



Revista Cubana de Ciencias Informáticas

ISSN: 2227-1899

Editorial Ediciones Futuro

Piñero González, Maidelyn; Marin Díaz, Aymara;
Trujillo Casañola, Yaimí; Buedo Hidalgo, Denys
Buenas prácticas para prevenir los riesgos de la eficiencia del desempeño en los productos de software
Revista Cubana de Ciencias Informáticas, vol. 15, núm. 1, 2021, pp. 89-113
Editorial Ediciones Futuro

Disponible en: <https://www.redalyc.org/articulo.oa?id=378366538006>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org



Sistema de Información Científica Redalyc
Red de Revistas Científicas de América Latina y el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso
abierto

Tipo de artículo : Artículo original
Temática : Ingeniería y gestión de software.
Recibido : 22/12/2020 | Aceptado : 15/01/2021

Buenas prácticas para prevenir los riesgos de la eficiencia del desempeño en los productos de software

Good practices to prevent risks of performance efficiency in software products

Maidelyn Piñero González¹ <https://orcid.org/0000-0002-6534-132X>

Aymara Marin Diaz² <https://orcid.org/0000-0001-5101-7804>

Yaimí Trujillo Casañola³ <https://orcid.org/0000-0002-3138-011x>

Denys Buedo Hidalgo⁴ <https://orcid.org/0000-0002-5031-2040>

¹ Universidad de las Ciencias Informáticas, Cuba. Carretera San Antonio Km 2 ½. {[mpinero](mailto:mpinero@uci.cu), [amarin](mailto:amarin@uci.cu), yaimi.dbuedo@uci.cu}@uci.cu

*Autor para la correspondencia. (mpinero@uci.cu)

RESUMEN

El comportamiento de la eficiencia del desempeño como característica de calidad en las aplicaciones, ha ganado valor a nivel nacional e internacional para la aceptación del software. Siendo un factor importante a tener en cuenta desde inicio del desarrollo del sistema, pues el mal desempeño de esta característica afecta la satisfacción del usuario. En el presente artículo se analizan normas, estándares y criterios de autores reconocidos a nivel internacional con el objetivo identificar características de la eficiencia del desempeño,

actividades de calidad, formas de medirla y los riesgos más comunes asociados a esta. A partir de ello, se proponen buenas prácticas a incorporar desde inicios del ciclo de vida del software que permitan prevenir los riesgos asociados a esta característica y aumentar la calidad del producto. Para valorar la contribución de la propuesta dada, se utiliza el método Delphi y criterios de expertos, la cual fue exitosamente aceptada y aprobada por mismos.

Palabras clave: eficiencia del desempeño; buenas prácticas; riegos; satisfacción; calidad.

ABSTRACT

The behavior of performance efficiency as a quality characteristic in applications has gained value at a national and international level for the acceptance of software. Being an important factor to take into account from the beginning of the development of the system, since the poor performance of this characteristic affects user satisfaction. In this article, norms, standards and criteria of internationally recognized authors are analyzed in order to identify characteristics of performance efficiency, quality activities, ways to measure it and the most common risks associated with it. Based on this, good practices are proposed to be incorporated from the beginning of the software life cycle to prevent the risks associated with this characteristic and increase the quality of the product. To assess the contribution of the given proposal, the Delphi method and expert criteria are used, which was successfully accepted and approved by them.

Keywords: performance efficiency; good practices; risks; satisfaction.

Introducción

La administración total de la calidad alimenta la cultura de mejora continua, desarrollando enfoques cada vez más eficaces en la Ingeniería de Software, que propicia costos más bajos, ingresos más altos, clientes y

empleados satisfechos y su fundamento se apoya en el compromiso con la calidad (Godfrey, 1998; Pressman, 2010; Socarras Ramírez, Trujillo Casañola, & Vega Prieto, 2018). El término calidad de software se refiere al grado de desempeño de las principales características con las que debe cumplir un sistema computacional durante su ciclo de vida, dichas características de cierta manera garantizan que el cliente cuente con un sistema confiable, lo cual aumenta su satisfacción frente a la funcionalidad y eficiencia del sistema construido (Estayno, Dapozo, Liliana, & Greiner, 2009; Mauro Callejas Cuervo, Andrea Catherine Alarcón Aldana, & Carreño., 2017; Pressman, 2010).

En la actualidad los sistemas son cada vez más complejos, por eso se hace muy difícil coordinar que todos los implicados estén disponibles y estables, puesto que cada uno de los subsistemas que contiene es gestionado. La calidad de la eficiencia en el desempeño se ve notablemente reducida, incurriendo en la seguridad y el aumento de fallos que puedan ocurrir. El desconocimiento del comportamiento del software en una situación de estrés, al límite del uso de recursos y el tiempo de respuesta afectado, propicia descontento e insatisfacción para las partes interesadas (Globe, 2017). La calidad subjetiva de un sistema de software se basa principalmente en sus características no funcionales. Dentro de los atributos de calidad se incluye la eficiencia del desempeño y es considerada como significativa para el software, ya que los usuarios rechazarán un software que sea demasiado lento (Sommerville, 2011).

La eficiencia del sistema puede ser una competencia en el mercado de software, pues obtener una respuesta correcta en el menor tiempo posible provee mayor calidad en un producto y satisfacción en los usuarios. La eficiencia del desempeño de la aplicación ahora es esencial para el éxito empresarial. Esta característica está relacionada con el rendimiento de un dispositivo en función de su comportamiento temporal, el uso de recursos y la capacidad o límites máximos de funcionamiento. Para medir la eficiencia en el desempeño hay que conocer los tiempos de respuesta y procesamiento, las tasas de rendimiento, los límites máximos de un producto o parámetro del sistema, las cantidades y los tipos de recursos utilizados, todo ello debe cumplir con los requisitos al realizar sus funciones (25010:2011, 2011; Castillo, Mora, Elizaldes, & Orozco, 2017; Fierro, Manosalvas, Rodríguez, & Landeta, 2019; Paz, Gómez, & Rosas, 2017).

En el reporte realizado por el Comité Internacional de Cualificación de Pruebas de Software (International Software Testing Qualifications Board por sus siglas en inglés, ISTQB), el cual está formado por más de 50 asociaciones nacionales de Control de Calidad del software y que estudia y da seguimiento al avance de las

pruebas de software en más de 120 países, durante el 2013 al 2014 identificó que las pruebas del futuro eran las pruebas de eficiencia del desempeño, de usabilidad y pruebas para dispositivos móviles (ISTQB, 2014). Para el 2017 – 2018 arrojó que las tendencias para la profesión de pruebas de software serán la automatización de pruebas, las pruebas ágiles y las pruebas de seguridad; y que, de las pruebas más importantes para la organización, se tiene las pruebas funcionales con un 83% y las pruebas de eficiencia del desempeño con un 60.7% (International Software Testing Qualifications Board ISTQB®, 2018; Marin Diaz, Trujillo Casañola, & Buedo Hidalgo, 2020).

