

Metodología para elaborar el modelo heurístico empírico en la formación online del ingeniero

Methodology to elaborate the empirical heuristic model in the online training of the engineer

Arasay Padrón Álvarez^{1*} <https://orcid.org/0000-0002-2848-7776>

Percy Huertas Niquén² <https://orcid.org/0000-0002-5970-055X>

Liuris Rodríguez Castilla³ <https://orcid.org/0000-0002-9788-0686>

¹ Universidad Internacional de la Rioja, (UNIR), España.

² Universidad Nacional de San Agustín (UNSA), Arequipa, Perú.

³ Centro de Referencia para la Educación de Avanzada, CREA, Universidad Tecnológica de La Habana «José Antonio Echeverría», CUJAE, Cuba.

* Autor para correspondencia: arasaybia@gmail.com

Resumen

El catálogo de requisitos es una tarea en el desarrollo de software que requiere de abundante documentación y se traduce en pérdida de tiempo para los desarrolladores. Muchas veces es sustituida por un simple listado de necesidades del usuario, donde insertan inconsistencias y ambigüedades y provoca el fracaso de los proyectos. Se hace necesario investigar una nueva estructuración de los requisitos que resuelva esta problemática, y formalicen un catálogo de requisitos donde la información quede ordenada eficientemente. Elaborar un modelo heurístico empírico por medio del uso de un conjunto de plantillas diseñadas y basadas en casos de uso, para estructurar la información referida a los requisitos de software, constituye el objetivo de esta investigación. Para tal fin se investigan 40 proyectos de desarrollo de software aplicándoles el modelo propuesto y la metodología elaborada, obteniendo el incremento de la calidad de los requisitos en un 47 % y una reducción de los tiempos de construcción en un 66 %.

Palabras clave: metodología, modelo, formación online, ingeniería, informática.

Abstract

The requirements catalog is a task in software development that requires abundant documentation and results in loss of time for developers. Many times it is replaced by a simple list of user needs, where inconsistencies and ambiguities are inserted and cause the failure of projects. It is necessary to investigate a new structuring of the requirements that resolves this problem, and formalizes a catalog of requirements where the information is efficiently organized. Developing an empirical heuristic model through the use of a set of templates designed and based on use cases, to structure the information referring to software requirements, constitutes the objective of this research. For this purpose, 40 software development projects are investigated by applying the proposed model and the developed methodology, obtaining an increase in the quality of the requirements by 47 % and a reduction in construction times by 66 %.

Keywords: methodology, model, online training, engineering, computing.

Recibido: 24/9/2023

Aprobado: 6/11/2023

INTRODUCCIÓN

Las investigaciones del Standish Group International (2014) aluden que la falta de atención a los requisitos genera brechas enormes de insatisfacción del usuario al momento de ingresar el producto en producción. El fracaso de los proyectos de software se debe, entre otros factores, a la falta de visión del desarrollador en los requisitos del producto y al poco uso de técnicas, modelos y metodologías adecuadas para el normal enfrentamiento de los requisitos según Boehm y Papaccio (1988), quienes, además, sustentan en sus investigaciones que corregir los errores en la primera etapa de construcción del software genera pérdidas.

La presente investigación es de carácter descriptivo y se encuentra enfocando en el área de Ciencias de la Ingeniería y línea de investigación de Sistemas, Computación e Informática. En ella se propone un conjunto de heurísticas basadas en el empirismo para construir un modelo que permita elaborar catálogos de requisitos adecuados cuando se construye software a medida. Para lograr el objetivo se emplea el método cuantitativo y la generación de estructuras o plantillas basadas en casos de uso sustentando la trazabilidad mediante casos de estudio.

A partir de los años 90, la industria obligó a tomar importancia a la ingeniería de requerimientos comenzando con la aparición de técnicas, métodos y metodologías de trabajo. Además, se resalta la importancia de saber formular los requerimientos, ya que una mala apreciación de ellos, puede conducir a una mala formulación del producto (Arias, 2007). En esta situación problemática también se alude al fracaso de los proyectos de desarrollo de software causado esencialmente, por la falta de visión en la conformación del equipo de desarrollo y específicamente en el trabajo en equipo (Rivas et al., 2009).

