número de operaciones que realiza, los encuestados desconocen esta información en el 100% de las respuestas.

En la pregunta 12, el 33.3% de los trabajadores consideran el costo de los viajes de negocios asociados al desarrollo del software y los viajes diarios al centro de trabajo son altos, un 46.2% cree que no lo son, y un 20.5% no sabe. Estas respuestas varían en función de la organización de cada empresa, puesto que no todas otorgan viáticos para viajes y otras garantizan el transporte de sus trabajadores.

	Frecuencia	Porcentaje
Válido Si	13	33,3
No	18	46,2
No se	8	20,5
Total	39	100,0

Tabla 11: Respuesta Pregunta 12

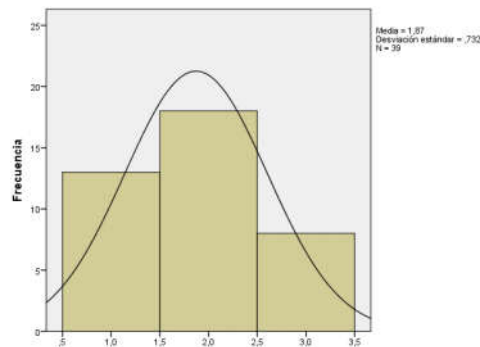


Figura 12: Histograma Respuesta Pregunta 12

En la pregunta 13, el 23.1% de los trabajadores consideran que durante el ciclo de vida del software ocurre alguna emisión de CO2 a causa del desarrollo y uso de este, un 66.7% cree que no, y un 10.3% no sabe.

	Frecuencia	Porcentaje
Válido Si	9	23,1
No	26	66,7
No se	4	10,3
Total	39	100,0

Tabla 12: Respuesta Pregunta 13

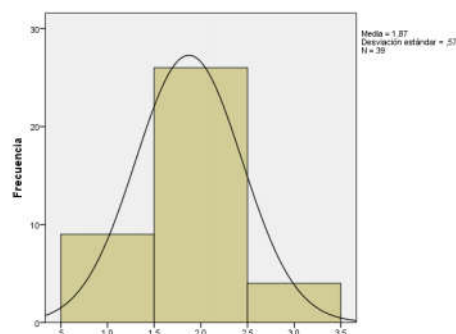


Figura 13: Histograma Respuesta Pregunta 13

En la pregunta 14, el 20.5% de los trabajadores consideran que la cantidad de energía consumida en el desarrollo del software es alta, un 64.1% cree que no, y un 15.4% no

sabe. Esto puede deberse a que la mayoría de los encuestados no se dedican directamente al desarrollo de software.

	Frecuencia	Porcentaje
Válido Si	8	20,5
No	25	64,1
No se	6	15,4
Total	39	100,0

Tabla 13: Respuesta Pregunta 14

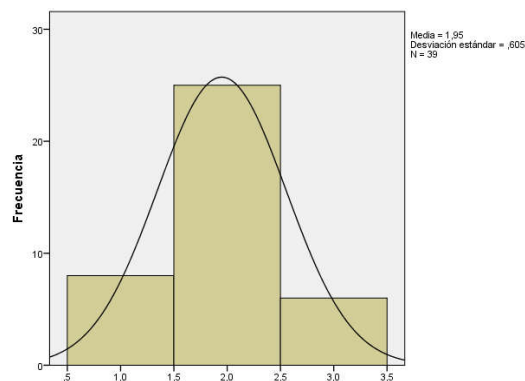


Figura 14: Histograma Respuesta Pregunta 14

En la pregunta 15, el 33.3% de los trabajadores consideran la cantidad de recursos consumidos durante el desarrollo del software por actividades que no tienen un valor adicional para consumidores o usuarios finales es alta, un 64.1% cree que no, y un 2.6% no sabe.

	Frecuencia	Porcentaje
Válido Si	13	33,3
No	25	64,1
No se	1	2,6
Total	39	100,0

Tabla 14: Respuesta Pregunta 15

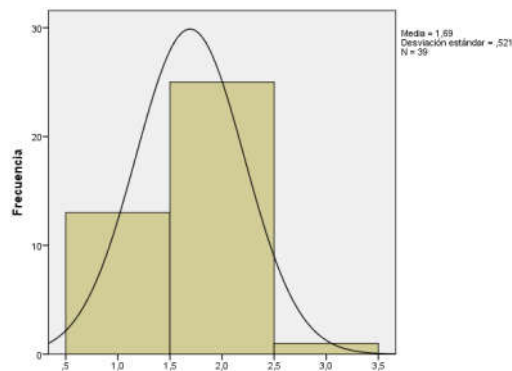


Figura 15: Histograma Respuesta Pregunta 15

En la pregunta 16, el 66.7% cree que el software puede ser usado por la mayor cantidad de persona posibles sin restricciones y un 33.3% cree que no. Se debe considerar que los encuestados en la mayoría de los casos son graduados de nivel superior y en carreras afines a la informática, lo que permite que tengan una mayor capacidad para trabajar con software.

	Frecuencia	Porcentaje
Válido Si	26	66,7
No	13	33,3
Total	39	100,0

Tabla 15: Respuesta Pregunta 16

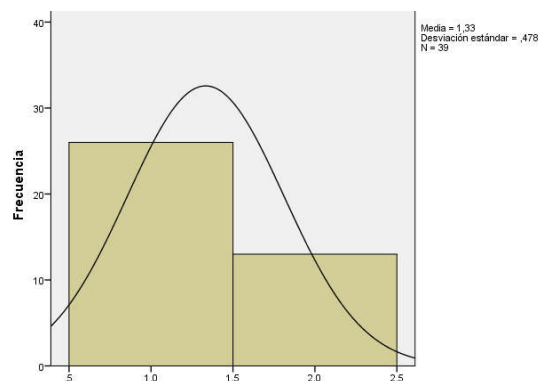


Figura 16: Histograma Respuesta Pregunta 16

En la pregunta 17, el 87.2% cree que el software es entendible, operable y de fácil aprendizaje y un 12.8% cree que no.

	Frecuencia	Porcentaje
Válido Si	34	87,2
No	5	12,8
Total	39	100,0

Tabla 16: Respuesta Pregunta 17

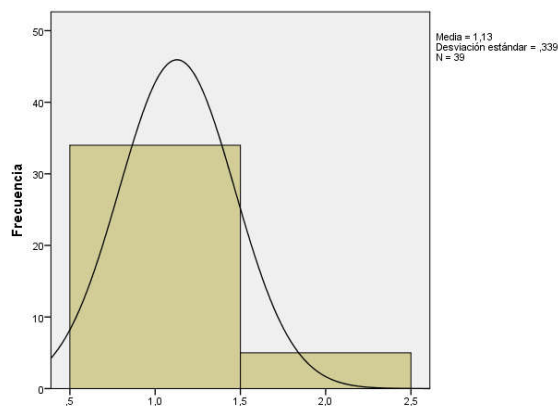


Figura 17: Histograma Respuesta Pregunta 17

En la pregunta 18, el 56.4% opina que las condiciones de trabajo en general en su empresa son buenas, un 41% cree que no y un 2.6% no sabe.