Por otra parte, MICRO FOCUS¹ junto Sogeti² y Capgemini³, realizan reportes sobre la calidad en el desarrollo de software y los principales desafíos en esta área a través del informe anual World Quality Report. Estos reportes cubren las América del Norte, Reino Unido, Francia, Alemania, Europa del Sur (Italia, España y Portugal) y Europa del Este (Polonia, Hungría y República Czen), China, Australia, Nueva Zelanda, el Medio Oriente y la Región Nórdica. En la edición de período del 2019-2020 destacó que las organizaciones consideran reasignar responsabilidades para la calidad técnica en los aspectos de pruebas de eficiencia del desempeño y seguridad. Teniendo como base que las pruebas de eficiencia del desempeño son las más usadas con un 63%, seguido las pruebas de seguridad con un 61% y los servicios en la nube con un 58%. Dentro de las habilidades que necesitan cambios sobresale la ingeniería de eficiencia del desempeño con un 48%, enfocado a la demanda de calidad en la rapidez y dándole la responsabilidad de garantizar la satisfacción del usuario final, siendo este el objetivo principal dentro de la estrategia de garantía de calidad y pruebas de software. (Micro Focus, 2020)

A partir de lo señalado anteriormente, se evidencia que el comportamiento en el uso de las pruebas de software para conocer el comportamiento de la eficiencia del desempeño en las aplicaciones, ha ganado valor a nivel nacional e internacional para la aceptación del software. Siendo una característica esencial a tener en cuenta en el ciclo de vida del software (Micro Focus, 2020; Sommerville, 2011). La buena eficiencia del desempeño de las aplicaciones es esencial para el éxito y aprobación de las partes interesadas. La demanda del cumplimiento de este requisito en organizaciones, instituciones y empresas de desarrollo de software, hace que sea imprescindible para lograr la satisfacción de los clientes.

En la presente investigación se realiza un análisis de los criterios dados por autores reconocidos en la materia, normas, estándares y guías referenciadas a nivel nacional e internacional, posibilitando identificar

las actividades de calidad que se realizan para conocer el comportamiento de la eficiencia del desempeño en el software, las mediciones y los riesgos más comunes asociados a esta característica. Se proponen buenas prácticas con el objetivo prevenir y/o minimizar los principales riesgos asociados a la eficiencia del desempeño y en consecuencia las insatisfacciones en los usuarios finales una vez que el software esté a su disposición.

Métodos o Metodología Computacional

Para la obtención de información de la investigación se emplearon varios métodos científicos, a continuación, se mencionan y se explica su uso:

Métodos teóricos

1. Analítico – Sintético: se emplea en el análisis de los elementos que brinda la bibliografía sobre la característica eficiencia del desempeño, posibilitando identificar factores comunes para obtener resultados sobre la base de los datos previamente analizados y los riesgos asociados a esta.
2. Inductivo – Deductivo: se utiliza en el diagnóstico de la tendencia del uso de la eficiencia del desempeño a nivel nacional e internacional, para identificar lo común en fenómenos individuales y a partir de ello realizar un razonamiento deductivo que aporten el uso y comportamiento de esta característica durante el ciclo de vida del desarrollo.
3. Análisis histórico – lógico: se utiliza en el estudio la trayectoria histórica sobre la característica eficiencia del desempeño que han sido plasmadas por diferentes autores, modelos, normas y estándares, de forma tal que se proporcionen una base de conocimiento a partir de lo histórico incorporar el empleo de actividades que garanticen una correcta eficiencia del desempeño en el software.

Métodos empíricos

1. Entrevista: se realizan entrevistas formal e informalmente para obtener datos e informaciones para argumentar la situación problemática y selección de las buenas prácticas en el uso e incorporación de la característica eficiencia del desempeño en el ciclo de desarrollo del software.
2. Encuesta: se realizaron encuestas para obtener información y experiencias de los especialistas, profesores, y las organizaciones sobre el uso de la eficiencia del desempeño en el ciclo de desarrollo del software.

Para conocer el comportamiento de la eficiencia del desempeño a nivel nacional, se realizó una encuesta a 37 especialistas e ingenieros de diferentes entidades de Cuba que participan en el desarrollo de software como parte de la política de informatización del país. Entre estas se encuentran: la Empresa de Informática-Automática-Telecomunicaciones (Xetid), Empresa de Desarrollo de Software (DESOFT), Empresa Productora de Software para la Técnica Electrónica (SOFTEL), la Universidad de las Ciencias Informáticas (UCI), el Banco Nacional de Cuba (Bandec), la Empresa de Seguridad Informática (Segurmática) y Centro Nacional de Calidad de Software (CALISOFT).

Los encuestados de las diferentes organizaciones desarrolladoras de software a nivel nacional mencionadas anteriormente, poseen desde 10 a 21 años de experiencia en desarrollo de software y entre los roles que han desempeñado se destacan: líder de proyecto, analista, desarrollador, probador, administrador de la configuración y administrador de la calidad. Al analizar los datos de la encuesta realizada se evidencia que para el 100% de las personas encuestadas el uso de las pruebas de eficiencia del desempeño es importante para la aceptación del producto por los usuarios. Sin embargo, 54% de los encuestados no incluyen las pruebas de eficiencia del desempeño en el proyecto al que pertenece. De los 17 especialistas encuestados que incluyen la realización de pruebas para comprobar la eficiencia del desempeño del software (representan el 46%), se confirmó que un 71% de estos las tienen en cuenta en las etapas finales del producto (37% la realizan al culminar el desarrollo y otro 34% en la fase de cierre); siendo sólo un 29% los que la tienen en cuenta durante la fase de ejecución del software. Para la selección de las tecnologías del desarrollo de software en los proyectos el 33% tienen en cuenta el cumplimiento de la eficiencia del desempeño de dichas tecnologías, destacando elementos como: el uso de recursos, compatibilidad entre

componentes, tiempo de ejecución, flexibilidad, estabilidad, escalabilidad, réplicas entre nodos (web, base de datos, caché en memoria), la alta disponibilidad y el balanceo de carga de las herramientas.

En conclusión, según los datos obtenidos de encuesta realizada se evidencia que se considera importante la eficiencia del desempeño para todos los miembros de los equipos de desarrollo de software. Aunque para un total de 37 especialistas que representan a 7 entidades de desarrollo de software en Cuba, sólo 17 de ellos mencionan que incluyen las pruebas de eficiencia del desempeño para selección de las tecnologías de desarrollo y la comprobación del correcto rendimiento del software, se muestra en la Figura 1: Uso de las Pruebas de Eficiencia del Desempeño. Además, se evidencia a continuación en la Figura 2: Ejecución de las Pruebas de Eficiencia del Desempeño, que sólo el 11% de los encuestados realizan estas pruebas durante el desarrollo del software, mientras que el 35 % las realizan al finalizar el desarrollo o en la fase de cierre del software y el resto no la realiza.

