



**UNIVERSIDAD
DE ANTIOQUIA**

**MÉTODO PARA LA EVALUACIÓN DE LA
SOSTENIBILIDAD DEL SOFTWARE PARA EL
PROCESO DE COMPRA Y CONTRATACIÓN DEL
SECTOR PÚBLICO.**

Autor
Carlos David Carrascal Vergara

Universidad de Antioquia
Facultad de Ingeniería
Medellín, Colombia
2021



Método para la Evaluación de la Sostenibilidad del Software para el Proceso de Compra Y
Contratación del Sector Público.

Carlos David Carrascal Vergara

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:
Ingeniero de Sistemas

Asesor:

Ing. León Daniel Jaramillo Ramírez, M.Sc.

Línea de Investigación: Ingeniería de Software

Grupo de Investigación:

ITOS

Universidad de Antioquia

Facultad de Ingeniería

Medellín, Colombia

2021

A mi familia por el apoyo incondicional y el esfuerzo de recorrer conmigo los caminos que conllevaron a convertirme en el ser que soy hoy.

CONTENIDO

1. INTRODUCCIÓN	1
1.1. Planteamiento del Problema.	1
1.2. Objetivos	4
1.2.1 Objetivo General	4
1.2.3 Objetivos Específicos	4
2. MARCO TEÓRICO	5
2.1 Modelo de Calidad de Software	5
2.2 Modelo de Calidad ISO/IEC 25000	6
2.3. Sostenibilidad en Ingeniería de Software	11
2.2. Procesos de contratación de Software en el Sector Público.	21
2.2.1. Procesos de contratación de Software en Colombia.	21
2.2.2. Otros Procesos de Contratación.	26
2.2.2.1. Navegando por la adquisición de software en una industria "como servicio". Guía práctica para profesionales de las TIC y la adquisición. - Australia.	26
2.2.2.2. Guía Del Usuario para la Adquisición de Productos y Servicios De Tecnología de La Información - Banco Asiático de Desarrollo.	28
3. METODOLOGÍA	31
3.1. Identificación del Problema y Motivación	31
3.2. Objetivos de una Solución	32
3.3. Diseño y Desarrollo	32
3.4. Demostración	32
3.5. Comunicación	33
4. RESULTADOS Y ANÁLISIS	34
4.1 ESTADO DEL ARTE	34
4.1.1. Sostenibilidad de Software	34
4.1.2. Estándares y Guías en Procesos de Adquisición de Software de compras.	41
4.1.2.1. ISO/IEC 12207	41
4.1.2.2. IEEE 1062-2019 - IEEE Recommended Practice for Software Acquisition.	42
4.1.2.3. Modelo CMMI-ACQ.	43
4.1.2.4. Modelo ESCM-CL	43
4.2. DISEÑO Y DESARROLLO	45
4.3. CARACTERIZACIÓN DE MODELOS EXISTENTES	46
4.4. MÉTODO PROPUESTO	48

4.4.1 MÉTODO PARA LA EVALUACIÓN DE LA SOSTENIBILIDAD DEL SOFTWARE PARA EL PROCESO DE COMPRA Y CONTRATACIÓN DEL SECTOR PÚBLICO.	52
4.4.1.1. Objetivo	52
4.4.1.2. Requisitos de la evaluación	52
4.4.1.3. Especificación de la Evaluación	53
4.4.1.4. Planificación de la Evaluación.	58
4.4.1.5. Ejecutar la evaluación	58
4.4.1.6. Concluir la Evaluación.	59
4.5. HERRAMIENTA DE EVALUACIÓN	60
4.6. DEMOSTRACIÓN DE APLICACIÓN Y VALIDACIÓN EXTERNA	65
4.6.1 VALIDACIÓN EXTERNA	65
4.6.2 APLICACIÓN DEL MÉTODO DE EVALUACIÓN	74
5. CONCLUSIONES Y RECOMENDACIONES.	87
6. REFERENCIAS	90
7. ANEXOS	96
ANEXO I - PREGUNTAS Y RESPUESTAS – EVALUACIÓN DE LOS ATRIBUTOS DE CALIDAD PARA LA SOSTENIBILIDAD DE SOFTWARE CASO I.	96
ANEXO II. PREGUNTAS Y RESPUESTAS – EVALUACIÓN DE LOS ATRIBUTOS DE CALIDAD PARA LA SOSTENIBILIDAD DE SOFTWARE CASO II.	109
ANEXO III. MEDICIÓN ATRIBUTOS DE CALIDAD DE LA FAMILIA ISO 25000.	122
ANEXO IV. HERRAMIENTA DE EVALUACIÓN.	131

LISTA DE TABLAS

Tabla 1. Descripción de los principios del Manifiesto de Karlskrona adaptado de (Becker, 2014).	14
Tabla 2. Principios del manifiesto de Karlskrona en relación con las fases del ciclo de vida del desarrollo de software adaptado de (Becker, 2014).	15
Tabla 3. Atributos de calidad para la sostenibilidad social. (N. Condori-Fernandez, P. Lago, 2018).	39
Tabla 4. Requisitos de calidad para la sostenibilidad técnica (N. Condori-Fernandez, P. Lago, 2018).	40
Tabla 5. Requisitos de calidad para la sostenibilidad económica (N. Condori-Fernandez, P. Lago, 2018).	41
Tabla 6. Atributos de calidad para la sostenibilidad ambiental. (N. Condori-Fernandez, P. Lago, 2018).	42
Tabla 7. Trabajos de Sostenibilidad de Software.	48
Tabla 8. Características y Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software.	49
Tabla 9. Descripción de la tabla de métricas de acuerdo al estándar ISO/IEC 25023 (ISO/IEC 25023 System and Software Quality Requirements and Evaluation [SQuaRE] — Measurement of system and software product quality, 2016).	54
Tabla 10. Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de la Seguridad (ISO/IEC 25023 System and Software Quality Requirements and Evaluation [SQuaRE] — Measurement of system and software product quality, 2016).	55
Tabla 11 Grado de importancia de Atributos de Calidad de acuerdo al Estándar ISO/IEC 25040. (ISO/IEC 25040 System and Software Quality Requirements and Evaluation [SQuaRE] — Evaluation process, 2011).	55
Tabla 12. Características y Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software.	57
Tabla 13. Niveles de puntuación de evaluación de sostenibilidad con base a las recomendaciones de estándar ISO/IEC 25040 (ISO/IEC 25040 System and Software Quality Requirements and Evaluation [SQuaRE] — Evaluation process, 2011).	58
Tabla 14. Preguntas definidas para evaluación de la Confidencialidad (J.Calabrese, R. Muñoz, 2018).	75
Tabla 15. Criterios de evaluación para la confidencialidad (J.Calabrese, R. Muñoz, 2018).	76
Tabla 16. Resultado de evaluación de atributos de calidad relevantes para la sostenibilidad Caso 1.	78
Tabla 17. Resultado de evaluación de atributos de calidad relevantes para la sostenibilidad Caso 2.	81

LISTA DE FIGURAS

Figura 1. Familia de normas ISO/IEC 25000 SQuaRE (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).	6
Figura 2. Modelo de calidad de producto según la norma ISO/IEC 25010 (ISO/IEC 25010 System and Software Quality Requirements and Evaluation [SQuaRE] — System and software quality models, 2011)	8
Figura 3. Modelo de Calidad de Uso ISO/IEC 25010. (ISO/IEC 25010 System and Software Quality Requirements and Evaluation [SQuaRE] — System and software quality models, 2011)	9
Figura 4. Flujo del Catálogo de diseño de sostenibilidad de software (SSDC) adaptado de (Oyedeki, S., Seffah, A., & Penzenstadler, B. 2018)).	18
Figura 5. Sostenibilidad integrada a lo largo de todo el ciclo de vida de software (Saputri, Lee, 2020).	20
Figura 6. Esquema Gráfico de la Metodología.	31
Figura 7. Esquema Genérico de un Proceso de Compra o contratación Pública (MinTIC Colombia. 2020)	50
Figura 8. Actividades del proceso de evaluación del software según el estándar ISO/IEC 25040.	50
Figura 9. Actividades del proceso de evaluación del software según el estándar ISO/IEC 25040.	58
Figura 10. Esquema de herramienta de Evaluación de la Sostenibilidad de Software.	60
Figura 11. Esquema de los datos generales de la herramienta de evaluación.	61
Figura 12. Esquema del encabezado de la herramienta de evaluación.	61
Figura 13. Ejemplo ponderación Experticia.	62
Figura 14 Ejemplo de Subtotal por Ítem.	62
Figura 15. Resultados de la sostenibilidad de software.	63
Figura 16. Encuesta a panel de expertos.	65
Figura 17. Respuestas a la primera pregunta.	65
Figura 18. Respuestas a la segunda pregunta de la encuesta.	66
Figura 19. Respuestas a la tercera pregunta de la encuesta.	66
Figura 20. Respuestas a la cuarta pregunta.	67

Figura 21. Respuestas a la quinta pregunta.	67
Figura 22. Respuestas a la sexta pregunta	68
Figura 23. Respuestas a la séptima pregunta.	68
Figura 24. Respuestas a la octava.	69
Figura 25. Respuestas a la novena pregunta.	70
Figura 26. Respuestas a la décima pregunta.	70
Figura 27. Pregunta undécima.	71
Figura 28. Pregunta duodécima.	72
Figura 29. Resultados de la sostenibilidad de software (J.Calabrese, R. Muñoz, 2018).	75
Figura 30. Resultado de Evaluación para la Dimensión Social Caso 1.	77
Figura 31. Resultado de Evaluación para la Dimensión Ambiental Caso 1.	77
Figura 32. Resultado de Evaluación para la Económica Caso 1.	78
Figura 33. Resultado de Evaluación para la Económica Caso 1.	78
Figura 34. Resumen de Evaluación de la Sostenibilidad de Software para el Caso 1.	78
Figura 35. Resultado de Evaluación para la Dimensión Social Caso 2.	81
Figura 36. Resultado de Evaluación para la Dimensión Ambiental Caso 2.	81
Figura 37. Resultado de Evaluación para la Dimensión Económica Social Caso 2.	81
Figura 38. Resultado de Evaluación para la Técnica Social Caso 2.	82
Figura 39 Resumen de Evaluación de la Sostenibilidad de Software para el Caso 2.	82

GLOSARIO

- **Atributo:** Propiedad o característica inherente de una entidad que puede distinguirse cuantitativa o cualitativamente por medios humanos o automatizados.
- **Calidad de Software:** Capacidad del producto de software para satisfacer las necesidades cuando se utiliza en condiciones especificadas.
- **Característica de Calidad:** Categoría de atributos en el cual está basada la calidad del software.
- **Componente:** Entidad con estructura discreta, como un módulo de ensamblaje o software, dentro de un sistema considerado en un nivel de análisis determinado.
- **Elemento de medición de calidad:** Medida definida en términos de una propiedad y el método de medición para cuantificarla.
- **Medida:** Variable a la que se asigna un valor como resultado de la medición.
- **Medición:** Conjunto de operaciones que tienen el objeto de determinar un valor de una medida.
- **Producto Software:** Conjunto de programas informáticos, procedimientos, documentación y datos asociados.
- **Operación:** Ejecución de una acción o proceso.
- **Valor:** Número o categoría asignado a un atributo de una entidad mediante la realización de una medición

RESUMEN

Día a día interactuamos con estos sistemas en casi todos los aspectos de nuestras vidas. A medida que usamos estos sistemas nos centramos únicamente en los beneficios tangibles del software, y es importante tener en cuenta las consideraciones a mediano y largo plazo. El desarrollo sostenible se considera un campo multidisciplinario que necesita integrar varios esfuerzos de la comunidad científica. Sin embargo, la noción de sostenibilidad y sus dimensiones o características es un tema abierto entre la comunidad de ingeniería de software.

El concepto de sostenibilidad está inicialmente asociado con el medio ambiente. Últimamente este concepto se ha integrado ampliamente al desarrollo de software y cada vez los proyectos son más complejos, y el producto resultante debe satisfacer los estándares de calidad acorde al mercado. La sostenibilidad se define como la capacidad de perdurar y preservar la función de un sistema durante un período de tiempo prolongado. Esta está basada en ciertas características o parámetros. La evaluación de estas características determina el grado de sostenibilidad del producto y determinar la mejor opción para los requerimientos de usuario.

En el caso de las entidades del sector público se busca hacer inversiones inteligentes con las que pueda acceder a tecnología que realmente se ajuste a las necesidades de las entidades y que corresponda con las mejores opciones que ofrece el mercado en términos de calidad y costo. Con el fin de fortalecer los procesos de compra y contratación de software, este trabajo propone un método que permita a las entidades del sector público la evaluación de sostenibilidad de software facilitando la toma de decisiones en la adquisición de nuevos productos, que permitan a las entidades estatales establecer condiciones generales para asegurar la calidad y que responda a las necesidades y características del entorno, generando un verdadero apalancamiento de los recursos invertidos y alejándose del detrimento patrimonial.

Palabras clave: Sostenibilidad, Calidad de Software, Método de evaluación.

ABSTRACT

Every day we interact with these systems in almost every aspect of our lives. As we use these systems we focus solely on the tangible benefits of software, and it is important to consider medium- and long-term considerations. Sustainable development is considered a multidisciplinary field that needs to integrate various efforts of the scientific community. However, the notion of sustainability and its dimensions or features is an open theme among the software engineering community.

The concept of sustainability is initially associated with the environment. Lately this concept has been widely integrated into software development and projects are becoming more complex, and the resulting product must meet market-consistent quality standards. Sustainability is defined as the ability to endure and preserve the function of a system for an extended period of time. This is based on certain characteristics or parameters. The evaluation of these characteristics determines the degree of sustainability of the product and determines the best option for user requirements.

In the case of public sector entities, we seek to make smart investments with which you can access technology that really fits the needs of the entities and that corresponds to the best options offered by the market in terms of quality and cost. In order to strengthen software procurement and procurement processes, this work proposes a method that allows public sector entities to evaluate software sustainability by facilitating decision-making in the procurement of new products, enabling state entities to establish general conditions to ensure quality and responding to the needs and characteristics of the environment, generating a real leverage of the resources invested and moving away from the economic detriment.

Keywords: Sustainability, Software Quality, Evaluation Method

1. INTRODUCCIÓN

1.1. Planteamiento del Problema.

La tecnología de la información juega un papel fundamental al abordar las preocupaciones ambientales, económicas y sociales. El uso de esta no solo puede inmateralizar actividades que de otro modo consumiría recursos, como por ejemplo correo electrónico y correo postal, reuniones virtuales y viajes. Además, se tiene la capacidad de obtener datos para optimizar los recursos, como puede ser análisis de consumo de agua y redes inteligentes. Por otro lado, es poco frecuente que se evalúe el impacto causado por el desarrollo de productos de sistemas de información. Por ejemplo, se estima que una computadora se vuelve obsoleta por cada computadora nueva en el mercado (Silicon Valley Toxics Coalition 2001), en otras palabras, el hardware de la computadora todavía es utilizable, pero el software lo vuelve inútil u obsoleto. Si los desarrolladores de software tuvieran en cuenta ese hecho, tal vez los productos se diseñarían desde cero para que sigan funcionando con el hardware antiguo (Silicon Valley Toxics Coalition 2001). El diseño del software como factor clave puede ayudar a reducir el consumo de energía entre un 30% y un 90% (Musthaler, 2018). Por lo tanto, para el diseño de software se debe tener en cuenta diferentes factores y la sostenibilidad debe considerarse como una cualidad importante de los sistemas de software, tal como la seguridad y la usabilidad, etc. (Lago, P.; Koçak, S.A.; Crnkovic, I.; Penzenstadler, B. 2012)

Según el Marco de Referencia de Arquitectura Empresarial de Tecnologías de Información de MinTIC, permite que las entidades públicas puedan fortalecer las capacidades institucionales para la prestación de servicios a usuarios mediante el uso de las TIC (MinTIC Colombia. 2020). La modernización del estado es fundamental para el desarrollo económico del país. El estado colombiano no es ajeno a ello, por lo que intenta regular los procesos de compra mediante la reglamentación nacional, por ejemplo:

El estado para su funcionamiento requiere servicios y bienes que adquieren a través de los procesos de compra y contratación. El artículo 2.2.1.1.1.6.1 del Decreto 1082 de 2015 establece que previo a una compra se debe realizar análisis de mercado, estudios técnicos y

financieros, que debe incluirse en los documentos en el proceso de contratación (MinTIC Colombia. 2020).

Las compras deben estar orientadas a satisfacer las necesidades de la Entidad Estatal y a obtener el mayor valor por el dinero público. Colombia Compra Eficiente, junto con el Ministerio de Tecnologías de la Información y Comunicaciones – MinTIC encontraron falencias y oportunidades de mejora en los procesos de contratación en las Entidades Estatales, tales como (MinTIC Colombia. 2020):

- Desconocimiento generalizado de los criterios de sostenibilidad en compra y contratación pública y privada (Cristancho, Martínez, 2018).
- Falta una visión integral donde se analice el ciclo de vida del software adquirido (Cristancho, Martínez, 2018).
- Adquisición de software sin tener en cuenta mínimos estándares de calidad, lo que ocasiona una baja eficiencia en el uso de los recursos públicos.
- Deficiencias en los procesos de selección que aseguren una relación costo-beneficio.

Por lo tanto, una forma de suplir estas falencias sería con la implementación de una directriz para la adquisición de software sostenible que facilite el uso eficiente de los recursos.

La tecnología disponible hoy, apunta a lograr un desarrollo de software donde la sostenibilidad sea un punto importante en el desarrollo de un proyecto, por lo tanto, dado que no existen lineamientos claros en la evaluación de calidad de productos en procesos de adquisición de software cobra importancia la implementación de modelos de calidad o métodos de evaluación de productos MinTIC (MinTIC Colombia. 2020). Esto nos permite hacer un control sobre el proceso de adquisición, y evaluar técnica y económicamente el producto que se va a comprar.

En los últimos años los estudios sobre la sostenibilidad de los sistemas de software han crecido notablemente. Algunos trabajos se enfocan en los requisitos de ingeniería y evaluación, otros trabajos se enfocan en el estudio de todo el proceso de desarrollo de software (Theresia Ratih Dewi Saputri, Seok-Won Lee, 2020). A pesar de que no existe un método integrado para evaluación de la sostenibilidad de software, son cada vez más los

estudios enfocados a desarrollar software sostenible (J. Kienzle, G. Mussbacher, B. Combemale, L. Bastin, N. Bencomo, J.-M. Bruel, C. Becker, S. Betz, R. Chitchyan, B. Cheng. 2019).

La sostenibilidad en las compras y contratación del sector público se ven afectadas por la obsolescencia tecnológica y el vertiginoso cambio de la misma, lo cual requiere herramientas que apoye a los tomadores de decisiones a generar compras inteligentes y sostenibles.

Con el fin de suplir las deficiencias en procesos de compra contratación en el sector público, este trabajo propone un método que permita evaluar, desde el proceso de compra y contratación, la adquisición del software indicado para el desarrollo de las actividades de las entidades del estado y que además sea sostenible de manera funcional en el tiempo, permitiendo una inversión óptima de los recursos del estado.

1.2. Objetivos

1.2.1 Objetivo General

- Proponer un método para la evaluación de la sostenibilidad del software para el proceso de compra y contratación del sector público.

1.2.3 Objetivos Específicos

- Revisar los modelos de calidad de software con énfasis en la sostenibilidad del producto.
- Caracterizar los parámetros de calidad que apunten a la sostenibilidad de software.
- Construir un método de calidad de software para la sostenibilidad en procesos de compra y contratación.
- Demostrar el método para la evaluación de sostenibilidad del software en procesos de compra y contratación.

2. MARCO TEÓRICO

Desarrollar un software con calidad implica la utilización de estándares, metodología y procesos para análisis, diseño, programación y pruebas, con el fin de lograr confiabilidad, efectividad y productividad para el control de la calidad del software.

2.1 Modelo de Calidad de Software

Actualmente las organizaciones desarrolladoras de software ven la necesidad no solo de entregar sus productos según la necesidad del cliente, sino con características y atributos de calidad, por tanto, la mayoría de ellas tienen implementado en sus procesos en donde la satisfacción del usuario y un producto confiable son la prioridad (Roa, P. A., Morales, C., & Gutiérrez, P, 2015).

Por definición, un modelo de calidad de software es “el conjunto de características y relaciones entre ellas que proporcionan la base para especificar requisitos de calidad y evaluación de la calidad” (IEEE 24765:2017, 2017). Para garantizar la calidad de software es importante implementar algún modelo o estándar de calidad que permita la gestión de atributos en el proceso de construcción de software, teniendo en cuenta que la concordancia entre los requisitos y su construcción son la base de las medidas de calidad establecidas. La principal finalidad del modelo de calidad de producto es especificar y evaluar el cumplimiento de criterios del producto (Callejas, M., Alarcón, A. y Álvarez, A. 2017).

2.2 Modelo de Calidad ISO/IEC 25000

El modelo de calidad ISO/IEC 25000 tiene como objetivo especificar los requisitos de calidad y evaluación de software. Esta norma es conocida como SQuaRE (*System and Software Quality Requirements and Evaluation*).

El modelo está compuesto por 5 divisiones: 2500n, 2501n, 2502n, 2503n y 2504n. La letra n denota que está compuesta por varias normas (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

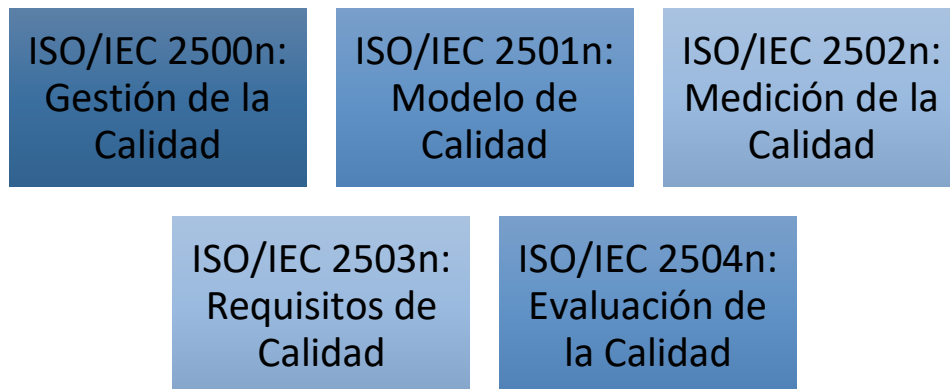


Figura 1. Familia de normas ISO/IEC 25000 SQuaRE (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

- **ISO/IEC 2500n Gestión de Calidad:** Esta división de referencia contiene las definiciones y términos en torno a los componentes utilizados en la ISO/IEC 25000 (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

- **ISO/IEC 2501n Modelo de Calidad:** Esta división contiene las características, subcaracterística y las definiciones del modelo de calidad de software (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).
- **ISO/IEC 2502n Medición de Calidad:** Esta división contiene las referencias para la medición, evaluación y aplicación de los atributos de calidad (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).
- **ISO/IEC 2503n Requisitos de calidad:** Esta división contiene las referencias para especificar los requisitos de calidad en el desarrollo de software (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).
- **ISO/IEC 2504n Evaluación de calidad:** Esta división proporciona recomendaciones y bases para la evaluación de la calidad de software. (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

ISO/IEC 25010 - Modelos de Sistema y Calidad del Software

Este estándar detalla el modelo de la calidad del producto, describe ocho características y subcaracterísticas para evaluar el software como se muestra en la figura 2. (ISO/IEC 25010 System and Software Quality Requirements and Evaluation [SQuaRE] — System and software quality models, 2011).

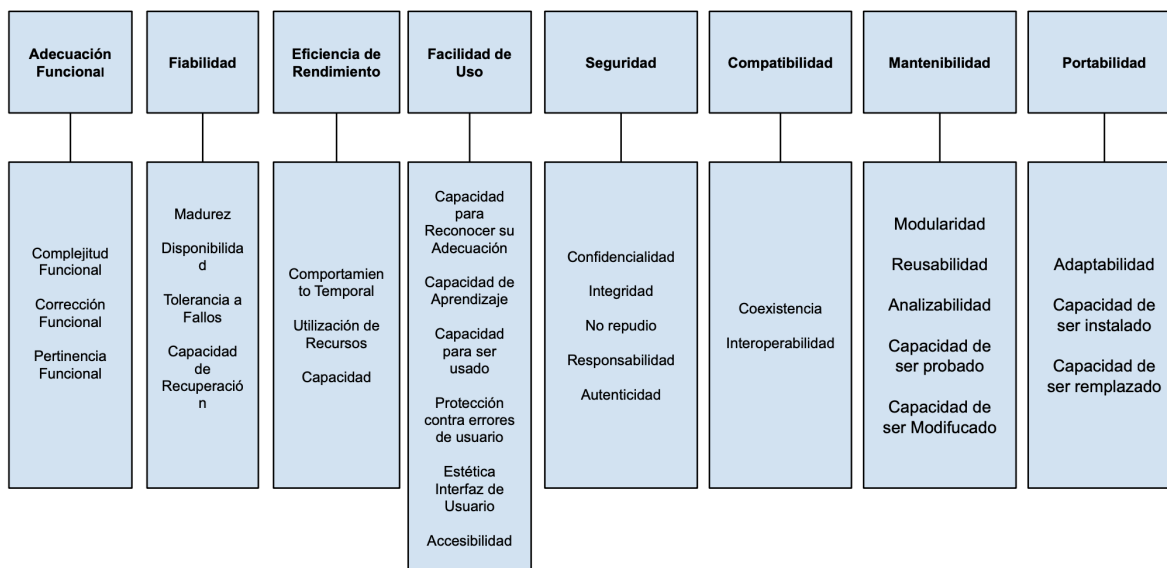


Figura 2. Modelo de calidad de producto según la norma ISO/IEC 25010 (ISO/IEC 25010 System and Software Quality Requirements and Evaluation [SQuaRE] — System and software quality models, 2011)

Las 8 características de calidad del producto se pueden definir de la siguiente forma. (ISO/IEC 25010 System and Software Quality Requirements and Evaluation [SQuaRE] — System and software quality models, 2011):

- **Adecuación Funcional:** Representa la capacidad del producto software para proporcionar las funciones necesarias para satisfacer al usuario.
- **Fiabilidad:** Capacidad del producto software para realizar las funciones específicas cuando se utiliza bajo ciertas condiciones y periodos de tiempo determinados.
- **Eficiencia en el Desempeño:** Capacidad del producto software para proporcionar un rendimiento apropiado, respecto a la cantidad recursos utilizados bajo determinadas condiciones.
- **Facilidad de Uso:** Capacidad del producto software para que sea entendido, aprendido y usado por el usuario.

- **Seguridad:** Capacidad de proteger la información y los datos, de manera que personas o sistemas no autorizados puedan tener acceso para consultas o actualizaciones.
- **Compatibilidad:** Capacidad del producto software para llevar a cabo sus funciones intercambiando información.
- **Mantenibilidad:** Capacidad del producto software para ser modificado o actualizado debido a necesidades evolutivas y correctivas.
- **Portabilidad:** Capacidad de un sistema o componente software de ser trasladado de un entorno a otro sin que esto afecte la funcionalidad de cada sistema.

Otro modelo es el de Calidad de Uso. Las características de este modelo determinan el nivel de calidad requerido desde el punto de vista del usuario. Estos requisitos son los que determinan la validación del software por parte del usuario. Este modelo se define con las siguientes características. (ISO/IEC 25010 System and Software Quality Requirements and Evaluation [SQuaRE] — System and software quality models, 2011), Ver figura 3.



Figura 3. Modelo de Calidad de Uso ISO/IEC 25010. (ISO/IEC 25010 System and Software Quality Requirements and Evaluation [SQuaRE] — System and software quality models, 2011)

- **Efectividad:** Capacidad del producto software para alcanzar los objetivos o necesidades del usuario, al momento de utilizar el sistema.
- **Eficiencia:** Capacidad del producto software para alcanzar los objetivos del usuario, utilizando los recursos mínimos.

- **Satisfacción:** Capacidad del producto software para satisfacer las diferentes necesidades mínimas de los usuarios al utilizarlo.
- **Utilidad:** Grado en que un usuario está satisfecho cuando logra alcanzar sus objetivos planteados.
- **Libertad de Riesgo:** Capacidad que tiene un producto o sistema software en reducir el riesgo potencial relacionado con la situación económica, vida humana, salud o medio ambiente. Esto incluye la salud y seguridad, tanto del usuario y aquellos afectados por el uso, así como las consecuencias materiales o económicas no deseadas.
- **Cobertura de Contexto:** capacidad de un producto o sistema software para ser utilizado con efectividad, eficiencia, libertad de riesgo y satisfacción en ámbitos de uso que fueron definidos.

Métricas para la calidad - ISO/IEC 25022 y 25023.

Las métricas de calidad se definen en las normas ISO/IEC 25023 e ISO/IEC 25022, las cuales contienen recomendaciones para la evaluación de los atributos de calidad. La evaluación de estos determina la calidad del producto software definido en la ISO/IEC 25010. (ISO/IEC 25010 System and Software Quality Requirements and Evaluation [SQuaRE] — System and software quality models, 2011).