En momentos más actuales se desarrollan estudios sobre el estado actual de la sostenibilidad de la práctica de la ingeniería de requerimientos encontrando que tanto a nivel individual, profesional y de práctica profesional existe una falta de conocimiento, apoyo y responsabilidad por el trato de los requerimientos, sugiriendo que la solución se encuentra en el proceso de enseñanza-aprendizaje universitario (Chitchyan et al., 2016). En este orden de ideas se hace referencia a que todo proyecto de software debe tratar con requerimientos y que no deben ser difusos, ya que conducen a una mala concepción del producto de software y esto afecta a la calidad del producto final siendo esta una de las principales etapas para su construcción. También se hace mención a la deficiente información que posee el catálogo de requisitos y se indica que el 56 % de los defectos encontrados para las pruebas de software tienen su base en los requisitos de software (Hussain et al., 2016; Rodríguez, 2017).

Con respecto a la evaluación de las metodologías orientadas a la ingeniería de requerimientos, Teja et al. (2017) proponen una forma de evaluación donde admiten que es un proceso muy importante en la construcción de software, pero que elegir una de ellas es una tarea compleja que depende de las características del problema a resolver por sus consideraciones de seguridad. Otros autores como Pollo (2017) desarrolla una metodología

con la finalidad de gestionar los problemas de la ingeniería de requerimientos cuando se lleva a cabo una explotación de información y plantea el modelo basado en procesos definiendo fases, actividades y tareas con la finalidad de llevar de la mano al analista de sistemas. Mientras que se insiste en la necesidad de métodos con basamentos científicos para evitar las deficiencias referidas (Anu et al., 2018).

Más recientemente se proponen modelos que incluyen tres etapas que definen la metodología para elaborar un catálogo de requisitos, sugieren las etapas de educación, elicitación y especificación de requerimientos de software y los validan experimentalmente con estudiantes que llevan el curso de Construcción de Software (Cornejo et al., 2019). En este orden se defiende el método de priorización de requerimientos y un algoritmo basado en lógica difusa con la finalidad de obtener requerimientos que contengan la mínima cantidad de errores (Luján y Casanova, 2019).

Hotomski y Glinz (2019) hacen énfasis en las diferencias en los requisitos con respecto a lo que desea el cliente y el entendimiento del analista, debilidades fatales para la construcción de software; e indican que se adolece de relacionadas sobre cómo actuar con respecto a la documentación afectada. López y Mejías (2020) refieren las principales actividades de la ingeniería de requerimientos sobre IPO, SCRUM, Design Thinking y Kanban; y mencionan que todos ellos valoran esta etapa de la construcción de software; pero que a través de las metodologías tradicionales el proceso resulta engorroso. Con respecto a personas discapacitadas, se enfatiza en que la Ingeniería de Requerimientos no se encuentra en la capacidad de asumir este reto y la comunicación resulta un problema más contundente (Silva et al., 2020).

El estudio sobre la influencia de las culturas en las actividades de la Ingeniería de Requerimientos, se concluye que existen algunas que influyen muy poco en las actividades de elicitación y validación de requisitos, aunque otras si intervienen con un alto grado, en las actividades de responsabilidad y honestidad. También se corrobora que las emociones, la diversidad cultural y la distribución geográfica influyen moderadamente en la actividades de la ingeniería de requerimientos (Alsanoosy y Harland, 2020; Hidellaarachchi et al., 2022).

Carballo y Barrientos (2020) refieren que el fracaso de los proyectos de desarrollo de software se debe esencialmente a los cambios que se insertan en los requerimientos de

software. Otro factor importante es la falta de visión o entendimiento de los requisitos de software y su propia gestión como lo son los cambios en los requisitos de software. Alhazmi y Huang (2020) tratan sobre la integración de la ingeniería de requerimientos en SCRUM mediante el modelo del pensamiento de diseño. Llevan a cabo un análisis por los cinco pasos del pensamiento de diseño, concluyendo que el pensamiento de diseño cobra severo impacto sobre SCRUM con dos actividades que no pertenecen a la ingeniería de requerimientos.

Las investigaciones mostradas aluden a la importancia de la Ingeniería de Requerimientos refiriendo sus defectos en la construcción del producto de software, pero no suministran soluciones para su tratamiento y efectividad. Existen intentos por estructurar el catálogo de requisitos sin lograr una adecuada trazabilidad con los artefactos asociados que permitan entender el dominio del problema y el conocimiento del modelo del negocio con la finalidad de lograr un adecuado mantenimiento del producto y una alta satisfacción de los clientes y de los usuarios finales.