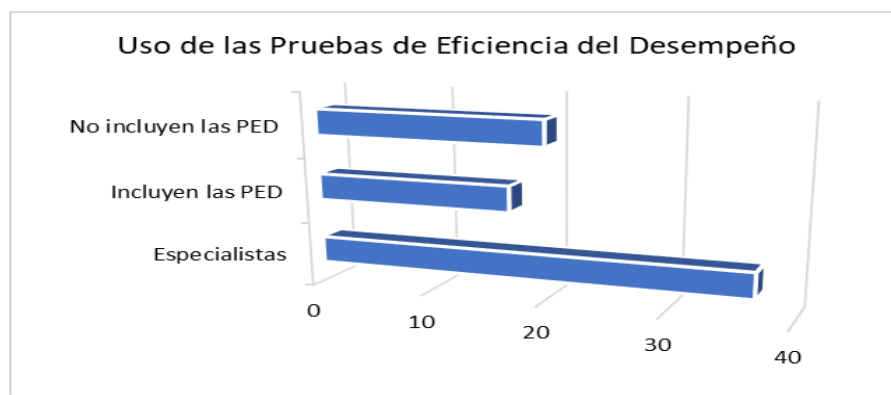


Fig. 1 - Uso de las Pruebas de Eficiencia del Desempeño (PED) a nivel nacional.

Elaboración propia.

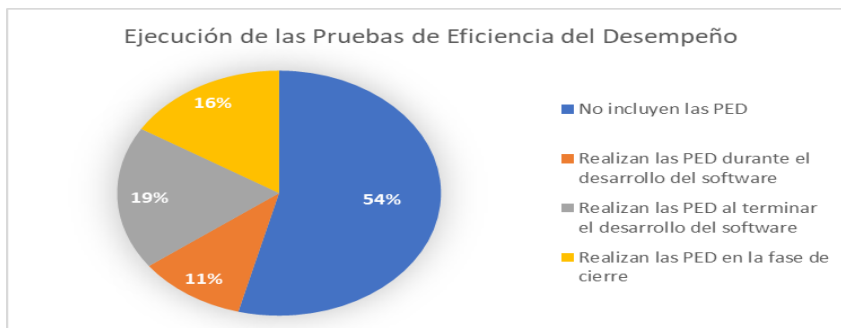


Fig. 2 - Ejecución de las Pruebas de Eficiencia del Desempeño (PED).

Elaboración propia.

Se realizó una entrevista a 6 asesores de la UCI: 1 de la Dirección de Negocio, 2 de la Dirección de Calidad, 2 del Centro de Desarrollo de Seguridad Informática y Servicios Telemáticos (Telemática) y 1 del Centro de Desarrollo de Software Interactivo (VERTEX). Todos los especialistas entrevistados, consideran que una de las principales insatisfacciones de los usuarios finales de los productos que son desarrollados por la universidad es la baja eficiencia del desempeño. Argumentan que la carga de datos, el consumo de memoria, el tiempo de respuesta, la concurrencia de usuarios, la estabilidad, la capacidad y velocidad de procesamiento para las peticiones e hilos de ejecución son problemas identificados que afectan a los sistemas y reflejan insatisfacciones en los usuarios finales y clientes. Un ejemplo mencionado fueron las aplicaciones móviles, donde la demandada de esta característica por los usuarios es un medidor para el software. A través de la calificación de las aplicaciones y opiniones enviadas por los usuarios se evidencia que la eficiencia de desempeño es un objetivo a mejorar dentro de los equipos de desarrollo de la universidad.

A partir de la información obtenida en las encuestas y entrevistas es posible concluir que la eficiencia del desempeño es una característica de calidad importante y que aporta valor al software. Además, se considera que tenerla en cuenta desde etapas tempranas del desarrollo software aporta ventajas significativas respecto a la calidad del producto y mitiga riesgos que afectan la satisfacción del cliente. No obstante, resulta complejo el uso de buenas prácticas y métodos para lograr la medición de la eficiencia del desempeño de un producto en cada una de las fases del ciclo de vida. En consecuencia, conocer la evolución del concepto de

la eficiencia del desempeño y su desarrollo en la actualidad es necesario para incorporar esta característica desde inicio del software.

La IEEE Std. 610.12-1990 define la eficiencia como el grado con que un sistema o componente desempeña las funciones asignadas con el mínimo consumo de recurso(Society, 1990). Pressman en la 5ta Edición de su libro “Ingeniería de Software: Un enfoque práctico” define a la eficiencia como la cantidad de recursos informáticos y de códigos necesarios para que un programa realice su función, mientras que el término eficiencia de ejecución es para el rendimiento del funcionamiento de un programa, teniendo en consideración los requisitos de tiempo de respuesta y de procesamiento(Pressman, 2002). Este autor en la 7ma Edición del libro plantea que para conocer la eficiencia del desempeño de un software es necesario tener en consideración el comportamiento en el tiempo, la utilización de los recursos, la eficacia y conformidad del sistema teniendo en cuenta la cantidad de recursos de cómputo y de código requeridos por un programa para llevar a cabo su función(Pressman, 2010). Sommerville alega que el software no tiene que desperdiciar los recursos del sistema, como la memoria y los ciclos del procesador. Por lo tanto, la eficiencia incluye capacidad de respuesta, tiempo de procesamiento, utilización de memoria, entre otros(Sommerville, 2011).La Norma Internacional ISO/IEC 25010:2011 se refiere a la eficiencia del desempeño como la eficiencia relativa para el costo de los recursos usados bajo condiciones estables, la define como característica de calidad del producto de software que tiene en cuenta el rendimiento de un dispositivo en función de su comportamiento temporal, el uso de recursos y la capacidad o límites máximos de funcionamiento(25010:2011, 2011). La definición dada por la ISO/IEC 25010:2011 ha sido mencionada en varios artículos(Fierro et al., 2019; Kaur, Grover, & Dixit, 2019; Paz et al., 2017) y el autor de la presente investigación considera que es la completa para la emplear durante la misma.

En la literatura sobre calidad de software y eficiencia del desempeño se encuentran varios autores, modelos, normas y estándares que guían a los desarrolladores a ejecutar prácticas y métodos para su medición durante el ciclo de vida del producto. Algunos son: Roger Pressman (Pressman, 2002, 2005, 2010), Ian Sommerville (Sommerville, 2011), la Norma ISO/IEC 25000 (25010:2011, 2011; 25023, 2016; JTC1/SC7, 2005) y ISTQB (ISTQB, 2014; International Software Testing Qualifications Board ISTQB®, 2018; International

Software Testing Qualifications Board ISTQB®, 2018). Para conocer las actividades, métodos, herramientas y el momento que consideran que es necesario para medir la eficiencia del desempeño se realiza un análisis de los mismos.