	Frecuencia	Porcentaje
Válido Si	22	56,4
No	16	41,0
No se	1	2,6
Total	39	100,0

Tabla 17: Respuesta Pregunta 18

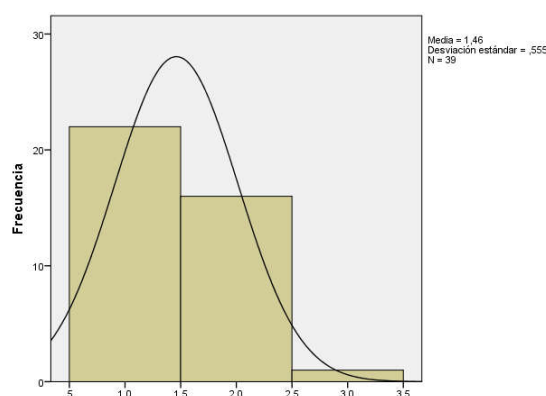


Figura 18: Histograma Respuesta Pregunta 18

En la pregunta 19, el 53.8% opina que el hardware utilizado en la empresa esta obsoleto, un 41% cree que no y un 5.1% no sabe.

	Frecuencia	Porcentaje
Válido Si	21	53,8
No	16	41,0
No se	2	5,1
Total	39	100,0

Tabla 18: Respuesta Pregunta 19

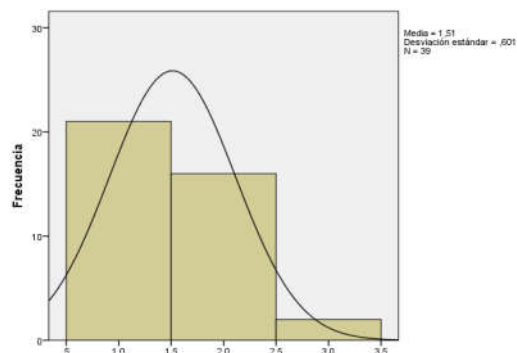


Figura 19: Histograma Respuesta Pregunta 19

En la pregunta 20, el 84.6% opina que el software es adaptable a las necesidades específicas del usuario y el 15,4% cree que no.

	Frecuencia	Porcentaje
Válido Si	33	84,6
No	6	15,4
Total	39	100,0

Tabla 19: Respuesta Pregunta 20

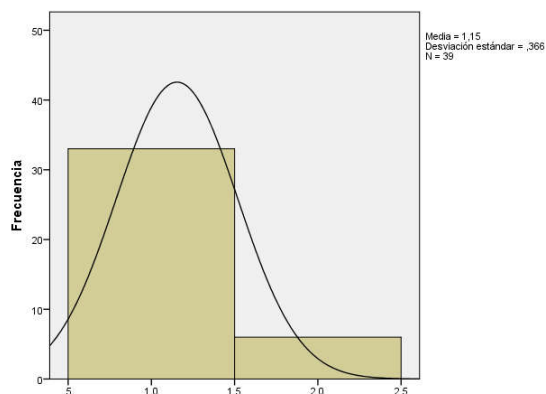


Figura 20: Histograma Respuesta Pregunta 20

En la pregunta 21, el 61.5% opina que el software para su funcionamiento consulta datos ubicados en librerías externas u otro software, el 28.2% cree que no y el 10.3%

no sabe. Sin embargo ningún encuestado considera que este indicador sea importante, ya que las librerías externas o software que se consultan pueden trabajar de forma independiente.

	Frecuencia	Porcentaje
Válido Si	24	61,5
No	11	28,2
No se	4	10,3
Total	39	100,0

Tabla 20: Respuesta Pregunta 21

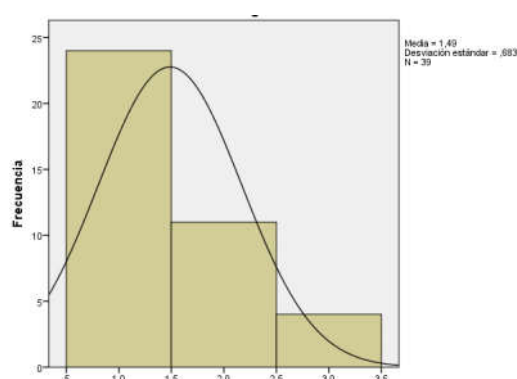


Figura 21: Histograma Respuesta Pregunta 21

En la pregunta 21A, el 5.1% sabe cuánta energía se consume al usar el software y el 94.9% no sabe. Se debe considerar que en las empresas cubanas disminuir el consumo de energía es una prioridad, y sin embargo los encuestados no relaciona el uso de software con este propósito.

	Frecuencia	Porcentaje
Válido Si	2	5,1
No	37	94,9
Total	39	100,0

Tabla 21: Respuesta Pregunta 21A

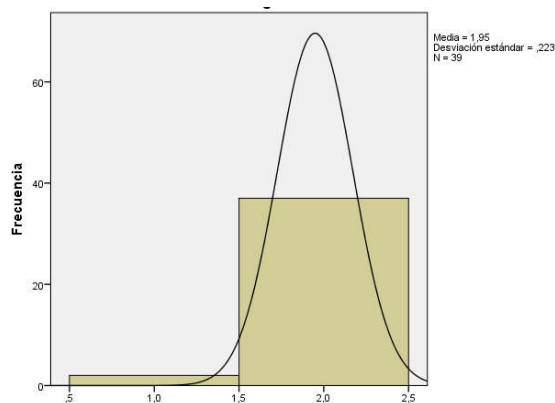


Figura 22: Histograma Respuesta Pregunta 21A

3.4- Modelo de Procesos

El modelo de procesos propuesto está basado en el Modelo de Proceso para una Ingeniería de Software Ágil y Sostenible presentado por Stefan Naumann, Eva Kern, Markus Dick y Timo Johann en 2015 el cual a su vez está basado en el modelo GREENSOFT. Se escoge este modelo de procesos como base porque en este se define un grupo de roles y sus tareas en cada fase del proceso, lo que permite una mayor organización a lo largo de todo el desarrollo. También muestra los aspectos cualitativos de la ingeniería de software y su objetivo principal es hacer que el software por sí mismo sea tan sostenible como el proceso de ingeniería.

Al GSSE se le realizarán algunas modificaciones para adaptarlo a las necesidades y particularidades de la empresa cubana actual, basándose en los criterios y métricas planteados en el Modelo de Calidad para un Software Sostenible y en los resultados de las encuestas aplicadas a los informáticos de las empresas de la localidad. [4]

3.4.1- Modelo de Procesos adaptado a la Empresa Cubana Actual

El modelo propone tres actividades fundamentales que constituyen el proceso: Capacitación Organizacional, la Valoración del Proceso y las Evaluaciones de Sostenibilidad y Pre Visualizaciones. La Capacitación Organizacional es un plan de capacitación tanto para trabajadores como directivos de la organización basado en los niveles de capacitación planteados por Marcello Thiry y Liliane Frez en el GreenRM Model, y adaptado a las particularidades de la empresa cubana actual. El enfoque de la Valoración de Proceso es la sostenibilidad del proceso de desarrollo de software

mismo. La sostenibilidad del producto de software es dirigida con Evaluaciones y Pre visualizaciones.