Esta situación problemática exige la atención científica del problema y en esa dirección se desarrolla la presente investigación que se propone como objetivo elaborar el modelo heurístico empírico por medio del uso de un conjunto de plantillas diseñadas y basadas en casos de uso, para estructurar la información referida a los requisitos de software.

MATERIALES Y MÉTODOS

Lugar de estudio

La investigación se realiza en la Escuela Profesional de Ingeniería de Sistemas a través del curso de Ingeniería de Requerimientos y se implementa y perfecciona la propuesta durante 6 años consecutivos. Para lograr el objetivo, y como metodología de investigación, los proyectos fueron trabajados por dos equipos de desarrollo; el primero se dedica a resolver el problema de manera tradicional y el segundo aplica el modelo propuesto. A partir de la información obtenida se lleva a cabo una analítica descriptiva y predictiva para conocer causalidades del caso. Esta consiste en proponer heurísticas que conllevan a estructurar la obtención de información de los proyectos para diseñar el catálogo de requisitos, corregir las inconsistencias y ambigüedades.

Los métodos y técnicas de investigación para la recolección de datos son el estudio de casos y la observación participante y no participante. Se lleva a cabo una analítica descriptiva que culmina con el uso de la prueba estadística F de Snedecor y de correlación. Para la analítica predictiva se realiza el análisis a partir del modelo del árbol de decisión. Específicamente se emplean métodos para obtener patrones de trabajo en las etapas de educación, ilación y especificación de requisitos con la finalidad de notar el comportamiento de los requisitos en estas etapas.

En promedio, cada equipo de desarrollo de software queda conformado por 8 estudiantes dedicados al desarrollo de software, a quienes se les entrena previamente, por un periodo de 5 días, sobre la forma de cómo aplicar el modelo propuesto mientras que el otro equipo se dedicaba a elaborar el producto final, bajo un modelo tradicional o bajo el esquema definido por el propio líder del proyecto. Finalmente, se evalúan los resultados del modelo en función de la información obtenida en el estudio de casos con la finalidad de determinar la validez del mismo. La selección de los estudiantes que componen los equipos de trabajo se hace bajo el mismo perfil.

Estructura del catálogo de requisitos, modelo para la elaboración de catálogos de requisitos en informática y ciclo de vida del modelo

El catálogo de requisitos consta de ocho secciones que se deben integrar de manera adecuada con la finalidad de organizar la información pertinente para construir un buen producto de software. Estas secciones son: organización, actores, fuentes, autores, educación de requisitos, ilación de requisitos, especificación de requisitos, y requisitos no funcionales. El modelo toma como punto de entrada las necesidades del cliente y entrega como salida el catálogo de requisitos que está conformado por los artefactos de educación, ilación, especificación (Figura 1).

El modelo contiene metodologías independientes en cada una de las etapas con la finalidad de otorgar claridad al proceso de desarrollo en general, el ciclo de vida se muestra en la Figura 2. Las etapas y metodologías a seguir según el modelo propuesto se explican a continuación de las representaciones gráficas que permiten la integración de todo el proceso.

Diseño de artefactos

Se definen artefactos versionados para organizaciones, autores, actores, fuentes, expertos, educación de requisitos, ilación de requisitos, especificación de requisitos. El artefacto de educación de requisitos permite que se obtengan los módulos que conforman el software a construir. La ilación de requisitos permite que el analista logre entender el modelo de negocios a automatizar y la especificación de requisitos que viene a ser la primera concepción de la codificación del producto.

Etapas de educación de requisitos

La relación con el cliente o usuario debe ser cordial con la finalidad de que el analista logre entender y comprender sus necesidades, pero traduciéndolas en términos que entienda el propio analista manteniendo los criterios del cliente. Esta actividad, considerada una de las más complejas, debe ser desarrollada profesionalidad y documentar hechos o sucesos que surgen durante el trayecto.

Etapas de ilación de requisitos

Una vez que el ingeniero ha deducido los módulos que conforman el producto a construir pasa a comprender cómo entiende esta lógica que le proporciona el modelo del negocio. Esta etapa es iterativa ya que si no logra comprender algún componente de esta etapa debe recurrir a la etapa de educación para determinar lo que se necesita, lo que también implica reuniones con el cliente hasta entender el modelo del negocio entregado por el cliente.