2.3. Sostenibilidad en Ingeniería de Software

En el campo de la ingeniería de software no hay un consenso sobre la definición de sostenibilidad. Cada investigador tiene su comprensión de lo que significa este concepto. Se puede considerar que el software es sostenible cuando los impactos negativos, directos e indirectos en la economía, la sociedad, los seres humanos y el medio ambiente que resultan del desarrollo, implementación y uso del software son mínimos y/o tienen un efecto positivo en el desarrollo sostenible (S. Naumann, M. Dick, E. Kern, T. Johann, 2011). El Instituto de Sostenibilidad de Software define la sostenibilidad como la capacidad del software que se usa hoy de estar disponible para mejorarse y soportarse en un futuro (Software Sustainability Institute, 2020). En pocas palabras, puede definirse de forma sencilla como la capacidad del software para perdurar en el tiempo (E.Y. Nakagawa, R. Capilla, E. Woods, P. Kruchten, 2018).

La sostenibilidad es un término con múltiples significados, todos ellos se refieren generalmente a la "capacidad de durar mucho tiempo" (The Sustainability Institute by ERM, 2020). Como se dijo anteriormente, no hay un consenso en la definición del término, esto se debe a que la comunidad en general piensa que la sostenibilidad puede verse de manera específica o global, por lo tanto, la discusión puede centrarse en un sistema en concreto o un sistema global.

Un sistema concreto puede ser un sistema de software, o cualquier otro sistema artificial o natural, mientras que un sistema global implica la sostenibilidad de todos los sistemas específicos relacionados entre sí (Penzenstadler, 2014).

El concepto de sostenibilidad global está directamente relacionado con la definición de desarrollo sostenible, "El desarrollo sostenible es el desarrollo que satisface las necesidades del presente sin comprometer la capacidad de las generaciones futuras de satisfacer sus propias necesidades" (United-Nation, 1987), y en el contexto de la ingeniería de software, la sostenibilidad se centra en las preocupaciones sobre el impacto de los sistemas de software en la sostenibilidad global (Penzenstadler, 2014).

En el desarrollo de software, los ingenieros y diseñadores a veces tienen problemas con la refactorización para implementar cambios impredecibles en las plataformas de tecnología o el negocio de la organización, esto se debe a que generalmente la sostenibilidad no se considera en el diseño del software.

La longevidad de un sistema y su sostenibilidad están intrínsecamente relacionadas, estos dos factores, son esencialmente dos lados del mismo problema. La sostenibilidad en la arquitectura de software se puede lograr a través de buenas decisiones de diseño que se conservan a largo plazo. La sostenibilidad en la arquitectura de software es el conjunto de factores que promueven la estabilidad y la longevidad de una arquitectura durante la evolución del sistema (Capilla, Nakagawa, Zdun, 2017).

La Ingeniería de Requisitos es un punto clave para los profesionales que desean desarrollar sistemas de software sostenibles. La Ingeniería de Requisitos para la Sostenibilidad (RE4S) se puede definir como la utilización de requisitos de ingeniería, en conjunto con técnicas de desarrollo para mejorar la sostenibilidad ambiental, social y económica de los sistemas de software y sus efectos directos e indirectos en el negocio. Para esto el RE4S se basa en varios modelos para la determinación de estos requisitos. (Penzenstadler, 2018).

Algunos modelos para la determinación de RE4S son el Modelo de Negocio Sostenible Canvas y el Modelo de Partes Interesadas. El Modelo Canvas fue desarrollado por Osterwalder y Pigneur. Según este modelo se puede describir a través de la construcción de bloques básicos que muestran la lógica de cómo una empresa puede generar dinero. Los nueve bloques están dados por (Osterwalder, Pigneur, 2010).

1. Segmento de clientes
2. Propuesta de valor
3. Canales de comunicación y ventas
4. Relaciones con clientes
5. Flujo de ingreso
6. Recursos clave

7. Actividades clave
8. Socios clave
9. Estructura de costes

Estos cubren las cuatro áreas principales de un negocio: clientes, oferta, infraestructura y viabilidad financiera.

El modelo de Partes Interesadas, define una parte interesada como una persona u organización que influye en los requisitos de un sistema o que se ve afectada por el mismo. Este modelo se compone de un diagrama y una matriz. El diagrama muestra una visión general de todas las partes interesadas activas y pasivas en un proyecto. Existen diferentes enfoques para identificar a las partes interesadas, entre ellos se tienen (Osterwalder, Pigneur, 2010).

- Fases: Análisis de los aspectos y fases de desarrollo del desarrollo de sistemas de software para encontrar los roles responsables.
- Lista de referencia: Creación de instancias de listas de referencia genéricas de las partes interesadas para el contexto concreto del proyecto.
- Contexto: Inspección del contexto empresarial y operativo del sistema en desarrollo, y la comprensión de qué roles concretos están involucrados.
- Objetivos: Analizar y refinar iterativamente un modelo de objetivo genérico y deducir los roles relacionados.

El Modelo de Metas es otro modelo para la determinación de RE4S. Las metas son objetivos de alto nivel del negocio, organización o sistema. Capturan las razones por las que se necesita un sistema y guían las decisiones en varios niveles dentro de la empresa. Un modelo de Metas es la base para la identificación temprana, la resolución de conflictos, y el modelado del comportamiento del sistema en diferentes niveles. Se pueden diferenciar tres tipos de objetivos (Anton, 1996):

- Objetivos de negocio: Son todos los objetivos relevantes para el negocio.
- Objetivos de uso: Tienen relación directa con el contexto funcional y el uso del sistema.

- **Objetivos relacionados con el sistema:** Son los objetivos relacionados con el sistema que determina las características del mismo.

La sostenibilidad de software es un tema que ha crecido últimamente en la comunidad científica, uno de los primeros trabajos es El Manifiesto Karlskrona para el Diseño Sostenible (KMSD), tiene sus raíces en el Tercer Taller Internacional Ingeniería de Requisitos para Sistemas Sostenibles (RE4SuSy) (Penzenstadler, B, 2012). Celebrado en Karlskrona, Suecia, el objetivo clave era combinar los diversos aspectos de la sostenibilidad con el fin de aclarar su alcance, objetivos y desafíos. El manifiesto reúne aportes en el estudio de la sostenibilidad de investigadores de diversas disciplinas en el campo de la ingeniería de software (Becker, 2014). El Manifiesto Karlskrona para el Diseño Sostenible incluye nueve principios de diseño sostenible, estos proporcionan la base para crear un punto de referencia para que las diferentes partes interesadas para aplicar durante el diseño de software (Becker, 2014).

Tabla 1. Descripción de los principios del Manifiesto de Karlskrona adaptado de (Becker, 2014).

Número Principio	Principio	Descripción
P1	La sostenibilidad es sistémica	La sostenibilidad nunca es una propiedad aislada. Requiere un terreno común interdisciplinar de sostenibilidad, así como una imagen global de la sostenibilidad dentro de otras propiedades.
P2	La Sostenibilidad tiene múltiples dimensiones	Se tiene que incluir esas dimensiones en el análisis si se quiere entender la naturaleza de la sostenibilidad en cualquier situación dada.
P3	La sostenibilidad trasciende múltiples disciplinas	Trabajar en sostenibilidad significa trabajar con personas de muchas disciplinas, abordando los desafíos desde múltiples perspectivas.
P4	La sostenibilidad es una preocupación independiente del propósito del sistema.	La sostenibilidad debe ser considerada incluso si el enfoque principal del sistema bajo diseño no es la sostenibilidad.
P5	La sostenibilidad se aplica tanto a un sistema como a sus contextos más amplios	Hay al menos dos esferas a tener en cuenta en el diseño del sistema: la sostenibilidad del propio sistema y cómo afecta a la sostenibilidad del sistema más amplio del que formará parte.
P6	La visibilidad del sistema es una condición previa necesaria y un habilitador del diseño de sostenibilidad.	Esforzarse por hacer visible el estatus del sistema y su contexto en diferentes niveles de abstracción y perspectivas.
P7	La sostenibilidad requiere acción en múltiples niveles	Buscar intervenciones que tengan más influencia en un sistema y considerar los costos de oportunidad: siempre que tome medidas hacia la sostenibilidad, considere si esta es la forma más efectiva de intervenir en comparación con acciones alternativas.

Número Principio	Principio	Descripción
P8	La sostenibilidad requiere satisfacer las necesidades de las generaciones futuras sin comprometer la prosperidad de la generación actual	La innovación en sostenibilidad puede ser un desacoplamiento de las necesidades presentes y futuras. Al alejarnos del lenguaje del conflicto y de la mentalidad de compensación, podemos identificar y promulgar opciones que beneficien tanto al presente como al futuro
P9	La sostenibilidad requiere pensamiento a largo plazo	Deben considerarse varios plazos, incluidos indicadores a largo plazo en la evaluación y las decisiones.

Los principios del manifiesto de Karlskrona pretenden ser una guía práctica para toda la comunidad, como lo es por ejemplo el Manifiesto Ágil (Fowler, M.; Highsmith, J 2001). Estos principios tienen como objetivo apoyar a las partes interesadas dentro de la industria y el mundo académico como pueden ser empresas, profesionales del software, investigadores y estudiantes para promover y desarrollar el diseño y las prácticas de sostenibilidad en el desarrollo de software (Christop, 2014). El Manifiesto de Karlskrona también sirve como un facilitador para pensar sobre los efectos generales del software en la sociedad y la necesidad de incorporar el pensamiento a largo plazo, la responsabilidad ética y la comprensión de cómo integrar la sostenibilidad en el diseño de sistemas de software (Becker, 2014).

Los principios de Karlskrona pueden relacionarse con las fases de desarrollo de software, la relación de estos con las fases de desarrollo de software proporciona una base para su uso, especialmente para los diferentes sistemas de software (Langer, 2016). La relación se muestra en la tabla 2.

Tabla 2. Principios del manifiesto de Karlskrona en relación con las fases del ciclo de vida del desarrollo de software adaptado de (Becker, 2014).

Fases de Diseño de Software	Principios del Manifiesto
Definición del proyecto	<p>P1. Esto garantiza que en la iniciación del proyecto se considere la sostenibilidad en la definición general del proyecto desde el principio.</p> <p>P2. La sostenibilidad del software tiene diferentes dimensiones que deben ser consideradas desde el principio para una mejor gestión de proyectos con diferentes partes interesadas.</p> <p>P3. El proyecto de software generalmente involucra a partes interesadas de diferentes dominios, incorporando sus preocupaciones de sostenibilidad proporciona una mejor gestión de esas preocupaciones desde múltiples perspectivas que pueden ayudar a la incorporación de la sostenibilidad para el software.</p>
Definición de los requisitos de usuario.	P2. Es importante tomar nota de los requisitos del usuario en relación con cada una de

	las dimensiones de sostenibilidad para tener un mejor análisis de sostenibilidad durante la fase de análisis y diseño
Definición de los requerimientos del sistema	P4. Durante la definición de los requisitos del sistema, los ingenieros de requisitos deben considerar la sostenibilidad como un factor a tener en cuenta. P5. Evaluar los impactos consecuentes de los requisitos de sostenibilidad del sistema y el entorno en el que funcionará el sistema
Análisis y Diseño	P2- La aplicación de este principio proporciona un plan para la evaluación del sistema desde todas las dimensiones de la sostenibilidad (económica, ambiental, social, individual y técnica). P4- En esta fase, este principio ayuda a fomentar el análisis y diseño del sistema basado en la sostenibilidad. P6- La aplicación de este principio permite una mejor visión del sistema desde diferentes niveles de abstracción. P8- Esto proporcionará una mejor comprensión durante el análisis para tomar mejores decisiones que ayudarán a los usuarios potenciales del sistema en el presente y en el futuro cuando el sistema evolucione.
Desarrollo	P2. Esto animará a los desarrolladores durante esta fase a considerar diferentes dimensiones de sostenibilidad, especialmente las dimensiones técnicas, sociales e individuales. P4. Fomentar la búsqueda de mejores vías para hacer sostenible el sistema desde la perspectiva del desarrollo (desarrolladores) teniendo en cuenta las funciones del sistema para ayudar a la longevidad.
Integración y Pruebas	P2 Proporciona integración y para que el equipo de pruebas tenga una plantilla de sostenibilidad que se pueda utilizar para probar el sistema para todas las dimensiones de sostenibilidad en función de la salida de los requisitos de sostenibilidad de las fases definición del proyecto. P4- La aplicación de este principio ayudará a considerar la sostenibilidad en esta fase, incluso si el enfoque principal del sistema no se centra en la sostenibilidad.
Implementación	P5. Proporciona razonamiento de antemano para que el equipo de desarrollo considere la sostenibilidad del sistema, su entorno de producción y propiciar el uso en vivo. P7. Sobre la base del principio 5 (P5), este principio ayudará a considerar buscar la participación de diferentes partes interesadas para hacer posible la actualización del sistema en el entorno de producción y cuando se impulse en vivo.
Sostenimiento / Mantenimiento	P9. Este principio en esta etapa ayuda a crear la conciencia para cuando el sistema esté en un entorno vivo, haya una evaluación continua para evaluar la sostenibilidad del sistema y pensar en formas de optimizar y mejorar la sostenibilidad del sistema desde las diferentes dimensiones

La relación de estos principios con las fases de desarrollo de software proporciona una base para comprender mejor su funcionamiento. Sin embargo, el manifiesto de Karlskrona se centró en principios en un alto nivel, lo que significa que es necesario ejemplificar los principios para mostrar su uso práctico. Las siguientes son las limitaciones de los principios citados (Oyedeji, S., Seffah, A., & Penzenstadler, B. 2018).

1. Los principios son abstractos y genéricos para servir a todos los posibles interesados en la sostenibilidad en todas las etapas de desarrollo y gestión del software.
2. Los principios se encuentran en un alto nivel de abstracción, faltan muchos detalles para su uso práctico.
3. Los principios están estrechamente relacionados, lo que dificulta el intercambio entre ellos.
4. Los principios no están relacionados con medidas tangibles, pero sirven como guía para crear medidas.
5. Los indicadores se utilizan para responder las preguntas como una forma de evaluar si se lograron las metas.
6. Uso actual del principio en software: cubre la aplicación actual del principio en el diseño y desarrollo de sistemas existentes, incluso si no se indica explícitamente en la documentación del sistema actual.
7. Uso futuro del principio en software: basado en la evaluación de la aplicación del principio actual en el diseño y desarrollo del sistema existente, se sugiere un uso potencial del principio en el diseño y mejora del sistema futuro (Oyedeji, S., Seffah, A., & Penzenstadler, B. 2018).

Para complementar el análisis de la sostenibilidad, Oyedeji, S., Seffah, A., & Penzenstadler, B (2018), propusieron El Catálogo de Diseño de Sostenibilidad de Software (SSDC) como una herramienta que puede facilitar la integración de la sostenibilidad en las prácticas de diseño, así como conducir a una mejor comprensión de la sostenibilidad por parte de los profesionales. El Catálogo es un conjunto de criterios derivados de los nueve principios del manifiesto de Karlskrona basados en el análisis cruzado de diferentes sistemas. Para cada criterio, también se derivan indicadores de sostenibilidad (Oyedeji, S., Seffah, A., & Penzenstadler, B. 2018).

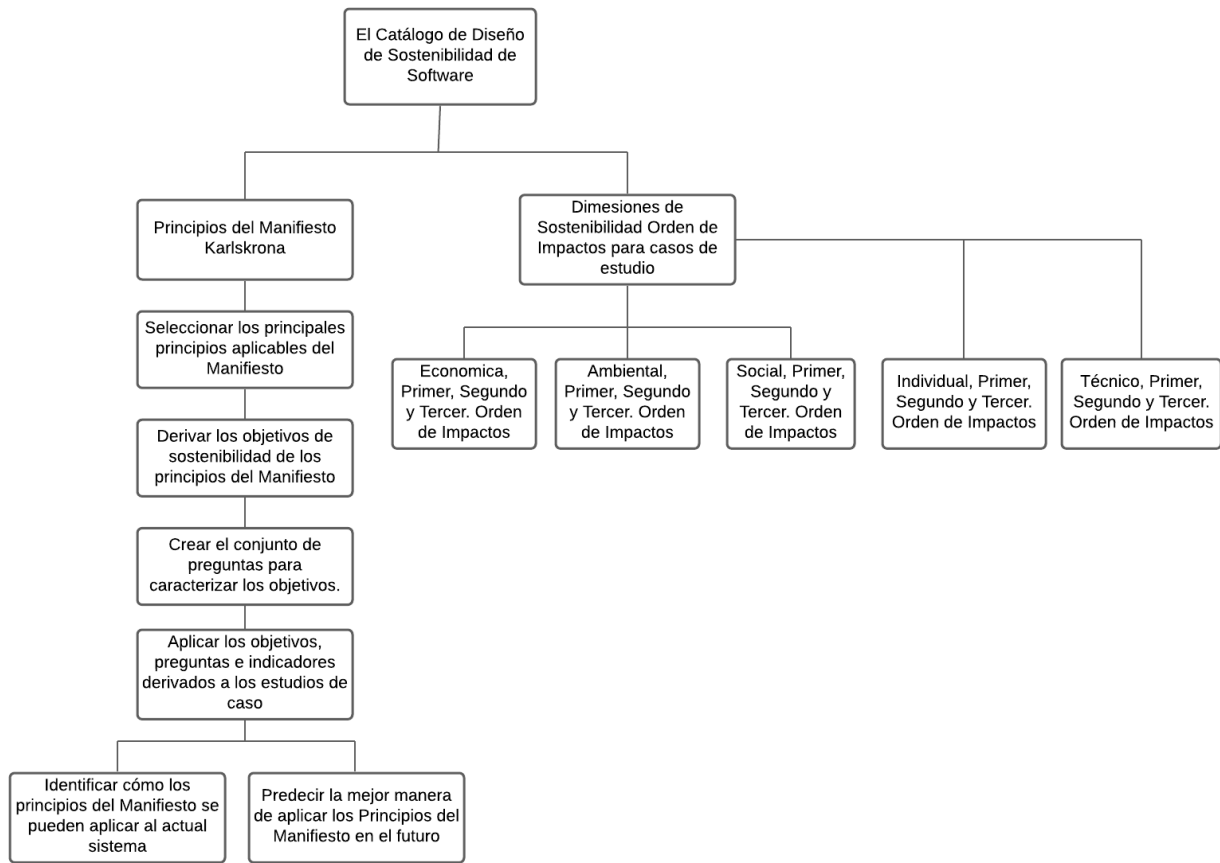


Figura 4. Flujo del Catálogo de diseño de sostenibilidad de software (SSDC) adaptado de (Oyedeji, S., Seffah, A., & Penzenstadler, B. 2018)).

El SSDC distingue principalmente dos componentes.

El primero es el análisis de sostenibilidad de sistemas, que es principalmente un conjunto de criterios para evaluar la sostenibilidad de los sistemas de software. Cada criterio se caracteriza por los siguientes elementos básicos que se utilizan para evaluar los sistemas de software:

1. Uno o más principios de Karlskrona los cuales se utilizan en la evaluación de cada categoría de sistema. No todos los principios se pueden aplicar necesariamente a todos los sistemas.
2. Objetivo/requisito: El resultado final deseado para cada categoría de sistema en función de la consideración de sostenibilidad.

3. Grupos de interés: los responsables de implementar los objetivos / requisitos.
4. Preguntas que caracterizan cada objetivo. De cada objetivo se derivan un conjunto de preguntas que determinarán si se está cumpliendo cada objetivo

El segundo componente son los indicadores de sostenibilidad asociados a cada criterio.

Estos indicadores están relacionados con las dimensiones de sostenibilidad y su orden de impactos (Oyedeki, S., Seffah, A., & Penzenstadler, B. 2018). Los órdenes de impacto cubren todos los efectos positivos y negativos del software en el ambiente, los cuales se descomponen en tres órdenes de magnitud. Los impactos de primer orden (efectos inmediatos) se refieren a los efectos directos del desarrollo y uso del sistema de software. Los impactos de segundo orden (efectos habilitadores) se refieren a los impactos indirectos relacionados con los efectos del uso del sistema de software en su dominio de aplicación. Los impactos de tercer orden (efectos estructurales) son los efectos acumulativos a largo plazo que resultan de la acumulación de impactos de primer y segundo orden a través del tiempo. (Becker, 2014). La sostenibilidad puede definirse en varias dimensiones (Becker, 2015).

- La dimensión individual abarca la libertad y el albedrío individuales (la capacidad de actuar en un entorno), la dignidad humana y la realización. Incluye la capacidad de las personas para prosperar, ejercer sus derechos y desarrollarse libremente.
- La dimensión social abarca las relaciones entre individuos y grupos. Por ejemplo, abarca las estructuras de confianza mutua y comunicación en un sistema social y el equilibrio entre intereses contradictorios.
- La dimensión económica abarca los aspectos financieros y el valor empresarial. Incluye el crecimiento de capital y la liquidez, las cuestiones de inversión y las operaciones financieras.
- La dimensión técnica cubre la capacidad de mantener y evolucionar sistemas artificiales (como el software) a lo largo del tiempo. Se refiere al mantenimiento y la evolución, la resiliencia y la facilidad de las transiciones del sistema.
- La dimensión ambiental abarca el uso y la administración de los recursos naturales. Incluye preguntas que van desde la producción inmediata de residuos y el consumo

de energía hasta el equilibrio de los ecosistemas locales y las preocupaciones relacionadas con el cambio climático.

Theresia Ratih Dewi Saputri y Seok-Won Lee proporcionan directrices sistemáticas para apoyar el desarrollo de sistemas de software sostenibles, un marco integrado que consta de una metodología de cuatro fases (Saputri, Lee, 2020). La primera fase está destinada a determinar los requisitos de sostenibilidad. El objetivo es desarrollar una metodología que defina los atributos de calidad y un modelo que satisfaga todas las dimensiones de sostenibilidad. En la segunda fase, se hace la descomposición de características, que incluye tareas, características y conjunto de clases. Estas características se descomponen en implementaciones funcionales para posteriormente ser analizadas. En la tercera fase se realiza la construcción de pruebas para la evaluación de la sostenibilidad mediante la definición de una línea base. Finalmente, en la última fase se realiza la evaluación de las pruebas definidas en la etapa anterior (Saputri, Lee, 2020).

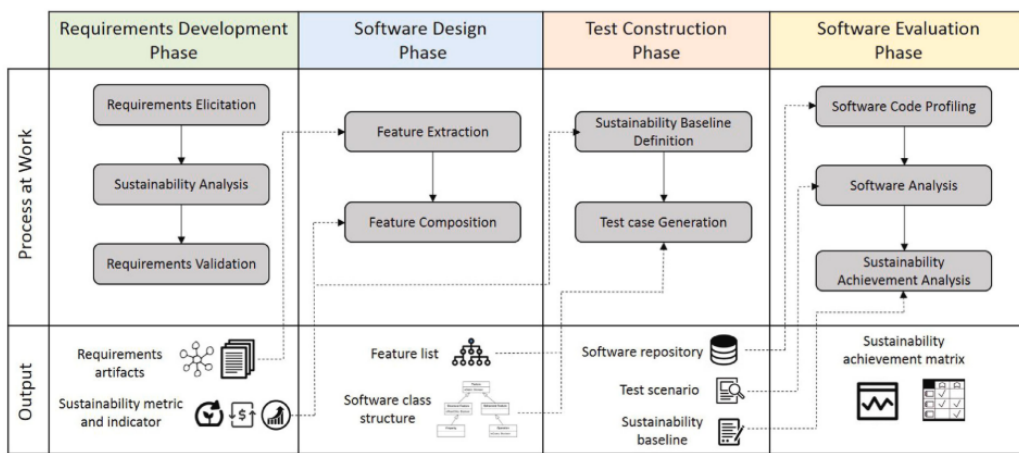


Figura 5. Sostenibilidad integrada a lo largo de todo el ciclo de vida de software (Saputri, Lee, 2020).

Para caracterizar la sostenibilidad de software se han utilizado los modelos de calidad de los programas informáticos estándar, como la norma ISO/IEC 25010. Hay varios trabajos que proponen diferentes características de sostenibilidad como la fiabilidad, la mantenibilidad, la portabilidad y la modificabilidad, el rendimiento, la usabilidad, la interoperabilidad y la adaptabilidad (SQuaRE, 2014). La evaluación de cada uno de estos parámetros o métricas puede darnos una idea del grado de sostenibilidad del software, estas se miden en función de atributos de calidad, al mejorar estas métricas aportan mayores

beneficios a las otras dimensiones como la económica, social y ambiental (Albertao, J. Xiao, C. Tian, Y. Lu, K. Q. Zhang, C. Liu, 2010).

2.2. Procesos de contratación de Software en el Sector Público.

Las Tecnologías de la Información y Comunicación en el Sector Público ayudan a desarrollar una gestión transparente. El software se ha convertido en un componente esencial para el éxito de las empresas. Por lo tanto, se debe prestar especial atención a este activo, además los servidores públicos son los responsables de dar cumplimiento a las políticas de adquisición de software que provee el estado (MinTIC Colombia. 2020). A continuación, se presentan algunas de recomendaciones o guías de los entes gubernamentales de Colombia y el mundo para la adquisición de software.

2.2.1. Procesos de contratación de Software en Colombia.

El Ministerio de Tecnologías de la Información y las Comunicaciones es el encargado de promover las políticas y proyectos en el sector de las tecnologías y la información (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, 2020). Tiene como función facilitar el acceso a las tecnologías de la información en todo el territorio colombiano. (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, 2020)

Colombia Compra Eficiente fue creada por el Gobierno Nacional por medio del Decreto Ley 4170 de noviembre 3 de 2011. Entre algunas de las funciones se tiene (Colombia Compra Eficiente, 2015):

- Dar soporte a las entidades públicas en procesos de compra y contratación.
- Elaborar políticas que permitan la gestión del mercado en oferta y demanda.
- Promover políticas que faciliten la eficiencia en compras públicas.

El Ministerio de las Tecnologías de la Información y Comunicaciones – MinTIC y Colombia Compra Eficiente elaboraron una guía con recomendaciones considerando los aspectos del proceso de compra y contratación. Esto con el fin de garantizar un mínimo de calidad en la adquisición de productos y servicios. (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente 2020.). La guía hace recomendaciones en términos económicos, técnico y legal para la compra de software según la Ley 80 1993, la Ley 1150 de 2011, el Decreto 1082 de 2015, la Ley 1882 del 2018 (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente 2020). Recomendaciones según la etapa contractual como planeación, selección, operación y liquidación (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente 2020).

- **Etapa de planeación:** En esta etapa se tiene en cuenta los requerimientos y necesidades para la adquisición de productos y servicios (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020).
- **En la etapa de selección:** Se debe tener en cuenta elementos de tipo técnico, económico y legal para la evaluación y selección del proveedor según las necesidades y requerimientos del proyecto. (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020).
- **Etapa de Operación:** El puede ser por Contrato de Licencia de Software, a través del cual, el desarrollador cede los derechos de uso de software a otra persona o entidad, definiendo las condiciones de pago, alcance y uso del mismo. Otro tipo de contrato es Contrato de Desarrollo de Software, donde el cliente contrata a una persona o empresa para la elaboración del producto a medida. (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020).
- **Etapa de Liquidación:** Se debe dar termino a la ejecución del contrato en esta etapa. Se debe determinar si se cumplieron con los requerimientos estipulados o si hay obligaciones pendientes. (Colombia Compra Eficiente, 2016).

En los procesos de compra y contratación de software se debe tener en cuenta un previo proceso de planeación, para establecer el mínimo técnico del producto, además de otros factores como el alcance, presupuesto y tiempo de ejecución. (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020).

Se recomienda a las Entidades del Estado en los procesos de compra y contratación hacer un estudio del mercado de los productos existentes según los requerimientos establecidos. El análisis debe estar enfocado en lo técnico, económico teniendo en cuenta un mínimo de calidad del producto (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020).

Se debe analizar la complejidad del producto para determinar los requerimientos necesarios para la ejecución del proyecto, para esto se debe tener en cuenta aspectos como (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020):

- Duración del proyecto.
- Costos del proyecto.
- Tamaño del equipo del proyecto.
- Composición del equipo del proyecto.
- Desempeño del equipo del proyecto.
- Urgencia e importancia del proyecto.
- Flexibilidad en el alcance.
- Recursos y tiempo.
- Claridad en el problema u oportunidad.
- Importancia estratégica del proyecto.
- Nivel de impacto dentro de la organización.
- Riesgos
- Limitaciones y dependencias del proyecto.

En la determinación de requerimientos se debe tener en cuenta, ciertos aspectos como metodologías y procedimientos. Esto con el fin de que los proveedores se ajusten a las necesidades del proyecto de acuerdo a la metodología del mismo. (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020).

En los documentos de estudios de mercado y pliego de condiciones se debe tener claro los requerimientos del proyecto para que los proponentes puedan elaborar una oferta específica con el fin de garantizar un mínimo de calidad del producto y asegurar los tiempos de ejecución del proyecto (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020).

En el proceso de planeación se debe tener en cuenta los posibles riesgos del proyecto, como pueden ser, retrasos en tiempos de ejecución, incumplimiento, cambios de personal etc. (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020).

Para participar en el proceso de contratación con las entidades del sector público las empresas privadas deben cumplir ciertos requisitos.

El requisito habilitante es el que determina si el proponente puede ser parte del proceso de contratación, se debe tener en cuenta aspectos económicos, financiero, jurídico, experiencia etc. (Colombia Compra Eficiente, 2014). Según el párrafo 2 del artículo 5 de la Ley 1150 del 2017 La calidad de software no está definida por sus certificaciones a nivel internacional, en consecuencia, no está permitido los requerimientos de certificaciones internacionales como requisito habilitante. Colombia Compra Eficiente y MinTIC recomiendan garantizar un mínimo técnico de calidad, por lo cual se podrá solicitar al proveedor que cuente con una metodología para el ciclo de vida del software. (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020).