El modelo propone tres fases fundamentales a lo largo del ciclo de vida del software:

Fase de Desarrollo, Fase de Uso y Fase de Fin de la Vida.

Fase de Desarrollo: esta fase explica los impactos sobre el desarrollo sostenible que resultan directamente de las actividades involucradas en el desarrollo de software, así como las actividades indirectamente implicadas.

La Fase de Distribución, que pertenece a la **Fase de Desarrollo**, explica los impactos sobre el desarrollo sostenible que resultan de distribuir el producto de software.

Fase de Uso: Esta fase considera los impactos que resultan de desplegar, usar, y mantener actualizado el producto de software.

Fase Fin de la Vida: Si un producto de software es sacado del servicio, es necesario cambiar los datos disponibles por un formato que puede ser procesado por el producto de software siguiente, o hacerlo accesible de alguna otra manera. Si los datos no pueden ser convertidos fácilmente, por ejemplo porque es guardado en un formato de datos reservado, esto podría tener un impacto sobre la sostenibilidad económica de una organización.

La Fase de Traspaso, que pertenece a la **Fase de Fin de la Vida**, explica los impactos sobre el desarrollo sostenible que resultan del traspaso y el reciclaje de los subproductos materiales mencionados en las **Fases de Desarrollo y Uso**.

El proceso distingue cuatro roles:

1. El Representante del Cliente: representa la organización o los usuarios que financian y usan el software.
2. El Encargado de Capacitación: es responsable de llevar a cabo la Capacitación Organizacional.
3. El Equipo de Desarrollo: es responsable de desarrollar el software, y de mejorar el software con respecto a los requisitos no funcionales relacionados con la sostenibilidad.
4. El Ejecutivo de Sostenibilidad: es responsable de la continua Valoración del Proceso, organizando las Evaluaciones y las Pre visualizaciones, y haciendo un

informe final con las cifras claves y conclusiones de las Evaluaciones, Pre visualizaciones y Valoraciones del Proceso.

Para la dimensión de capacitación del modelo adaptado se presentan dos niveles de capacitación y 9 atributos de procesos definidos. La mayor parte de estos atributos están relacionados con las necesidades de la empresa cubana, y se basan en las detectadas a través de la encuesta a las empresas de la localidad.

El nivel de capacitación 1 está dirigido a la organización del proceso:

1. Se instruye a los trabajadores sobre los temas de sostenibilidad, desarrollo sostenible y software sostenible con el objetivo de alcanzar un conocimiento general y un nivel de concientización sobre el mismo.
2. Se lleva a cabo un proceso de instrucción a todos los niveles jerárquicos de la empresa, con el fin de instruir al personal sobre los criterios y métricas que miden la sostenibilidad del software a lo largo su ciclo de vida.
3. Se establece la documentación del proceso y se informa a todos los involucrados, en este caso el Representante del Cliente, el Equipo de Desarrollo, y el Ejecutivo de Sostenibilidad.
4. Se lleva a cabo la planificación del proceso de ingeniería.
5. Se asegura que el personal que lleva a cabo el proceso es competente en base a una educación apropiada, entrenamiento y experiencia.
6. Se especifica el proceso en la política medioambiental de la empresa.

El nivel de capacitación 2 es definido como el proceso institucionalizado:

1. Se toman los criterios de calidad de software existentes o modelos de calidad, y se interpretan en el contexto de la sostenibilidad ambiental de la empresa.
2. Un grupo de mejora del proceso es formalmente establecido para discutir y revisar el proceso.
3. Los resultados del proceso son revisados en diferentes niveles jerárquicos de la organización, incluyendo el más alto nivel de dirección.
4. La medición del proceso es recolectada y analizada para respaldar la decisión a tomar sobre las soluciones medioambientales.

La Valoraciones del Proceso recolecta y edita datos desde el proceso que pueden utilizarse para calcular la huella de carbón o el consumo de energía y se realiza en la **Fase de Desarrollo**. Los siguientes datos pueden ser monitoreados y recolectados:

- Infraestructura de las tecnologías, que las condiciones de las maquinas usadas sean óptimas o al menos que se encuentren en buen estado.
- Intensidad de la CPU de las máquinas usadas para desarrollar el software.
- Eficiencia en el tiempo de corrida del software.
- Energía que es necesaria para aire acondicionado o para luces de las oficinas.
- Energía eléctrica que es necesaria para suministrarla a las estaciones de trabajo de desarrolladores y otros empleados.
- Monitorear los medios de transporte para los viajes de negocios y para desplazarse diariamente al lugar de trabajo.
- Cantidad de recursos usados durante el desarrollo del software como papel, bolígrafos, tinta para impresoras. El software especializado para la captura de datos disminuye el uso excesivo de papel en las oficinas, salvando así millones de árboles al año.
- Eficiencia energética que tiene como objetivo optimizar el consumo de energía en relación con los servicios dados por el sistema. Para lograr esto se podría rebajar la cantidad de servicio requerido hasta un nivel que todavía es aceptable para los usuarios o dependiendo del área de solicitud.

Los impactos sociales pueden ser las condiciones de trabajo y el pago a los trabajadores, lo cual tiene consecuencia tanto para los trabajadores como para sus familias. La fase de desarrollo también explica los impactos del mantenimiento de software en el sentido de solucionar problemas de este, porque esto es también una actividad de desarrollo de software y debe pertenecer a la fase de desarrollo.

También se deben medir los impactos sobre el desarrollo sostenible que resultan de distribuir el producto de software, como son:

- Manuales impresos (tipo de papel y tinta).

- Los medios de transporte elegidos, tipos y diseños del embalaje a la hora de transportar (por ejemplo plástico, espuma de poliuretano, material biodegradable)
- Medios de almacenamiento de datos (por ejemplo CD / DVD, memoria USB).
- Además, si el producto de software es ofrecido como una descarga, que es común hoy, entonces el tamaño de la descarga debe ser considerado, igual que la energía eléctrica y los recursos materiales que son necesarios para operar la infraestructura de las TIC.

Durante la **Fase de Desarrollo** de software, la Valoración del Proceso debe presentar un pequeño reporte a Evaluaciones y Pre visualizaciones para mantener informados a los desarrolladores.

Las Evaluaciones de sostenibilidad y las Pre visualizaciones se concentran en el software bajo desarrollo para optimizarlo en relación con sus impactos y efectos sobre el desarrollo sostenible. Se realizan varias evaluaciones durante el desarrollo lo cual asegura un ciclo continuo de mejoras. En las revisiones, se examina el trabajo reciente en busca de problemas de sostenibilidad como pueden ser:

- El número de métodos del software que expresa el tamaño de la aplicación.
- La entropía.
- La accesibilidad.
- Usabilidad.
- Portabilidad.
- Que el software se ajuste al propósito para el que está siendo creado.
- Que dentro de su dominio de aplicación reduzca desperdicios.

Luego se preparan soluciones junto con la tasa de mejora esperada, que es la parte de pre visualización. En la siguiente sesión de evaluación y pre visualización, se vuelve a los problemas planteados en la anterior sesión y se evalúa si se han abordado o no.