Etapas de especificación de requisitos

La tarea de especificar los requisitos comienza con la etapa de educación, continúa con el refinamiento sucesivo del conocimiento del modelo del negocio producto de las sucesivas entrevistas con el cliente y los *stakeholders* en la etapa de ilación, y se culmina con una especificación de la información orientada al codificador. En esta especificación se definen la metodología y los datos del aplicativo a construir; además de lograr el esquema de la base de datos, así como las estructuras de datos a trabajar.

Resolución de conflictos

El proceso de corrección implica un análisis de las tablas de educación, ilación y especificación. Cuando en las tablas de educación se encuentran estos defectos se procede a corregirlos versionando las mismas hasta solventar los errores. Los contenidos de las tablas de educación comúnmente se encuentran relacionados con los módulos a construir y deben refinarse hasta quedar definidos los límites del sistema.

Las tablas de ilación deben ser analizadas bajo dos ángulos: la primera de manera interna y la segunda buscado su asociación con las tablas de educación. En la forma interna se buscan los defectos y estos son corregidos versionando las correspondientes tablas; en la segunda se estipula que si se encuentra el error interno esto debe ser corroborado con la tabla de educación asociada, si se encuentra un error esta tabla de educación también debe ser corregida y versionada.

Las tablas de especificación siguen la misma metodología. Primero se analizan y corrigen los errores de manera interna versionando estas; segundo se corrigen los errores asociados con las tablas de ilación y si estas presentan correcciones se busca corregir también las tablas de educación. Siempre se emplea la versión que permite guardar el historial de los cambios propuestos.

Implementación del dataset

Para obtener resultados que den soporte al modelo, se tomaron en cuenta 40 proyectos de desarrollo de software, durante 12 semestres académicos, en los que fueron distribuidos los estudiantes del curso de Ingeniería de Requerimientos. En los 40 proyectos de desarrollo de software se trabajó por el modelo propuesto y se recolectaron 4,787 requisitos de software; todos los proyectos conforman el caso de estudio en mención. Cada proyecto fue desarrollado en el periodo de cuatro meses, 8 horas diarias de trabajo en 5 días a la semana. Para una mayor precisión, los tiempos fueron convertidos en minutos.

RESULTADOS Y DISCUSIÓN

El tiempo promedio en obtener un requisito es de 13 minutos y su costo 13 soles, lo que implica que por cada minuto que se consume en obtener un requisito su costo es un sol (S/

1.00). El promedio para obtener un requisito en la etapa de especificación es de 12 minutos con un costo promedio por requisito de 10 soles. Los analistas se demoran, en promedio, 2 minutos para corregir inconsistencias en la etapa de educación con un costo promedio de 7 soles por requisito. En cambio, para corregir ambigüedades se tardan en promedio 2,6 minutos con un costo de 8,9 soles.

Con respecto a las inconsistencias, el promedio para corregir inconsistencias es de 2,6 minutos con un costo promedio de 9 soles; en cambio para corregir ambigüedades en la etapa de ilación emplean, en promedio, 0.9 minutos con un costo promedio de 3 soles. Es importante que cuando se corrige una inconsistencia de manera indirecta, se encuentra la corrección de subsecuente ambigüedad; es decir que las correcciones se van realizando en cascada, de tal forma que estas se cumplen a lo largo de la línea de trazabilidad.

La corrección de inconsistencias en la etapa de especificación tiene un tiempo promedio de 1,1 minutos a un costo de 4 soles por requisito. Para corregir las ambigüedades se toma un tiempo promedio de 0,47 minutos con un costo promedio de 1,6 soles. Por otro lado, las entrevistas tienen una duración promedio de 33,5 minutos, lo que implica que la corrección de errores en esta etapa es afectada en la línea sucesiva con respecto a la educación e ilación; esto significa que, si se detecta un error en la etapa de especificación, debe ser corregido también en la etapa de ilación y luego pasar a la etapa de educación de requisitos de software.

El tiempo de ejecución del proceso original es de 122 minutos, es decir 2 horas y 2 minutos. En el caso de los nuevos procesos implementados, estos consumen en promedio 39 minutos. Asimismo, existen 3 versiones en promedio en la educación, 3 versiones en la ilación y 2 versiones en la especificación. La cantidad de versiones implican y afectan la calidad del requisito. Cuanto mayor cantidad de versiones se tienen entonces significa que el requisito no ha sido comprendido o entendido en su totalidad o que su redacción no ha sido la adecuada; el caso contrario implica que el modelo de negocio asociado al requisito de software ha sido comprendido en su totalidad.