Las Entidades en los procesos de compra y contratación debe tener en cuenta el Código Estándar de Productos y Servicios de Naciones Unidas (UNSPSC), el cual es “una metodología uniforme de codificación utilizada para clasificar productos y servicios fundamentada en un arreglo jerárquico y en una estructura lógica” (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, 2020). Según el proyecto a ejecutar el proveedor deberá certificar el registro, ya sea personal natural o jurídica que aspire a realizar contratos con el estado en el Registro Único de Proponentes (Cámara de Comercio de Bogotá, 2020).

Las entidades públicas deberán definir procesos de evaluación con base a los requerimientos del proyecto con el fin de que los proveedores puedan soportar las capacidades técnicas del producto. Se debe tener en cuenta un mínimo de calidad del producto. (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020).

MinTIC sugiere tener en cuenta para procesos de compra y contratación de software las modalidades de licitación pública, mínima cuantía o concurso de méritos. No se recomienda procesos de compra por Selección Abreviada por Subasta Inversa por posibles problemas de ejecución de contrato. (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente 2020).

Las siguientes son algunas de las normas para la compra y contratación de software en Colombia.

- Ley 23 de 1982, fundamenta la protección jurídica al Software. Decreto 1360 de 1989, incorpora el concepto de Software en la legislación colombiana.
- Ley 565 de 2000, en la cual se aprueba el "Tratado de la OMPI -Organización Mundial de la Propiedad Intelectual- sobre Derechos de Autor (WCT)", adoptado en Ginebra, el veinte (20) de diciembre de mil novecientos noventa y seis (1996) en la cual se establece expresamente que “los programas de ordenador están protegidos

como obras literarias en el marco de lo dispuesto en el Artículo 2 del Convenio de Berna” (García, 2007).

- Decisión 486 de 2000, se refiere a los requisitos de patentabilidad.
- Ley 1450 de 2011 se refiere a la naturaleza de los contratos de cesión de derechos patrimoniales de autor. (Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC, Colombia Compra Eficiente, 2020).

Los siguientes son algunas de las guías para procesos de compra y contratación de las tecnologías de la información en algunos países.

2.2.2. Otros Procesos de Contratación.

2.2.2.1. Navegando por la adquisición de software en una industria "como servicio". Guía práctica para profesionales de las TIC y la adquisición. - Australia.

Navigating Software Procurement in an “as-a-Service” Industry. A Practice Guide for ICT and Procurement Professionals. Su traducción al español Navegando por la adquisición de software en una industria "como servicio" Guía práctica para profesionales de las TIC y la adquisición. Es una guía para ayudar a altos directivos de las agencias gubernamentales de TI australianas a optimizar la adquisición de software. Desarrollada por Business Aspect, una empresa australiana de consultoría empresarial y tecnológica en colaboración con las agencias gubernamentales (Business Aspect, 2012).

Esta guía comprende tres partes principales (Business Aspect, 2012).

A. Un inventario de requisitos federales de adquisiciones.

En esta parte de la guía se delinean los 50 requisitos que todo el gobierno debe considerar de manera regular durante la toma de decisiones sobre las TIC, incluida la contratación de software.

Los organismos que deseen adquirir programas de software, independientemente del tipo de software, deben adherirse al Commonwealth Procurement Guidelines del

gobierno de Australia, haciendo especial hincapié en los siguientes principios generales de contratación.

1. Valor por dinero.
2. Fomentar la competencia.
3. Uso eficiente, eficaz y ético de los recursos.
4. Responsabilidad y Transparencia

Si el proceso de adquisición supera los \$80.000 dólares se deben seguir los siguientes procedimientos obligatorios.

1. Umbrales de adquisición.
2. Valorar la contratación.
3. Acercarse al mercado, incluyendo Licitación Abierta, Listas Multiusos, Licitación Selecta, Abastecimiento Directo, Paneles y Adquisiciones de Agencias Cooperativas.
4. Solicitar documentación.
5. Condiciones para la participación.
6. Límites de tiempo mínimos.
7. Recibo y apertura de presentaciones
8. Adjudicación de contratos
9. Notificación de decisiones

B. Establecimiento de un enfoque único para la adquisición de software.

Esta parte proporciona una manera simplificada de reunir todos los requisitos pertinentes en un "marco unificado" que permita a los profesionales trabajar a través del proceso de adquisición de una manera bien estructurada, que respete todos los requisitos de contratación necesarios (Business Aspect, 2012).

La aplicación del marco unificado de adquisición de software se puede implementar de la siguiente manera (Business Aspect, 2012).

1. Compartir las definiciones de los distintos tipos de software con cualquier empresa o personal de TIC que se embarque en un esfuerzo de adquisición de software.
2. Aplicación de las listas de verificación durante el proceso de contratación, en particular durante el desarrollo de la Solicitud de Licitación.
3. Realizar la evaluación del riesgo durante la evaluación de la licitación para garantizar que el organismo esté plenamente informado sobre el vínculo entre una solución determinada, sus perfiles de riesgo y financieros.

C. Herramientas Prácticas y Listas de Verificación.

Esta parte proporciona un conjunto de preguntas útiles, plantillas y cláusulas modelo que los gerentes de TIC y los profesionales de compras pueden utilizar como inserciones en la licitación y como parte del proceso general de toma de decisiones de adquisiciones. Estos se basan en modelos de evaluación de costos, valor y riesgos bien aceptados que son agnósticos de productos y tecnología.

Las opciones típicas disponibles en el mercado australiano hoy en día para adquirir software de negocios se resumen de la siguiente manera (Business Aspect, 2012).

- Implementación de una solución local: Desarrollo de software personalizado o a la medida.
- Utilizar un proveedor de servicios de aplicaciones.
- Suscribirse a un software como servicio.

2.2.2.2. Guía Del Usuario para la Adquisición de Productos y Servicios De Tecnología de La Información - Banco Asiático de Desarrollo.

El Banco Asiático de Desarrollo (BAD) ha adoptado tres procedimientos a partir de los cuales el Comprador puede seleccionar para licitar productos y servicios de tecnología de la información (TI). Los procedimientos de licitación son los siguientes (Asian Development Bank, 2018):

- Adquisición de una Etapa - Una capa: Es el principal procedimiento de licitación utilizado para la mayor parte de la contratación financiada por el BAD.

- Adquisición de una Etapa - Dos capas: Permite evaluar las ofertas por motivos puramente técnicos y administrativos sin referencia del precio.
- Adquisición de dos Etapas: Permite al Comprador discutir las ofertas técnicas de la primera etapa con los licitadores y revisar las especificaciones, con ofertas de precios y ofertas técnicas revisadas a presentar en la segunda etapa.

Esta guía fue diseñada con los siguientes objetivos (Asian Development Bank, 2018):

1. Simplificar la preparación para el Comprador de un documento de licitación específico para la adquisición de productos y servicios de TI.
2. Reducir el tiempo y el esfuerzo de licitación por parte de los preparadores.
3. Facilitar y simplificar la evaluación y comparación del Comprador de las ofertas.
4. Minimizar el tiempo requerido por el BAD para la revisión previa del Documento de Licitación.

La adquisición de una sola etapa es adecuada para productos y servicios de TI sencillos. Si el Comprador considera que la relación calidad-precio es la apropiada se logrará adjudicando el contrato al licitador que ofrezca el precio de oferta evaluado más bajo. Si el Comprador considera que la relación calidad-precio se logrará únicamente mediante el uso de una metodología de puntuación, se dará mayor importancia a los aspectos de calidad para determinar el licitador al que se le concederá el contrato. La adquisición de una sola etapa no es adecuada para sistemas complejos, aplicaciones y proyectos de gran tamaño. Para estas adquisiciones, se debe utilizar la contratación en dos etapas (Asian Development Bank, 2018).

Entre los productos y servicios de TI más adecuados para las ofertas en dos etapas se incluyen los siguientes (Asian Development Bank, 2018):

- Soluciones empresariales complejas (por ejemplo, un sistema integrado de banca comercial, un sistema de gestión de tesorería, etc.).
- Un sistema que requiere un amplio desarrollo de software.

- Tecnologías de la información complejas (por ejemplo, equipos de procesamiento de datos a gran escala; sistemas que incluyen equipos altamente especializados como estaciones de trabajo gráficas, plotters, equipos fotográficos como para catastros avanzados; sistemas que requieren telecomunicaciones de alta velocidad a gran escala).
- Sistemas que implican amplios servicios técnicos para el diseño, desarrollo, personalización, instalación, capacitación, operaciones y soporte técnico.
- Una combinación de lo anterior.

3. METODOLOGÍA

El trabajo se realizará con base en una versión adaptada de la metodología Design Science (Peppers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V. & Bragge, J., 2006), que comprende:

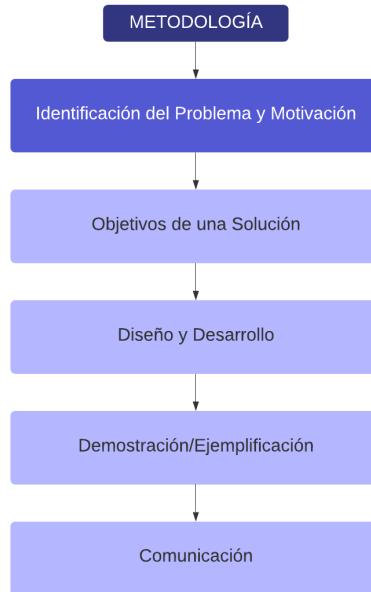


Figura 6. Esquema Gráfico de la Metodología.

3.1. Identificación del Problema y Motivación

Consiste en la definición y planteamiento del problema buscando en la literatura la sustentación del mismo, así como la identificación de posibles soluciones o trabajos de investigación adelantados en el tema seleccionado que permitan validar el enfoque seleccionado.

A pesar del avance de la tecnología, Colombia Compra Eficiente en su estudio de la oferta y la demanda de adquisición de Software en Entidades Estatales encontró potencial para mejorar la forma en cómo se adquieren dichos servicios (MinTIC Colombia. 2020), ¿cómo mejorar los procesos de compra y contratación para estos? El planteamiento de un método que permita evaluar técnicamente ciertos atributos de software como la sostenibilidad podría mejorar la forma en que se adquieren dichos servicios. En el numeral 1.1 se describe el planteamiento del problema,

3.2. Objetivos de una Solución

Consiste en definir los objetivos de la solución a partir de la definición del problema. El objetivo principal de este trabajo es proponer un método de evaluación de la sostenibilidad de software para los procesos de compra y contratación en las entidades del sector público. Los objetivos generales y específicos se encuentran en el numeral 1.2 de este documento.

3.3. Diseño y Desarrollo

Consiste en la creación de la posible solución. Esta puede ser potencialmente, construcciones, modelos, métodos o instancias.

Esta etapa la constituyen las actividades de construcción del marco teórico y estado del arte según la revisión bibliográfica. Posteriormente, se consultan los métodos de evaluación actuales, propuestas y trabajos desarrollados, identificando los aspectos y características de calidad que contribuyan a la sostenibilidad del software. La generación del método de evaluación será el resultado del análisis de las características de calidad enfocados a la calidad de software y a los procesos de compra y contratación en las entidades del sector público. Los resultados de esta etapa se evidencian en el numeral 4 del presente documento.

3.4. Demostración

Consiste en demostrar la eficacia de la posible solución para resolver el problema. Esto podría involucrar en su experimentación, simulación, un caso de estudio, prueba u otra actividad apropiada.

La demostración del método propuesto se hará a través de la simulación de un proceso de compra y contratación donde la aplicación del método nos determinará la mejor opción para el uso de los recursos, también se hará una validación externa a través de encuesta con personal experto en calidad de software y personal del sector público. Los resultados de esta etapa se encuentran en el numeral 4.6 de este documento.

3.5. Comunicación

Determinación e interpretación de los hallazgos obtenidos del trabajo realizado en relación con los objetivos determinados, consolidando los resultados, y realizando la entrega del artículo, un informe de investigación con el modelo propuesto y la sustentación del trabajo.

4. RESULTADOS Y ANÁLISIS

4.1 ESTADO DEL ARTE

4.1.1. Sostenibilidad de Software

La incorporación del concepto de sostenibilidad en la disciplina de la ingeniería de software implica la consideración del impacto del software en el entorno circundante a lo largo de las etapas del ciclo de vida: desarrollo, operación y mantenimiento (Calero, 2015). Estos impactos se pueden caracterizar en diferentes formas (Lago, 2015), la manera más común de abordar la sostenibilidad del software es a través de la eficiencia energética, sin embargo, se pueden considerar otros aspectos (Naumann, 2011). Se han realizado discusiones sobre la noción de sostenibilidad, Venters debatió si la sostenibilidad puede considerarse como un requisito funcional o una propiedad emergente, se concluye que se trata de un concepto polifacético y que se debería caracterizar cuantitativamente (Venters, 2014).

La sostenibilidad durante el desarrollo de software para su análisis requiere una “descomposición tangible del concepto de sostenibilidad” (Penzenstadler, 2013) la cual puede expresarse en varias dimensiones. Diferentes propuestas de investigación abordan la dimensión económica, social, ambiental y técnica (P. Lago, 2013) como mínimo a tener en cuenta, debido al hecho que “la sostenibilidad es alcanzable sólo cuando se tienen en cuenta todas las dimensiones” (P. Lago 2015). Desde un punto de vista práctico, las dimensiones de la sostenibilidad se pueden estudiar por separado antes de explorar una visión integrada (C. Becker, R. Chitchyan, L. Duboc, S. Easterbrook, B. Penzenstadler, N. Seyff, C.C. Venters, 2015).

En ingeniería una de las tareas importantes es comprender la naturaleza de los sistemas de software y su impacto en el desarrollo sostenible. A pesar de que la sostenibilidad es una de las preocupaciones importantes en la sociedad, considerar el diseño de esta no es una tarea trivial (Becker, 2015). Existen varios desafíos al considerar el diseño de la sostenibilidad, uno de ellos es la falta de un método para su evaluación. (Chitchyan, 2016).

La evaluación de la calidad del software como tal no es nueva, sin embargo, la evaluación basada en la noción de sostenibilidad como una propiedad de la calidad del software está aún en crecimiento y son muy pocos los estudios sobre ello. Por lo tanto, la forma en que se debe evaluar el software enfocado a la sostenibilidad está todavía en desarrollo (Condori-Fernandez y P. Lago, 2018).

La sostenibilidad del software se manifiesta intrínsecamente con el tiempo, por lo tanto, desde una perspectiva técnica se ha vinculado siempre a la noción de evolución del software. Sin embargo, este concepto es mucho más amplio (Avgeriou, 2013), ya que no debe centrarse solo en lo técnico, sino que debe considerar otros factores como lo ambiental, técnico y social (Fernandez, 2019).

Diferentes autores han propuesto varios enfoques para evaluar la sostenibilidad de un sistema de software. Kienzle en trabajo con otros autores, proponen un enfoque basado en modelos para la evaluación de la sostenibilidad. Este se enfocó en integrar el uso de la Ingeniería basada en modelos para incorporar varios puntos de vista para la toma de decisiones de sostenibilidad (Kienzle, Mussbacher, Combemale, Bastin, Bencomo, Bruel, Becker, Betz, Chitchyan, Cheng. 2019)

Fernández en trabajo con otros autores propusieron la evaluación de la sostenibilidad mediante un modelo de calidad. El enfoque se evalúa mediante la realización de una estrategia de investigación empírica en un proyecto de software. El resultado de estudio dio como resultado que la sostenibilidad tiene naturaleza de múltiples dimensiones (Fernandez, Lago, Luaces, Places, Folgueira, 2019)

Lami y Buglione proponen un método para medir la sostenibilidad utilizando un enfoque orientado a objetivos. Proponen cuatro dimensiones de medición de la sostenibilidad como lo son la infraestructura, personas, procesos y productos. (Lami, Buglione, 2012)

Los modelos de calidad de software estándar como ISO / IEC 25010 se utilizan para caracterizar la sostenibilidad del software. Hay varios trabajos que proponen diferentes

características de sostenibilidad como Fiabilidad, Mantenibilidad, Portabilidad y Modificabilidad, Desempeño, Usabilidad, y Adaptabilidad. (ISO 25000 Calidad del producto de software, 2014)

Johann y Kern proponen métodos que demuestran una forma práctica de evaluar los criterios de sostenibilidad de un sistema de software. El primer autor propuso una métrica genérica para medir el software y un método para aplicarlo en un proceso de ingeniería de software. Aunque este trabajo presentó la medición detallada de la sostenibilidad, solo se centró en el aspecto de consumo de energía de software El segundo autor propuso un Modelo de Calidad para Software Verde y Sostenible (Johann, 2012, Kern, 2013)

Algunos autores listan criterios que se pueden utilizar para medir la sostenibilidad, entre ellos el aspecto energético. Sin embargo, enfocarse solo en el consumo de energía no es adecuado para la evaluación de la sostenibilidad. Albertao en trabajo con otros autores muestra cómo cada métrica y medida podría aplicarse para evaluar el proyecto de software en función de sus beneficios en las dimensiones de sostenibilidad (Albertao F, Xiao J, Tian C, Lu Y, Zhang K.Q, Liu C. 2010). Evaluar el impacto de los requisitos de calidad en la sostenibilidad es el primer paso hacia el desarrollo de sistemas de software que satisfagan las preocupaciones de diseño de sostenibilidad (Condori-Fernandez, P. Lago, 2018).

Teniendo en cuenta la naturaleza multidimensional de la sostenibilidad, los esfuerzos están orientados a la definición de un modelo de sostenibilidad de software con el propósito de caracterizar cada dimensión de sostenibilidad. Según la norma ISO/IEC 25010 los atributos de calidad permiten caracterizar la calidad de software. Con base en esto, Condori Fernández y Lago realizaron un estudio donde se investigó la contribución y la relevancia de los atributos de calidad desde diferentes perspectivas. Por definición, un atributo de calidad es “el requisito que especifica el grado de afectación de la calidad de un sistema o software” (IEEE 24765, 2017), y dado que un atributo de calidad puede contribuir a una o más dimensiones de sostenibilidad, Fernández y Lago identificaron las dimensiones donde éste tiene más relevancia (N. C. Fernandez, P. Lago, 2018).

En la identificación de los atributos de calidad de software que pueden contribuir a la sostenibilidad del software, Fernández y Lago los caracterizaron según una lista de verificación definida por los modelos de calidad de la norma ISO/IEC 25010 (N. Condori-Fernández, P. Lago, 2018), Estos se centran en el modelo de calidad del producto y el modelo de calidad de uso.

A continuación, se muestran los principales hallazgos sobre la contribución de los atributos de calidad por dimensión de sostenibilidad social, técnica, económica y ambiental. Para determinar esta contribución, Fernández y Lago en su estudio evaluaron la importancia y relevancia percibida de cada dimensión de sostenibilidad de diferentes fuentes. Las siguientes tablas muestran el análisis de estabilidad realizado para comparar los niveles de contribución de atributos a la sostenibilidad, en negrilla se muestra el atributo de calidad más relevante para cada dimensión de sostenibilidad. La columna Ronda 1&2 de las tablas corresponde al resultado del análisis realizado en primera instancia y la columna Ronda 3 corresponde a una validación realizada por Condori-Fernández y Lago. De acuerdo a las rondas, el autor asigna niveles de contribución los cuales se expresan en una escala de 4 puntos: 1 altamente contributiva, 2 contributiva, 3 ligeramente contributiva y 4 no contributiva. La columna puntaje se asigna de la siguiente forma (N. Condori-Fernández, P. Lago, 2018):

- 2 (dos) puntos si no hay diferencia entre la ronda 1&2 y la ronda 3.
- 1 (un) punto si hay una unidad de diferencia entre la ronda 1&2 y la ronda 3.
- 0 (cero) puntos si la diferencia entre la ronda 1&2 y ronda 3 es mayor o igual a 2.

La dimensión de Sostenibilidad Social se centra en garantizar que las generaciones actuales y futuras tengan el mismo o mayor acceso a los recursos sociales mediante la búsqueda de equidad generacional (Fernandez, Lago, 2019). Según el estudio realizado la característica más importante para esta dimensión es la Seguridad y el atributo más relevante es la **Confidencialidad**. Otras características que aportan a la sostenibilidad social son la autenticidad, la satisfacción, la confianza y la utilidad. En menor grado de contribución la

usabilidad en términos de los atributos de operatividad y protección contra errores del usuario (N. Condori-Fernandez, P. Lago, 2018).

Tabla 3. Atributos de calidad para la sostenibilidad social. (N. Condori-Fernandez, P. Lago, 2018).

Características	Atributos de Calidad	Ronda 1&2	Ronda 3	Puntaje
Seguridad	Confidencialidad	1	1	2
Seguridad	Autenticidad	1	2	1
Seguridad	Responsabilidad	1	4	0
Satisfacción	Confiabilidad	1	2	1
Libertad de Riesgos	Mitigación de Riesgo para la salud	2	1	1
Seguridad	Integridad	2	1	1
Eficacia	Eficacia	2	1	1
Satisfacción	Usabilidad	2	2	2
Usabilidad	Operabilidad	2	4	0
Compatibilidad	Interoperabilidad	2	2	2
Libertad de Riesgo	Mitigación de riesgo ambiental	2	3	1
Usabilidad	Protección contra errores de usuario.	2	4	0
Usabilidad	Aprendizaje	2	3	1
Accesibilidad	Accesibilidad	2	1	1
Usabilidad	Capacidad de reconocimiento	3	4	1
Compatibilidad	Coexistencia	3	2	1

La accesibilidad y la libertad de riesgo son características adicionales de calidad que están relacionadas con la usabilidad y la satisfacción. Mediante la accesibilidad, el sistema de software puede ser utilizado por muchas más personas, con esto la igualdad y la equidad se favorecen positivamente. Sobre la libertad de riesgo, los sistemas de software también deben mitigar el riesgo potencial para las personas en el ámbito de uso. Por ejemplo, un juego de software puede ser muy usable, pero también tiene propiedades adictivas que animan a los usuarios a jugar más. (N. Condori-Fernandez, P. Lago, 2018).

La dimensión de la Sostenibilidad técnica busca apoyar el uso a largo plazo, la evolución o adaptación adecuada de los sistemas de software en entornos de ejecución en

constante cambio (Fernandez, Lago, 2019), **La Interoperabilidad** el atributo más importante para esta dimensión, seguida de la corrección funcional y la disponibilidad. Otro requisito importante es la confiabilidad en términos de disponibilidad, tolerancia a fallas y recuperabilidad. Esto conlleva a que la durabilidad sea un factor mucho más importante, al permitir que un sistema de software pueda realizar funciones específicas en ciertas condiciones durante un largo período de tiempo (N. Condori-Fernandez, P. Lago, 2018).

Tabla 4. Requisitos de calidad para la sostenibilidad técnica (N. Condori-Fernandez, P. Lago, 2018).

Características	Atributos de Calidad	Ronda 1&2	Ronda 3	Puntaje
Adecuación Funcional	Corrección funcional	1	2	1
Compatibilidad	Interoperabilidad	1	1	2
Fiabilidad	Disponibilidad	1	4	0
Adecuación Funcional	Pertinencia funcional	1	2	1
Satisfacción	Utilidad	2	4	0
Fiabilidad	Tolerancia a fallos	2	1	1
Mantenibilidad	Modificabilidad	2	1	1
Satisfacción	Confiar	2	1	1
Cobertura de Contexto	Complejidad del contexto	2	4	0
Eficacia	Eficacia	2	2	2
Robustez	Robustez	2	2	2
Portabilidad	Adaptabilidad	2	1	1
Eficiencia de rendimiento	Comportamiento temporal	2	4	1
Mantenibilidad	Modularidad	2	1	1
Mantenibilidad	Probabilidad	2	2	2
Fiabilidad	Recuperabilidad	2	1	1
Compatibilidad	Coexistencia	2	2	2
Fiabilidad	Madurez	2	3	2
Eficiencia	Eficiencia	3	3	2
Supervivencia	Capacidad de supervivencia	3	2	1
Eficiencia de rendimiento	Capacidad	3	2	1

La dimensión de la sostenibilidad económica tiene como objetivo garantizar que los sistemas de software puedan crear valor económico (Fernández, Lago, 2019), **La Efectividad** es el mejor contribuyente a esta dimensión, ver tabla 5. La disponibilidad, usabilidad y adecuación funcional también hacen su contribución en menor medida a esta dimensión.

Tabla 5. Requisitos de calidad para la sostenibilidad económica (N. Condori-Fernandez, P. Lago, 2018).

Características	Atributos de Calidad	Ronda 1&2	Ronda 3	Puntaje
Efectividad	Efectividad	1	1	2
Fiabilidad	Disponibilidad	1	2	1
Satisfacción	Confiar	2	2	2
Satisfacción	Utilidad	2	1	1
Libertad de riesgo	Mitigación del riesgo económico	2	1	1
Cobertura de contexto	Compleitud del contexto	2	3	1
Cobertura de contexto	Flexibilidad	2	2	2
Adecuación funcional	Pertinencia funcional	2	1	1
Adecuación funcional	Corrección funcional	2	2	2
Fiabilidad	Recuperabilidad	2	1	1
Eficiencia	Eficiencia	3	2	1
Adecuación Funcional	Compleitud Funcional	3	2	1

La dimensión de la sostenibilidad ambiental busca evitar que el uso de sistemas de software dañe el medio ambiente en el que operan (Fernandez, Lago, 2019). Para esta dimensión la característica más importante es la **Mantenibilidad** y el atributo más relevante es la **Reusabilidad** y la modificabilidad en una menor medida, ver tabla 6. (N. Condori-Fernandez, P. Lago, 2018).

Tabla 6. Atributos de calidad para la sostenibilidad ambiental. (N. Condori-Fernandez, P. Lago, 2018).

Características	Atributos de Calidad	Ronda 1&2	Ronda 3	Puntaje
Mantenibilidad	Reusabilidad	1	1	2
Mantenibilidad	Modificabilidad	1	4	0
Eficiencia de Rendimiento	Utilización de recursos	2	1	1
Libertad de Riesgo	Mitigación de riesgos ambientales	2	3	1
Eficiencia de Rendimiento	Comportamiento del tiempo	2	3	1
Fiabilidad	Disponibilidad	2	2	2
Eficiencia	Eficiencia	3	1	0
Compatibilidad	Convivencia	3	1	0

4.1.2. Estándares y Guías en Procesos de Adquisición de Software de compras.

La importancia del proceso de adquisición de software es mejorar la calidad de los productos o servicios que ofrece la organización y reducir los costos, en cuanto a modelos de calidad para procesos de compra y contratación software existen diferentes modelos que se pueden aplicar.

4.1.2.1. ISO/IEC 12207

Esta normal provee una referencia para mejorar procesos del ciclo de vida de software, se puede aplicar a compra, suministro, desarrollo entre otras actividades (International Organization for Standardization, 2017). Este estándar se puede utilizar en uno o más de los siguientes modos: (International Organization for Standardization, 2017):

- Por una organización, para ayudar a establecer un entorno de procesos deseados. Estos procesos pueden ser apoyados por una infraestructura de métodos, procedimientos, técnicas, herramientas y personal capacitado.
- Mediante un proyecto, para ayudar a seleccionar, estructurar y emplear los elementos de un entorno establecido para proporcionar productos y servicios

- Por un comprador y un proveedor, para ayudar a desarrollar un acuerdo relativo a los procesos y actividades.
- Por evaluadores de procesos, sirve como modelo de referencia de proceso para su uso en el desempeño de evaluaciones de procesos.

Dentro de las actividades del proceso de adquisición de este estándar se tiene: (International Organization for Standardization, 2017).

- Inicio. En esta etapa se definen y analizan los requerimientos del sistema.
- Preparación de la solicitud de propuestas.
- Preparación y actualización del contrato.
- Seguimiento del proveedor.
- Aceptación y finalización.
-

4.1.2.2. IEEE 1062-2019 - IEEE Recommended Practice for Software Acquisition.

Describe un conjunto de consideraciones de calidad útiles para aplicar durante uno o varios pasos de un proceso de adquisición de software.

Este estándar provee recomendaciones para cualquier tipo de software (Institute of Electrical and Electronics Engineers, 2019). Buscar estandarizar el proceso de compra en las empresas, teniendo en cuenta las características necesarias durante el proceso de adquisición. (Institute of Electrical and Electronics Engineers, 2019).

Las etapas recomendadas de este estándar son (Institute of Electrical and Electronics Engineers, 2019).

- Planear la estrategia de adquisición de software.
- Definir los requerimientos de adquisición de software.
- Identificar los proveedores potenciales
- Preparación de los requerimientos del contrato.
- Evaluación de la propuesta y selección del proveedor.
- Seguimiento del proveedor.

- Aceptación del software.
- Evaluar el proceso e Identificar oportunidades de Mejora.

4.1.2.3. Modelo CMMI-ACQ.

Desarrollado inicialmente por el Gobierno Estadounidense y el Instituto de Ingeniería de Software (SEI) para mejorar procesos en el ciclo de vida de software (Software Engineering Institute. 2011).

CMMI para la adquisición (CMMI – ACQ) se compone en varias categorías.