Al finalizar el proceso de desarrollo de software, se debe realiza un informe que incluya los resultados encontrados por la Valoración del Proceso, tanto como las evaluaciones y Pre visualizaciones. Este debe ser entregado al Representante del Cliente.

En la **Fase de Uso** los informáticos de las empresas son los encargados de instalar el

software y darles soporte a los usuarios en sus organizaciones. Por lo tanto, mantener actualizado incluye, por ejemplo la instalación de las actualizaciones del software, la configuración de software y sistemas de computadora, y capacitar a los empleados con respecto al uso de software siguiendo el plan de capacitación planteado anteriormente. Usuarios adecuadamente capacitados podrían necesitar menos tiempo para cumplir sus tareas, lo que resulta en menos consumo de energía, configurar el sistema para que consuma menos energía o solo cambiar la computadora a modo Suspendido cuando no se vaya a usar.

Para entregar sus servicios, un programa de computadora requiere un tiempo de procesamiento, que consume energía eléctrica. Este también puede requerir servicios ofrecidos por otros servidores (por ejemplo, sistemas de dirección de base de datos, sistemas de planificación de recurso de la empresa, o sólo el servicio de WWW), que causa el consumo de energía adicional. Además, la estrategia de actualización de un producto de software (por ejemplo tamaño y frecuencia de las actualizaciones) influyen en la transferencia de datos, el procesamiento, y la infraestructura de las TIC, que son necesarias para repartir las actualizaciones. Combinadas, éstas causan un consumo de energía y recursos adicionales.

En esta fase también se deben medir los indicadores siguientes:

- Inactividad, en caso de los software que se encuentran activos 24 horas, como es el caso de los servidores.
- Obsolescencia del Hardware, que quiere decir que el uso del hardware debe ser el final de su vida útil, en lugar de su sustitución temprana debido a requisitos impuestos por el software como es el cambio de hardware debido a que el sistema operativo evolucione y este quede obsoleto. En este caso se puede usar un sistema operativo que funcione bien independientemente de la arquitectura del hardware. Algunos expertos afirman que las computadoras de mesa deben tener una vida útil de hasta cinco años y las usadas como servidores 1 año, lo que claramente no ocurre en nuestras empresas debido a la falta de recursos y el bloqueo económico al que nos encontramos sometidos.
- Infraestructura de la empresa, que tenga las condiciones necesarias para que el

software pueda ser utilizado, que las maquinas se encuentren en buenas condiciones y el correcto cableado de la red. La situación social de la empresa también debe ser tenida en cuenta.

Para medir el consumo de energía o la eficiencia energética se pueden usar dos métodos, el primero es el planteado por Dick [31] basado en pruebas de unidad, y el segundo el publicado este año 2018 por Achim Guldner, Marcel Garling, Marlies Morgen, Stefan Naumann, Eva Kern y Lorenz M. Hilty, que plantea como medir el consumo de energía que el software induce al hardware. [32]

Para integrar aspectos de la huella de carbón y la eficiencia energética se puede utilizar el artículo de “Integrando Aspectos de la Huella de Carbón y mediciones Continuas de Eficiencia Energética en una Ingeniería de Software Sostenible” publicado en 2013. [43] Para calcular la huella de carbón se calcula sumando la totalidad de los Gases del efecto Invernadero emitidos de forma directa o indirecta por la actividad del proceso y se expresa en masa de CO₂ equivalente. Una vez se conoce la huella de carbón es posible poner en práctica una estrategia de reducción y/o de compensación de emisiones. La Norma ISO 14067 establece un marco de referencia internacionalmente reconocido para el cálculo de la Huella de carbón de un producto.

Las empresas cubanas, en este caso las de la localidad, que utilizan software en su día a día, deben comprometerse a la responsabilidad ambiental y social y a la aplicación de sistemas de dirección ambientales. Estos compromisos también deben cubrir los estándares ambientales y sociales en toda la cadena de suministro de productos y servicios, lo cual es necesario para producir, anunciar, distribuir, y reciclar el producto de software.

3.5- Sistema de Conocimiento de Referencias

Se propone un sistema de conocimientos con el objetivo de apoyar y guiar la formación de los informáticos del territorio en el desarrollo y uso de software sostenible. El mismo se organiza de acuerdo a la distribución por fases del modelo propuesto.

1. The GREENSOFT Model: A reference model for green and sustainable software and its engineering de Stefan Naumann, Markus Dick, Eva Kern, Timo Johann.
2. Process-Centric Software Sustainability de Lami.
3. Green Model for Sustainable Software Engineering de Mahmoud and Ahmad.
4. The Software Process Enhancement for Agile and Sustainable Software Engineering de Stefan Naumann, Markus Dick, Eva Kern, Timo Johann.
5. Sustainable Software Engineering: Process and Quality Models, Life Cycle, and Social Aspects de Stefan Naumann, Eva Kern, Markus Dick y Timo Johann.
6. GreenRM: Reference Model for Sustainable Software Development de Marcello Thiry, Liliane Frez y Alessandra Zoucas
7. Enhancing Software Engineering Processes towards Sustainable Software Product Design de Markus Dick y Stefan Naumann.
8. What does Sustainability mean in and for Software Engineering? de Birgit Penzenstadler.
9. El desarrollo Sostenible en el mundo contemporáneo de Antonio Pulido.
10. Desarrollo Sostenible en Cuba: Teoría y Práctica de Rafael Andrés Velázquez Pérez.
11. Indicadores de sostenibilidad ambiental y de desarrollo sostenible: estado del arte y perspectivas de Rayén Quiroga M.
12. La sostenibilidad y el software de la Dra. Anaisa Hernández González.
13. Educación para el desarrollo sustentable: primeros pasos para la toma de conciencia en el desarrollo de los Sistemas de Información de Mabel Torres
14. Information Technology and Sustainability de Lorenz M. Hilty
15. Ingeniería y Desarrollo Sustentable de Ing. Gustavo Giuliano.
16. ¿Por dónde empezar para aplicar prácticas verde/sostenibles en el proceso de desarrollo del software y obtener un producto verde/sostenible? De la Dra. Anaisa Hernández González.
17. Identification of Practices for the Development of Green and Sustainable Software Using Agile Methods in GSD Projects: A Systematic Literature Review

Protocol de Nasir Rashid y Siffat Ullah Khan.

18. Green Agile Maturity Model For Global Software Development Vendors de Nasir Rashid and Siffat Ullah Khan.
19. Development of a Green ICT Model for Sustainable Enterprise Strategy de Bokolo Anthony Jnr y Mazlina Abdul Majid.
20. Principles of Software Engineering Management de Tom Gilb y Susannah Finzi.
21. Education for a Sustainable Future de Jaimie P. Cloud.
22. The Impact Of Environmental Resource Learning Applied To Geometry In Education For Power And Citizenship de Roosevelt Barros Morales, Mara Cabanilla Guerra, Carlos Barros, Sunny Bores Froment.
23. Knowledge systems for sustainable development de David W. Cash, William C. Clark, Frank Alcock, Nancy M. Dickson, Noelle Eckley, David H. Guston, Jill Jaeger, and Ronald B. Mitchell.
24. Evaluating Equality Requirements for Software Systems de Maryam Al Hinai y Ruzanna Chitchyan.