Por otro lado, el 30 % de los estudiantes practican el arte o el deporte mientras que el 70 % se dedican al desarrollo de software. El hecho de practicar el arte permite tener una mejor forma de definir el diseño de los problemas. Con respecto a los atributos de calidad de los requisitos se encontró que, en promedio, el 51 % de ellos cumplían con los atributos de

calidad mientras que el 49 % de ellos no lo cumplían cuando el modelo propuesto era aplicado (Tabla 1).

Tabla 1. Atributos de calidad de los requisitos

Atributos de calidad de los requisitos	Frecuencia en %	
	No	Sí
Correcto	58	42
Modificable	49	51
Verificable	46	54
Priorizable	47	53
Comprensible	47	53
Coherente	46	54
Promedio	49	51

Fuente: elaboración propia.

Al inicio se encontró que solo el 6,6 % de los requisitos eran correctos mejorando sustancialmente el 35,5 %. El 10,7 % eran modificables y al final se alcanzó un 40 % de su mejoramiento; en los proyectos iniciales se encontró una pobre priorización de los requisitos alcanzando posteriormente el 48,8 %. El problema de redacción de los requisitos era notorio ya que solo se logró comprender el 1,3 % de ellos; después de la aplicación del modelo el 51,7 % de los mismos presentaban una mejor redacción y como consecuencia un mejor entendimiento. Otro importante atributo de calidad es la coherencia, encontrando que sólo el 1,2 % de ellos la tenían en los proyectos originales; después de la aplicación del modelo se encontró que el 52,9 % de ellos presentaban una mejor coherencia.

En la generalidad de los casos existe un decremento en los tiempos de construcción de más del 50 % y en el promedio general de todos los proyectos una reducción del tiempo del 66 %; es decir que los proyectos de desarrollo de software elaborados de manera normal consumen más tiempo que aquellos donde se aplica el modelo en investigación. Esto implica que el modelo propuesto logra una reducción de tiempos en la construcción de los proyectos de desarrollo de software y que el entendimiento del modelo de negocio se logra de manera coherente a partir de la etapa de educación de requisitos quedando plasmado en la etapa de ilación de requisitos y logrando alcanzar su primera etapa de codificación en la etapa de especificación de requisitos de software.

CONCLUSIONES

El modelo empírico propuesto y la metodología seguida estructura, transparentemente, el dominio de la información de las necesidades del usuario de tal manera que maneja la complejidad de está logrando un concepto integrado que permite el manejo de los requisitos de software. Los artefactos confeccionados capturan y manejan toda la complejidad de la información sobre las necesidades del cliente estructurándolas de manera que sean simples de interpretar y de hallar inconsistencias y ambigüedades.

El análisis del caso de estudio demuestra que se consume una mayor cantidad de tiempo en la etapa de educación, 28 minutos en promedio y con un costo promedio de 16,2 soles por requisito, con respecto a las otras etapas. Asimismo, la calidad de los requisitos de software se incrementó en un 47 % en promedio.

REFERENCIAS BIBLIOGRÁFICAS

- Alhazmi, A.; y Huang, S. (2020). Integrating Design Thinking into Scrum Framework in the Context of Requirements Engineering Management (ponencia). *CSSE 2020*, Beijing.
- Alsanoosy, T. M. S.; y Harland, J. (2020). Identification of Cultural Influences on Requirements Engineering Activities. Paper presented at the *42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*.
- Anu, V.; Hu, W.; Carver, J. (2018). Development of a Human Error Taxonomy for Software Requirements: A Systematic Literature Review. *Information and Software Technology* (103), 112-124. Recuperado de <https://www.sciencedirect.com/science/article/pii/S0950584916302373>
- Arias, M. (2007). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *Revista InterSedes*, 6 (10), 1-13. Recuperado de <https://www.redalyc.org/pdf/666/66612870011.pdf>
- Boehm, B.; y Papaccio, P. (1988). Understanding and Controlling Software Cost. *IEEE Transactions on Software Engineering*, 14 (10), 1462–1477. Recuperado de <https://ieeexplore.ieee.org/abstract/document/6191/>