- La categoría de proceso de Gestión de proyectos cubre las actividades de gestión de proyectos relacionadas con la planificación, el seguimiento y el control de proyectos. (Software Engineering Institute. 2011).
- La categoría de proceso de Ingeniería de Adquisición establece un conjunto consistente de requisitos y acuerdos que se derivan de las necesidades de las partes interesadas y declaraciones de capacidad operativa para que los productos de trabajo desarrollados, productos entregados y servicios de proveedores cumplan con satisfacer las necesidades del usuario final (Software Engineering Institute. 2011).
- La categoría de proceso de soporte incluye los procesos y herramientas necesarios para permitir que los proyectos apliquen eficazmente técnicas de medición, control y decisión para administrar el proyecto (Software Engineering Institute. 2011).

4.1.2.4. Modelo ESCM-CL

El IT Services Qualification Center (ITSqc) liderado por la Universidad Carnegie Mellon ha desarrollado el eSourcing Capability Model for Client Organizations (eSCM-CL). Este es un modelo proporciona a las organizaciones clientes, un conjunto de mejores prácticas para evaluar y mejorar su capacidad de fomentar las relaciones con los proveedores de forma más eficaz y a su vez gestionar mejor dichas relaciones (Hefley, Bill and Loesche, Ethel A. and Khera, Pawan and Siegel, 2005).

El ESCM-CL permite a las organizaciones y clientes, evolucionar, mejorar e innovar su capacidad de desarrollar relaciones fuertes, durables y de confianza con sus proveedores de servicios, a la vez que satisface las necesidades del negocio. Este modelo tiene dos objetivos:

1. Proporcionar a las organizaciones clientes un camino a seguir que ayudaría a mejorar su capacidad a lo largo del ciclo de vida de la adquisición.
2. Ofrecer a las organizaciones cliente el objetivo principal de evaluar su capacidad de adquirir (Hefley, Bill and Loesche, Ethel A. and Khera, Pawan and Siegel, 2005).

4.2. DISEÑO Y DESARROLLO

La revisión de literatura permitió tener una visión general sobre los modelos y prácticas de caracterización de la sostenibilidad del software, así también, como modelos y guías de procesos de compra más comunes para la adquisición del software en el mundo. La sostenibilidad como se pudo observar, es uno de los temas que a lo largo de los años ha tomado gran importancia en el desarrollo de software y aunque no exista un consenso general de cómo caracterizarla, son varios los trabajos que se han desarrollado, en los cuales se ha tratado de integrar el concepto de sostenibilidad a lo largo de todo el ciclo de vida de software (Penzenstadler, 2018. Becker, 2015. Saputri, Lee, 2020. Condori-Fernandez y P. Lago, 2018).

Los trabajos consultados tratan de tener en cuenta ciertos parámetros para determinar la sostenibilidad. Al no existir un marco integrado de sostenibilidad para los procesos de compra y contratación, se propone tomar los factores comunes de los diferentes trabajos de sostenibilidad para la adquisición de software. Según el análisis de los trabajos de sostenibilidad, varios autores están de acuerdo en que hay que tener en cuenta diferentes dimensiones de sostenibilidad tales como, social, ambiental, económica y técnica. Para la caracterización de los atributos de calidad, muchos autores tomaron en cuenta el estándar ISO/IEC 25010 de Calidad de Producto y Calidad de Uso. La mayoría de los trabajos determinaron que algunas características de calidad del software se pueden utilizar para caracterizar la sostenibilidad, como se muestra en la tabla 8. Estas características a su vez se componen de subcaracterísticas y atributos de calidad, algunos más relevantes que otros. Condori-Fernandez y Lago en su estudio determinaron cuáles atributos de calidad del estándar ISO/IEC 25010 son más relevantes para las dimensiones de sostenibilidad de software. Por lo tanto, al evaluar los atributos de calidad más relevantes nos dará una idea del grado de sostenibilidad del software. El método propuesto es evaluar los atributos del modelo de Calidad de Producto del estándar ISO / IEC 25010 más relevantes para la sostenibilidad de software aplicables a los procesos de adquisición de software en las entidades del sector público.

4.3. CARACTERIZACIÓN DE MODELOS EXISTENTES

En la tabla 7 muestra los factores a tener en cuenta para la evaluación de la sostenibilidad de Software, desde el ciclo de vida del software, la fase de diseño, desarrollo, así como los atributos de calidad a tener en cuenta. Para la construcción de esta se tomaron en cuenta los factores mencionados en el numeral 4.2 Diseño y Desarrollo. Las columnas indican los diferentes trabajos de sostenibilidad consultados, y las filas las características de calidad y elementos a tener en cuenta para la sostenibilidad de software. Las X indican las características de calidad o elementos de sostenibilidad asociado a los trabajos para caracterizar la sostenibilidad de software.

Tabla 7. Trabajos de Sostenibilidad de Software.

Trabajos/Sostenibilidad	A Catalogue Supporting Software Sustainability Design. 2018	Integrated framework for incorporating sustainability design in software engineering life-cycle: An empirical study. 2018	Characterizing the contribution of quality requirements to software sustainability. 2018	Towards a Software Sustainability-Quality Model: Insights from a Multi-Case Study. 2019	Classifying the Measures of Software Sustainability. 2018
The Karlskrona Manifesto for Sustainability Design - Principles	X	X	X		
Sostenibilidad Social	X	X	X	X	X
Sostenibilidad Económica	X	X	X	X	X
Sostenibilidad Ambiental	X	X	X	X	X
Sostenibilidad Técnica	X	X	X	X	X
Adecuación Funcional			X	X	X
Fiabilidad		X	X	X	X
Eficiencia de Rendimiento		X	X	X	X
Facilidad de Uso		X	X	X	X
Seguridad			X	X	X
Compatibilidad		X	X	X	X
Mantenibilidad		X	X	X	X
Portabilidad		X	X	X	X
Efectividad			X	X	X
Eficiencia			X	X	
Satisfacción			X		
Utilidad			X		
Libertad de Riesgo			X		
Cobertura de Contexto			X		

A continuación, se muestran las características y atributos de calidad más relevantes basados en las dimensiones de sostenibilidad (N. Condori-Fernandez, P. Lago, 2018).

Tabla 8. Características y Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software.

Dimensión	Característica	Atributo	Relevancia
Social	Seguridad	Confidencialidad	Altamente Contribuyente
		Autenticidad	Contribuyente
		Integridad	Contribuyente
	Satisfacción	Confiabilidad	Contribuyente
	Libertad de Riesgos	Libertad de Riesgos Salud	Contribuyente
	Accesibilidad	Accesibilidad	Contribuyente
Ambiental	Mantenibilidad	Reusabilidad	Altamente Contribuyente
	Rendimiento Eficiencia	Utilización de Recursos	Contribuyente
Económica	Efectividad	Efectividad	Altamente Contribuyente
	Fiabilidad	Disponibilidad	Contribuyente
		Recuperabilidad	Contribuyente
	Libertad de Riesgo	Mitigación del Riesgo Económico	Contribuyente
	Adecuación Funcional	Pertinencia Funcional	Contribuyente
Técnica	Compatibilidad	Interoperabilidad	Altamente Contribuyente
	Adecuación Funcional	Corrección Funcional	Contribuyente
	Fiabilidad	Tolerancia a Fallos	Contribuyente
	Mantenibilidad	Modificabilidad	Contribuyente
		Modularidad	Contribuyente
	Portabilidad	Adaptabilidad	Contribuyente

4.4. MÉTODO PROPUESTO

En general los procesos de adquisición de software constan de varias etapas para la compra y contratación del producto, ver figura 7. De acuerdo a los procesos de adquisición consultados, un proceso de compra consta de etapas iniciales para la determinación de los requisitos del software, la elaboración de la oferta, pliegos de condiciones e identificación de los posibles proveedores. Una etapa de evaluación técnica, económica del producto y finalmente las últimas etapas de selección y aceptación del producto. De acuerdo a MinTIC y al artículo 5, numeral 2 de la Ley 1150 de 2007, se establecen los requisitos puntuables en los pliegos de condiciones, estos deben establecer criterios como (MinTIC Colombia. 2020):

1. La ponderación de los elementos de calidad y precio soportados en puntajes o fórmulas.
2. La ponderación de los elementos de calidad y precio que representen la mejor relación de costo-beneficio.

Con esto, deben determinar las condiciones mínimas de la oferta del proceso de compra (MinTIC Colombia. 2020).

Dado que se deben establecer elementos mínimos de calidad de software para la determinación de estos, el método propuesto debe ser parte del proceso de adquisición de software en la evaluación técnica del producto. El no cumplimiento del método de evaluación de sostenibilidad no implica descalificación del proponente como se ha mencionado anteriormente. Los proponentes que estén dispuestos a suministrar información sobre la evaluación de los atributos de calidad y que cumplan los criterios de evaluación si la entidad lo determinar, tendrán una puntuación adicional en la evaluación técnica del producto.

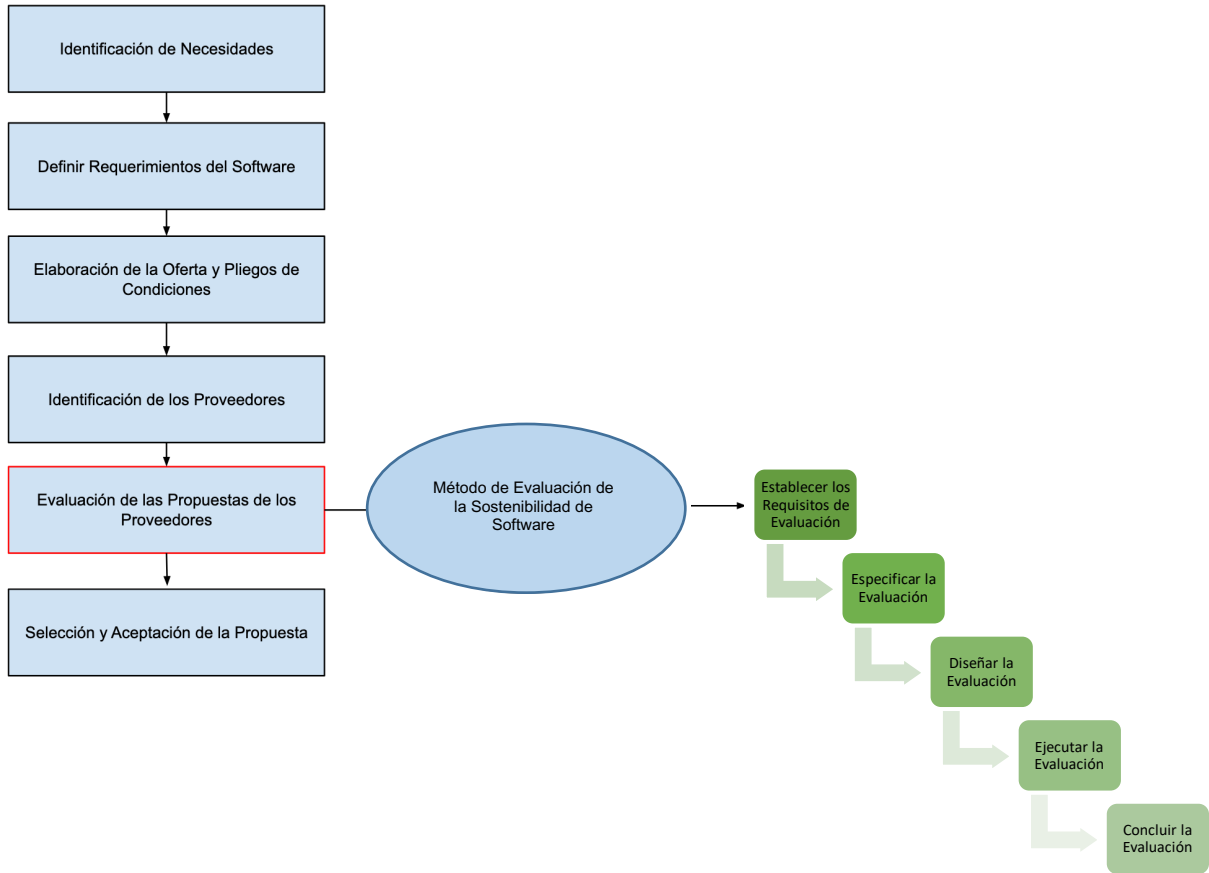


Figura 7. Esquema Genérico de un Proceso de Compra o contratación Pública (MinTIC Colombia. 2020)

Para la construcción del método de evaluación de la sostenibilidad toma como base el estándar ISO/IEC 25040, el cual define el proceso para la evaluación del software. Los parámetros a evaluar son los definidos en la tabla 8. Dicho proceso de evaluación consta de un total de cinco actividades 25040 (ISO/IEC 25040 System and Software Quality Requirements and Evaluation [SQuaRE] — Evaluation process, 2011).

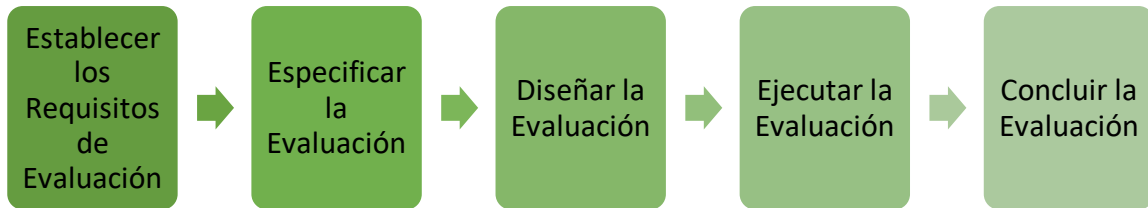


Figura 8. Actividades del proceso de evaluación del software según el estándar ISO/IEC 25040.

- **Actividad 1: Establecer los requisitos de la evaluación.**

En esta actividad se establecen los requisitos de evaluación del producto con base a los objetivos (ISO/IEC 25040 System and Software Quality Requirements and Evaluation [SQuaRE] — Evaluation process, 2011).

- **Actividad 2: Especificar la evaluación.**

En esta actividad se especifican los métodos de evaluación de la calidad del producto que incluye la revisión o evaluación de uno o más componentes del software. Para el proceso de adquisición, la evaluación de la calidad del software se define de los criterios de decisión para las métricas de calidad de producto y calidad de uso (ISO/IEC 25040 System and Software Quality Requirements and Evaluation [SQuaRE] — Evaluation process, 2011).

- **Actividad 3: Diseñar la evaluación.**

En esta actividad se define el plan con las actividades de evaluación que se deben realizar. Se programarán las actividades de evaluación de la calidad de los productos identificadas, teniendo en cuenta la disponibilidad de recursos como el personal y las herramientas de software. (ISO/IEC 25040 System and Software Quality Requirements and Evaluation [SQuaRE] — Evaluation process, 2011).

- **Actividad 4: Ejecutar la evaluación.**

En esta actividad se ejecuta la evaluación de las métricas del producto con base a los criterios de evaluación. (ISO/IEC 25040 System and Software Quality Requirements and Evaluation [SQuaRE] — Evaluation process, 2011).

- **Actividad 5: Concluir la evaluación.**

La conclusión debe incluir el reporte de cumplimiento de las métricas evaluadas con base a los criterios de evaluación (ISO/IEC 25040 System and Software Quality Requirements and Evaluation [SQuaRE] — Evaluation process, 2011).

4.4.1 MÉTODO PARA LA EVALUACIÓN DE LA SOSTENIBILIDAD DEL SOFTWARE PARA EL PROCESO DE COMPRA Y CONTRATACIÓN DEL SECTOR PÚBLICO.

El método propuesto proporciona requisitos, recomendaciones y directrices para evaluar la sostenibilidad del software. La evaluación está diseñada para adquirentes de software en procesos de compra y contratación del sector público para cualquier tipo de software. En cuanto a la determinación de las características de calidad, estas pueden ser determinadas por la entidad pública en caso de que ejecute la medición de las métricas, o los proponentes pueden suministrar la información para la evaluación de las mismas. Es importante aclarar que este no es un proceso de certificación.

4.4.1.1. Objetivo

El método propuesto evalúa la sostenibilidad de software para los procesos de compra y contratación en el sector público.

4.4.1.2. Requisitos de la evaluación

Para la evaluación de la Sostenibilidad de Software nos centraremos en las características y atributos más relevantes para la sostenibilidad de Software del estándar ISO/IEC 25010.

El propósito es evaluar la sostenibilidad del software en proceso de compra y contratación en entidades del sector público con el fin de asegurar la calidad de los productos a adquirir. Como propósito se tiene:

- Evaluar la sostenibilidad del software.
- Cumplir con un mínimo de calidad del producto a adquirir.
- En caso de haber dos o más ofertas, comparar los diferentes productos a través de la evaluación de la sostenibilidad del software.

Los requisitos del producto a adquirir son los determinados en el modelo de calidad de producto ISO/IEC 25010 que mayor relevancia tienen en la sostenibilidad de software especificados en la tabla 8.

4.4.1.3. Especificación de la Evaluación

En esta actividad se especifican los módulos de evaluación (compuestos por las métricas, herramientas y técnicas de medición) y los criterios de decisión que se aplicarán en la evaluación.

Los atributos de calidad más relevantes para la sostenibilidad de software están definidos en la tabla 8. Estas métricas se definen con base en el estándar ISO/IEC 25022 y 25023. El proponente si lo desea, puede suministrar información que soporte el cumplimiento de los atributos o suministrar la información suficiente para que la entidad contratante pueda realizar la evaluación los mismos. Es importante aclarar que el no cumplimiento de los atributos no implica descalificación del proponente, el método de evaluación se centra en la evaluación técnica de producto. La tabla 9 muestra la estructura de definición de los atributos de calidad de la ISO/IEC 25000.

Tabla 9. Descripción de la tabla de métricas de acuerdo al estándar ISO/IEC 25023 (ISO/IEC 25023 System and Software Quality Requirements and Evaluation [SQuaRE] — Measurement of system and software product quality, 2016).

Ítem	Descripción
Sub Característica	Sub Característica de calidad.
Métrica	Nombre asignado a la métrica de calidad
Propósito de la métrica de Calidad	Motivo por el cual se selecciona la métrica.
Método de aplicación	Manera de cómo se va a aplicar la métrica.
Fórmula y cálculo de Datos	Establece la fórmula de medición y especifica los significados de los datos que se van a utiliza
Valor deseado	Proporciona el rango y los valores preferibles y recomendados.

A continuación, a manera de ejemplo se ilustra cómo se define y se miden las métricas de calidad para evaluación de los atributos en la ISO/IEC 25000, Las demás métricas se pueden ver más en detalle en el Anexo 3 cabe destacar que la siguiente información se debe tomar como una referencia y la forma de medición puede variar según el caso.

Tabla 10. Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de la Seguridad (ISO/IEC 25023 System and Software Quality Requirements and Evaluation [SQuaRE] — Measurement of system and software product quality, 2016).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Confidencialidad	Capacidad de control de accesos	¿Qué tan controlable son los accesos al sistema?	Contar el número de diferentes tipos de operaciones ilegales detectado.	$X = A / B$ A=Número de diferentes tipos de operaciones ilegales detectados B = Número de tipos de operaciones ilegales en la especificación Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor

Los criterios de decisión se determinan de acuerdo a valor deseado de las métricas especificadas en las tablas anteriores. Para esto se define el grado de importancia y ponderación de las métricas de acuerdo a la relevancia de la sostenibilidad y el estándar ISO/IEC 25040. En la tabla 11 se define el grado de importancia para los atributos de calidad.

Tabla 11 Grado de importancia de Atributos de Calidad de acuerdo al Estándar ISO/IEC 25040. (ISO/IEC 25040 System and Software Quality Requirements and Evaluation [SQuaRE] — Evaluation process, 2011).

Significado	Grado de Importancia
Criterio Relevante	Alto
Criterio con Relevancia Media	Medio
Criterio No Relevante	Bajo
No Aplica	N/A

El grado de importancia para los atributos de calidad se realiza de acuerdo a las dimensiones de sostenibilidad de software con base en los atributos más relevantes. Los

atributos se seleccionan de tal manera que sean aplicables a procesos de compra de acuerdo al estándar ISO/IEC 25040.

Los pesos específicos y ponderación se otorgan a las características de calidad más relevantes para la sostenibilidad definidas en la tabla 8. A los atributos de calidad más relevantes para la sostenibilidad se le dará el doble del peso específico. La sumatoria de los pesos específicos de la evaluación de los atributos determinarán la proporción de cumplimiento de las características de calidad y dimensión de sostenibilidad. La finalidad es calcular el cumplimiento de los atributos de calidad con base a la proporción total, por lo tanto, para facilitar los cálculos se tomará como base de peso específico:

- Peso específico de atributo de calidad: 50
- Peso específico de atributo de calidad con grado de importancia alta: 100

Los pesos específicos de cada atributo de calidad podrán variar de acuerdo a criterio del evaluador y del tipo de producto. La tabla 12 muestra la ponderación de las características y atributos de calidad. La ponderación por característica muestra el cumplimiento de los atributos por característica de calidad y la ponderación por dimensión muestra el cumplimiento de los atributos por dimensión de sostenibilidad. Con el fin de dar flexibilidad al método, en este se tendrá en cuenta el peso específico determinado por el usuario, el cual se definirá con más detalle en la herramienta de evaluación.

Tabla 12. Características y Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software.

Dimensión	Característica	Atributo	Relevancia	Peso específico Experticia	Peso específico Usuario	Ponderación /Característica	Ponderación/Dimensión
Social	Seguridad	Confidencialidad	Alto	100		100%	100%
		Autenticidad	Medio	50			
		Integridad	Medio	50			
	Usabilidad	Accesibilidad	Medio	50		100%	
Ambiental	Mantenibilidad	Reusabilidad	Alto	100		100%	100%
	Rendimiento Eficiencia	Utilización de Recursos	Medio	50		100%	
Económica	Efectividad	Efectividad	Alto	100		100%	100%
	Fiabilidad	Disponibilidad	Medio	50		100%	
		Recuperabilidad	Medio	50			
	Adecuación Funcional	Pertinencia Funcional	Medio	50		100%	
Técnica	Compatibilidad	Interoperabilidad	Alto	100		100%	100%
	Adecuación Funcional	Corrección Funcional	Medio	50		100%	
	Fiabilidad	Tolerancia a Fallos	Medio	50		100%	
	Mantenibilidad	Modificabilidad	Medio	50		100%	
		Modularidad	Medio	50			
	Portabilidad	Adaptabilidad	Medio	50		100%	

El nivel de puntuación de la ponderación por característica y ponderación por dimensión se determinarán de acuerdo a la tabla 13.

Tabla 13. Niveles de puntuación de evaluación de sostenibilidad con base a las recomendaciones de estándar ISO/IEC 25040 (ISO/IEC 25040 System and Software Quality Requirements and Evaluation [SQuaRE] — Evaluation process, 2011).

Escala	Niveles de Puntuación	Grado de Satisfacción
75.0%-100.0%	Cumple con los requisitos	Muy Satisfactorio
50.0% - 74.9%	Aceptable	Ni Satisfactorio Ni Insatisfactorio
25.1% - 49.9%	Mínimamente Aceptable	Insatisfactorio
0.0 - 25.0%	Inaceptable	

Los niveles de puntuación de las métricas se determinan en la tabla 14 según las recomendaciones del estándar ISO/IEC 2504.

Tabla 14. Niveles de puntuación de los atributos de calidad. (ISO/IEC 25040 System and Software Quality Requirements and Evaluation [SQuaRE] — Evaluation process, 2011).

Cumplimiento de Atributos de Calidad	Resultado
75.0%-100.0%	Cumple
50.0% - 74.9%	Cumple Parcialmente
0.0% - 49.9%	No Cumple

Los criterios de decisión de la evaluación de la sostenibilidad se harán de acuerdo a los criterios del evaluador, los resultados obtenidos en las pruebas y niveles de puntuación. Para realizar el análisis de la sostenibilidad del software se utilizará la herramienta de evaluación que se definirá en el numeral 4.6.

4.4.1.4. Planificación de la Evaluación.

De acuerdo a la disponibilidad y tiempos del proceso de contratación se debe planificar el uso de los recursos como, talento humano, materiales, que puedan ser necesarios. Se debe tener en cuenta la herramienta de evaluación de acuerdo a los criterios del evaluador de los numerales anteriores.

4.4.1.5. Ejecutar la evaluación

En esta etapa se ejecutan las actividades de evaluación de la sostenibilidad del producto obteniendo los atributos de calidad aplicando los criterios de evaluación de acuerdo a la herramienta de evaluación. Es importante destacar, que el método propuesto hace parte del proceso de evaluación técnica del producto. Se debe hacer uso de la herramienta de evaluación, en la cual, se debe determinar los pesos específicos definidos por el usuario previo a la evaluación del producto. La entidad pública debe requerir a los proponentes los atributos de calidad de la tabla 8. Los proponentes deben realizar las mediciones de las métricas y deben suministrar los resultados, los cuales se deben evaluar con base a los porcentajes de cumplimiento de la tabla 14. Finalmente se aplica la herramienta de evaluación ingresando los resultados de los atributos de acuerdo al nivel de cumplimiento, la herramienta determinará los niveles de sostenibilidad.

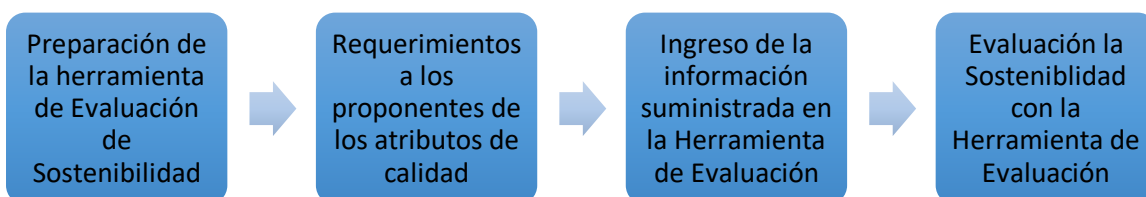


Figura 9. Actividades del proceso de evaluación del software según el estándar ISO/IEC 25040.

4.4.1.6. Concluir la Evaluación.

La herramienta de evaluación determinará los niveles de cumplimiento de sostenibilidad de software de acuerdo a la tabla 13. La entidad pública deberá determinar el puntaje adicional o bonus obtenido por el proponente por el cumplimiento de sostenibilidad de software.

4.5. HERRAMIENTA DE EVALUACIÓN

La herramienta de evaluación se construyó con base a varios criterios como lo son, los atributos de calidad más relevantes para la sostenibilidad de software, la ponderación de los mismos y la ponderación por usuario con el fin de dar flexibilidad al método de evaluación de acuerdo al criterio del evaluador. La herramienta se encuentra disponible en los anexos “*Herramienta de Evaluación.xlsx*”.

La figura 9 muestra el esquema de la herramienta de evaluación de la sostenibilidad de software.

HERRAMIENTA DE EVALUACIÓN DE LA SOSTENIBILIDAD DE SOFTWARE																
DATOS GENERALES																
Proponente:																
Producto Software:																
Observaciones:																
Herramienta de Evaluación del Modelo de Calidad																
Dimensión Social																
Numeral	Característica	Peso Específico Experta	Peso Específico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Justificación	Subtotal por ítem	Cumplimiento por ítem	Cumplimiento por Característica	Resultados					
1	Seguridad	100	20	120	Confidencialidad	Cumple		120	100.00%	100.00%		Dimensión Social	100.00%			
		50	10	60	Autenticidad	Cumple		60	100.00%			Dimensión Ambiental	100.00%			
		50	0	50	Higiene	Cumple		50	100.00%			Dimensión Económica	0.00%			
	Usabilidad	50	20	70	Accesibilidad	Cumple		70	100.00%			Dimensión Técnica	100.00%			
								Cumplimiento	100.00%		Sostenibilidad	76.19%				
Dimensión Ambiental																
Numeral	Característica	Peso Específico Experta	Peso Específico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Justificación	Subtotal por ítem	Cumplimiento por ítem	Cumplimiento por Característica	Escala	Niveles de Puntuación	Grado de Satisfacción			
2	Mantenibilidad	100	0	100	Reusabilidad	Cumple		100	100.0%	100.0%				75.0%-100.0%	Cumple con los requisitos	Muy Satisfactorio Satisfactorio No Satisfactorio Insatisfactorio
	Eficiencia Rendimiento	50	0	50	Utilización de Recursos	Cumple		50	100.0%	100.0%						
								Cumplimiento	100.00%		50.0%-74.9%	Aceptable				
											25.1%-49.9%	Minimamente Aceptable	0.0%-25.0%	Inaceptable		
Dimensión Económica																
Numeral	Característica	Peso Específico Experta	Peso Específico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Justificación	Subtotal por ítem	Cumplimiento por ítem	Cumplimiento por Característica						
3	Efectividad	100	0	100	Efectividad	No Cumple		0	0.0%	0.0%						
	Fiabilidad	50	0	50	Disponibilidad	No Cumple		0	0.0%	0.0%						
		50	0	50	Recuperabilidad	No Cumple		0	0.0%	0.0%						
	Adecuación Funcional	50	0	50	Perinencia Funcional	No Cumple		0	0.0%	0.0%						
								Cumplimiento	0.00%							
Dimensión Técnica																
Numeral	Característica	Peso Específico Experta	Peso Específico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Justificación	Subtotal por ítem	Cumplimiento por ítem	Cumplimiento por Característica						
4	Compatibilidad	100	0	100	Interoperabilidad	Cumple		100.0	100.0%	100.0%						
	Adecuación Funcional	50	0	50	Corrección Funcional	Cumple		50.0	100.0%	100.0%						
	Fiabilidad	50	0	50	Tolerancia a Fallos	Cumple		50.0	100.0%	100.0%						
		50	0	50	Modificabilidad	Cumple		50.0	100.0%	100.0%						
	Mantenibilidad	50	0	50	Modularidad	Cumple		50.0	100.0%	100.0%						
	Portabilidad	50	0	50	Adaptabilidad	Cumple		50.0	100.0%	100.0%						
								Cumplimiento	100.00%							

Figura 10. Esquema de herramienta de Evaluación de la Sostenibilidad de Software.