Roger Pressman

Pressman hace alusión dentro de las características de calidad a los términos rendimiento y eficiencia, este último lo menciona dentro de la calidad en las aplicaciones web. Define que el rendimiento se mide con base en la velocidad de procesamiento, el tiempo de respuesta, el uso de recursos, el conjunto e incluye la eficiencia. Mientras señala que la eficiencia es el grado en el que el software emplea óptimamente los recursos del sistema, según el comportamiento del tiempo y de los recursos. Ambos requisitos se refieren a la percepción de comportamiento del software respecto al desempeño del tiempo de respuesta, el uso de recursos y velocidad de generación de respuestas y pueden medirse sólo de manera indirecta. Por tanto, se refieren a la eficiencia del rendimiento o eficiencia del desempeño al uso de recursos (como la memoria de almacenamiento), sincronía o velocidad, protocolo de redes y requerimientos de servicio, incluidos interfaces del sistema operativo y apoyo de otros componentes.

Lo considera como un requerimiento que debe definirse con el cliente desde el inicio del desarrollo del software y como un criterio que se aplican a todas las fases de prueba del software. Para esto se realizan pruebas diseñadas para valorar el tiempo de respuesta y la confiabilidad conforme aumenta la demanda en la capacidad de recursos y verificar los límites de la eficiencia del desempeño establecidos durante el diseño del software. Dentro de las pruebas que mencionan se encuentran: las pruebas del rendimiento, las pruebas de carga, las pruebas de transacción, las pruebas de esfuerzo, las pruebas de sistema y las pruebas de validación.

Algunos de los problemas o riesgos que causa fallos en la eficiencia del rendimiento en un sistema son: falta de recursos en el lado servidor, red con ancho de banda inadecuada, capacidades de base de datos inadecuadas, capacidades de sistema operativo deficientes o débiles, funcionalidades pobremente diseñadas y otros conflictos de hardware o software que pueden conducir a rendimiento del software degradado. Para

ello, las pruebas de eficiencia del desempeño deben de realizarse lo largo de todos los pasos del proceso de prueba y la intención es doble: 1) comprender cómo responde el sistema conforme aumenta la carga: conociendo el número de usuarios, número de transacciones o volumen de datos global y 2) recopilar mediciones que conducirán a modificaciones de diseño para mejorar el rendimiento.

Cuando todos los elementos del sistema están plenamente integrados puede determinarse el verdadero rendimiento de un sistema para los diferentes niveles de prueba. La prueba del sistema verifica que todos los elementos se mezclan de manera adecuada y que se logra el funcionamiento/rendimiento global del sistema. Las pruebas de transacción se enfocan en comprobar lo correcto del procesamiento y también en los conflictos de eficiencia del desempeño (por ejemplo, tiempos de procesamiento de transacción y volumen de transacción). La prueba de carga examina la carga del mundo real en varios niveles de carga y en varias combinaciones, y la prueba de esfuerzo fuerza a aumentar la carga hasta el punto de rompimiento para determinar cuánta capacidad puede manejar el entorno. La prueba de validación proporciona la garantía final de que el software cumple con todos los requerimientos informativos, funcionales, de comportamiento y de rendimiento.

La base de todo software es la tecnología y a esta se le realizan pruebas de eficiencia del desempeño garantizando los primeros pasos de la prueba del software para una evaluación extensa de las funciones y el rendimiento normal del programa. Analizar cómo puede interpretar de manera diferente la eficiencia del desempeño, depende del área de aplicación del software. La eficiencia del desempeño es uno de los cuatro componentes de riesgo se promedian para determinar un valor de impacto global.(Pressman, 2002, 2005, 2010)

Ian Sommerville

Sommerville menciona la eficiencia como un atributo esencial dentro de las características del producto. Forma parte de los requerimientos no funcionales que afectan más la arquitectura global de un sistema que los componentes individuales, y a menudo es más significativos que los requerimientos funcionales. Incluye los requerimientos de rendimiento y de espacio. Asocia la eficiencia a la cantidad de recursos del sistema,

como la memoria y los ciclos del procesador, conteniendo la capacidad de respuesta, tiempo de procesamiento, utilización de memoria y otros. Mientras que el rendimiento está sobre la base del tiempo de ejecución del sistema y cuánta memoria requiere. Por ende, se refiere al término eficiencia del rendimiento o eficiencia del desempeño.

Considera la eficiencia del desempeño una propiedad emergente de un sistema puesto que es una característica del sistema como un todo, y no de sus partes componentes. Alerta que se debe tener en cuenta desde las etapas tempranas en el desarrollo del software, porque las decisiones de diseño arquitectónico tienen un efecto profundo sobre si el sistema puede o no cubrir requerimientos críticos. Valora en un sistema distribuido que los riesgos que afectan la eficiencia del desempeño son: la velocidad de ejecución de un procesador, el ancho de banda de la red, la carga de la red y la velocidad de todas las computadoras que forman parte del sistema. Por ello, se considera la arquitectura como requerimientos de calidad de servicio y debe diseñarse para localizar operaciones críticas dentro de un pequeño número de componentes.

El modelado de interacción de componentes ayuda a entender si es probable que una estructura de un sistema propuesto obtenga la eficiencia del desempeño requerida. Para medir la eficiencia del desempeño menciona el uso métricas dinámicas, realizando mediciones al programa en ejecución. Dichas métricas pueden recopilarse durante las pruebas realizadas al sistema. Propone las pruebas de rendimiento, de esfuerzo, de sistema (incluyendo las pruebas de versión) y las pruebas de usuario (incluyendo las pruebas alfa, beta y de aceptación). Las pruebas de rendimiento deben diseñarse para garantizar que el sistema procese su carga pretendida. Las pruebas de esfuerzo ayudan a descubrir cuándo comienza la degradación, de manera que se puedan adicionar comprobaciones al sistema para rechazar transacciones más allá de este punto. Las pruebas de sistema demuestran cuando la eficiencia del desempeño puede producir salidas inesperadas o no deseadas rendimiento. Las pruebas de usuario son una etapa en el proceso de pruebas donde los clientes proporcionan entrada y asesoría sobre las pruebas del sistema por interacciones inesperadas entre diferentes partes de un programa, problemas de temporización o dificultades con el rendimiento del sistema para decidir si está o no listo para ser aceptado por los desarrolladores del sistema y desplegado en el entorno del cliente. (Sommerville, 2011)