Fase de Uso

1. Energy Consumption and Hardware Utilization of Standard Software: Methods and Measurements for Software Sustainability de Achim Guldner, Marcel Garling, Marlies Morgen, Stefan Naumann, Eva Kern y Lorenz M. Hilty.
2. Sustainable software products—Towards assessment criteria for resource and energy efficiency de Eva Kern, Lorenz M. Hilty, Achim Guldner, Yuliy V. Maksimov, Andreas Filler, Jens Gröger y Stefan Naumann.
3. Integrating Aspects of Carbon Footprints and Continuous Energy Efficiency Measurements into Green and Sustainable Software Engineering de Eva Kern, Markus Dick, Jakob Drangmeister, Tim Hiller, Stefan Naumann y Achim Guldner.
4. Norma ISO 14067.
5. Green SW Engineering: Ideas for Including Energy Efficiency into your Software Projects de Gerald Kaefer.
6. An Empirical Evaluation of Database Software Features on Energy Consumption

de Sedef Akınlı, Gulfem Isiklar Alptekin, Andriy Miransky, Ayse Basar Bener, and Enzo Cialini.

7. Sustainable Software Engineering: Process and Quality Models, Life Cycle, and Social Aspects de Stefan Naumann, Eva Kern, Markus Dick y Timo Johann.
8. A Model and Selected Instances of Green and Sustainable Software de Markus Dick, Stefan Naumann, and Norbert Kuhn.
9. The Status Quo and the Prospect of Green IT and Green IS: A Systematic Literature Review de Mohammad Dalvi Esfahani, Azizah Abdul Rahman y Nor Hidayati Zakaria.
10. Position Paper: The Social Dimension of Sustainability in Requirements Engineering de Timo Johann y Walid Maalej.
11. Integrating the Complexity of Sustainability in Requirements Engineering de Martin Mahaux y Caroline Canon.
12. Sustainability Science for Strong Sustainability de Tom Dedeurwaerdere.
13. Responsible Living: Concepts, Education and Future Perspectives de Victoria W. Thoresen, Robert J. Didham, Jørgen Klein y Declan Doyle.

Fase de Fin de la Vida

1. The GREENSOFT Model: A reference model for green and sustainable software and its engineering de Stefan Naumann, Markus Dick, Eva Kern, Timo Johann.
2. Sustainable Software Engineering: Process and Quality Models, Life Cycle, and Social Aspects de Stefan Naumann, Eva Kern, Markus Dick y Timo Johann.

3.6 Validación del modelo propuesto

3.6.1 Método Delphi

El método utilizado, Método Delphi para Evaluación de Alternativas, es variado al poder combinarse en él técnicas cuantitativas y cualitativas [44] y consiste en la presentación a expertos o conocedores del tema, en este caso profesores de la Universidad de Cienfuegos, con vistas a la obtención de un acuerdo consensuado sobre el modelo propuesto. El método se aplica en dos fases: la selección de los expertos con la

preparación necesaria para opinar sobre los resultados de esta investigación y una segunda fase en que los expertos evalúan el modelo de procesos mediante una encuesta elaborada con este fin.

El cuestionario está compuesto por 11 categorías generales a ser evaluadas en una escala desde Inadecuado hasta Muy Adecuado (Anexo B). Se ofreció la posibilidad de que los expertos brindaran recomendaciones o cualquier observación adicional sobre el tema. El Delphi se considera como uno de los métodos subjetivos más confiables y con la elaboración estadística de las opiniones de expertos en el tema tratado. El conjunto de respuestas que se obtiene de la encuesta es sometido a un procesamiento estadístico en el Paquete SPSS versión 22.

3.6.2 Caracterización de los expertos

Se seleccionan 11 profesores de la Facultad de Ingeniería que poseen conocimientos sobre el tema. El coeficiente como experto se mide a partir de obtener un valor medio entre el conocimiento y la argumentación del experto sobre el tema (coeficiente de competencia del experto) mediante la siguiente expresión:

$$K = \frac{1}{2} (K_c + K_a)$$

Dónde:

K_c – Coeficiente de conocimiento del experto sobre el tema.

K_a – Coeficiente de argumentación del experto sobre el tema.

Se establece que cuando K se encuentra entre los valores de 0.8 y 1 ($0.8 < K < 1$) es confiable la selección realizada.

Se les aplica un cuestionario (Anexo D) que contiene las categorías evaluativas de los indicadores. El Anexo E exhibe el procesamiento estadístico de los perfiles de competencia. Se observa en los datos que los expertos seleccionados poseen un alto nivel de competencia, con un promedio de $K = 0.8727$.

Los 11 expertos seleccionados son graduados del nivel superior, 9 con categoría académica de Máster y 2 Doctores, siendo 9 Profesores Auxiliares y 2 Profesores Titulares. Como promedio tienen 14 años de experiencia en la docencia y su labor investigativa ha estado dirigida fundamentalmente a la informática y temas afines.

3.6.3 Resultados de los criterios dado por los expertos

Luego de seleccionados los 11 expertos se aplica el Método Delphi, los resultados obtenidos se presentan en el Anexo C. Los expertos evaluaron los diferentes aspectos del modelo. Las variables se describieron mediante el análisis de frecuencias absolutas, acumuladas, relativas acumuladas, y sus imágenes en la tabla de la inversa de la función de distribución normal, procedimiento descrito por Cerezal y Fiallo [44]. Los puntos de corte para la comparación con los promedios de cada variable evaluada se muestran en el Anexo C.

El aspecto “Fases” obtuvo un 81.8% de calificación de Muy Adecuado, lo que demuestra que las fases asumidas en el propuesta permiten una organización en la distribución del proceso. Las “Iteraciones” definidas, con un 90.9% de calificaciones de Muy Adecuado, permiten al informático establecer un ciclo continuo de mejoras.

Los aspectos “Análisis de Sostenibilidad”, “Capacitación” y “Objetivos de sostenibilidad” con un 100%, 63.63% y 81.8% de calificaciones de Muy Adecuado respectivamente, donde establecen los criterios para la superación del informático y la empresa con el objetivo de alcanzar un conocimiento general y un nivel de concientización sobre el tema cumpliendo con los indicadores de sostenibilidad. “Técnicas y métodos”, que comprende las técnicas propuestas para medir el cumplimiento de los criterios de sostenibilidad, obtuvo un 90.9% de evaluación de Muy Adecuado, los “Roles” definidos un 72.72% de Muy Adecuado y las “Actividades” propuestas un 81.8%.

“Consumo de energía” con un 90.9% de calificación de Muy Adecuado y “Huella de carbón y eficiencia energética” con un 81.8%, son considerados como propuestas novedosas por parte de los expertos, ya que ningún modelo anterior propone su uso como parte del proceso de desarrollo y uso de software sostenible. El “Sistema de conocimientos” definido, con un 100% de calificación de Muy Adecuado, contribuye a la organización en la formación del informático para la implementación del modelo de procesos propuesto.