La herramienta de evaluación determinará el cumplimiento del software a adquirir con los atributos de calidad más relevantes para la sostenibilidad de software. Para esto se

evalúan las 4 dimensiones, Social, Ambiental, Económica y Técnica de acuerdo al nivel de puntuación de la tabla 13. La herramienta de evaluación está compuesta por.

La herramienta consta de varias partes de la siguiente forma:

- **Datos Generales:**

DATOS GENERALES	
Proponente:	
Producto Software:	
Observaciones:	

Figura 11. Esquema de los datos generales de la herramienta de evaluación.

En esta sección se debe ingresar la información general del proponente, así como el nombre del producto y las respectivas observaciones.

- **Aplicación:**

El encabezado está determinado por los siguientes campos:

Numeral	Característica	Peso Especifico Experticia	Peso Especifico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Justificación	Subtotal por Item	Cumplimiento por Item	Cumplimiento por Característica
---------	----------------	----------------------------	---	------------	---------------------	-----------	---------------	-------------------	-----------------------	---------------------------------

Figura 12. Esquema del encabezado de la herramienta de evaluación.

- **Dimensión de sostenibilidad - Numerales:** Esta se determinará de acuerdo al nivel de cumplimiento de las características de calidad.
- **Características de calidad:** Esta se determinará de acuerdo al nivel de cumplimiento de los atributos de calidad.
- **Peso específico Experticia:** Valor numérico el cual determina la importancia del atributo de calidad. 100 para los atributos de calidad con más relevancia para cada dimensión y 50 para el resto de atributos.
- **Peso específico Usuario:** Valor número el cual determina la importancia de cada atributo según la experticia del evaluador. El valor máximo está determinado por el 20% de la suma de los valores específicos. Por ejemplo, en la figura 10 si la suma total de los pesos específicos de los atributos en una dimensión de sostenibilidad es 250, el valor máximo a asignar por experticia será de 50 entre todos los atributos de calidad. Por lo tanto, el evaluador podrá asignar como considere los 50 puntos máximo por experticia.

Numeral	Característica	Peso Específico Experticia	Peso Específico Usuario (Distribuido según necesidad)	Peso Total
1	Seguridad	100	20	120
		50	10	60
		50	0	50
	Usabilidad	50	20	70

Figura 13. Ejemplo ponderación Experticia.

- **Peso total:** Es la suma de Pesos Específicos más Peso Específico de Experticia
- **Resultado:** Es el resultado de la evaluación de los atributos de calidad, este campo se determina de acuerdo al nivel de cumplimiento de la tabla 14.
- **Justificación:** Es una breve descripción de los resultados de cumplimiento de los atributos de calidad.
- **Subtotal por Ítem:** Es el valor del peso específico del atributo de calidad según el resultado de cumplimiento.
 - Si el resultado es Cumple se tendrá un 100% del valor del peso específico.
 - Si el resultado es Cumple Parcialmente se tendrá un 50% del valor del peso específico.
 - Si el resultado es No Cumple se tendrá un 0% del valor del peso específico.

La figura 14 muestra un ejemplo de subtotal por ítem para los diferentes valores de resultado.

Dimensión Social										
Numeral	Característica	Peso Específico Experticia	Peso Específico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Justificación	Subtotal por Ítem	Cumplimiento por Ítem	Cumplimiento por Característica
1	Seguridad	100	20	120	Confidencialidad	Cumple Parcialmente		60	50.00%	52.17%
		50	10	60	Autenticidad	Cumple		60	100.00%	
		50	0	50	Integridad	No Cumple		0	0.00%	
	Usabilidad	50	20	70	Accesibilidad	Cumple		70	100.00%	100.0%
								Cumplimiento	63.33%	

Figura 14 Ejemplo de Subtotal por Ítem.

- Cumplimiento por Ítem: Es el porcentaje de cumplimiento del atributo de calidad con base al resultado.
- Cumplimiento por Característica: Es el porcentaje de cumplimiento ponderado de las características de calidad con base al resultado de los atributos de calidad.
- Cumplimiento: Es el porcentaje de cumplimiento ponderado de la dimensión de sostenibilidad con base al resultado de los atributos de calidad.

Para el uso de la herramienta el usuario deberá como primera medida, especificar el campo de peso específico del usuario, esto con el fin de ajustar los atributos de calidad según sus requerimientos. El peso específico de experticia más el peso específico usuario determinará el peso total del atributo de calidad. $Peso\ Total = Peso\ Específico\ Experticia + Peso\ Específico\ Usuario$.

Posteriormente deberá ingresar los valores del campo resultado, los valores dependen del nivel de cumplimiento de los atributos de calidad suministrados por el proponente según la tabla 14. $Resultado = Cumple \mid Cumple\ Parcialmente \mid No\ Cumple$

Dependiendo del cumplimiento del campo resultado se le asignará el 100%, 50% o 0% del campo peso total al campo subtotal por ítem. $Subtotal\ por\ Ítem = (\% \text{ de cumplimiento}) \times Peso\ Total$.

El campo cumplimiento por ítem es el porcentaje de cumplimiento de subtotal por ítem entre el peso total. $Cumplimiento\ por\ Ítem = Subtotal\ por\ Ítem / Peso\ Total$

El cumplimiento por característica es la suma de todos los subtotales de los ítems sobre el peso total de todos los atributos de la característica. $Cumplimiento\ por\ Característica = Suma\ de\ Subtotales\ por\ ítem\ de\ la\ característica / Suma\ de\ Pesos\ Totales\ de\ la\ característica$

El campo cumplimiento es la suma de todos los subtotales por ítem entre el peso total de todos los atributos en la dimensión de sostenibilidad. $Cumplimiento = Suma\ de\ Subtotales$

por ítem de la dimensión de sostenibilidad / Suma de Pesos Totales de la dimensión de sostenibilidad.

El resultado final será el porcentaje de cumplimiento ponderado de las dimensiones de sostenibilidad de software, ver figura 15. El nivel de cumplimiento está determinado por según la tabla 13 de niveles de puntuación y de acuerdo al resultado se mostrará el siguiente código de colores. Verde por encima del 75.00%, Amarillo entre el 50.00%-74.99% y rojo por debajo del 50%. *Sostenibilidad = Suma de Subtotales por ítem de todas las dimensiones de sostenibilidad / Suma de Pesos Totales de todas las dimensiones de sostenibilidad.*

Resultado	Justificación	Subtotal por Ítem	Cumplimiento por Ítem	Cumplimiento por Característica	Resultados	
Cumple Parcialmente		60	50.00%	52.17%	Dimensión Social	63.33%
Cumple		60	100.00%		Dimensión Ambiental	100.00%
No Cumple		0	0.00%		Dimensión Económica	0.00%
Cumple		70	100.00%	100.0%	Dimensión Técnica	100.00%
		Cumplimiento	63.33%		Sostenibilidad	65.71%

Figura 15. Resultados de la sostenibilidad de software.

4.6. DEMOSTRACIÓN DE APLICACIÓN Y VALIDACIÓN EXTERNA

Con el fin de mostrar la funcionalidad y validar el método de evaluación se realizaron dos etapas con este fin.

4.6.1 VALIDACIÓN EXTERNA

Con el fin de validar el método de evaluación de la sostenibilidad de software se realizó la validación externa con un panel de expertos en calidad de software en personal del sector público. El mecanismo utilizado para la validación fue por medio de encuesta. Las preguntas propuestas para la validación fueron las siguientes.

1. ¿Está de acuerdo que en las Empresas del Sector Público se implementen métodos de evaluación de la Sostenibilidad de Software en procesos de compra y contratación?
2. ¿El Método de evaluación propuesto contiene todos los elementos relevantes para su propósito?
3. ¿El método propuesto contiene sólo los elementos relevantes para su propósito?
4. ¿Fue posible entender completamente el método de evaluación?
5. ¿Es posible determinar la calidad de un producto dentro del dominio objetivo usando el método propuesto?
6. ¿El método de evaluación es ajustable según los criterios y necesidades de cada Empresa del Sector Público?
7. ¿El método de evaluación propuesto arrojará el mismo resultado si se aplica varias veces en el mismo producto?
8. ¿El método de evaluación propuesto es funcional para su finalidad según el objetivo planteado?
9. ¿Considera que la ponderación del método de evaluación es la adecuada?
10. ¿Es importante la evaluación de la sostenibilidad a través de los atributos de calidad en un proceso de compra y contratación en el sector público?
11. ¿Qué considera que hace falta para un mejor desempeño del método? Pregunta abierta

12. ¿Qué considera que se puede mejorar para un mejor desempeño del método? Pregunta abierta
13. Sugerencias o comentarios.

Se envía la documentación referente al trabajo de grado, la encuesta fue formulada y enviada a los expertos en calidad de software de diferentes universidades de la región y entidades del sector público, a los cuales se les indicó la siguiente información:

<h2>Método para la Evaluación de la Sostenibilidad del Software para el Proceso de Compra Y Contratación del Sector Público.</h2> <p>*Obligatorio</p>
<p>Objetivo Proponer un método para la evaluación de la sostenibilidad del software para el proceso de compra y contratación del sector público</p>
<p>Encuesta Agradecemos su colaboración al responder esta encuesta y le solicitamos que lo haga de la manera más objetiva. Sus respuestas son fundamentales para el proceso de investigación que adelantamos. Conteste las siguientes preguntas de acuerdo a su nivel de experticia en el tema, eligiendo la respuesta de la escala que más se adecue a su juicio.</p>

Figura 16. Encuesta a panel de expertos.

Los resultados de las encuestas se muestran a continuación en las siguientes figuras.

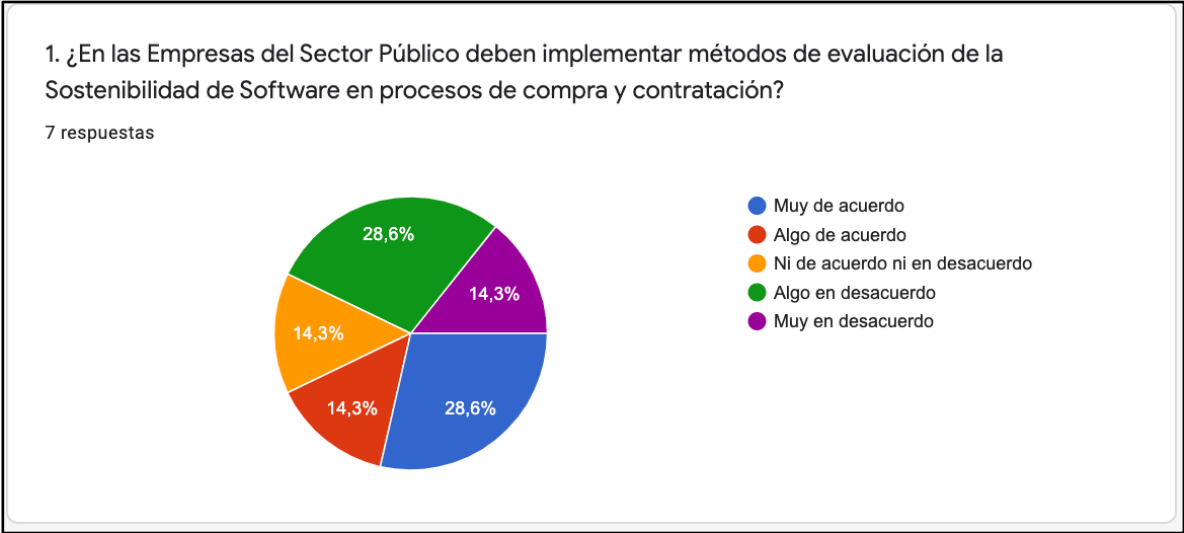


Figura 17. Respuestas a la primera pregunta.

Referente a la primera pregunta se obtuvieron respuestas variadas, no hay consenso si se deben implementar métodos de evaluación de la sostenibilidad de software en el sector público en procesos de compra y contratación.

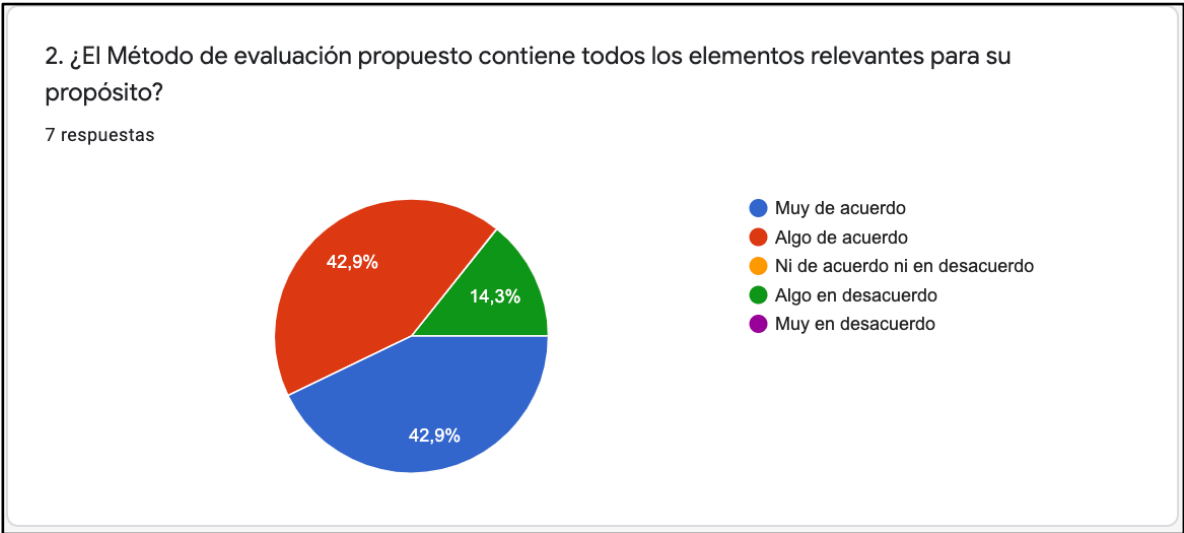


Figura 18. Respuestas a la segunda pregunta de la encuesta.



Figura 19. Respuestas a la tercera pregunta de la encuesta.

En las preguntas 2 y 3 en general los expertos indican que en el método se tuvo en cuenta lo mínimo y todo lo necesario para la evaluación de la sostenibilidad, lo cual muestra que en el método se tuvo en cuenta aspectos básicos y además se consideraron todos los elementos que son pertinentes para su objetivo.

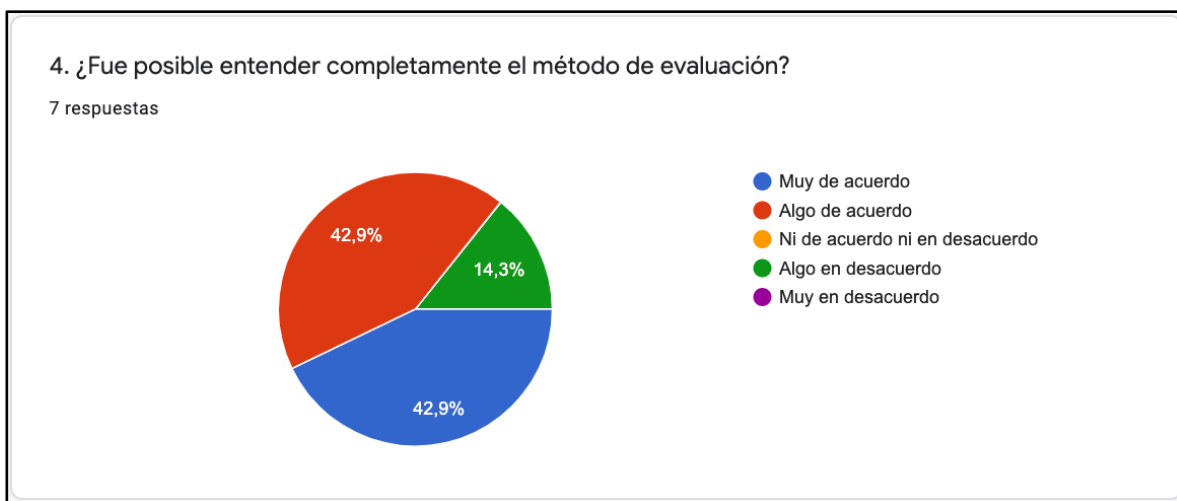


Figura 20. Respuestas a la cuarta pregunta.

La mayoría de los expertos indicaron estar “Muy de acuerdo” y “Algo de acuerdo” en entender el método de evaluación. Lo cual muestra que el método planteado es entendible y aplicable por un usuario del sector público.



Figura 21. Respuestas a la quinta pregunta.

En la quinta pregunta los expertos indicaron que es posible determinar la calidad del producto dentro del dominio objetivo usando el método propuesto. Dado que el método de evaluación de la sostenibilidad está basado en atributos de calidad, en la evaluación de estos se determina la calidad del producto. Con esto, si el resultado de la evaluación es satisfactorio se estará estableciendo un mínimo de calidad del software.



Figura 22. Respuestas a la sexta pregunta

Los expertos indicaron que el método de evaluación es adaptable según las necesidades de cada Empresa del Sector Público. Lo cual muestra flexibilidad en el método para hacer ajustes en los parámetros de evaluación según los requerimientos del software a adquirir.

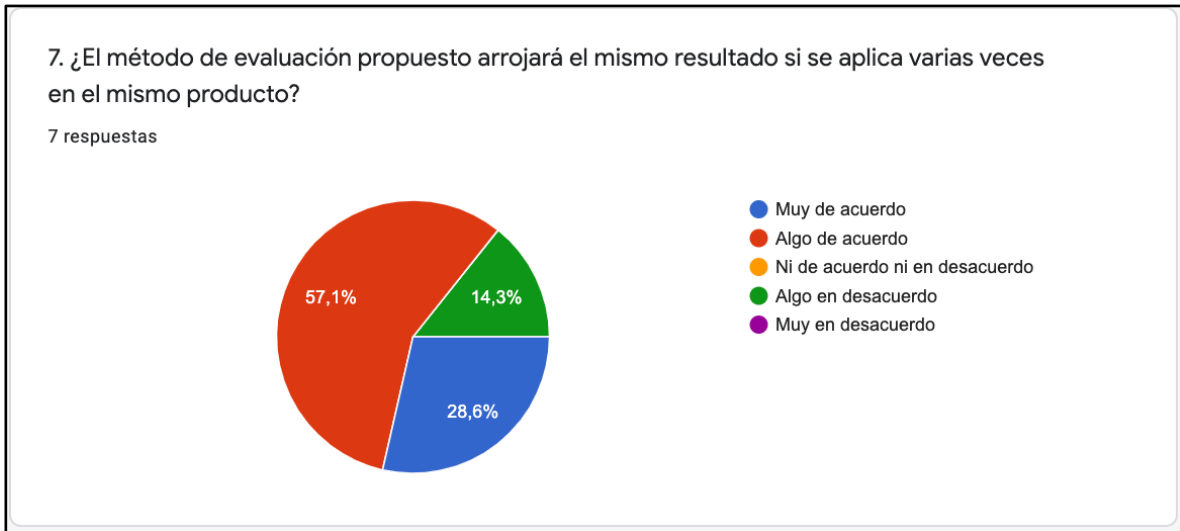


Figura 23. Respuestas a la séptima pregunta.

Según las respuestas de los expertos, estos determinan que el método propuesto arrojará el mismo resultado si se ejecuta varias veces, lo cual señala que está construido de tal forma que sea eficiente en su ejecución y no necesite varias repeticiones para validar el resultado, optimizando así el tiempo del usuario.

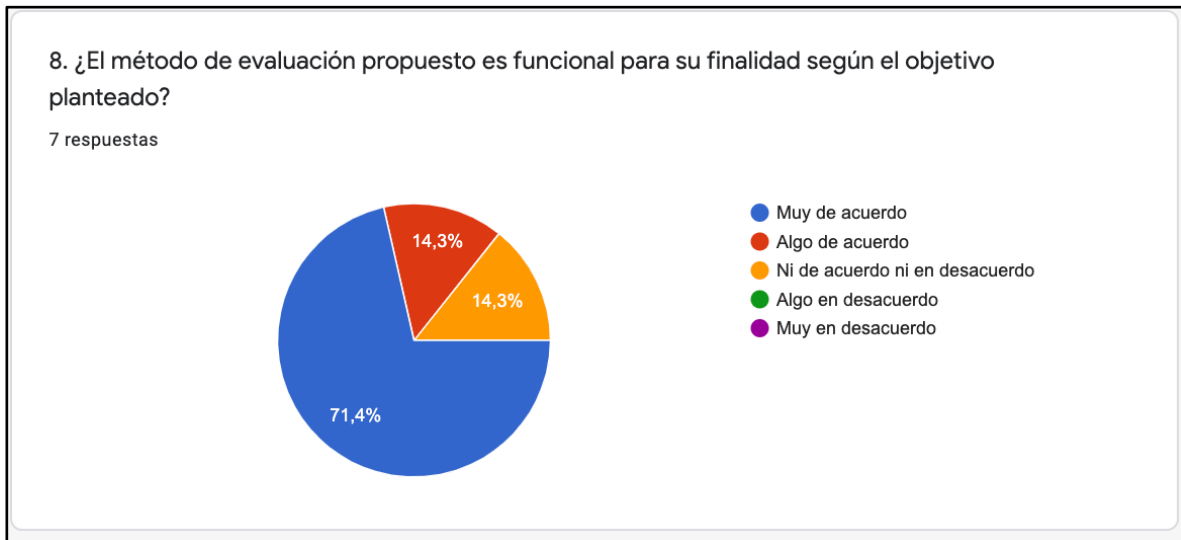


Figura 24. Respuestas a la octava.

Las respuestas de los expertos señalan que el método propuesto fue construido adecuadamente. Contiene los elementos y características necesarias para la determinación de la sostenibilidad de software en compras del sector público. Este será aplicable y funcional para las entidades que lo implemente.

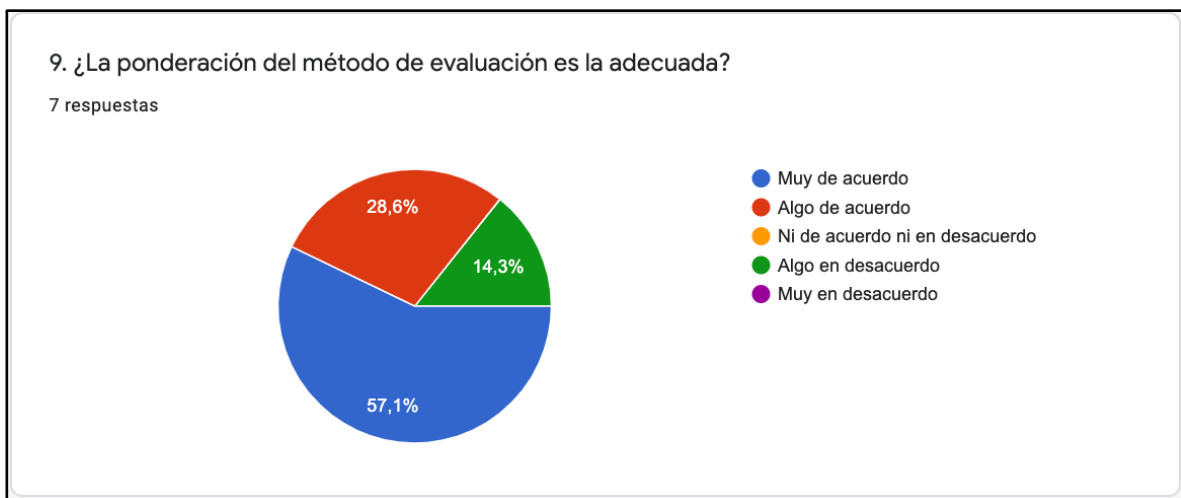


Figura 25. Respuestas a la novena pregunta.

En la novena pregunta los expertos validaron la ponderación en el método propuesto, lo cual indica que los pesos específicos por experticia, por usuario, así como la ponderación

del cumplimiento de las características, numerales y dimensiones de sostenibilidad es la correcta, mostrando que ésta forma calificación es la adecuada en la evaluación de los atributos de calidad para determinar la sostenibilidad de software.

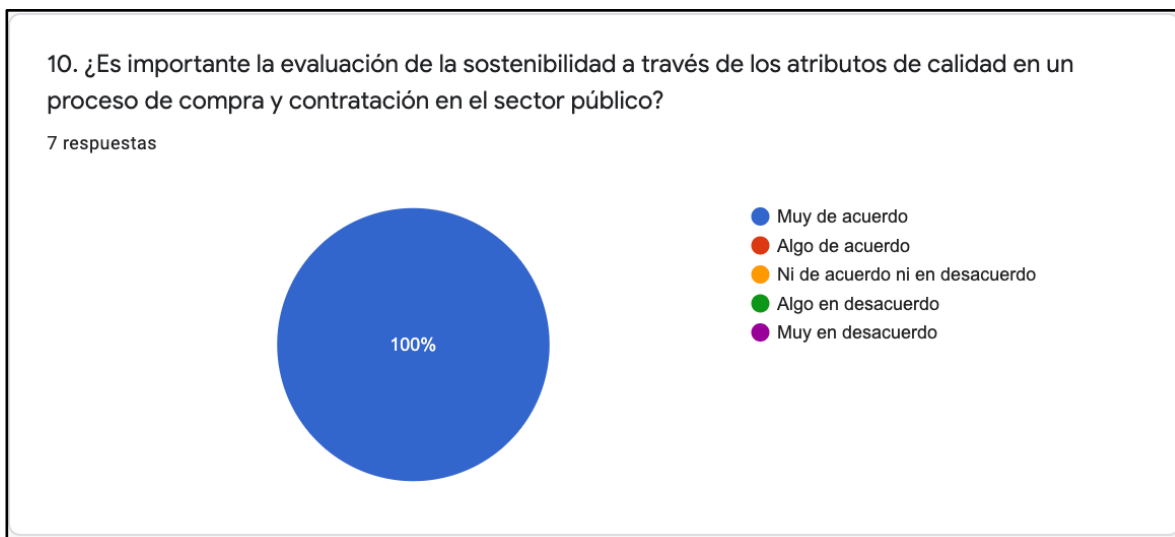


Figura 26. Respuestas a la décima pregunta.

Los expertos indicaron que es importante que se evalúe la sostenibilidad de software a través de los atributos de calidad de software, el uso de estos puede determinar un mínimo de calidad de software en los procesos de compra y contratación, cumpliendo con las recomendaciones dadas por Colombia Compra Eficiente en procesos de adquisición.

11. ¿Qué considera que hace falta para un mejor desempeño del modelo o método? - Pregunta abierta

Figura 27. Pregunta undécima.

En las preguntas abiertas se tomaron las sugerencias y recomendaciones para el método de evaluación propuesto.

Algunos encuestados sugirieron que se podría complementar con “criterios de evaluación que se hagan de acuerdo a un framework/autoridad existente” en los estudios en sostenibilidad varios autores concuerdan en que no existe un marco integrado para su evaluación, cada uno tiene conceptos diferentes como lo evidencia la teoría, la construcción

de este trabajo puede sentar la bases para medición de la sostenibilidad de software de forma unificada de acuerdo a los atributos de calidad. En el presente trabajo, en la aplicación del método propuesto se utilizó como framework el Asistente de evaluación de la calidad de producto de software de la ISO/IEC 25000 con enfoque GQM (Goal, Question, Metric (J.Calabrese, R. Muñoz, 2018). Su uso se mostrará más en detalle en la siguiente sección.

Otros encuestados sugirieron incluir elementos como “Propósito del software a contratar, capacidad y garantía del proveedor”, si bien estos elementos se pueden tener en cuenta en la evaluación de los atributos de calidad, algunos de estos aspectos generalmente hacen parte del proceso de contratación, las obligaciones contractuales o la evaluación como tal del proveedor, los requerimientos de atributos de calidad no son requisitos habilitantes para los proponentes, por lo tanto la evaluación de este método sería un plus para los proveedores.

Otra sugerencia de los encuestados es que se tenga en cuenta el consumo energético, el método de evaluación de la sostenibilidad planteado está descrito de forma general, se podría incluir elementos de medición de consumo energético si la Entidad Pública así lo requiere en sus especificaciones, por ejemplo, los requerimientos de la app para móviles. El método de evaluación de puede modificar de acuerdo a las especificaciones adicionales si así lo necesita.

Otras sugerencias de los encuestados es mejorar la Usabilidad de la Herramienta de Evaluación planteada en el numeral 4.6, sugerencias de tipo intuitivo que ayuden al usuario a un mejor uso de la herramienta por lo tanto esta debe estar descrita con claridad, con ayudas gráficas, códigos de colores que faciliten el uso de funcionarios públicos. Estas recomendaciones fueron tenidas en cuenta y se implementaron en la elaboración final de la Herramienta de Evaluación.

12. ¿Qué considera que se puede mejorar para un mejor desempeño del modelo o método? -
Pregunta abierta

Figura 28. Pregunta duodécima.