Norma ISO/IEC 25000:2011

La Norma ISO/IEC 25010: 2011 establece un modelo de calidad que puede ser usado para identificar características de calidad relevantes que favorecen el establecimiento de requerimientos, criterios para la satisfacción y la medición correspondiente. En esta norma se determinan las características de calidad que se deben tener en cuenta en el momento de evaluar las propiedades de un producto software. Se define que la calidad del producto software se puede tomar como el grado en que satisface los requisitos de sus usuarios, aportando de esta forma valor. Además, precisa cinco características para el modelo de uso de la calidad (eficacia, eficiencia, satisfacción, libre de riesgo y cobertura del contexto) y ocho para el modelo de calidad del producto (adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad). Cada una de las características en el modelo de calidad del producto está conformada por un conjunto de subcaracterísticas que permiten la evaluación de un producto de software: adecuación funcional (completitud funcional, corrección funcional, pertinencia funcional), compatibilidad (coexistencia, interoperabilidad), eficiencia de desempeño (capacidad, comportamiento temporal, utilización de recursos), mantenibilidad (analizabilidad, capacidad de ser modificado, capacidad de ser probado, modularidad, reusabilidad), fiabilidad (capacidad de recuperación, disponibilidad, madurez, tolerancia a fallos), portabilidad (adaptabilidad, capacidad de ser reemplazado, facilidad de instalación), seguridad (autenticidad, confidencialidad, integridad, no-repudio, responsabilidad) y usabilidad (accesibilidad, aprendizaje, estética, inteligibilidad, operabilidad, protección frente a errores de usuario). (25010:2011, 2011; 25023, 2016; JTC1/SC7, 2005; Muñoz, 2018)

Norma ISO/IEC 25023:2016

La Norma ISO/IEC 25023:2016 define las propiedades medibles de calidad relacionadas de un producto de sistema/software. Estas propiedades se miden aplicando un método de medición. El resultado de aplicar un método de medición se denomina elemento de medida de calidad. La característica eficiencia de desempeño está relacionada con el rendimiento de un dispositivo en función de su comportamiento temporal, el uso de recursos y la capacidad o límites máximos de funcionamiento. La subcaracterística comportamiento en el

tiempo está constituida por los tiempos de respuesta y procesamiento de una aplicación en ejecución en condiciones determinadas, su valoración se obtiene a través de tres medidas de calidad: i) tiempo medio de respuesta, ii) tiempo de respuesta adecuado, iii) rendimiento promedio. La subcaracterística utilización de recursos se refiere al uso de éstos durante la ejecución de una tarea en condiciones preestablecidas. Se mide a través de cuatro medidas de calidad: i) media de utilización del procesador, ii) utilización media de la memoria, iii) uso medio de los dispositivos entrada y salida y iv) utilización del ancho de banda. La subcaracterísticas capacidad o límites máximos de funcionamiento se refiere a la capacidad de respuesta del software cumplen los requisitos. Se mide a través de tres medidas de calidad: i) capacidad de procesamiento de transacciones, ii) capacidad de acceso de usuario y iii) accesibilidad de acceso de usuario. (25010:2011, 2011; 25023, 2016; Alain Abran, 2016; Fierro et al., 2019)

ISTQB

El Comité Internacional de Cualificación de Pruebas de Software ISTQB® realiza la certificación de Nivel Básico de Prueba de Rendimiento a cualquier persona involucrada en la ingeniería del rendimiento. La eficiencia de desempeño es una parte esencial para proporcionar una "buena experiencia" a los usuarios cuando utilizan sus aplicaciones en distintas plataformas fijas y móviles. Técnicamente clasifica la eficiencia de desempeño como una característica del modelo de calidad de producto de la Norma ISO/IEC 25010:2011, que incluye las subcaracterísticas mencionadas por esta: comportamiento temporal, utilización de recursos y capacidad.

Alega que el comportamiento temporal es un aspecto de la prueba de rendimiento que evalúa la capacidad de un componente o sistema para responder a las entradas del usuario o del sistema en un tiempo y bajo unas condiciones específicas. Las mediciones del comportamiento temporal pueden variar desde el tiempo de "extremo a extremo" que tarda el sistema en responder a la entrada del usuario, hasta el número de ciclos de CPU requeridos por un componente de software para ejecutar una tarea particular. La utilización de recursos se determina si la disponibilidad de los recursos del sistema es un riesgo, la utilización de esos recursos (por ejemplo, la asignación de memoria RAM limitada) puede investigarse mediante la realización de pruebas de rendimiento específicas. En cuanto a la capacidad se identifican como problemas de

comportamiento del sistema en los límites de la capacidad requerida del sistema (por ejemplo, el número de usuarios o el volumen de datos), se pueden realizar pruebas de rendimiento para evaluar la adecuación de la arquitectura del sistema.

Se tienen en cuenta varios tipos de pruebas de rendimiento, estas incluyen, las propias pruebas de rendimiento, las pruebas de carga, de estrés, de escalabilidad, de resistencia, de capacidad, de picos y de concurrencia. La prueba de rendimiento juega un papel crítico en el establecimiento de niveles de calidad aceptables para el usuario final en cuanto a la eficiencia de desempeño del software. Estas pueden llevarse a cabo de manera iterativa en apoyo del análisis, diseño e implementación del sistema para permitir que se tomen decisiones arquitectónicas y para ayudar a dar forma a las expectativas de los implicados. Los equipos deben incorporar la prueba de rendimiento estático y dinámico en iteraciones tempranas, en lugar de esperar a las iteraciones finales para abordar los riesgos de rendimiento.

Menciona los riesgos asociados para la eficiencia del desempeño según las arquitecturas de los sistemas, dígame, sistemas individuales, sistemas multicapas, sistemas distribuidos, sistemas virtualizados, sistemas clientes-servidor, sistemas en tiempo real embebidos, aplicaciones móviles y aplicaciones mainframe. Dentro de los riesgos característicos se evidencian: el consumo excesivo, incluyendo la fuga de memoria, actividades en segundo plano como software de seguridad, subsistemas de almacenamiento lento (por ejemplo, dispositivos externos de baja velocidad o fragmentación del disco), mala administración del sistema operativo, diseño de base datos deficiente o no escalable, cuellos de botellas en la red y ancho de banda o capacidad inadecuados en un único servidor. Además, el rendimiento puede degradarse debido al flujos de trabajos críticos o datos hacia, desde o a través de servidores remotos poco fiables o impredecibles, principalmente cuando existen problemas de conexión o periodo de carga intensa e intermitentes. Asociados a deficiencias en cuanto a la velocidad y fiabilidad de la conexión, congestión en la red y problemas potenciales debido a cortafuegos, inspección de paquetes, equilibrio de carga del servidor o por la carga excesiva en el hardware.