Se determinó el grado de acuerdo entre los expertos, con la prueba no paramétrica del coeficiente de concordancia de rangos W de Kendall, con N muestras relacionadas. El coeficiente W–Kendall funciona como medida de relación entre k ordenaciones que

pueden ser ligadas de m objetos o individuos, en N muestras relacionadas [44]. Los resultados en la ronda con una significación asintótica es $0,001 < 0,5$ y un coeficiente W–Kendall de 0,559 permiten inferir que los expertos concuerdan significativamente, con un consenso significativo del 55,9% y con un margen del 2,03% de error.

De los resultados del método se puede concluir que la fundamentación y la estructura de la propuesta son evaluadas de Muy Adecuadas en la mayoría de los aspectos. Los expertos consideran que el modelo de procesos adaptado responde al problema de investigación, con acciones en cada etapa que se corresponden con el objetivo del mismo y que permiten a los informáticos alcanzar el éxito en el desarrollo de software sostenible. No sugieren cambios adicionales y proponen su aplicación.

3.7– Conclusiones

Se realiza una caracterización del uso del software sostenible en empresas del territorio, que arroja como resultado que los informáticos no consideran los “Criterios y métricas de sostenibilidad” durante el desarrollo y uso de software. Basado en ello y en el modelo GSSE, se propone un modelo de procesos adaptado y un sistema de conocimientos que sirvan como base para la formación de los ingenieros informáticos del territorio hacia el desarrollo de software sostenible. Además se lleva a cabo la validación de la propuesta mediante el método de valoración de expertos, que la calificaron como Muy Adecuada en la mayoría de los aspectos evaluados.

Conclusiones

Los resultados del trabajo responden satisfactoriamente a los objetivos propuestos, por lo que se concluye:

- Se realiza un estudio de los aspectos teóricos sobre el desarrollo sostenible, profundizando en el desarrollo de software sostenible y los modelos de procesos de ingeniería de software sostenible.
- Se realiza un análisis de modelos de procesos planteados por diferentes autores y se comparan considerando diversos aspectos, como las fases en las que se organiza el proceso, los objetivos de sostenibilidad y los roles definidos.
- Se realiza una caracterización del estado actual del desarrollo de software sostenible en empresas del territorio cienfueguero, que arroja como resultado que durante el ciclo de vida del software no se consideran los “Criterios y métricas de sostenibilidad”.
- Se selecciona como base de la propuesta el Modelo de Proceso para una Ingeniería de Software Ágil y Sostenible, y se adapta al contexto cubano considerando las características de las empresas del territorio. Se definen fases, roles e indicadores para su posterior implementación.
- Se propone un sistema de conocimientos como base para la formación de los informáticos del territorio hacia el desarrollo de un software sostenible.
- Se valida el modelo propuesto a través del método Delphi con encuestas procesadas en el SPSS v22 a 11 expertos que evaluaron los diferentes aspectos del mismo considerando actividades, roles, fases, entre otros. Los expertos corroboraron su concordancia y validez al calificar con los valores Bastante Adecuado y Muy Adecuado todos los aspectos analizados.

Recomendaciones

Se recomienda que:

- El modelo adaptado para el desarrollo de ingeniería del software sostenible sea implementado en las empresas del territorio.
- Se cree un espacio virtual de aprendizaje donde se pueda consultar e intercambiar información y experiencias sobre el desarrollo de software sostenible y la implementación del modelo propuesto.

Referencias bibliográficas

- [1] Gro Harlem Brundtland, "Informe de la Comisión Mundial de Desarrollo y Medio Ambiente," Mar. 1987.
- [2] R. W. Bybee, "Planet earth in crisis: how should science educators respond?," The American Biology Teacher, vol. 53, pp. 143–153, 1991.
- [3] Maria Novo, El desarrollo sostenible. Su dimensión ambiental y educativa. Madrid: UNESCO-Pearson, 2006.
- [4] Stefan Naumann, Eva Kern, Markus Dick, Timo Johann, "Sustainable Software Engineering: Process and Quality Models, Life Cycle, and Social Aspects," Advances in Intelligent Systems and Computing, pp. 191–205, 2015.
- [5] Markus Dick, Stefan Naumann, "Enhancing software engineering processes towards sustainable software product design.," EnviroInfo 2010: Integration of Environmental Information in Europe. Proceedings of the 24th International Conference on Informatics for Environmental Protection, Oct. 2010.
- [6] G. Lami, F. Fabbrini, M. Fusani, "Software sustainability from a process-centric perspective," Winkler, D., O'Connor, R.V., Messnarz R, Berlin, pp. 97–108, 2012.
- [7] S.S. Mahmoud, I. Ahmad, "A green model for sustainable software engineering," Int. J. Soft. Eng. Appl, vol. 7(4), pp. 55–74, 2013.
- [8] P. Bozzelli, Q. Gu, & P. Lago, "A systematic literature review a green software metrics.," Technician report. VU University Amsterdam. [En línea]. Disponible en: <http://www.s2group.es.vu.nl/cup-content/uploads/2013/01/LS.PaoloBozzelli.GreenMetrics.pdf>
- [9] Nasir Rashid, Siffat Ullah Khan, "Green agility for global software development vendor: a systematic literature review protocol.," in Pakistan Academy of Science, vol. 52(4), pp. 301–313.
- [10] J. Otegi, M. Aguilar, C. Cruz, L. Fuentes, "Towards sustainable Project management. A literature review," presented at the 19th International Congress on Project Management and Engineering. 43-56p.
- [11] María Rita Concepción García, Félix Rodríguez Expósito, Sergio Cleger Tamayo, José Pablo Suárez Rivero, Pilar Abad Real, "Educación para la sostenibilidad en docencia de ingeniería informática," Revista Iberoamericana de Educación, vol. 59/2, Jun. 2012.
- [12] Anaisa Hernández González, "La sostenibilidad y el software," Dilemas Contemporáneos: Educación, Política y Valores, vol. 2, p. 44, Oct. 2017. [En línea]. Disponible en: <http://www.dilemascontemporaneoseduccionpoliticaayvalores.com>