La pregunta 12 se refiere a mejoras en el desempeño del método ¿Qué considera que se puede mejorar para un mejor desempeño del modelo o método? La mayoría de los encuestados concuerdan en que se debía mejorar la Usabilidad de la herramienta de evaluación, con ayudas visuales, mejor visualización de las métricas, códigos de colores y descripciones de los atributos de calidad, sugerencias similares a las descritas en la pregunta 11. Cabe destacar que estas sugerencias se implementaron en la versión final de la herramienta de evaluación.

En relación a la pregunta 13 Sugerencias o Comentarios, se hacen comentarios tales como “El método y la herramienta desarrollada resulta muy útil y fácil de aplicar. “y algunas sugerencias de forma de los documentos suministrados.

En general los expertos consultados en calidad de software validaron el método de evaluación en diferentes puntos e hicieron sugerencias dependiendo del tipo de software a evaluar, en general el método está acorde a su propósito y puede ser utilizado y adaptado de acuerdo al criterio y necesidad de las empresas del sector público. El método de evaluación determina la calidad del producto y es funcional para determinar la sostenibilidad del software. Los expertos también validaron que es correcto el uso de los atributos de calidad de la ISO/IEC 25000 para la evaluación del software y además la ponderación propuesta en la herramienta de evaluación es la adecuada. Se hicieron algunas recomendaciones las cuales fueron tenidas en cuenta en la versión final de este documento.

4.6.2 APLICACIÓN DEL MÉTODO DE EVALUACIÓN

Para la aplicación se evaluaron dos aplicaciones web con potencial uso en las empresas del sector público. En un proceso de compra y contratación de software las Entidades deben requerir a los proponentes los atributos de calidad más relevantes para la sostenibilidad de software, éstos deben realizar las mediciones de los atributos y suministrar la información a la entidad pública sobre los niveles de cumplimiento de las métricas.

Dado que los desarrolladores de los productos a evaluar no suministraron mediciones de los atributos de calidad, estos facilitarán información que permita la medición de las mismas en el presente trabajo.

Para realizar las mediciones con la información suministrada se utilizará el Asistente de evaluación de la calidad de producto de software de la ISO/IEC 25000 con enfoque GQM (Goal, Question, Metric). Este enfoque está orientado a medir métricas o atributos de calidad para cierto objetivo a través de la utilización de preguntas (J.Calabrese, R. Muñoz, 2018). El asistente proporciona elementos necesarios que ayudan a evaluar los atributos de calidad del software de la ISO/IEC 25000, define un conjunto de preguntas cuyas respuestas permiten obtener una métrica aplicable a las características de calidad de la ISO/IEC 25010 (J.Calabrese, R. Muñoz, 2018).

Para ilustrar la funcionalidad del Asistente de Evaluación con enfoque GQM de los atributos de calidad se define a manera de ejemplo la medición de la Seguridad – Confidencialidad.

Métrica a Evaluar: Confidencialidad.

El objetivo es evaluar la confidencialidad del software definida en ISO/IEC 25000. Para esto se definen las siguientes preguntas orientadas a este objetivo, ver tabla 15.

Tabla 15. Preguntas definidas para evaluación de la Confidencialidad (J.Calabrese, R. Muñoz, 2018).

ID	PREGUNTA
P1	¿Se requiere que la contraseña posea al menos 8 caracteres?
P2	¿Se requiere que la contraseña posea letras mayúsculas y minúsculas?
P3	¿Se requiere que la contraseña posea números y letras?
P4	¿Se requiere que la contraseña posea caracteres especiales?
P5	¿El sistema utiliza conexión segura mediante HTTPS?
P6	¿La base de datos posee los datos encriptados?
P7	¿El sistema permite acceder a funcionalidades en las cuales no se tiene permiso?
P8	¿El sistema permite que cualquier persona tenga acceso a la base de datos?
P9	¿El sistema permite que cualquier persona tenga acceso al código del servidor de la aplicación?

P10	¿Cualquier persona tiene acceso al servidor físico?
P11	¿Cualquier persona tiene acceso al servidor remoto?
P12	¿El sistema posee redireccionamientos hacia sitios no seguros?

Las preguntas de la tabla 15 deben ser respondidas SI o NO. Las respuestas se deben combinar de forma lógica generando criterios de evaluación para transformarlos en valores cualitativos, ver tabla 16. (J.Calabrese, R. Muñoz, 2018).

Tabla 16. Criterios de evaluación para la confidencialidad (J.Calabrese, R. Muñoz, 2018).

ID	NOMBRE	DESCRIPCIÓN	FÓRMULA	PTOS
C-1	Conexiones seguras	Una conexión se considera segura si se utiliza HTTPS y si no se tienen redireccionamientos hacia sitios no seguros	$P5 \ \& \ \neg P12 = V$	1
C-2	Control de acceso	Se debe controlar que no se permita acceder a funcionalidades sin autorización, tampoco a la base de datos, al código de la aplicación ni a los servidores, físico ni remoto	$si \ P7 \ \ P8 \ \ P9 \ \ P10 \ \ P11 = F$	1
C-3	Encriptación de datos	Los datos de la base de datos deben estar encriptados	$P6 = V$	1
C-4	Contraseña debajo nivel	La contraseña se considera de bajo nivel si posee menos de 8 caracteres, no posee letras mayúsculas y minúsculas, no posee letras y números y no posee caracteres especiales	$P1 \ \ P2 \ \ P3 \ \ P4 = F$	0
	Contraseña de medio nivel	La contraseña se considera de medio nivel si posee al menos 8 caracteres o letras mayúsculas y minúsculas o letras y números o	$P1 \ \ P2 \ \ P3 \ \ P4 = V$	0.5
	Contraseña de alto nivel	La contraseña se considera de alto nivel si posee al menos 8 caracteres, letras mayúsculas y minúsculas, letras y números y caracteres especiales	$P1 \ \& \ P2 \ \& \ P3 \ \& \ P4 = V$	1

Los criterios de evaluación se agrupan y combinan para definir la métrica. La propuesta de la medición de la confidencialidad se muestra en la figura 29. El resultado final será el cumplimiento de la métrica de confidencialidad, de acuerdo al nivel de cumplimiento de la tabla 14 se le asignará el valor en el campo Resultado de la Herramienta de Evaluación.

Confidencialidad.
Métrica: *Confidencialidad*
Propósito: *¿Cuán eficiente es el sistema a la hora de proteger el acceso de datos e información no autorizados, ya sea accidental o deliberadamente?*
Método de aplicación: *Contestar las preguntas de los CE correspondientes a la subcaracterística "Confidencialidad" y calcular la puntuación obtenida, sumando los puntajes de los CE referenciados que cumplan con la meta esperada. "Puntaje total" hace referencia al máximo puntaje que se puede obtener.*
Entradas: *A = Puntaje obtenido. B = Puntaje total.*
Fórmula: $X = A/B$

Figura 29. Resultados de la sostenibilidad de software (J.Calabrese, R. Muñoz, 2018).

Con la implementación del Asistente de Evaluación con enfoque GQM se evalúan los atributos de calidad de la ISO/IEC 25000 de manera sencilla de acuerdo a los parámetros descritos anteriormente. De forma similar se define la evaluación de los demás atributos de calidad más relevantes para la sostenibilidad de software definidos en la Herramienta de Evaluación. Las preguntas de evaluación y criterios de evaluación de cada una de las métricas se muestran en el Anexo I y II.

Para mostrar la funcionalidad del método de evaluación de sostenibilidad de software, se realizaron las preguntas del asistente de evaluación a los desarrolladores de las dos aplicaciones web con potencial uso en el sector público. Las respuestas fueron suministradas por los desarrolladores y estas se pueden ver más en detalle en el anexo I y II del presente trabajo.

Evaluación de la Sostenibilidad de Software Caso 1.

El primer caso a analizar es un proyecto dedicado a la generación de documentos online para entidades gubernamentales, Para la evaluación de la sostenibilidad del proyecto se analizó el código fuente y la documentación publicada por el desarrollador. El proceso de evaluación se hizo a través del Asistente de Evaluación con enfoque GQM mencionado anteriormente. Se respondieron las preguntas relacionadas con la medición de los atributos de calidad más relevantes para la sostenibilidad. Las preguntas y las respuestas para la evaluación de los atributos de calidad se muestran en detalle en el anexo I.

Se realizó la evaluación siguiendo el procedimiento del Asistente de Evaluación con enfoque GQM para ISO/IEC 25000, los detalles de la evaluación de cada métrica de calidad se muestran en detalle en el anexo I.

El resumen de los resultados de evaluación de cada uno de los atributos de calidad se muestra en la tabla 16. La columna Resultado indica si el atributo cumplió o no con la evaluación o si cumplió parcialmente de acuerdo al grado de satisfacción definido anteriormente, esta se ingresó con base en las evaluaciones de los atributos de calidad de la tabla 14.

Tabla 16. Resultado de evaluación de atributos de calidad relevantes para la sostenibilidad Caso 1.

Característica	Atributo de Calidad	Resultado	Observación
Seguridad	Confidencialidad	Cumple Parcialmente	Se cumple parcialmente con lo requerido en la herramienta.
	Autenticidad	Cumple	El sistema cuenta con un sistema de autenticación de credenciales
	Integridad	Cumple	El sistema cuenta con protecciones a la integridad de los datos.
Usabilidad	Accesibilidad	No Cumple	El sistema no cumple con los requerimientos de accesibilidad para los usuarios.
Mantenibilidad	Reusabilidad	No Cumple	El sistema no cuenta con elementos reutilizables en su estructura de código.
Rendimiento Eficiencia	Utilización de Recursos	Cumple	Es sistema es eficiente con el uso de los recursos.
Efectividad	Efectividad	Cumple	El sistema puede ejecutar todas las tareas eficientemente.
Fiabilidad	Disponibilidad	Cumple	El sistema estará disponible cuando el usuario lo requiera.
	Recuperabilidad	Cumple Parcialmente	El sistema cumple parcialmente con los elementos de recuperabilidad
Adecuación Funcional	Pertinencia Funcional	Cumple	El sistema cumple con los criterios de pertinencia funcional
Compatibilidad	Interoperabilidad	No Cumple	El sistema no cumple con los parámetros de Interoperabilidad
Adecuación Funcional	Corrección Funcional	Cumple	El sistema cumple con los criterios de corrección funcional.

Fiabilidad	Tolerancia a Fallos	Cumple Parcialmente	El sistema tiene solo algunos elementos de tolerancia a fallos
Mantenibilidad	Modificabilidad	Cumple	El sistema tiene elementos que facilitan la modificación de sus componentes
	Modularidad	Cumple	El sistema tiene implementado elementos en su confirmación de forma modular.
Portabilidad	Adaptabilidad	Cumple	El sistema puede adaptarse a la mayoría de los sistemas operativos y entornos.

Los resultados de los atributos de calidad fueron ingresados en la herramienta de evaluación de la sostenibilidad de software y la evaluación de las diferentes dimensiones de sostenibilidad se muestra en las siguientes figuras.

Dimensión Social									
Numeral	Característica	Peso Especifico Experticia	Peso Especifico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Subtotal por ítem	Cumplimiento por ítem	Cumplimiento por Característica
1	Seguridad	100	0	100	Confidencialidad	Cumple Parc	50	50.00%	75.00%
		50	0	50	Autenticidad	Cumple	50	100.00%	
		50	0	50	Integridad	Cumple	50	100.00%	
	Usabilidad	50	0	50	Accesibilidad	No Cumple	0	0.00%	0.0%
								60.00%	

Figura 30. Resultado de Evaluación para la Dimensión Social Caso 1.

Dimensión Ambiental									
Numeral	Característica	Peso Especifico Experticia	Peso Especifico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Subtotal por ítem	Cumplimiento por ítem	Cumplimiento por Característica
2	Mantenibilidad	100	0	100	Reusabilidad	No Cumple	0	0.0%	0.0%
	Eficiencia Rendimiento	50	0	50	Utilización de Recursos	Cumple	50	100.0%	100.0%
								33.33%	

Figura 31. Resultado de Evaluación para la Dimensión Ambiental Caso 1.

Dimensión Económica									
Numeral	Característica	Peso Especifico Experticia	Peso Especifico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Subtotal por ítem	Cumplimiento por ítem	Cumplimiento por Característica
3	Efectividad	100	0	100	Efectividad	Cumple	100	100.0%	100.0%
	Fiabilidad	50	0	50	Disponibilidad	Cumple	50	100.0%	75.0%
		50	0	50	Recuperabilidad	Cumple Parc	25	50.0%	
	Adecuación Funcional	50	0	50	Complejitud Funcional	Cumple	50	100.0%	100.0%
								90.00%	

Figura 32. Resultado de Evaluación para la Económica Caso 1.

Dimensión Técnica									
Numeral	Característica	Peso Especifico Experticia	Peso Especifico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Subtotal por ítem	Cumplimiento por ítem	Cumplimiento por Característica
4	Compatibilidad	100	0	100	Interoperabilidad	No Cumple	0.0	0.0%	0.0%
	Adecuación Funcional	50	0	50	Corrección Funcional	Cumple	50.0	100.0%	100.0%
	Fiabilidad	50	0	50	Tolerancia a Fallos	Cumple Parc	25.0	50.0%	50.0%
	Mantenibilidad	50	0	50	Modificabilidad	Cumple	50.0	100.0%	100.0%
		50	0	50	Modularidad	Cumple	50.0	100.0%	
	Portabilidad	50	0	50	Adaptabilidad	Cumple	50.0	100.0%	100.0%
								64.29%	

Figura 33. Resultado de Evaluación para la Económica Caso 1.

La herramienta de evaluación arrojó como resultado para el Caso 1 tiene un 65% de Sostenibilidad de Software, tal como se muestra en la figura 34.

Resultados	
Dimensión Social	60.00%
Dimensión Ambiental	33.33%
Dimensión Económica	90.00%
Dimensión Técnica	64.29%
Sostenibilidad	65.00%

Figura 34. Resumen de Evaluación de la Sostenibilidad de Software para el Caso 1.

Evaluación de la Sostenibilidad de Software Caso 2.

El segundo caso a evaluar es un proyecto desarrollado por la Universidad de Antioquia en conjunto con una entidad del sector público, por motivos de confidencialidad no menciona el nombre del proyecto. La evaluación se hizo de forma similar al caso 1, se

utilizó para la medición de los atributos de calidad el Asistente de Evaluación con enfoque GQM para ISO/IEC 25000. Las preguntas fueron enviadas al desarrollador y este las respondió de acuerdo a su criterio. Las preguntas y las respuestas del desarrollador se muestran en detalle en el Anexo II.

Las respuestas a las preguntas fueron ingresadas en el Asistente de Evaluación con enfoque GQM para ISO/IEC 25000 de forma similar al descrito en el caso I. El resultado se muestra en la tabla 17. El detalle de evaluación de cada una de las métricas se muestra en el anexo II.

Tabla 17. Resultado de evaluación de atributos de calidad relevantes para la sostenibilidad Caso 2.

Característica	Atributo de Calidad	Resultado	Observación
Seguridad	Confidencialidad	Cumple	El sistema cumple con los elementos de confidencialidad de datos.
	Autenticidad	Cumple	El sistema cuenta con un sistema de autenticación de credenciales
	Integridad	Cumple	El sistema cuenta con protecciones a la integridad de los datos.
Usabilidad	Accesibilidad	No Cumple	El sistema no cumple con los requerimientos de accesibilidad para los usuarios.
Mantenibilidad	Reusabilidad	Cumple	El sistema cuenta con elementos reutilizables en su estructura de código.
Rendimiento Eficiencia	Utilización de Recursos	Cumple	Este sistema es eficiente con el uso de los recursos.
Efectividad	Efectividad	Cumple	El sistema puede ejecutar todas las tareas eficientemente.
Fiabilidad	Disponibilidad	Cumple	El sistema estará disponible cuando el usuario lo requiera.
	Recuperabilidad	Cumple	El sistema cumple con los elementos de recuperabilidad
Adecuación Funcional	Pertinencia Funcional	Cumple	El sistema cumple con los criterios de pertinencia funcional
Compatibilidad	Interoperabilidad	Cumple	El sistema cumple con los parámetros de Interoperabilidad
Adecuación Funcional	Corrección Funcional	Cumple	El sistema cumple con los criterios de corrección funcional.
Fiabilidad	Tolerancia a Fallos	Cumple	El sistema cumple con los parámetros de tolerancia a fallos.
Mantenibilidad	Modificabilidad	Cumple	El sistema tiene elementos que facilitan la modificación de sus componentes

	Modularidad	Cumple	El sistema tiene implementado elementos en su confirmación de forma modular.
Portabilidad	Adaptabilidad	Cumple	El sistema puede adaptarse a la mayoría de los sistemas operativos y entornos.

Los resultados de los atributos de calidad fueron ingresados en la herramienta de evaluación de la sostenibilidad de software, la evaluación del caso 2 para las diferentes dimensiones de sostenibilidad se muestran en las siguientes figuras.

Dimensión Social									
Numeral	Característica	Peso Especifico Experticia	Peso Especifico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Subtotal por ítem	Cumplimiento por ítem	Cumplimiento por Característica
1	Seguridad	100	0	100	Confidencialidad	Cumple	100	100.00%	100.00%
		50	0	50	Autenticidad	Cumple	50	100.00%	
		50	0	50	Integridad	Cumple	50	100.00%	
	Usabilidad	50	0	50	Accesibilidad	No Cumple	0	0.00%	0.0%
								80.00%	

Figura 35. Resultado de Evaluación para la Dimensión Social Caso 2.

Dimensión Ambiental									
Numeral	Característica	Peso Especifico Experticia	Peso Especifico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Subtotal por ítem	Cumplimiento por ítem	Cumplimiento por Característica
2	Mantenibilidad	100	0	100	Reusabilidad	Cumple	100	100.0%	100.0%
	Eficiencia Rendimiento	50	0	50	Utilización de Recursos	Cumple	50	100.0%	100.0%
								100.00%	

Figura 36. Resultado de Evaluación para la Dimensión Ambiental Caso 2.

Dimensión Económica									
Numeral	Característica	Peso Especifico Experticia	Peso Especifico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Subtotal por ítem	Cumplimiento por ítem	Cumplimiento por Característica
3	Efectividad	100	0	100	Efectividad	Cumple	100	100.0%	100.0%
	Fiabilidad	50	0	50	Disponibilidad	Cumple	50	100.0%	
		50	0	50	Recuperabilidad	Cumple	50	100.0%	
	Adeuación Funcional	50	0	50	Complejitud Funcional	Cumple	50	100.0%	100.0%
								100.00%	

Figura 37. Resultado de Evaluación para la Dimensión Económica Social Caso 2.

Dimensión Técnica										
Numeral	Característica	Peso Especifico Experticia	Peso Especifico Usuario (Distribuido según necesidad)	Peso Total	Atributo de Calidad	Resultado	Subtotal por Item	Cumplimiento por Item	Cumplimiento por Característica	
4	Compatibilidad	100	0	100	Interoperabilidad	Cumple	100.0	100.0%	100.0%	
	Adecuación Funcional	50	0	50	Corrección Funcional	Cumple	50.0	100.0%	100.0%	
	Fiabilidad	50	0	50	Tolerancia a Fallos	Cumple	50.0	100.0%	100.0%	
	Mantenibilidad		50	0	50	Modificabilidad	Cumple	50.0	100.0%	100.0%
			50	0	50	Modularidad	Cumple	50.0	100.0%	
	Portabilidad	50	0	50	Adaptabilidad	Cumple	50.0	100.0%	100.0%	
								100.00%		

Figura 38. Resultado de Evaluación para la Técnica Social Caso 2.

Resultados	
Dimensión Social	80.00%
Dimensión Ambiental	100.00%
Dimensión Económica	100.00%
Dimensión Técnica	100.00%
Sostenibilidad	95.00%

Figura 39 Resumen de Evaluación de la Sostenibilidad de Software para el Caso 2.

La herramienta de evaluación mostró como resultado que para el Caso 2 tiene un 95% de Sostenibilidad de Software, tal como se muestra en la figura 39.

Al ejecutar el método de evaluación de la sostenibilidad para procesos de compra y contratación del sector público, nos muestra para el caso 1 un resultado de 65% de sostenibilidad de software, lo cual es Ni satisfactorio Ni insatisfactorio de acuerdo a la tabla 25 de niveles de puntuación de sostenibilidad de software. Mientras que para el caso dos mostró como resultado un 95% de sostenibilidad de software lo cual es Muy Satisfactorio de acuerdo a la tabla 12.

En cuanto al Caso I, ver figura 30, en la evaluación de la Dimensión Social, la Seguridad es la característica más importante, la cual cumple con los atributos de Integridad de los datos y la Accesibilidad, en cuanto a la Confidencialidad cumple parcialmente, debido a que en el manejo de las contraseñas no se especifican los requerimientos que esta debe tener para tener un acceso seguro a los datos, si bien se implementan métodos de autenticación

mediante contraseñas un potencial usuario puede crear contraseñas no seguras lo cual puede comprometer la el acceso a los datos. En cuanto a la Accesibilidad no se especifica las funciones o ayudas por las cuales un usuario con determinada característica o condición pueda utilizar el programa. Por lo tanto, no se cumple con el atributo de a Accesibilidad. Al evaluar la Sostenibilidad en la Dimensión Social muestra que cumple con el 60%.

En cuanto al Caso II la Dimensión Social, ver figura 35, mostró mejores resultados que en el Caso I, esta dimensión cumple con todos los atributos de calidad de Seguridad. De forma similar al Caso I, también mostró falencia en la Accesibilidad, dando a que no se implementan ayudas para el uso de usuarios con características especiales. El nivel de sostenibilidad para la dimensión social fue de un 80%.

En la Dimensión Ambiental las características importantes son la Mantenibilidad con el atributo de Reusabilidad y la Eficiencia en el rendimiento con el atributo de Utilización de Recursos. En el Caso I, ver figura 31, no se especifican los elementos reutilizables en el software, por lo tanto, no se cumple con las especificaciones de este atributo de calidad. Mientras que la Utilización de Recursos cumple satisfactoriamente. En el Caso II, ver figura 36, mostró como resultado que cumplen con ambos atributos de calidad. En esta dimensión el cumplimiento del Caso I fue de un 33%, mientras que en el Caso II fue de un 100%. Es importante anotar que la reusabilidad para la dimensión social es el atributo de mayor importancia, por lo tanto, el no cumplir con los requerimientos de este representa un gran impacto en la evaluación de esta dimensión.

En la Dimensión Económica las características más importantes son la Efectividad, Fiabilidad y la Adecuación Funcional. En la evaluación del Caso I, ver figura 32, cumple con los atributos de Efectividad, Disponibilidad y Pertinencia Funcional, en el caso de la Recuperabilidad, en la evaluación no se especifica claramente la recuperación del sistema después de un fallo, por lo tanto, la evaluación de este ítem mostró que cumple parcialmente. En el Caso II, en la evaluación mostró que cumple en su totalidad con los atributos de calidad para esta dimensión ver figura 37, la puntuación en este caso fue del 100%. Mientras que el Caso I obtuvo una puntuación del 90%.

La Dimensión Técnica es la dimensión con mayor número de atributos, por lo tanto, la de mayor peso entre las otras dimensiones. La característica más importante es la Compatibilidad con el atributo de la Interoperabilidad, en el Caso I, ver figura 33, en su evaluación no cumple con este atributo, dado que no se especifica la capacidad del sistema o componente para intercambiar información con otros. En cuanto a la Tolerancia a Fallos solo tiene implementado algunas medidas, por lo tanto, en su evaluación cumple parcialmente. Los atributos de Corrección Funcional, Modificabilidad, Modularidad y Adaptabilidad mostraron como resultado que se cumple en su totalidad. El Caso I mostró como resultado un 64.29%. El Caso II en su evaluación de los atributos de calidad, ver figura 38, mostró que cumple completamente con estos ítems, por tanto, en la Evaluación de la Dimensión Técnica obtuvo una puntuación del 100%. En la evaluación de sostenibilidad el Caso I obtuvo una puntuación total de 65%, ver figura 18. Mientras que el Caso II, obtuvo una puntuación de 95%, ver figura 23.

Dado un proceso de compra y contratación para la puntuación adicional técnica, solo se tendrían en cuenta las aplicaciones con grado mayor de sostenibilidad al 75%. En el caso hipotético que ambas aplicaciones fueran parte del mismo proceso de compra y contratación la aplicación del Caso II sería la que obtendría un puntaje adicional por cumplir con los atributos de calidad enfocados a la sostenibilidad de software. Como se mencionó anteriormente el método planteado para la evaluación de sostenibilidad debe ser usado en la evaluación técnica del producto como un componente adicional.

5. CONCLUSIONES Y RECOMENDACIONES.

La sostenibilidad de software es un tema que en los últimos años ha tomado importancia dentro del ámbito científico. Según la teoría consultada, varios autores concuerdan en que no existen un marco integrado para la medición de la sostenibilidad de software. Algunos de los autores se centraron en caracterizar la sostenibilidad de por medio de los atributos de calidad de software del Modelo de Calidad de Producto ISO/IEC 25000. Se determinó que esta se compone en varias dimensiones y algunos atributos de calidad de software son más relevantes que otros. Esto nos permitió evaluar estos atributos de calidad para medir la sostenibilidad de software en el método propuesto.

En general en los procesos de compra y contratación, en los contratos de menor presupuesto no se realiza evaluación técnica previo a adquisición del producto. Normalmente se evalúa al proponente sobre los requisitos contractuales del proceso de contratación pública.

Dado que el requerimiento de certificaciones de calidad no es un requisito habilitante en los procesos de contratación de software en Colombia, no se puede exigir la certificación de estos para participar en los procesos de adquisición. Sin embargo, Colombia Compra Eficiente sugiere que las Entidades del Sector Público deben garantizar un mínimo de calidad de productos de software.

Si bien no se puede exigir la medición de los atributos de calidad de la norma ISO/IEC 25000, en el proceso de compra y contratación el método de evaluación de la sostenibilidad haría parte de la evaluación técnica del producto. En dado caso que el proponente no cumpla con el mínimo de evaluación de la sostenibilidad del producto o no suministre información para la evaluación del mismo, no sería descalificado. En cambio, si cumple con el mínimo de sostenibilidad tendría un plus o puntuación adicional en la evaluación técnica del producto. El peso de la puntuación adicional debe ser determinado por la entidad pública.

Para la evaluación de los atributos de calidad más relevantes para la sostenibilidad de software, los proponentes podrán suministrar la documentación que soporte el cumplimiento de los mismos. Las Entidades Públicas también podrá realizar la medición de los atributos si

cuentan con el personal y equipo correspondiente requiriendo a los proponentes los soportes que faciliten la evaluación.

La evaluación de los atributos de calidad de la ISO 25000 en ciertos casos resulta compleja si no se dispone de personal y equipo especializado. En general los programas de software de análisis estático miden las características de calidad de forma global y no especifican los atributos de los mismos. Por lo tanto, dependerá de los criterios de la Entidad Pública el evaluar las características de calidad de forma global o específica cómo se plantea en el Método de Evaluación Propuesto.

La validación del método de evaluación propuesto en este documento se realizó en dos etapas, una validación externa por medio de una encuesta a personal experto en calidad de software y personal del sector público, lo que permitió dar valor agregado al método de evaluación y la herramienta de demostración. Con la retroalimentación obtenida en este proceso, el método propuesto paso por una etapa de complemento y ajustes de los parámetros aportando una mayor relevancia al método de evaluación.

La segunda etapa por medio de la aplicación del método propuesto a dos aplicaciones con potencial uso en las Entidades Públicas.

En la aplicación del método, debido a que la evaluación de los atributos de calidad resulta compleja, se utilizó el Asistente de Evaluación desarrollado por Calabrese y Muñoz en el 2018. Se realizó el procedimiento descrito para ambas aplicaciones y se utilizó la Herramienta de Evaluación para determinar el grado de Sostenibilidad de Software. Dependiendo de los requerimientos de cada Entidad Pública se puede modificar las preguntas del asistente de evaluación con el fin de adaptarlo a las necesidades del producto.

En el Caso I obtuvo un 65% de sostenibilidad. El caso I mostró deficiencias en las Dimensiones Social 60%, técnica 64%, ambiental 33%, solo cumplió en la dimensión económica con un 90%, mostrando deficiencias especialmente en la accesibilidad, reusabilidad e interoperabilidad. Cumpliendo parcialmente con la confidencialidad, recuperabilidad y la tolerancia a fallos. Mientras que el Caso II obtuvo un 95% de sostenibilidad, en general el Caso II cumplió con casi todos los estándares establecidos en los atributos de calidad, mostrando deficiencias solo en la accesibilidad. La accesibilidad fue

una deficiencia que se presentó en ambas aplicaciones lo que indica que la usabilidad no es una característica que se tenga muy en cuenta en el desarrollo de aplicaciones.

La validación externa se realizó con personal experto en calidad de software, suministrada la documentación y todo lo referente al proceso de evaluación de la sostenibilidad. Los expertos concuerdan en que el método es funcional, la ponderación planteada es la correcta y además contiene elementos necesarios para evaluar la sostenibilidad de software, este puede ser adaptado a las necesidades y criterios de las entidades públicas. Los expertos también indican que es correcto el uso de los atributos de calidad de software de la ISO/IEC 25000. Además, ven importante la evaluación de estos atributos para determinar la sostenibilidad de software en un proceso de compra y contratación. Se hicieron sugerencias sobre la usabilidad de la herramienta de evaluación las cuales se tuvieron en cuenta para el producto final.