ISTQB plantea que la arquitectura es claramente una decisión técnica importante con un profundo impacto en los riesgos del rendimiento del software. Propone métricas de rendimiento entre las que se encuentra: el tiempo de respuesta de la transacción, transacciones por segundo, capacidad de procesamiento, respuestas por segundo. Para la monitorización del rendimiento se tiene en cuenta el uso del CPU y de memoria empleada. Argumenta que el uso de herramientas de monitorización puede apoyar eficazmente el análisis de la causa raíz del rendimiento sujeto a prueba y también se puede utilizar para controlar un entorno de producción cuando se entrega el producto. La prueba de rendimiento puede formar parte de las pruebas automatizadas realizadas en la Integración Continua si las pruebas se diseñan a nivel de construcción y durante estas también se pueden ejecutar monitores de ejecución en el propio generador de carga. (ISTQB, 2014; International Software Testing Qualifications Board ISTQB®, 2018; International Software Testing Qualifications Board ISTQB®, 2018)

Resultados y discusión

Del análisis realizado sobre la eficiencia del desempeño de autores, estándares, normas y guías se puede concretar que todos ofrecen un concepto que incluyen subcaracterísticas. Proponen diferentes tipos de pruebas como actividades de calidad para el control y medición, incluyendo métricas a tener en cuenta, además de los riesgos asociado a la deficiencia del rendimiento en los sistemas. Es significativo destacar que reconocen la necesidad de tener en cuenta la característica desde la concepción del software, donde las decisiones en la arquitectura, el diseño e implementación afectan el rendimiento desde una perspectiva técnica, creando riesgos e influyendo en la probabilidad de defectos con un alto impacto en el sistema. Por lo tanto, es importante la identificación, evaluación, mitigación y gestión de los riesgos de la eficiencia del desempeño a lo largo del ciclo de vida del software. Los requisitos de eficiencia del desempeño es una información de entrada fundamental para comenzar el diseño, implementación y ejecución de las pruebas que se realizarán en todos los niveles del sistema. Es relevante mencionar que las Normas ISO establece características, subcaracterísticas y medidas, pero no la relación con los riesgos, o el momento del ciclo de vida que puede ser medida. A continuación, se muestra una tabla comparativa para identificar semejanzas y

diferencias entre los criterios aportados sobre la eficiencia del desempeño de los autores, estándares, normas y guías analizadas:

Tabla 1 - Análisis de la eficiencia del desempeño. Resumen Comparativo.

Aspectos de la eficiencia del desempeño	Pressman	Sommerville	ISO/IEC 25010:2011	ISO/IEC 25023:2016	ISTQB
Características y subcaracterísticas	Velocidad de procesamiento, comportamiento del tiempo de respuesta y el uso de recursos.	Utilización de recursos del sistema, capacidad de respuesta, tiempo de procesamiento.	Comportamiento temporal, el uso de recursos y la capacidad o límites máximos de funcionamiento.	Se rige por la Norma ISO/IEC 25010:2011.	Se rige por la Norma ISO/IEC 25010:2011.
Etapas de prueba y evaluación	Se aplican a todas las fases de prueba del software.	Desde la concepción y diseño del software.	Desde etapas tempranas del desarrollo del software.	Se rige por la Norma ISO/IEC 25010:2011.	Desde etapas tempranas del desarrollo del software. De manera iterativa en apoyo del análisis, diseño e implementación del sistema.
Actividades de calidad	Pruebas de rendimiento, de carga, de transacción, de esfuerzo, de sistema y de validación.	Pruebas de rendimiento, de esfuerzo, de sistema: pruebas de versión. Pruebas de usuario: las alfa, beta y de aceptación.	No mencionan	No mencionan	Pruebas de rendimiento, de carga, de estrés (o prueba de esfuerzo), de escalabilidad, de picos, de resistencia, de concurrencia, de capacidad.
Medidas y/o mediciones	A través de pruebas diseñadas para valorar el tiempo de respuesta y la confiabilidad conforme aumenta la demanda en la capacidad de recursos y verificar los límites de la eficiencia del desempeño establecidos durante el diseño del software.	Métricas dinámicas, realizadas con el programa en ejecución. Dichas métricas pueden recopilarse durante las pruebas realizadas al sistema.	Se rige por la Norma ISO/IEC 25023:2016.	Se aplican un método de medición para obtener medida de calidad. Cada subcaracterística tiene asociado medidas de calidad.	Valora el comportamiento de cada subcaracterística de la eficiencia del desempeño a través de mediciones realizadas por pruebas de rendimiento específicas. Varían según los factores del contexto del negocio, el operativo y los objetivos de las pruebas.
Riesgos	Falta de recursos en el lado servidor. Red con ancho de banda inadecuada.	Riesgos que afectan la eficiencia del desempeño son: la velocidad de ejecución de un	No mencionan	No mencionan	Vienen asociados al consumo excesivo, mala administración del sistema operativo, diseño de base datos deficiente o no escalable, cuellos de botellas

	Capacidades de base de datos inadecuadas. Capacidades de sistema operativo deficientes o débiles. Funcionalidades pobremente diseñadas y otros conflictos de hardware o software que pueden conducir a rendimiento del software degradado	procesador, el ancho de banda de la red, la carga de la red y la velocidad de todas las computadoras que forman parte del sistema.			en la red, ancho de banda o capacidad inadecuados en un único servidor, flujos de trabajos críticos, deficiencias en cuanto a la velocidad y fiabilidad de la conexión, congestión en la red y problemas potenciales debido a cortafuegos, inspección de paquetes, equilibrio de carga del servidor o por la carga excesiva en el hardware.
--	---	--	--	--	---

Elaboración Propia.

Durante el desarrollo del software pueden ocurrir modificaciones en funcionales que afecten directamente a la eficiencia del desempeño del mismo, pues se realizan un conjunto de cambios y en ocasiones no se mide el impacto y las consecuencias que traen consigo. Por ello, una correcta gestión de riesgo desde inicio del software de los requisitos de la eficiencia del desempeño propicia mayor claridad para la identificación y análisis de los riesgos asociados a esta característica y la forma de evaluar las alternativas para resolverlos. Para los riesgos relacionados con la eficiencia del desempeño que afectan la calidad del producto, el proceso es (International Software Testing Qualifications Board ISTQB®, 2018):

1. Identificar los riesgos de la calidad del producto, centrándose en características como comportamiento temporal, el uso de recursos y la capacidad.
2. Evaluar los riesgos identificados, asegurándose de que se abordan las categorías de arquitectura pertinentes. Evaluar el nivel global riesgo para cada riesgo identificado en términos de probabilidad e impacto utilizando criterios claramente definidos.
3. Adoptar las medidas adecuadas para mitigar los riesgos de cada elemento de riesgo en función de la naturaleza del elemento del riesgo y del nivel del riesgo.
4. Gestionar los riesgos de forma continua para garantizar que los riesgos se mitiguen adecuadamente antes de su entrega.

A continuación, se evidencia conjunto de riesgos identificados en el análisis realizado en la literatura y por un grupo de especialistas que inspeccionan el comportamiento de esta característica en la Dirección de Calidad de Software de la UCI, permitiendo, además, identificar y proponer buenas prácticas a tener en cuenta para prevenir los riesgos.

Tabla 2- Riesgos y buenas prácticas asociados a la eficiencia del desempeño.