- [13] Inés Restrepo Tarquino, "Concepto de sostenibilidad," presented at the Seminario Taller Internacional Saneamiento en Asentamientos Formales e Informales con Énfasis en Alcantarillados Condominiales, Santiago de Cali, 2000.
- [14] Effrey Sachs, "The African Green Revolution," Sustainable Developments. [En línea]. Disponible en: http://www.earth.columbia.edu/sitefiles/file/about/director/2009/SciAm_May%202008.pdf
- [15] C. Folke, "Respetar los límites del planeta y recuperar la conexión con la biosfera," in *The State of the World 2013: Is Sustainability Still Possible?*, New York: W.W. Norton.
- [16] Jordi García, Helena García, David López, Fermín Sánchez, Eva Vidal, Marc Aliet, Jose Cabré, "La sostenibilidad en los proyectos de ingeniería," *ReV AENUI*, vol. 6, 2013. [En línea]. Disponible en: <http://www.aenui.net>
- [17] B. G. Norton, "Evaluating ecosystem states: Two competing paradigms," *Ecological Economics*, vol. 14, pp. 113–127. [En línea]. Disponible en: <https://www.raco.cat/index.php/Ensenanza/article/viewFile/21701/21535>
- [18] J. M. Naredo, "Sobre el origen, el uso y el contenido del término sostenible," *Documentación Social*, 1996, vol. 102, pp. 129–147.
- [19] M. Luffiego, M.F. Bastida, F. Ramos, J. Soto, "Epistemología, caos y enseñanza de las ciencias," *Enseñanza de las ciencias*, 1994. vol. 12(1), pp. 89–96.
- [20] C. Becker, "The Karlskrona manifesto for sustainability design." [En línea]. Disponible en: <http://arxiv.org/pdf/140.6968.pdf>.
- [21] Paula Serrano Yuste, "Sostenibilidad: estándares, calificación y certificación energética de edificios, y software," *Certificados Energéticos*. [En línea]. Disponible en: <https://www.certificadosenergeticos.com/sostenibilidad-diseno-verde-estandares-calificacion-certificacion-energetica-edificios>.
- [22] Luis Eduardo Sánchez Alejo, "La vinculación de la informática con el desarrollo sustentable," diciembre-2012. [En línea]. Disponible en: <https://prezi.com/or7gikqqaxw6/la-vinculacion-de-la-informatica-con-el-desarrollo-sustentable/>.
- [23] Ayse Basar Bener, Maurizio Morisio, Andny Miranskyy, Sedef Akinii Kocak, "TI Verde y Software Verde," *IEEECS*, Oct-2014. [En línea]. Disponible en: <https://www.computer.org/web/computingnow/archive/october2014-spanish>.
- [24] Rusli Abdullah; Salfarina Abdullah; Jamilah Din; Mxin Tee, "A systematic literature review of green software development in collaborative knowledge management environment," *International Journal of Advanced Computer Technology (IJACT)*. [En línea]. Disponible en: <http://ijact.org/volume4issue1/IJ0410057.pdf>

- [25] Eva Kern, Markus Dick, Stefan Naumann, Tim Hiller, "Impacts of software and its engineering on the carbon footprint of ICT," *Environmental Impact Assessment Review*. [En línea]. Disponible en: <http://dx.doi.org/10.1016/j.eiar.2014.07.003>
- [26] Andrés Mauricio Méndez, "VIII Taller Internacional de Tecnologías de Software Libre y Código Abierto," *Informática 2018*.). [En línea]. Disponible en: <http://www.informaticahabana.cu/es/node/1848>.
- [27] R.Ahmad, F. Barahom, & A. Hussain, "A systematic review on sustainability studies in software engineering," *Knowledge Management International Conference (KMICe)*. [En línea]. Disponible en: <http://www.kmice.csms.net.my>.
- [28] Rita. Concepción, Félix Rodríguez, José Pablo Suárez, "Procedimiento para valorar la sostenibilidad en la ingeniería informática," presentado en *Memorias XV III Congreso Universitario de Innovación en las Enseñanzas Técnicas*, Santander, España, 2010.
- [29] Nasir Rashid, Siffat Ullah Khan, "Developing Green and Sustainable Software using Agile Methods in Global Software Development: Risk Factors for Vendors," *Science and technology publications*. [En línea]. Disponible en: <http://www.scitepress.org/Papers/2016/59138/59138.pdf>.
- [30] Marcello Thiry, Liliane Frez, "GreenRM: Reference Model for Sustainable Software Development," 2015. [En línea]. Disponible en: https://ksiresearchorg.ipage.com/seke/seke14paper/seke14paper_234.pdf.
- [31] Eva Kern, Markus Dick, Stefan Naumann, J. Drangmeister, Timo Johann, "Measurement and rating of software-induced energy consumption of desktop pcs and servers," presented at the Pillmann W, Schade S, Smits P (eds) *Innovations in sharing environmental observations and information: proceedings of the 25th international conference enviroido*, Italy, Oct-2011.
- [32] Achim Guldner, Marcel Garling, Marlies Morgen, Stefan Naumann,, "Energy Consumption and Hardware Utilization of Standard Software: Methods and Measurements for Software Sustainability," *Springer International Publishing AG*, 2018. DOI 10.1007/978-3-319-65687-8_22
- [33] Eva Kern, Lorenz M. Hilty , Achim Guldner, Yuliy V. Maksimov, Andreas Filler, Jens Gröger, Stefan Naumann, "Sustainable software products—Towards assessment criteria for resource and energy efficiency," *Future generation computer systems*, 2018. [En línea]. Disponible en: <http://www.green-software-engineering.de>.
- [34] Stefan Naumann , Markus Dick, Eva Kern, Timo Johann, "The GREENSOFT Model: A reference model for green and sustainable software and its engineering," *Academia*, 2010. [En línea]. Disponible en: http://www.academia.edu/6573845/The_GREENSOFT_Model_A_reference_model_for_green_and_sustainable_software_and_its_engineering

- [35] International Organization for Standardization, "International Organization for Standardization, Software Engineering – Software Product Quality Requirements and Evaluation," 2011. [En línea]. Disponible en: <http://https://www.iso.org/standard/35683.html>.
- [36] C.U. Smith, L.G. Williams, Performance solutions. Practical Guide to Creating Responsive, Scalable Software. Boston: Addison-Wesley, 2002.
- [37] "OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2," Unified Modelling Language. [En línea]. Disponible en: <ftp://calhau.dca.fee.unicamp.br/pub/docs/ea977/UMLSuper2.1.2.pdf>.
- [38] S. Wang, H. Chen, W. Shi, "SPAN: a software power analyzer for multicore computer systems," ScienceDirect. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S221053791000003X>.
- [39] F. Butollo, J. Kusch, T. Laufer, "Buy IT Fair, Guideline for Sustainable Procurement of Computers," Berlin, 2009.
- [40] "Intel Corporation, Intel® Energy Checker SDK." [En línea]. Disponible en: <http://software.intel.com/enus/articles/intel-energy-checker-sdk/>.
- [41] Robert Johnson, Patricia Kuby, Estadística Elemental: Los Esencial, 10th ed. Thomson.
- [42] SPSS org, "SPSS v.22 for ingeniers." [En línea]. Disponible en: <http://www.spss.com>.
- [43] Eva Kern, Markus Dick, Jakob Drangmeister, Tim Hiller, and Stefan Naumann, Achim Guldner, "Integrating Aspects of Carbon Footprints and Continuous Energy Efficiency Measurements into Green and Sustainable Software Engineering," EnviroInfo 2013: Environmental Informatics and Renewable Energies, 2013.
- [44] J. Cerezal, J. Fiallo, El método Delphi. Los métodos científicos en las investigaciones pedagógicas. La Habana, 2002.
- [45] E. Egaña, La Estadística: herramienta fundamental en la investigación pedagógica. La Habana: Pueblo y Educación, 2003.

Anexos

Anexo A – Encuesta para la evaluación del desarrollo y uso de software sostenible

Estimado usuario, la encuesta que completará a continuación forma parte de una investigación de la Facultad de Ingeniería de la Universidad de Cienfuegos sobre el desarrollo y uso del software en la empresa cubana, características, impacto y necesidades del mismo. En las preguntas de tipo cerrada puede añadir cualquier observación o comentario que considere necesario. De antemano se le agradece su colaboración.