Finalmente, con base en la información aportada por el personal experto y la aplicación del método a dos proyectos de software se puede afirmar que el método de evaluación de la sostenibilidad propuesto, es funcional y adaptable a las necesidades y ayudará a las Entidades Públicas a determinar la calidad del producto con un mínimo de requerimientos para la preservación de los recursos públicos cumpliendo con las recomendaciones planteadas por Colombia Compra Eficiente al principio de este trabajo..

6. REFERENCIAS

Albertao F, Xiao J, Tian C, Lu Y, Zhang K.Q, Liu C, Measuring the sustainability performance of software projects, in: 2010 IEEE 7th International Conference on E-Business Engineering, IEEE, 2010, pp. 369–373.

Anton, A. I., 1996, Goal based requirements analysis. Proceedings of the 2nd International Conference on Requirements Engineering ICRE'96, pp. 136–144.

Asian Development Bank. User's Guide To Procurement Of Information Technology Products And Services, 2018.

Avgeriou, P., Stal, M., Hilliard, R., 2013. Architecture sustainability [guest editors' introduction]. *Softw. IEEE* 30 (6), 40–44.

Becker, C. Betz, S. Chitchyan, R. Duboc, R, Easterbrook S.M, Penzenstadler B, Seyff N, Requirements: The key to sustainability, *IEEE Softw.* 33 (2015) 56–65.

Becker, C. Chitchyan, R. Easterbrook, L. Penzenstadler, B. N. Seyff, C.C. Venters, Sustainability design and software: the Karlskrona Manifesto, Proceedings of International Conference on Software Engineering, 2015, pp. 467–476.

Becker, C. Website for The Karlskrona Manifesto for Sustainability Design. 2014. Available online: <http://sustainabilitydesign.org/karlskrona-manifesto/> (accesado en 10 October 2020).

Berkhout, F. Hertin, J. Impacts of Information and Communication Technologies on Environmental Sustainability: Speculations and Evidence. Available online: <http://www.oecd.org/sti/inno/1897156.pdf> (accessed on 30 october 2020).

Business Aspect -Software Procurement Practice Guide.

- Calabrese, Julieta. Muñoz, Rocío. Asistente para la evaluación de calidad de producto de software según la familia de normas ISO/IEC 25000 utilizando el enfoque GQM, 2018.
- Calero, C. Piattini, M. Introduction to green in software engineering, in: C. Calero, M. Piattini (Eds.), *Green in Software Engineering*, Springer, Cham, 2015, pp. 3–27.
- Capilla, R., Nakagawa, E. Y., Zdun, U., & Carrillo, C. (2017). Toward Architecture Knowledge Sustainability: Extending System Longevity. *IEEE Software*, 34(2), 108–111. <https://doi.org/10.1109/MS.2017.54>
- Chitchyan, R. Becker, C. Betz, S. Duboc, L. Penzenstadler, B. Seyff, N. Venters, C. Sustainability design in requirements engineering: state of practice, in: *Proceedings of the 38th International Conference on Software Engineering Companion*, ACM, 2016, pp. 533–542.
- Christoph, B. Sustainability and longevity: Two sides of the same quality? *CEUR Workshop Proc.* 2014, 1216, 1–6.
- Cristancho Rodríguez, N. E., Martínez Sánchez, Z. C. (2018). *Compras Sostenibles, Una Herramienta Que Promueve La Enseñanza, Transferencia Y Empoderamiento De La Sostenibilidad Por Los Grupos De Interés Como Multiplicadores Positivos*. In *Bogotá : Universidad Externado de Colombia, 2018* (Vol. 53, Issue 9).
- Colombia Compra Eficiente, Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC. *Guía de Buenas Prácticas en la Adquisición de Software*, 2020).

Dick, M. Naumann, S, and Kuhn, N. A Model and Selected Instances of Green and Sustainable Software, in *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience*, J. Berleur, M. Hercheui, and L. Hilty, Editors. 2010, Springer Berlin Heidelberg. p. 248-259.

Fernandez N.C, Lago P, Luaces M.R, Places A.S, Folgueira L.G, Using participatory technical-action-research to validate a software sustainability model, in: 2019 6th International Conference on ICT for Sustainability, ICT4S, 2019.

Fernandez, N.C. and Lago, P. “Characterizing the contribution of quality requirements to software sustainability,” *Journal of systems and software*, vol. 137, pp. 289–305, Mar. 2018.

García, A. (2007). Aspectos jurídicos del software libre en Colombia. *Revista Virtual Universidad Católica del Norte*, (22),1-15.[fecha de Consulta 1 de Febrero de 2021].

García-Mireles, G. A., Moraga, M. Á., García, F., Calero, C., & Piattini, M. (2018). Interactions between environmental sustainability goals and software product quality: A mapping study. *Information and Software Technology*, 95(October 2017), 108–129.

Hefley, Bill and Loesche, Ethel A. and Khera, Pawan and Siegel, Jane A.L., *A Framework for Best Practices in the Sourcing Life-Cycle: The Architecture of the Esourcing Capability Model for Client Organizations* (November 10, 2005).

Hilty, L.M. Aebischer, B. ICT for sustainability: an emerging research field, in: L. Hilty, B. Aebischer (Eds.), *ICT Innovations for Sustainability. Advances in Intelligent Systems and Computing*, vol. 310, Springer, Cham, 2015, pp. 3–36.

- I.O. for Standardization, Systems and Software Engineering: Systems and Software Quality Requirements and Evaluation (SQuaRE): Measurement of System and Software Product Quality, ISO, 2016.
- ISO (International Organization for Standardization). (2017). ISO/IEC/IEEE 24765: 2017 Systems and software engineering-Vocabulary.
- ISO (International Organization for Standardization). (2017). ISO/IEC/IEEE 12207:2017 Systems and software engineering — Software life cycle processes.
- Institute of Electrical and Electronics Engineers. 2019. 1062-2019 - IEEE Recommended Practice for Software Acquisition.
- Johann, T., Dick, M. Naumann, S. Kern, E. How to measure energy-efficiency of software: Metrics and measurement results, in: Proceedings of the First International Workshop on Green and Sustainable Software, IEEE Press, 2012, pp. 51–54.
- Kern, E. Dick M. Naumann, S., A. Guldner, T. Johann, Green Software and Green Software Engineering—Definitions, Measurements, and Quality Aspects, 2013, pp. 87–94, Hilty et al.(2013).
- Kienzle J, Mussbacher G, Combemale B, Bastin L, Bencomo N, Bruel J, Becker C, Betz, Chitchyan R, Cheng B, et al., Towards model-driven sustainability evaluation, Commun. ACM (2019) 1–10.
- Lago, P. Koçak, S. Crnkovic, I. Penzenstadler, B. Framing sustainability as a property of software quality, Commun. ACM 58 (2015) 70–78.
- Lami, G. Buglione, L. Measuring software sustainability from a process-centric perspective, Proceedings of the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA) and 2012 Joint Conference

of the 22nd International Workshop on Software Measurement, IEEE, 2012, pp. 53–59.

Langer, A.M. Guide to Software Development—Designing and Managing the Life Cycle. 2016, p. 402. Available online: <https://www.springer.com/la/book/9781447167976> (accesado en 28 October 2020)

Manotas, I. Bird, C. Zhang, R. Shepherd, D. Jaspan, Sadowski, C. Pollock, L. Clause, J. An empirical study of practitioners' perspectives on green software engineering, Proceedings of the 38th International Conference on Software Engineering, ACM, Austin, Texas, 2016, pp. 237–248.

Ministerio de Tecnologías de la Información y las Comunicaciones MinTIC. Acerca del MinTIC, 2020.

Naumann, S. Dick, M. Kern, E. Johann, T. The GREENSOFT model: a reference model for green and sustainable software and its engineering, *Sustain. Comput.: Inform. Syst.* 1 (2011) 294–304

Oyedeji, S., Seffah, A., & Penzenstadler, B. (2018). A catalogue supporting software sustainability design. *Sustainability (Switzerland)*, 10(7), 1–30.

Osterwalder, A, Pigneur, Y, *Business Model Generation*, John Wiley & Sons, 2010.

Penzenstadler, B. Femmer, H, A generic model for sustainability with process- and product-specific instances, Proceedings of the 2013 Workshop on Green in Software Engineering, Green by Software Engineering, GIBSE 2013, 2013, pp. 3–7.

Penzenstadler, B. Raturi, A. Richardson, D. Calero, C. Femmer. Franch, H. Systematic mapping study on software engineering for sustainability (SE4S), Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (p. 14), ACM, 2014, pp. 1–10.

- Penzenstadler, B.; Martin, M.; Camille, S. RE4SuSy: Requirements Engineering for Sustainable Systems. *J. Syst. Softw.* 2013, 995.
- Penzenstadler, B. (2018). Sustainability analysis and ease of learning in artifact-based requirements engineering: The newest member of the family of studies (It's a girl!). *Information and Software Technology*, 95(April 2017), 130–146.
- Peffers, K., Tuunanen, T., Gengler, C., Rossi, M., Hui, W., Virtanen, V. y Bragge, J. (2006). *The design science research process: A model for producing and presenting information systems research*. Claremont, USA: DESRIST 2006
- Robinson W.N., Pawlowski S.S, V. Volkov, Requirement's interaction management, *ACM Comput. Surv.* 35 (2003) 132–190.
- Software Engineering Institute. 2011. CMMI® for Acquisition (CMMI-ACQ) Primer, Version 1.3
- Santacruz Espinoza, J. J., Vega Abad, C. R., Pinos Castillo, L. F., & Cárdenas Villavicencio, O. E. (2017). Sistema cobit en los procesos de auditorías de los de sistemas informáticos. *Journal of Science and Research: Revista Ciencia E Investigación. ISSN 2528-8083*, 2(8), 65-68. <https://doi.org/10.26910/issn.2528-8083vol2iss8.2017pp65-68>
- Saputri, T. R. D., & Lee, S. W. (2020). Integrated framework for incorporating sustainability design in software engineering life-cycle: An empirical study. *Information and Software Technology*, xxxx, 106407.
- United-Nation, W.C.o.E.D., Report of the world commission on environment and development: our common future, in United Nation conference on environmental G.A. García-Mireles et al. development. (1987)

7. ANEXOS

ANEXO I - PREGUNTAS Y RESPUESTAS – EVALUACIÓN DE LOS ATRIBUTOS DE CALIDAD PARA LA SOSTENIBILIDAD DE SOFTWARE CASO I.

Preguntas utilizadas para la medición de la característica Usabilidad.

ID	Preguntas de Usabilidad	(Si / No)
U1	¿El sistema permite cambiar los colores del mismo para adecuarse a las necesidades de los usuarios?	n
U2	¿El sistema permite cambiar el tamaño de la letra de sus textos?	n
U3	¿El sistema está preparado para la lectura de pantalla con voz?	n
U4	¿El sistema presenta textos difíciles de comprender?	n
U5	¿El sistema posee textos con información irrelevante?	n
U6	¿El sistema posee palabras y/o textos con faltas ortográficas?	n
U7	¿El sistema permite deshacer una acción realizada?	s
U8	¿El sistema presenta textos escritos en diferentes idiomas?	n
U9	¿El sistema brinda la opción de cambiar el lenguaje del sitio a otro idioma?	n
U10	¿El sistema posee una interfaz amigable? (El sitio puede entenderse y usarse fácilmente)	s
U11	¿El sistema indica la sección en la que se encuentra el usuario?	s
U12	¿El sistema indica las secciones accedidas hasta el momento?	n
U13	¿El sistema posee más de un término para referirse a una misma acción? (Ej: Botón aceptar, botón confirmar, botón ok)	n
U14	¿El contenido de los listados del sistema se organiza en páginas?	s
U15	¿El sistema presenta consistencia de colores en todas sus secciones?	s
U16	¿El sistema posee errores visuales? (Ej: elementos solapados, menús desplegables sin funcionar, textos en lugares no destinados a ello, etc.)	n
U17	¿El sistema informa mediante un mensaje si una operación fue realizada con éxito/sin éxito?	s
U18	¿El sistema permite salir de alguna manera de cada sección? (Ej: Atrás, Cancelar, Salir, Volver)	s
U19	¿El sistema posee atajos de teclado para el acceso a las diferentes funcionalidades?	n
U20	¿El sistema posee íconos para el acceso a las diferentes funcionalidades?	s
U21	Ante una situación de error, ¿el sistema explica claramente cómo solucionar el error ocurrido?	n
U22	Ante una situación de error, ¿el sistema explica claramente el error ocurrido?	n

U23	Ante una situación de error, ¿el sistema explica claramente cómo prevenir que vuelva a ocurrir?	n
U24	Ante varias situaciones de error, ¿la interfaz del mensaje de error se mantiene consistente?	s
U25	A la hora de completar un formulario, ¿el sistema indica el tipo de información que se espera en cada uno de los campos?	s
U26	A la hora de completar un formulario, ¿el sistema indica cuáles de sus campos son obligatorios?	s
U27	A la hora de completar un formulario, ¿el sistema permite ingresar un tipo de información que difiere con el esperado en un campo? (Ej: El sistema permite ingresar letras en un campo DNI)	n
U28	A la hora de completar un formulario, ¿existe información precargada en alguno de sus campos? (Ej: El campo país posee una lista desplegable con los diferentes países)	s
U29	En cada sección del sistema, ¿se brinda una pequeña ayuda sobre las acciones que el usuario puede realizar?	n
U30	¿El sistema posee una sección de ayuda? (Ej: Manual de usuario)	n
U31	¿El sistema posee una sección de preguntas frecuentes?	n
U32	Al utilizar la ayuda provista por el sistema, ¿Se pudo resolver la inquietud exitosamente?	n
U33	¿El sistema provee un acceso rápido a la ayuda?	n

Preguntas utilizadas para la medición de la característica Seguridad.

ID	Preguntas de Seguridad	(Si / No)
S1	¿Se requiere que la contraseña posea al menos 8 caracteres?	n
S2	¿Se requiere que la contraseña posea letras mayúsculas y minúsculas?	n
S3	¿Se requiere que la contraseña posea números y letras?	n
S4	¿Se requiere que la contraseña posea caracteres especiales?	n
S5	¿El sistema utiliza conexión segura mediante HTTPS?	s
S6	¿La base de datos posee los datos encriptados?	s
S7	¿El sistema permite acceder a funcionalidades en las cuales no se tiene permiso?	n
S8	¿El sistema permite que cualquier persona tenga acceso a la base de datos?	n
S9	¿El sistema permite que cualquier persona tenga acceso al código del servidor de la aplicación?	n
S10	¿Cualquier persona tiene acceso al servidor físico?	n
S11	¿Cualquier persona tiene acceso al servidor remoto?	n
S12	¿El sistema posee redireccionamientos hacia sitios no seguros?	n
S13	¿El sistema solicita una confirmación de registro mediante un mail a la hora de registrarse?	s
S14	¿El sistema permite que cualquier persona pueda modificar la base de datos?	n
S15	¿El sistema permite que cualquier persona pueda modificar el código del servidor de la aplicación?	n
S16	¿El sistema permite inyecciones SQL?	n
S17	¿El sistema posee un historial de acciones realizadas?	s
S18	¿El sistema posee algoritmos de cifrado de datos?	n
S19	¿El sistema posee un mecanismo criptográfico, como firma digital?	n
S20	¿El sistema solicita confirmación a la hora de realizar una acción?	n
S21	¿El sistema posee una protección con certificados SSL?	s
S22	¿El sistema da aviso cuando se es accedido desde una ubicación desconocida?	n
S23	¿El sistema informa vía mail las operaciones realizadas?	n
S24	¿El sistema guarda un registro de fecha y hora de ingreso al mismo?	s
S25	¿El sistema registra el tipo de navegador y sistema de operación utilizado para ingresar al sitio?	s
S26	¿El sistema registra la dirección IP desde la cual se ingresa al sitio?	s

S27	¿El sistema realiza una comprobación de identidad mediante un certificado digital?	s
S28	¿El sistema posee un sistema de verificación en dos pasos?	n
S29	¿Es requerida una clave de segundo nivel para el ingreso al sistema?	n
S30	¿El sistema realiza una comprobación de identidad mediante datos biométricos?	n
S31	¿El sistema realiza una comprobación de identidad mediante tarjeta de coordenadas?	n
S32	¿El sistema realiza una comprobación de identidad mediante credenciales?	s
S33	¿El sistema realiza una comprobación de identidad mediante una firma electrónica?	n

Preguntas utilizadas para la medición de la característica Portabilidad.

ID	Preguntas de Portabilidad	(Si / No)
P1	¿El sistema posee una aplicación móvil?	n
P2	¿El sistema funciona correctamente en diferentes computadoras con diferentes características?	s
P3	¿El sistema se utiliza correctamente en un navegador de una Tablet?	s
P4	¿El sistema puede ser utilizado en dispositivos con cualquier sistema operativo?	s
P5	¿El sistema funciona correctamente en cualquier navegador de internet?	s
P6	¿El sistema funciona correctamente en el navegador de un dispositivo móvil?	s

Preguntas utilizadas para la medición de la característica Compatibilidad.

ID	Preguntas de Compatibilidad	(Si / No)
C1	¿Es posible utilizar los recursos de la computadora por otros sistemas mientras se utiliza el sistema?	s
C2	¿Es posible ejecutar el sistema si se están ejecutando otros?	s
C3	¿Se producen errores inesperados al ejecutar el sistema cuando hay otros ejecutándose?	n
C4	¿El sistema permite intercambiar información con otros sistemas?	n
C5	¿Es posible utilizar información brindada por otro sistema?	n
C6	¿La información del sistema puede ser utilizada por otro sistema?	n
C7	¿Se producen errores de algún tipo al intentar utilizar información compartida con otros sistemas?	n

C8	¿La funcionalidad del sistema se ve alterada por el uso de otro sistema al mismo tiempo?	n
----	--	---

Preguntas utilizadas para la medición de la característica Mantenibilidad.

ID	Preguntas de Mantenibilidad	(Si / No)
M1	¿El sistema posee componentes que se pueden reutilizar dentro del mismo software?	n
M2	¿El sistema posee componentes reutilizados dentro del mismo software?	n
M3	¿El sistema posee componentes que se pueden utilizar para la construcción de otro software?	n
M4	¿El sistema está construido de forma modular en todos sus componentes?	s
M5	¿El sistema permite cambios en los módulos de tal manera que no impacte su funcionalidad del programa?	s
M6	¿El sistema permite cambios en los módulos de tal manera que no impacte su funcionalidad de otros módulos?	s

Preguntas utilizadas para la medición de la característica Eficiencia de Rendimiento.

ID	Preguntas de Eficiencia en Rendimiento	(Si / No)
ER1	¿El programa hace uso de la CPU de tal manera que permita la ejecución de otros programas?	s
ER2	¿El programa hace uso de la memoria Ram de tal manera que permita la ejecución de otros programas?	s
ER2	¿Al hacer uso intensivo del programa el sistema permite la ejecución de otros programas?	s

Preguntas utilizadas para la medición de la característica Efectividad.

ID	Preguntas de Efectividad	(Si / No)
E1	¿Al ejecutar una tarea del programa esta se ejecuta correctamente?	s
E2	¿Al ejecutar una tarea una misma tarea varias veces estas se completan correctamente?	s
E2	¿Al ejecutar las tareas del programa estas alcanzan el objetivo deseado?	s

Preguntas utilizadas para la medición de la característica Fiabilidad.

ID	Preguntas de Fiabilidad	(Si / No)
F1	¿El programa estará disponible para su uso las 24 horas al día?	s
F2	¿El programa estará disponible para su uso los 7 días de la semana?	s
F3	¿El programa soporta el uso de varios usuarios a la vez?	s

F4	¿El programa se bloquea si eventualmente ocurre un fallo?	n
F5	¿El programa se reinicia si eventualmente ocurre un fallo?	s
F6	¿El programa recupera los datos afectados si eventualmente ocurre un fallo?	n
F7	¿El programa regresa a un estado deseado si eventualmente ocurre un fallo?	s
F8	¿El programa puede operar si falla algunos de sus componentes de Software?	s
F9	¿El programa tiene implementado en su software algún componente para la prevención de fallos?	n
F10	¿El programa tiene implementado en su hardware algún componente para la prevención de fallos?	s

Preguntas utilizadas para la medición de la característica Adecuación Funcional.

ID	Preguntas de Adecuación Funcional	(Si / No)
AF1	¿El programa provee los resultados correctos cuando se ejecutan determinadas tareas o funciones?	s
AF2	¿El programa provee los resultados precisos cuando se ejecutan determinadas tareas o funciones?	s
AF3	¿El programa tiene las mínimas tareas para su objetivo?	s
AF4	¿El programa tiene todas las tareas para su objetivo?	s

EVALUACIÓN DE MÉTRCIAS CASO I.

Medición de la Confidencialidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación	
Confidencialidad	Conexiones seguras	$S5 \ \& \ \neg S12 = V$	1	1	
	Control de acceso	$S7 \ \ S8 \ \ S9 \ \ S10 \ \ S11 = F$	1	1	
	Encriptación de datos	$S6 = V$	1	1	
	Contraseña		$S1 \ \ S2 \ \ S3 \ \ S4 = F$	0	0
			$S1 \ \ S2 \ \ S3 \ \ S4 = V$	0.5	0
			$S1 \ \& \ S2 \ \& \ S3 \ \& \ S4 = V$	1	0
Total (Evaluación / Puntaje)			4.5	3	
			0.66 - Cumple Parcialmente		

Medición de la Autenticidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Autenticidad	Comprobación de identidad	$S27 \ \ S30 \ \ S31 \ \ S32 \ \ S33 = V$	1	1
	Comprobaciones adicionales	$S28 \ \ S29 \ \ S13 = V$	1	1
Total (Evaluación / Puntaje)			2	2
			1.00 - Cumple	

Medición de la Integridad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Integridad	Prevención de accesos	$S7 S8 S9 S16 = F$	1	1
	Prevención de modificaciones	$S14 S15 = F$	1	1
	Confirmación de datos	$S13 = V$	1	1
Total (Evaluación / Puntaje)			3	3
			1.00 - Cumple	

Medición de la Accesibilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Accesibilidad	Accesibilidad para usuarios con dificultad visual	$U3 = V$	0.5	0
		$U3 \& (U1 U2) = V$	0.75	0
		$(U1 U2) \& \neg(U3) = V$	0.25	0
		$U1 \& U2 \& U3 = V$	1	0
	Atajos desde teclado	$U19 = V$	1	0
	Manejo de idiomas	$\neg(U8) \& U9 = V$	1	0
		$\neg(U8) U9 = V$	0.75	0
Total (Evaluación / Puntaje)			5.25	0
			0.00 - No Cumple	

Medición de la Reusabilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Mantenibilidad	Reusabilidad de Componentes internos	$M1 \mid M2 \mid = V$	0.5	0
		$M1 \ \& \ M2 \mid = V$	1	0
	Reusabilidad de Componentes externos	$M3 = V$	1	0
Total (Evaluación / Puntaje)			2.5	0
			0.00 - No Cumple	

Medición de la Utilización de Recursos

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Utilización de Recursos	Rendimiento CPU y RAM	$ER1 \ \& \ ER2 \mid = V$	1	1
		$ER1 \mid ER2 = V$	0.5	0.5
	Rendimiento Otros Programas	$ER3 = V$	1	1
Total (Evaluación / Puntaje)			2.5	0
			1.00 - Cumple	

Medición de la Efectividad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Efectividad	Ejecución de Tareas	$E1 \mid E2 \mid = V$	1	1
		$E1 \ \& \ E2 = V$	0.5	0.5
	Cumplimiento de Tareas	$E3 = V$	1	1

Total (Evaluación / Puntaje)	2.5	2
	1.00 - Cumple	

Medición de la Disponibilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Disponibilidad	Disponibilidad del Sistema	$F1 \ \& \ F2 = V$	1	1
		$F1 F2 = V$	0.5	0.5
	Disponibilidad Multiusuario	$F3 = V$	1	1
Total (Evaluación / Puntaje)			2.5	2.5
			1.00 - Cumple	

Medición de la Recuperabilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Recuperabilidad	Reinicio Bloqueo del Sistema por fallo	$\neg F4 \ \& \ \neg F5 = V$	1	1
		$\neg F4 \neg F5 = V$	0.5	0.5
	Recuperación de Datos por Fallo	$F6 \ \& \ F7 = V$	1	0
Total (Evaluación / Puntaje)			2.5	1.5
			0.60 – Cumple Parcialmente	

Medición de la Adecuación Funcional

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Correctitud Funcional	Ejecución de Tareas Correctas y Precisas	$AF1 \ \& \ AF2 = V$	1	1
Perterencia Funcional	Compleitud de Tareas para los usuarios	$AF3 \ \ AF4 = V$	1	1
Total (Evaluación / Puntaje)			2	2
			1.00 - Cumple	

Medición de la Interoperabilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Interoperabilidad	Intercambio de información	$C4 = F$	0	0
		$C4 \ \& \ C5 \ \& \ C6 = V$	1	0
		$(C4 \ \& \ C5) \ \ (C4 \ \& \ C6) = V$	0.75	0
	Errores inesperados	$C7 = F$	1	1
Total (Evaluación / Puntaje)			2.75	1
			0.36 - No Cumple	

Medición de la Tolerancia a Fallos

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Tolerancia a Fallos	Operación Frente a Fallo	$F8 = V$	1	1
	Protección Frente a Fallos	$F9 \ \& \ F10 = V$	1	0
		$F9 \ \ F10 = V$	0.5	0.5
Total (Evaluación / Puntaje)			2.5	1.5

	0.60 - Cumple Parcialmente
--	----------------------------

Medición de la Modificabilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Modificabilidad	Modificación de componentes sin afectar otros módulos	$M5 = V$	1	1
	Modificación de componentes sin afectar el software	$M6 = V$	1	1
Total (Evaluación / Puntaje)			2	2
			1.00 - Cumple	

Medición de la Modularidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Modularidad	Modularidad de componentes	$M4 = V$	1	1
Total (Evaluación / Puntaje)			1	1
			1.00 - Cumple	

Medición de la Adaptabilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Adaptabilidad	Web Responsive	$P3 \& P6 = V$	1	1
		$P3 P6 = V$	0.5	0.5

	Aplicación Móvil	P1 = V	1	0
	Adaptación al Hardware	P2 = V	1	1
	Adaptación al Software	P4 & P5 = V	1	1
		P4 P5 = V	0.75	0.75
Total (Evaluación / Puntaje)			5.25	4.25
			0.80 - Cumple	

ANEXO II. PREGUNTAS Y RESPUESTAS – EVALUACIÓN DE LOS ATRIBUTOS DE CALIDAD PARA LA SOSTENIBILIDAD DE SOFTWARE CASO II.

Preguntas utilizadas para la medición de la característica Usabilidad.

ID	Preguntas de Usabilidad	(Si / No) Móvil	(Si / No) Web
U1	¿El sistema permite cambiar los colores del mismo para adecuarse a las necesidades de los usuarios?	n	s
U2	¿El sistema permite cambiar el tamaño de la letra de sus textos?	n	n
U3	¿El sistema está preparado para la lectura de pantalla con voz?	n	n
U4	¿El sistema presenta textos difíciles de comprender?	n	n
U5	¿El sistema posee textos con información irrelevante?	n	n
U6	¿El sistema posee palabras y/o textos con faltas ortográficas?	n	n
U7	¿El sistema permite deshacer una acción realizada?	Si, manual, no ^z	Si, manual, no ^z
U8	¿El sistema presenta textos escritos en diferentes idiomas?	n	n
U9	¿El sistema brinda la opción de cambiar el lenguaje del sitio a otro idioma?	n	n
U10	¿El sistema posee una interfaz amigable? (El sitio puede entenderse y usarse fácilmente)	s	s
U11	¿El sistema indica la sección en la que se encuentra el usuario?	s	s
U12	¿El sistema indica las secciones accedidas hasta el momento?	s	n
U13	¿El sistema posee más de un término para referirse a una misma acción? (Ej: Botón aceptar, botón confirmar, botón ok)	n	n
U14	¿El contenido de los listados del sistema se organiza en páginas?	s	s
U15	¿El sistema presenta consistencia de colores en todas sus secciones?	s	s
U16	¿El sistema posee errores visuales? (Ej: elementos solapados, menús desplegables sin funcionar, textos en lugares no destinados a ello, etc.)	n	S - pero parece ser que es normal actualmente. Me refiero a

			alguna falta de contraste
U17	¿El sistema informa mediante un mensaje si una operación fue realizada con éxito/sin éxito?	s	s
U18	¿El sistema permite salir de alguna manera de cada sección? (Ej: Atrás, Cancelar, Salir, Volver)	s	s
U19	¿El sistema posee atajos de teclado para el acceso a las diferentes funcionalidades?	n	n
U20	¿El sistema posee íconos para el acceso a las diferentes funcionalidades?	s	s
U21	Ante una situación de error, ¿el sistema explica claramente cómo solucionar el error ocurrido?	n	n
U22	Ante una situación de error, ¿el sistema explica claramente el error ocurrido?	s	s
U23	Ante una situación de error, ¿el sistema explica claramente cómo prevenir que vuelva a ocurrir?	n	n
U24	Ante varias situaciones de error, ¿la interfaz del mensaje de error se mantiene consistente?	s	s
U25	A la hora de completar un formulario, ¿el sistema indica el tipo de información que se espera en cada uno de los campos?	s	s
U26	A la hora de completar un formulario, ¿el sistema indica cuáles de sus campos son obligatorios?	s	s
U27	A la hora de completar un formulario, ¿el sistema permite ingresar un tipo de información que difiere con el esperado en un campo? (Ej: El sistema permite ingresar letras en un campo DNI)	n	n
U28	A la hora de completar un formulario, ¿existe información precargada en alguno de sus campos? (Ej: El campo país posee una lista desplegable con los diferentes países)	s	s
U29	En cada sección del sistema, ¿se brinda una pequeña ayuda sobre las acciones que el usuario puede realizar?	n	n
U30	¿El sistema posee una sección de ayuda? (Ej: Manual de usuario)	n	n
U31	¿El sistema posee una sección de preguntas frecuentes?	n	n
U32	Al utilizar la ayuda provista por el sistema, ¿Se pudo resolver la inquietud exitosamente?	n	n
U33	¿El sistema provee un acceso rápido a la ayuda?	n	n

Preguntas utilizadas para la medición de la característica Seguridad.