Riesgos	Buenas Prácticas
1. Pocos recursos asociados al hardware y/o al software, dígase PC con pocas prestaciones de velocidad de ejecución del procesador, de memoria, poco almacenamiento en disco duro, capacidades de sistema operativo deficientes o mala administración del mismo. 2. Asignación de capacidades inadecuadas para servidores, base de datos, ancho de banda y/o sistemas asociados. 3. Funcionalidades pobremente diseñadas e implementadas que demandan un consumo excesivo del hardware y/o software. 4. Deficiencias en la arquitectura del software y la red. 5. Deficiencias en cuanto a la velocidad y fiabilidad de la conexión dada por congestión en la red, el poco ancho de banda, la generación de cuellos de botellas, problemas debido al cortafuegos, inspección de paquetes, carga del servidor o por la carga excesiva en el hardware.	1. Identificación de los riesgos asociados a los recursos de hardware y/o software para definir las posibles deficiencias que puedan influir en el rendimiento del sistema. 2. Definir desde inicio del software los requisitos de eficiencia del desempeño, teniendo en cuenta las características asociadas al comportamiento temporal, el uso de recursos y la capacidad. (Disciplina Levantamiento de requisitos del software) 3. Ejecución de pruebas de rendimiento, de carga, de estrés, de escalabilidad, de concurrencia, de capacidad, de resistencia y de picos periódicamente para obtener el estado de la eficiencia del desempeño en el software. Siendo recomendable realizar estas pruebas en el hardware real lo antes posible que permitan capturar las restricciones de recursos y comportamientos relacionados con el rendimiento. 4. Recopilación de datos e información sobre el comportamiento de la eficiencia del desempeño del software en diferentes ambientes de ejecución, para un análisis detallado e identificación de posibles mejoras. 5. Utilización de herramientas dedicadas a las pruebas, la monitorización y el análisis de registros de la eficiencia del desempeño en el software.

Elaboración Propia.

La eficiencia del desempeño es importante para proporcionar una “buena experiencia” a los usuarios en las aplicaciones en distintas plataformas fijas o móviles. Las pruebas propuestas juegan un papel fundamental en el establecimiento de niveles de calidad aceptables para el usuario final. El enfoque y el orden de

prioridad adecuados dependen de los riesgos y las necesidades de los implicados. Las pruebas deben estar alineadas con las expectativas definidas en los diferentes grupos implicados, deben ser reproducibles, aportar resultados que sean comprensibles y que puedan compararse fácilmente, además, asequibles desde el punto vista práctico y ejecutable en el plazo establecido. (International Software Testing Qualifications Board ISTQB®, 2018)

La identificación de los riesgos asociados a los recursos de hardware y software para definir las posibles deficiencias que puedan influir en el rendimiento del sistema desde realizarse desde la concepción del negocio, de forma que para el levantamiento de los requisitos se tengan en cuenta las prestaciones de los recursos que se tienen. Los requisitos de eficiencia del desempeño, estarán asociadas al comportamiento temporal, el uso de recursos y la capacidad o límites máximos de funcionamiento; estos deben definirse en el inicio del desarrollo del software para conocer si los niveles de rendimiento son aceptables para cumplir con los objetivos operativos. Cada prueba propuesta (de rendimiento, de carga, de estrés, de escalabilidad, de concurrencia, de capacidad, de resistencia y de pico) proporciona información valiosa sobre el rendimiento del software, estas deben ejecutarse iterativamente y de modo continuo para cada nivel de prueba (componente, integración, sistema, integración del sistema y pruebas de aceptación). Para ello, se deben planificar, monitorizar, controlar, realizar el análisis y diseño de la prueba para luego en implementación definir los casos de pruebas de rendimiento que serán ejecutados en el software. Se obtienen resultados que serán expresados a través de métricas definidas en el proyecto y de fácil comprensión para todos los involucrados. Para el análisis de dichas métricas es necesario la recopilación de datos e información sobre el comportamiento de la eficiencia del desempeño del software en diferentes ambientes de ejecución, para un análisis detallado e identificación de posibles mejoras. La ejecución de pruebas de rendimiento en ocasiones requiere de hardware y una herramienta para la generación de carga, volumen de datos, velocidad y precisión que proporcionen mediciones y métricas para facilitar la evaluación de la prueba y registro de información.

Se debe tener en cuenta que estos tipos de pruebas propuestas están asociadas a las actividades de calidad para conocer y mitigar los riesgos relacionados con el mal desempeño de las aplicaciones en los entornos de

producción y realizar las correcciones necesarias antes de salir al mercado (Castillo et al., 2017). Para la valoración de la propuesta dada, fue necesario realizar una encuesta que permitiera obtener criterios de expertos. Para la selección de estos se aplicó la técnica de análisis curricular. Participaron 15 expertos con más de diez años en la industria del software y de diferentes organizaciones desarrolladoras de software a nivel nacional.

Proceso de selección de expertos

Se realizó una valoración de los posibles expertos, se consideraron como los criterios de selección iniciales: los conocimientos relacionados con la calidad y la gestión de la calidad en proyectos de software, experiencia laboral en la industria de software de 8 años en adelante, experiencia práctica como el principal factor y la producción científica enfocada al objetivo a evaluar. Al tener en cuenta estos criterios se realizó un cuestionario para el conocimiento curricular. Como resultado se seleccionaron 15 expertos a nivel nacional, de las instituciones: Segurmática, DESOFT, XETID, UCI, CUJAE y CALISOFT. Luego de seleccionados los expertos, teniendo en cuenta los conocimientos sobre las características de calidad del producto y la experiencia práctica que poseen sobre el valor del cumplimiento de las mismas, se aplicó el método Delphi para aprovechar los elementos comunes en el grupo de expertos. Se preservó el anonimato mediante el uso de flujos de comunicación que permite la participación de los expertos aunque se encuentren geográficamente dispersos (Trujillo, 2014). Se realizaron 3 rondas para combinar y concretar los resultados emitidos. Se arrojaron resultados satisfactorios a partir del análisis de las respuestas dadas, pues todas las categorías fueron evaluadas de Muy altas o Altas, validando la contribución de las buenas prácticas asociadas a minimizar riesgos de la eficiencia del desempeño, además como característica necesaria a incorporar al inicio del desarrollo del software, como fundamental para la satisfacción de los usuarios y aceptación del producto. Se obtuvo una moda de Alta o Muy Alta y los expertos no emitieron votos en la escala de Baja o Ninguna. A partir de los votos emitidos por los expertos se obtienen los siguientes resultados:

Tabla 3- Por ciento del criterio de los expertos.

Criterio	Por ciento	Moda
Relevancia	95	5
Pertinencia	94	5
Coherencia	84	4
Comprensión	96	5
Exactitud	70	4

Elaboración propia.

A partir de los estos resultados se pueden asegurar que los expertos concuerdan en que la eficiencia del desempeño es necesario tenerla en cuenta para la satisfacción de los usuarios finales. Que para ellos las buenas prácticas propuestas inciden positivamente en la prevención y control de los riesgos asociados a la eficiencia del desempeño y que es necesario incorporarla desde inicio del desarrollo del software, siendo necesario realizar una correcta gestión de riesgo de esta característica para que no repercuta negativamente en la explotación del sistema y aceptación del mismo.