- Carballo, L.; y Barrientos, I. (2020). Las causas del cambio en los requerimientos de software. *Revista Cubana de Ciencias Informáticas*, 14 (2), 131-144. Recuperado de https://scielo.sld.cu/scielo.php?pid=S2227-18992020000200131&script=sci_arttext
- Cornejo, V.; Flores, S.; Bedregal, N.; y Tupacyupanqui, D. (2019). Modelo Procedimental para la Especificación de Requisitos Funcionales en la Construcción de Software. *RISTI: Revista Ibérica de Sistemas e Tecnologias de Informação Iberian Journal of Information Systems and Technologies* (E26), 571-586. Recuperado de <https://www.proquest.com/openview/5615fedd1bf087239cec9598033911e5/1?pq-origsite=gscholar&cbl=1006393>
- Chitchyan, R.; Becker, C.; Betz, S.; Duboc, L.; Penzenstadler, B.; Seyff, N.; y Venters, C. (2016). Sustainability Design in Requirements Engineering: State of Practice (ponencia). *38th IEEE International Conference on Software Engineering Companion*, USA.
- Hidellaarachchi, D.; Grundy, J.; Hoda, R.; y Mueller, I. (2022). The Influence of Human Aspects on Requirements Engineering-related Activities: Software Practitioners' Perspective. *ACM Trans. Softw. Eng. Methodol*, 1 (1), 1-27.
- Hotomski, S.; y Glinz, M. (2019). GuideGen: An approach for keeping requirements and acceptance tests aligned via automatically generated guidance. *Information and Software Technology* (110), 17-38. Recuperado de <https://www.sciencedirect.com/science/article/pii/S0950584918301599>
- Hussain, A.; Mkpojiogu, E.; y Muhamad, F. (2016, 11-13 April). The Role of Requirements in the Success or Failure of Software Projects (ponencia). *International Soft Science Conference (ISSC 2016)*, Malaysia.
- López, C.; y Mejías, J. (2020). Guía para la Ingeniería de Requerimientos bajo un enfoque ágil integrando técnicas de usabilidad. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação* (E32), 546-558. Recuperado de <https://www.proquest.com/openview/8d965b8c754de2de90f7a678eb5cb450/1?pq-origsite=gscholar&cbl=1006393>
- Luján, M.; y Casanova, C. (2019). *Priorización en Ingeniería de Requerimientos con Preferencias Difusas*. Universidad Tecnológica Nacional, Facultad Regional

- Concepción del Uruguay, Grupo de Investigación sobre Inteligencia Computacional e Ingeniería de Software, Departamento Ingeniería en Sistemas de Información.
- Pollo, M. (2017). *Modelo de proceso para elicitación de requerimientos en proyectos de explotación de información* (tesis doctoral). Universidad de la Plata, Argentina. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/96315>
- Rivas, C.; López, J.; y Barato, J. (2009). ¿Por qué fracasan los proyectos de software? (ponencia). *VII Congreso del Comité de Calidad en los Sistemas, Tecnologías de la Información y Comunicaciones*, España.
- Rodríguez, C. (2017). Impacto de los requerimientos en la calidad de software. *Tecnología, Investigación y Academia*, 5 (2), 161-173. Recuperado de <https://revistas.udistrital.edu.co/index.php/tia/article/view/7607>
- Silva, A.; Rodríguez, C.; Cheveiro, N.; García, R.; Duarte, S.; Araujo, M.; Silva, C. (2020, marzo). Lessons Learned about Oral–auditory and Visual–spatial Communication in Requirements Engineering with Deaf Stakeholders (ponencia). *35th Annual ACM Symposium on Applied Computing*, Czech Republic.
- Standish Group International (2014). *CHAOS Chronicle (2014) Report*. The Standish Group International.
- Teja, S.; Laborde, R.; Samer, A.; Barrére, F.; y Benzekri, A. (2017). Which Security Requirements Engineering Methodology Should I Choose? Towards a Requirements Engineering-based Evaluation Approach (ponencia). *12th International Conference on Availability, Reliability and Security*, Italy.

Conflictos de intereses

Los autores declaramos que no existen conflictos de intereses y están de acuerdo con la versión final del texto.

Contribución de los autores

Arasay Padrón Alvarez: participó en la concepción de la idea, la elaboración del modelo heurístico, elaboración del texto, aplicación de la metodología, recolección de datos, procesamiento, análisis de la información.

Percy Huertas Niquén: participó en la elaboración del texto, aplicación de la metodología, recolección de datos, procesamiento, análisis de la información.

Liuris Rodríguez Castilla: participó en la elaboración del texto, evaluación de la revista, gestión bibliográfica, adaptación de normas de redacción y gestión de la publicación en línea.