Empresa: _____

Graduado de: _____

1. ¿Qué entiende usted del término Sostenibilidad?

2. ¿Cuál es su concepto de Desarrollo Sostenible?

3. ¿Qué es para usted software sostenible?

4. ¿Cree usted que su empresa respeta los principios de sostenibilidad?

Si___ No___ Comentarios_____

5. ¿El software que usted utiliza en su trabajo considera que es adecuado a sus propósitos de operación específicos?

Si___ No___ Comentarios_____

6. ¿Dentro de su dominio de aplicación, el software produce algún desperdicio?

Si___ No___ Comentarios_____

7. ¿Cree que el software que utiliza cumple con los principios de software sostenible?

Si___ No___ Comentarios_____

8. ¿El software que usted utiliza se ejecuta en un tiempo prudente, o sea, funciona rápido?

Si___ No___ Comentarios_____

9. ¿El hardware de la computadora que usted utiliza permite la correcta ejecución del software?

Si___ No___ Comentarios_____

10. ¿Cuántas horas del día usted está trabajando con el software? _____ horas

Comentarios_____

11. ¿Cuántas operaciones puede realizar usted con el software? _____

Comentarios_____

12. ¿El costo de los viajes de negocios, o diarios al centro de trabajo, que se corresponden con el uso o desarrollo del software son altos?

Si___ No___

Comentarios_____

13. ¿Ocurre alguna emisión de CO₂ al medio ambiente durante el ciclo de vida del software?

Si___ No___

Comentarios_____

14. ¿La cantidad de energía consumida en el desarrollo del software es alta?

Si___ No___

Comentarios_____

15. ¿La cantidad de recursos consumidos durante el desarrollo del software por actividades que no tienen un valor adicional para consumidores o usuarios finales es alta?

Si___ No___

Comentarios_____

16. ¿Puede el software ser usado por la mayor cantidad de personas posibles sin restricciones?

Si___ No___

Comentarios_____

17. ¿El software es entendible, operable y de fácil aprendizaje?

Si___ No___

Comentarios_____

18. ¿Las condiciones de trabajo en la empresa son buenas?

Si___ No___

Comentarios_____

19. ¿El hardware utilizado en la empresa se corresponde con las tecnologías y necesidades actuales?

Si___ No___

Comentarios_____

20. ¿El software es adaptable a las necesidades específicas del usuario?

Si___ No___

21. ¿El software para su funcionamiento consulta datos ubicados en librerías externas u otro software?

Si___ No___

Comentarios_____

- a) ¿Sabe cuánta energía se consume cuando usted utiliza ese software?

Si___ Cantidad:_____ kW

No___

Comentarios_____

Otros datos que considere importantes:

Muchas gracias.



Anexo B

Encuesta a Expertos

Estimado colega: Por medio de la presente lo estamos invitando a colaborar como experto en una investigación que tributa a la Carrera de Ingeniería Informática perteneciente a la Universidad de Cienfuegos y cuya temática fundamental son los modelos de procesos de software sostenible. Conocemos su prestigio profesional y nos sentiríamos honrados de contar con su juicio y aportes para el éxito de dicha investigación.

Le solicitamos que siga los pasos que a continuación relacionamos.

Será enviado a vuelta de correo el modelo de procesos de software sostenible adaptado a las características de la empresa cubana actual, el cual debe valorar según los criterios expresados a continuación.

Cualquier sugerencia, duda o crítica debe hacerlo a través de este correo.

Muchas gracias.

Responda marcando con una X su consideración sobre estos elementos:

1. Fases

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	Inadecuado

2. Iteraciones

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	Inadecuado

3. Análisis de Sostenibilidad

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	Inadecuado

4. Capacitación

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	Inadecuado

5. Objetivos de sostenibilidad

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	Inadecuado

6. Técnicas y métodos

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	Inadecuado

7. Roles

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	Inadecuado

--	--	--	--	--

8. Actividades

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	Inadecuado

9. Medición del consumo de energía

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	Inadecuado

10. Medición de Huella de carbón y eficiencia energética

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	Inadecuado

11. Sistema de conocimientos

Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	Inadecuado

Anexo C

Resultados del Método Delphi

Tabla: Frecuencia Observada

Aspectos a Evaluar	Inadecuado	Poco Adecuado	Adecuado	Bastante Adecuado	Muy Adecuado
Fases	0	0	0	2	9
Iteraciones	0	0	0	1	10
Análisis de Sostenibilidad	0	0	0	0	11
Capacitación	0	0	0	4	7
Objetivos de sostenibilidad	0	0	0	2	9
Técnicas y métodos	0	0	0	1	10
Roles	0	0	0	3	8
Actividades	0	0	0	2	9
Consumo de energía	0	0	0	1	10
Huella de carbón y la eficiencia energética	0	0	0	2	9
Sistema de conocimientos	0	0	0	0	11

El coeficiente de concordancia entre los expertos es de: **W = 0,559**

Anexo D

Cuestionario de Expertos

Nombre y apellidos: _____

Años de experiencia en la docencia: _____

Categoría Docente: Auxiliar. _____ Titular: _____

La estrategia didáctica para el aprendizaje interactivo en ambientes en línea en el posgrado debe ser validada su aplicación; para ello se utiliza el método de expertos dentro de los cuales Ud. ha sido escogido por su relación y conocimiento del tema en cuestión por lo que debe seguir los pasos que a continuación relacionamos. De ser positiva su disposición responda este cuestionario en su primera fase correspondiente al criterio de expertos.

Cuestionario: Valore en una escala de valor del 0 al 10, considerando el mínimo cero y el máximo 10, su conocimiento e información sobre:

1. Desarrollo de software sostenible: -----

2. Modelos de procesos para el desarrollo del software sostenible. -----

Marque con una (X) las fuentes que por su consideración han influido en el nivel de conocimiento que posee:

Fuentes de conocimiento	Grado de influencia		
	Alto	Medio	Bajo
Preparación que posee sobre el tema.			
Conocimiento empírico o teórico del tema			
Investigaciones acerca del mismo.			
Experiencia obtenida.			

Anexo E

Coeficiente de Competencia

Experto	Ka	Kc	Fórmula	K	Código
1	0.9	0.9	$1/2(0.9+0.9)$	0.9	Alto
2	0.9	0.9	$1/2(0.9+0.9)$	0.9	Alto
3	0.8	0.9	$1/2(0.8+0.9)$	0.85	Alto
4	0.9	0.9	$1/2(0.9+0.9)$	0.9	Alto
5	0.9	0.9	$1/2(0.9+0.9)$	0.9	Alto
6	0.9	0.9	$1/2(0.9+0.9)$	0.9	Alto
7	0.8	0.8	$1/2(0.8+0.8)$	0.8	Alto
8	0.9	0.9	$1/2(0.9+0.9)$	0.9	Alto
9	0.8	0.8	$1/2(0.8+0.8)$	0.8	Alto
10	0.9	0.9	$1/2(0.9+0.9)$	0.9	Alto
11	0.8	0.9	$1/2(0.8+0.9)$	0.85	Alto