Conclusiones

Las normas, estándares y autores reconocidos a nivel internaciones analizados, ofrecen un marco común en el que se establecen características, subcaracterísticas, mediciones, actividades de calidad y riesgos para la eficiencia del desempeño. Se concluye que

1. La eficiencia del desempeño se relaciona al comportamiento temporal (tiempo de procesamiento, de respuesta, de ejecución), el uso de recursos (tanto de hardware como de software) y la capacidad o límites máximos de funcionamiento (límites de velocidad de red, de procesamiento, de ancho de banda, de base de datos).
2. Es preciso definir los requisitos de la eficiencia del desempeño desde inicio del desarrollo de software para obtener un diseño y una arquitectura que se correspondan y analizar el

comportamiento de esta durante el desarrollo del mismo para minimizar los riesgos asociados con los defectos o fallas que estas pueden ocasionar.

3. La ejecución de las buenas prácticas propuestas debe incorporarse como parte del desarrollo del software y tienen que realizarse iterativo y continuo, para así prevenir riesgos, mejorar la calidad del producto y la satisfacción de los usuarios finales.

Referencias

- 25010:2011, I. S. I. I. (2011). ISO/IEC 25010 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. In (First edition ed., pp. 44): Switzerland.
- 25023, I. I. (2016). Medición de la calidad del producto del sistema y software. In (pp. 61). Switzerland.
- Alain Abran, J.-M. D., Rafa Al Qutaish, Naji Habra (2016). An Information Model for Software Quality Measurement with ISO Standards 18.
- Castillo, F. F. R., Mora, N. M. L., Elizaldes, K. D. C., & Orozco, J. I. P. J. C. T. g. d. i. a. a. l. p. (2017). Estado del Arte: Métricas de calidad para el desarrollo de aplicaciones web. 6(4), 1-12.
- Estayno, M., Dapozo, G., Liliana, C. P., & Greiner, C. (2009). MODELOS Y MÉTRICAS PARA EVALUAR CALIDAD DE SOFTWARE.
- Fierro, F. A. S., Manosalvas, C. A. P., Rodríguez, N. N. C., & Landeta, P. (2019). Análisis de la eficiencia de desempeño en aplicaciones de Realidad Aumentada utilizando la normativa ISO/IEC/25010. Revista Ibérica de Sistemas e Tecnologías de Informação, 13.
- Globe, (2017). Pruebas de rendimiento en eCommerce. Retrieved from <https://www.globetesting.com/2017/01/pruebas-de-rendimiento-para-ecommerce/>
- Godfrey, A. B. (1998). Gestión de la Calidad Total. Capítulo 14. In Gestión de la Calidad Total (pp. Pag. 39).
- ISTQB, I. S. T. Q. B. (2014). Effectiveness Survey. Retrieved from
- ISTQB®, I. S. T. Q. B. (2018). Programa de Estudio de Nivel Básico Especialidad – Prueba de Rendimiento. Retrieved from España:

- ISTQB®, I. S. T. Q. B. (2018). Worldwide Software Testing Practices Report. Retrieved from www.istqb.org
- JTC1/SC7, I. I. (2005). Software product quality requirements and evaluation (SQuaRE) - Guide to SQuaRE. In FCD 25000 - Software Engineering – Software product quality requirements and evaluation (SQuaRE) - Guide to SQuaRE (pp. 52). CANADA CANADA (SCC).
- Kaur, A., Grover, P. S., & Dixit, A. (2019). Performance Efficiency Assessment for Software Systems. 10.
- Marin Diaz, A., Trujillo Casañola, Y., & Buedo Hidalgo, D. J. R. C. d. C. I. (2020). Estrategia de pruebas para organizaciones desarrolladoras de software. 14(3), 83-104.
- Mauro Callejas Cuervo, Andrea Catherine Alarcón Aldana, & Carreño., A. M. Á. (2017). Modelos de calidad del software, un estado del arte. Ingenieria y Tecnología, Vol 13 pp. 15. doi:<http://dx.doi.org/10.18041/entramado.2017v13n1.25125>
- Micro Focus, C. G., Sogeti. (2020). World Quality Report Retrieved from
- Muñoz, J. C.-R. (2018). Asistente para la evaluación de la calidad del producto de software según la familia de Normas ISO/IEC 25000 utilizando el enfoque GQM.,
- Paz, J. A. M., Gómez, M. Y. M., & Rosas, S. C. (2017). Análisis sistemático de información de la Norma ISO 25010 como base para la implementación en un laboratorio de Testing de software en la Universidad Cooperativa de Colombia Sede Popayán. 4to Congreso Internacional AmITIC 2017, Popayán, Colombia., 6.
- Pressman, R. S. (2002). Ingeniería del Software. Un Enfoque Práctico (C. F. Madrid Ed. Quinta ed.).
- Pressman, R. S. (2005). Ingeniería de Software, un enfoque práctico (6ta ed.).
- Pressman, R. S. (2010). Ingeniería del Software. Un Enfoque Práctico (Séptima ed.).
- Socarras Ramírez, I., Trujillo Casañola, Y., & Vega Prieto, R. J. R. C. d. C. I. (2018). La mejora de procesos organizacionales para proyectos de desarrollo de software. 12(3), 177-191.
- Society, S. C. C. o. t. I. C. (1990). Std 610.12-1990 Standard Glossary of Software Engineering Terminology. In IEEE Computer Society (pp. pp. 94). USA: USA.
- Sommerville, I. (2011). Ingeniería de Software (Novena ed.).
- Trujillo, Y. C. (2014). " MODELO PARA VALORAR LAS ORGANIZACIONES DESARROLLADORAS DE SOFTWARE AL INICIAR LA MEJORA DE PROCESOS". (Tesis de doctorado), Universidad de las Ciencias Informáticas

Conflicto de interés

Los autores de este artículo autorizan la distribución y uso de su artículo.

Contribuciones de los autores

1. Conceptualización: Aymara Marin Díaz
2. Curación de datos: Yaimí Trujillo Casañola
3. Análisis formal: Maidelyn Piñero González
4. Adquisición de fondos: -
5. Investigación: Maidelyn Piñero González
6. Metodología: Denys Buedo Hidalgo
7. Administración del proyecto: Yaimí Trujillo Casañola
8. Recursos: Denys Buedo Hidalgo
9. Software: Denys Buedo Hidalgo
10. Supervisión: Yaimí Trujillo Casañola
11. Validación: Aymara Marin Díaz
12. Visualización: Maidelyn Piñero González
13. Redacción – borrador original: Maidelyn Piñero González
14. Redacción – revisión y edición: Aymara Marin Díaz

Financiación

No fue necesaria financiación para el desarrollo de la investigación.

¹ Compañía de software global con 40 años de experiencia en la entrega y soporte de software empresarial.

² Sociedad para la gestión y tratamiento de la información.

³ Corporación multinacional francesa de servicios de consultoría tecnológica.