ID	Preguntas de Seguridad	(Si / No) Móvil	(Si / No) Web
S1	¿Se requiere que la contraseña posea al menos 8 caracteres?	s	s
S2	¿Se requiere que la contraseña posea letras mayúsculas y minúsculas?	s	s
S3	¿Se requiere que la contraseña posea números y letras?	s	s
S4	¿Se requiere que la contraseña posea caracteres especiales?	s	s
S5	¿El sistema utiliza conexión segura mediante HTTPS?	n	s
S6	¿La base de datos posee los datos encriptados?	s	s
S7	¿El sistema permite acceder a funcionalidades en las cuales no se tiene permiso?	n	n
S8	¿El sistema permite que cualquier persona tenga acceso a la base de datos?	n	n
S9	¿El sistema permite que cualquier persona tenga acceso al código del servidor de la aplicación?	n	n
S10	¿Cualquier persona tiene acceso al servidor físico?	n	n
S11	¿Cualquier persona tiene acceso al servidor remoto?	n	n
S12	¿El sistema posee redireccionamientos hacia sitios no seguros?	n	n
S13	¿El sistema solicita una confirmación de registro mediante un mail a la hora de registrarse?	s	s
S14	¿El sistema permite que cualquier persona pueda modificar la base de datos?	n	n
S15	¿El sistema permite que cualquier persona pueda modificar el código del servidor de la aplicación?	n	n
S16	¿El sistema permite inyecciones SQL?	n	n
S17	¿El sistema posee un historial de acciones realizadas?	n	n
S18	¿El sistema posee algoritmos de cifrado de datos?	s	s
S19	¿El sistema posee un mecanismo criptográfico, como firma digital?	n	n
S20	¿El sistema solicita confirmación a la hora de realizar una acción?	s	s
S21	¿El sistema posee una protección con certificados SSL?	s	s
S22	¿El sistema da aviso cuando se es accedido desde una ubicación desconocida?	n	n
S23	¿El sistema informa vía mail las operaciones realizadas?	n	n

S24	¿El sistema guarda un registro de fecha y hora de ingreso al mismo?	s	s
S25	¿El sistema registra el tipo de navegador y sistema de operación utilizado para ingresar al sitio?	n	n
S26	¿El sistema registra la dirección IP desde la cual se ingresa al sitio?	n	n
S27	¿El sistema realiza una comprobación de identidad mediante un certificado digital?	n	n
S28	¿El sistema posee un sistema de verificación en dos pasos?	n	n
S29	¿Es requerida una clave de segundo nivel para el ingreso al sistema?	n	n
S30	¿El sistema realiza una comprobación de identidad mediante datos biométricos?	n	n
S31	¿El sistema realiza una comprobación de identidad mediante tarjeta de coordenadas?	n	n
S32	¿El sistema realiza una comprobación de identidad mediante credenciales?	S -psw	S -psw
S33	¿El sistema realiza una comprobación de identidad mediante una firma electrónica?	n	n

Preguntas utilizadas para la medición de la característica Portabilidad.

ID	Preguntas de Portabilidad	(Si / No) Móvil	(Si / No) Web
P1	¿El sistema posee una aplicación móvil?	s	s
P2	¿El sistema funciona correctamente en diferentes computadoras con diferentes características?	s	s
P3	¿El sistema se utiliza correctamente en un navegador de una Tablet?	s	s
P4	¿El sistema puede ser utilizado en dispositivos con cualquier sistema operativo?	Actualmente no, solo Android, pero se puede compilar para OIS, solo esos dos	n

		conozco en el mundo móvil	
P5	¿El sistema funciona correctamente en cualquier navegador de internet?	s	s
P6	¿El sistema funciona correctamente en el navegador de un dispositivo móvil?	s	s

Preguntas utilizadas para la medición de la característica Compatibilidad.

ID	Preguntas de Compatibilidad	(Si / No) Móvil	(Si / No) Web
C1	¿Es posible utilizar los recursos de la computadora por otros sistemas mientras se utiliza el sistema?	s	s
C2	¿Es posible ejecutar el sistema si se están ejecutando otros?	s	s
C3	¿Se producen errores inesperados al ejecutar el sistema cuando hay otros ejecutándose?	n	n
C4	¿El sistema permite intercambiar información con otros sistemas?	n	s
C5	¿Es posible utilizar información brindada por otro sistema?	s	s
C6	¿La información del sistema puede ser utilizada por otro sistema?	n	s
C7	¿Se producen errores de algún tipo al intentar utilizar información compartida con otros sistemas?	n	n
C8	¿La funcionalidad del sistema se ve alterada por el uso de otro sistema al mismo tiempo?	n	n

Preguntas utilizadas para la medición de la característica Mantenibilidad.

ID	Preguntas de Mantenibilidad	(Si / No) Móvil	(Si / No) Web
M1	¿El sistema posee componentes que se puede reutilizar dentro del mismo software?	s	s
M2	¿El sistema posee componentes reutilizados dentro del mismo software?	s	s
M3	¿El sistema posee componentes que se puede para la construcción de otro software?	s	s
M4	¿El sistema está construido de forma modular en todos sus componentes?	s	s
M5	¿El sistema permite cambios en los módulos de tal manera que no impacte su funcionalidad del programa?	s	s

M6	¿El sistema permite cambios en los módulos de tal manera que no impacte su funcionalidad de otros módulos?	s	s
----	--	---	---

Preguntas utilizadas para la medición de la característica Eficiencia de Rendimiento.

ID	Preguntas de Eficiencia en Rendimiento	(Si / No) Móvil	(Si / No) Web
ER1	¿El programa hace uso de la CPU de tal manera que permita la ejecución de otros programas?	s	s
ER2	¿El programa hace uso de la memoria Ram de tal manera que permita la ejecución de otros programas?	s	s
ER2	¿Al hacer uso intensivo del programa el sistema permite la ejecución de otros programas?	s	s

Preguntas utilizadas para la medición de la característica Efectividad.

ID	Preguntas de Efectividad	(Si / No) Móvil	(Si / No) Web
E1	¿Al ejecutar una tarea del programa esta se ejecuta correctamente?	s	s
E2	¿Al ejecutar una tarea una misma tarea varias veces estas se completan correctamente?	s	s
E2	¿Al ejecutar las tareas del programa estas alcanzan el objetivo deseado?	s	s

Preguntas utilizadas para la medición de la característica Fiabilidad.

ID	Preguntas de Fiabilidad	(Si / No) Móvil	(Si / No) Web
F1	¿El programa estará disponible para su uso las 24 horas al día?	s	s
F2	¿El programa estará disponible para su uso los 7 días de la semana?	s	s
F3	¿El programa soporta el uso de varios usuarios a la vez?	s	s
F4	¿El programa se bloquea si eventualmente ocurre un fallo?	n	n
F5	¿El programa se reinicia si eventualmente ocurre un fallo?	n	n
F6	¿El programa recupera los datos afectados si eventualmente ocurre un fallo?	s	s
F7	¿El programa regresa a un estado deseado si eventualmente ocurre un fallo?	s	s

F8	¿El programa puede operar si falla algunos de sus componentes de software?	s	s
F9	¿El programa tiene implementado en su software algún componente para la prevención de fallos?	s	s
F10	¿El programa tiene implementado en su hardware algún componente para la prevención de fallos? (UPS, Prevención de cortes de energía)	s	s
F11	¿El programa tiene implementado en su hardware algún componente para la prevención de fallos?	s	s

Preguntas utilizadas para la medición de la característica Adecuación Funcional.

ID	Preguntas de Adecuación Funcional	(Si / No) Móvil	(Si / No) Web
AF1	¿El programa provee los resultados correctos cuando se ejecutan determinadas tareas o funciones?	s	s
AF2	¿El programa provee los resultados precisos cuando se ejecutan determinadas tareas o funciones?	s	s
AF3	¿El programa tiene las mínimas tareas para su objetivo?	s	s
AF4	¿El programa tiene todas las tareas para su objetivo?	s	s

EVALUACIÓN DE MÉTRCIAS CASO II.

Medición de la Confidencialidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación	
Confidencialidad	Conexiones seguras	$S5 \ \& \ \neg S12 = V$	1	1	
	Control de acceso	$S7 \ \ S8 \ \ S9 \ \ S10 \ \ S11 = F$	1	1	
	Encriptación de datos	$S6 = V$	1	1	
	Contraseña		$S1 \ \ S2 \ \ S3 \ \ S4 = F$	0	0
			$S1 \ \ S2 \ \ S3 \ \ S4 = V$	0.5	0.5
			$S1 \ \& \ S2 \ \& \ S3 \ \& \ S4 = V$	1	1
Total (Evaluación / Puntaje)			4.5	4.5	
			1.00 - Cumple		

Medición de la Autenticidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Autenticidad	Comprobación de identidad	$S27 \ \ S30 \ \ S31 \ \ S32 \ \ S33 = V$	1	1
	Comprobaciones adicionales	$S28 \ \ S29 \ \ S13 = V$	1	1
Total (Evaluación / Puntaje)			2	2
			1.00 - Cumple	

Medición de la Integridad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Integridad	Prevención de accesos	$S7 \ \ S8 \ \ S9 \ \ S16$	1	1

		= F		
	Prevención de modificaciones	S14 S15 = F	1	1
	Confirmación de datos	S13 = V	1	1
Total (Evaluación / Puntaje)			3	3
			1.00 - Cumple	

Medición de la Accesibilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Accesibilidad	Accesibilidad para usuarios con dificultad visual	$U3 = V$	0.5	0
		$U3 \& (U1 U2) = V$	0.75	0
		$(U1 U2) \& \neg(U3) = V$	0.25	0.25
		$U1 \& U2 \& U3 = V$	1	0
	Atajos desde teclado	$U19 = V$	1	0
	Manejo de idiomas	$\neg(U8) \& U9 = V$	1	0
		$\neg(U8) U9 = V$	0.75	0.75
Total (Evaluación / Puntaje)			5.25	1
			0.20 - No Cumple	

Medición de la Reusabilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Reusabilidad	Reusabilidad de Componentes internos	$M1 M2 = V$	0.5	0.5

		$M1 \& M2 \neq V$	1	1
	Reusabilidad de Componentes externos	$M3 = V$	1	1
Total (Evaluación / Puntaje)			2.50	1
			1.00 - No Cumple	

Medición de la Utilización de Recursos

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Utilización de Recursos	Rendimiento CPU y RAM	$ER1 \& ER2 \neq V$	1	1
		$ER1 \mid ER2 = V$	0.5	0.5
	Rendimiento Otros Programas	$ER3 = V$	1	1
Total (Evaluación / Puntaje)			2.5	2.5
			1.00 - Cumple	

Medición de la Efectividad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Efectividad	Ejecución de Tareas	$E1 \mid E2 = V$	1	1
		$E1 \& E2 = V$	0.5	0.5
	Cumplimiento de Tareas	$E3 = V$	1	1
Total (Evaluación / Puntaje)			2.5	2.5
			1.00 - Cumple	

Medición de la Disponibilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
-------------------	---------------	---------	---------	------------

Fiabilidad	Disponibilidad del Sistema	$F1 \ \& \ F2 = V$	1	1
		$F1 F2 = V$	0.5	0.5
	Disponibilidad Multiusuario	$F3 = V$	1	1
Total (Evaluación / Puntaje)			2.5	2.5
			1.00 - Cumple	

Medición de la Recuperabilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Recuperabilidad	Reinicio Bloqueo del Sistema por fallo	$\neg F4 \ \& \ \neg F5 = V$	1	1
		$\neg F4 \neg F5 = V$	0.5	0.5
	Recuperación de Datos por Fallo	$F6 \ \& \ F7 = V$	1	0
Total (Evaluación / Puntaje)			2.5	2.5
			1.00 - Cumple Parcialmente	

Medición de Adecuación Funcional

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Corrección Funcional	Ejecución de Tareas Correctas y Precisas	$AF1 \ \& \ AF2 = V$	1	1
Pertinencia Funcional	Compleitud de Tareas para los usuarios	$AF3 AF4 = V$	1	1
Total (Evaluación / Puntaje)			2	2
			1.00 - Cumple	

Medición de la Interoperabilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
-------------------	---------------	---------	---------	------------

Interoperabilidad	Intercambio de información	$C4 = F$	0	0
		$C4 \& C5 \& C6 = V$	1	1
		$(C4 \& C5) (C4 \& C6) = V$	0.75	0.75
	Errores inesperados	$C7 = F$	1	1
Total (Evaluación / Puntaje)			2.75	2.75
			1.00 Cumple	

Medición de la Tolerancia a Fallos

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Tolerancia a Fallos	Operación Frente a Fallo	$F8 = V$	1	1
	Protección Frente a Fallos	$F9 \& F10 = V$	1	1
		$F9 F10 = V$	0.5	0.5
Total (Evaluación / Puntaje)			2.5	2.5
			1.00 - Cumple	

Medición de la Modificabilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Modificabilidad	Modificación de componentes	$M5 M6 = V$	1	1
		$M5 \& M6 = V$	1	1
Total (Evaluación / Puntaje)			2	2
			1.00 - Cumple	

Medición de la Modularidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Modularidad	Modularidad de componentes	$M4 = V$	1	1
Total (Evaluación / Puntaje)			1	1
			1.00 - Cumple	

Medición de la Adaptabilidad

Subcaracterística	Nombre del CE	Fórmula	Puntaje	Evaluación
Adaptabilidad	Web Responsive	$P3 \ \& \ P6 = V$	1	1
		$P3 \ \ P6 = V$	0.5	0.5
	Aplicación Móvil	$P1 = V$	1	1
	Adaptación al Hardware	$P2 = V$	1	1
	Adaptación al Software	$P4 \ \& \ P5 = V$	1	0
		$P4 \ \ P5 = V$	0.75	0.75
Total (Evaluación / Puntaje)			5.25	4.25
			0.80 - Cumple	

ANEXO III. MEDICIÓN ATRIBUTOS DE CALIDAD DE LA FAMILIA ISO 25000.

Sostenibilidad Social.

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de la Seguridad (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Confidencialidad	Capacidad de control de accesos	¿Qué tan controlable son los accesos al sistema?	Contar el número de diferentes tipos de operaciones ilegales detectado.	$X = A / B$ A=Número de diferentes tipos de operaciones ilegales detectados B = Número de tipos de operaciones ilegales en la especificación Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor
Confidencialidad	Encriptación de datos	¿Qué tan implementada o está la encriptación de datos?	Contar el número de elementos de datos encriptados/desencriptados correctamente	$X = A / B$ A=Número de elementos de datos encriptados/desencriptados Correctamente B=Número de elementos de datos totales	$0 \leq X \leq 1$ El más cercano a 1, es el mejor
Autenticidad	Métodos de autenticación	¿Qué tan bien el sistema auténtica la identidad de un sujeto?	Contar el número de métodos de autenticación previstos.	$X = A$ A = Número de métodos de autenticación previstos	$X \geq 0$ Donde X es mayor a 0, siendo X el mejor igual o mayor a 2
Integridad	Prevención de Datos Corruptos	¿Hasta qué punto se puede prevenir la corrupción de datos?	Contar el número de casos de corrupción de datos ocurridos en la actualidad y el número de accesos donde se espera que ocurran daños de datos.	$X=A/B$ A = Número de casos de corrupción de datos ocurridos en la actualidad. B = Número de accesos donde se espera que ocurran daños de datos Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 0, es el mejor

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica Facilidad de Uso (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Accesibilidad	Accesibilidad	¿A qué cantidad de funciones puede acceder un usuario con discapacidades físicas?	Contar el número de funciones a las que pueden acceder personas con discapacidad y contar el número total de funciones implementadas.	$X = A/B$ A= Número de funciones a las que pueden acceder personas con discapacidad. B = Número total de elementos de interfaz Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor

Sostenibilidad Ambiental.

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de la Mantenibilidad (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Reusabilidad	Ejecución de reusabilidad	¿Cuántos elementos pueden ser reutilizados?	Contar el número de elementos reutilizados y el número total de elementos de la biblioteca reutilizable.	$X = A / B$ A = Número de elementos reutilizados B = Número total de elementos de la biblioteca reutilizable Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1, es el mejor

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de Rendimiento Eficiencia (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Utilización de recursos	Utilización de CPU	¿Cuánto tiempo de CPU es usado para realizar una tarea dada?	Tomar el tiempo de operación y la cantidad de tiempo de CPU que se usa para realizar una tarea.	$X = B - A$ A = La cantidad de tiempo de CPU que realmente es usado para realizar una tarea B = Tiempo de operación	$0 \leq X \leq 1$ Cuanto más se acerque a 0 es lo mejor. Donde el peor caso es $\geq 15t$.
Utilización de recursos	Utilización de la memoria	¿Cuánto espacio de memoria es usado para realizar una tarea dada?	Medir la cantidad total de espacios de memoria y la cantidad de espacios de memoria que realmente es usado para realizar una tarea.	$X = B - A$ A = Cantidad de espacios de memoria que realmente es usado para realizar una tarea B = Cantidad total de espacios de memoria Dónde: $B > 0$	$0 \leq X \leq 15$ El más cercano a 0 es el mejor
Utilización de recursos	Utilización de los dispositivos de E/S	¿Cuánto tiempo los dispositivos de E/S utilizan para realizar una tarea?	Tomar el tiempo de operación y el tiempo que los dispositivos de E/S pasan ocupados para realizar la tarea.	$X = B - A$ A = Tiempo que los dispositivos de E/S pasan ocupados para realizar la tarea B = Tiempo de operación	$0 \leq X \leq 15$ El más cercano a 0 es el mejor

Sostenibilidad Económica.

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de Efectividad (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Efectividad	Completitud de la tarea	¿Qué cantidad de tareas son completadas correctamente?	Contar el número de tareas completadas y el número total de tareas intentadas.	$X = A/B$ A= Número de tareas completadas. B = Número total de tareas intentadas	$0 \leq X \leq 1$ El más cercano a 1, el mejor
Efectividad	Efectividad de la tarea	¿Qué cantidad de los objetivos de la tarea se realiza completamente?	Tomar el valor proporcional de cada componente faltante o incorrecto en la salida de la tarea.	$X = A/B$ A=Cantidad de objetivos completados por la tarea. B=Cantidad de objetivos planeados.	$0 \leq X \leq 1$ El más cercano a 1, el mejor
Efectividad	Frecuencia de error	¿Cuál es la frecuencia de los errores cometidos por el usuario en comparación con lo planeado?	Contar el número de errores cometidos por los usuarios y contar el número de tareas.	$X = A/B$ A = Número de errores cometidos por los usuarios B = Número de tareas	$0 \leq X \leq 1$ El más cercano a 0 es el mejor

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de Fiabilidad (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Disponibilidad	Tiempo de servicio	¿Cuál es el tiempo de servicio del sistema que proporciona realmente?	Tomar el tiempo de servicio del sistema que se proporciona actualmente y tomar el tiempo de servicio del sistema regulado en el cronograma operacional.	$X = A/B$ A = Tiempo de servicio del sistema que se proporciona actualmente B = Tiempo de servicio del sistema regulado en el cronograma operacional	$0 \leq X \leq 1$ Cuanto más se acerque a 1 es lo mejor
Recuperabilidad	Recuperabilidad	¿Cuál es el tiempo promedio que toma el sistema en recuperarse completamente después un fallo?	Tomar el tiempo que le tomó al sistema en recuperarse y contar el número de casos en los cuales se ha observado que el sistema entró en recuperación.	$X = A / T$ A = Número de casos en los cuales se ha observado que el sistema entró en recuperación. T = Tiempo que le tomó al sistema en recuperarse Dónde: $T > 0$	$X = A/T$ El más cercano a 0/t es el mejor. Donde el peor caso es $\geq 10/t$

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de Adecuación Funcional en la Dimensión Económica (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Pertinencia funcional	Completitud de la implementación funciona	¿Cuán completa es la implementación de acuerdo a la especificación de requerimientos?	Contar el número de las funciones indicadas en la especificación de requerimientos y el número de funciones que faltan o están incorrecta	$X = A / B$ A = Número de funciones que están incorrectas o que no fueron implementadas B = Número de las funciones establecidas en la especificación de requisitos	$0 \leq X \leq 1$ El más cercano a 0 es el mejor

Sostenibilidad Técnica.

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de Compatibilidad en la Dimensión Técnica (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Interoperabilidad	Conectividad con sistemas externos	¿Qué tan correctamente se ha implementado los protocolos de interfaz externa?	Contar el número de interfaces implementadas con otros sistemas y el número total de interfaces externas	$X = A/B$ A= Número de interfaces implementadas con otros sistemas B = Número total de interfaces externas Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor
Interoperabilidad	Capacidad de intercambiar de datos	¿Qué tan exacto es el intercambio de datos entre el sistema otros sistemas de enlace?	Contar el número de datos que se han intercambiado sin problemas con otro sistema y el número total de datos que se intercambiarán	$X = A/B$ A= Número de datos que se han intercambiado sin problemas con otro sistema B = Número total de datos que se intercambiarán Dónde: $B > 0$	$0 \leq X \leq 1$ El más cercano a 1 es el mejor

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de Adecuación Funcional en la Dimensión Técnica (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Pertinencia Funcional	Exactitud	¿Cuánto del estándar requerido de exactitud se cumple?	Contar el número de elementos de datos implementados con el estándar específico de exactitud y el número total de elementos de datos implementados	$X = A/B$ A=Número de elementos de datos implementados con el estándar específico de exactitud B = Número total de elementos de datos implementados	$0 \leq X \leq 1$ El más cercano a 0 es el mejor
Pertinencia Funcional	Precisión computacional	¿Con qué frecuencia ocurren los resultados inexactos?	Contar el número de Cálculos inexactos encontrados y tomar el tiempo de operación	$X = A/T$ A =Numero de cálculos inexactos encontrados T=Tiempo de operación	$X = A/T$ El más cercano a 0/t es el mejor. Donde el peor caso es $\geq 10/t$

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de Compatibilidad (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Interoperabilidad	Conectividad con sistemas externos	¿Qué tan correctamente se ha implementado los protocolos de interfaz externa?	Contar el número de interfaces implementadas con otros sistemas y el número total de interfaces externas	$X = A/B$ A= Número de interfaces implementadas con otros sistemas B = Número total de interfaces externas	$0 \leq X \leq 1$ El más cercano a 1 es el mejor
Interoperabilidad	Capacidad de intercambiar de datos	¿Qué tan exacto es el Intercambio de datos entre el sistema otros sistemas de enlace?	Contar el número de datos que se han intercambiado sin problemas con otro sistema y el número total de datos que se intercambiaran	$X = A/B$ A= Número de datos que se han Intercambiado sin problemas con otro sistema B = Número total de datos que se intercambiara	$0 \leq X \leq 1$ El más cercano a 1 es el mejor

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de Fiabilidad (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Tolerancia a fallos	Prevención de falla	Contar el número de ocurrencia de fallas serias y críticas evitadas contra los casos de pruebas de fallas iniciales y el número de casos de pruebas de fallas iniciales ejecutados durante las pruebas.	$X = A/B$ A = Número de ocurrencia de fallas evitadas contra los casos de pruebas de fallas iniciales. B = Número de casos de pruebas de fallas iniciales ejecutados durante las pruebas	$X = A/B$ A = Número de ocurrencia de fallas evitadas contra los casos de pruebas de fallas iniciales. B = Número de casos de pruebas de fallas iniciales ejecutados durante las pruebas.	$0 \leq X \leq 1$ Cuanto más se acerque a 1 es lo mejor
Tolerancia a fallos	Redundancia	¿Cuántos tipos de componentes/ sistemas del son instalados de forma redundante para evitar un fallo en el sistema?	Contar el número total de tipos de componentes y el número de tipos de componentes instalados de forma redundante	$X = A / B$ A= Número componentes/sistemas instalados de forma redundante B = Número total de componentes	$0 \leq X \leq 1$ Cuanto más se acerque a 1 es lo mejor
Tolerancia a fallos	Anulación de operación incorrecta	¿Cuántas funciones son implementada con capacidad de	Contar el número de Funciones implementadas que evitan fallas críticas y serias causadas por	$X = A/B$ A = Número de operaciones incorrectas Presentadas	$0 \leq X \leq 1$ Cuanto más se acerque a 0 es lo

		Anular operaciones incorrectas?	operaciones incorrectas y contar el número operaciones incorrectas presentadas	B = Número total de funciones implementadas para anular operaciones incorrectas	mejor
--	--	---------------------------------	--	---	-------

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de Mantenibilidad (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuARE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor Deseado
Modificabilidad	Complejidad ciclomática	¿Cuál es la complejidad estructural de un código fuente?	Contar las instrucciones condicionales, bucles, salidas de métodos y cláusulas AND y OR dentro de los condicionales	$X = A + 1$ A = Número de instrucciones condicionales que tiene una función	$1 \leq X \leq 15$ El más cercano a 1, es el mejor
Modificabilidad	Profundidad de herencia	¿Qué tan profunda es la jerarquía de la herencia de las clases involucradas en una determinada función?	Contar las jerarquías empleadas en una determinada función o método	$X = A$ A = Número de jerarquías empleadas para una determinada función	$0 \leq X \leq 4$ El más cercano a 0 es el mejor
Modificabilidad	Complejidad De modificación	¿Con qué facilidad el desarrollador puede modificar el software para resolver problemas?	Tomar el tiempo de trabajo que le toma al desarrollador modificar y contar el número de modificaciones	A = Número de modificaciones B = Tiempo de trabajo que le toma al desarrollador modificar Dónde: $T > 0$	$X = A/T$ El más lejano a 0/t es el mejor
Modificabilidad	Índice de éxito de modificación	¿Hasta qué punto puede el sistema ser operado sin fallas después del mantenimiento ?	Contar el número de Problemas dentro de un determinado período antes de mantenimiento y contar el número de problemas en el mismo periodo después del matenimiento.	$X = A/B$ A = Número de problemas dentro de un determinado período antes de mantenimiento B = Número de problemas en el	$0 \leq X \leq 1$ El más cercano a 0, es el mejor
Modularidad	Capacidad de condensación	¿Qué tan fuerte es la relación entre los componentes del sistema?	Contar el número de componentes que no son afectados por cambios de otros componentes y el número total de componentes específicos	$X = A / B$ A = Número de componentes que no son afectados por cambios de otros componentes B = Número total de componentes específicos	$0 \leq X \leq 1$ El más cercano a 0, es el mejor
Modularidad	Acoplamiento de clases	¿Qué tan fuerte es la relación entre una	Contar el número de relaciones que tiene una función con	$X = A$ A = Número de relaciones que tiene	$1 \leq X \leq 4$ El más cercano a 1,

		función del sistema con otras clases implementadas?	respecto a otras clases	una función con respecto a otras clases	es el mejor
--	--	---	-------------------------	---	-------------

Atributos de Calidad de Software más relevantes para la Sostenibilidad de Software para la característica de Portabilidad (ISO/IEC 25000 System and Software Quality Requirements and Evaluation [SQuaRE], 2014).

SubCaracterística	Métricas	Propósito	Método de Aplicación	Fórmula	Valor deseado
Adaptabilidad	Adaptabilidad en entorno hardware	¿Es el sistema lo suficientemente capaz de adaptarse al entorno hardware?	Contar el número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el entorno hardware y contar el número total de funciones las cuales han sido probadas.	$X = A/B$ A = Número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el entorno hardware. B = Número total de funciones que han sido probadas.	$0 \leq X \leq 1$ El más cercano a 0, es el mejor
Adaptabilidad	Adaptabilidad en entorno de software	¿Es el sistema lo suficientemente capaz de adaptarse al entorno del sistema software?	Contar el número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el sistema y contar el número total de funciones las cuales han sido probadas	$X = A/B$ A = Número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con el sistema B = Número total de funciones que han sido probadas	$0 \leq X \leq 1$ El más cercano a 0, es el mejor
Adaptabilidad	Adaptabilidad en entorno empresarial	¿Es el sistema lo suficientemente capaz de adaptarse al entorno operacional?	Contar el número funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con usuarios del entorno empresarial y contar el número total de funciones las cuales han sido probadas	$X = A/B$ A = Número de funciones operativas de las tareas que no se hayan completado durante las pruebas operativas con usuarios del entorno empresarial B = Número total de funciones que han sido probadas	$0 \leq X \leq 1$ El más cercano a 0, es el mejor

ANEXO IV. HERRAMIENTA DE EVALUACIÓN.

Anexo IV - Herramienta de Evaluación.xlsx