

# Nombramiento de prácticas relacionadas con requisitos no funcionales en el proceso de desarrollo de software

Naming Practices Related to Non-Functional Requirements in the Software Development Process

Carlos Mario Zapata Jaramillo<sup>1</sup>

<sup>1</sup>Facultad de Minas, Universidad Nacional de Colombia, Medellín, Colombia.  
Email: cmzapata@unal.edu.co

 OPEN ACCESS



## Copyright:

©2020. La revista *Ingenierías USBmed* proporciona acceso abierto a todos sus contenidos bajo los términos de la licencia creative commons Atribución no comercial SinDerivar 4.0 Internacional (CC BY-NC-ND 4.0)

**Tipo de artículo:** Investigación.

**Recibido:** 01-05-2020.

**Revisado:** 03-08-2020.

**Aprobado:** 17-06-2020.

**Doi:** 10.21500/20275846.4697

## Referenciar así:

C. M. Zapata-Jaramillo, "Nombramiento de prácticas relacionadas con requisitos no funcionales en el proceso de desarrollo de software," *Ingenierías USBMed*, vol. 11, no. 2, pp. 39-47, 2020.

## Disponibilidad de datos:

todos los datos relevantes están dentro del artículo, así como los archivos de soporte de información.

## Conflicto de intereses:

los autores han declarado que no hay conflicto de intereses.

**Editor:** Andrés Felipe Hernández.

Universidad de San Buenaventura,  
Medellín, Colombia.

**Resumen.** En ingeniería de software es usual hablar de prácticas en diferentes niveles como grupos de actividades que se repiten y entregan buenos resultados con su uso. Los requisitos no funcionales (entendidos como restricciones en los servicios o funciones del sistema) también tienen prácticas asociadas. Sin embargo, existen formas de nombramiento que no permiten generar un consenso alrededor de dichas prácticas, pues se utilizan productos de trabajo, restricciones, descripciones y otros elementos que impiden reconocer la práctica y su uso. Por ello, en este artículo se nombra un conjunto de prácticas sobre requisitos no funcionales a partir de su descripción en la literatura y tomando en consideración los elementos que requiere el buen nombramiento. Se espera que este conjunto de prácticas de requisitos no funcionales sirva como punto de partida para la representación detallada en estándares gráficos que se definan para ese fin.

**Palabras Clave.** Requisitos no funcionales, prácticas, nombramiento de prácticas.

**Abstract.** In software engineering we can talk about practices at different levels. Practices are understood as sets of repeatable activities for achieving good results when they are used. Non-functional requirements –constraints on services/functions of a system– also have associated practices. However, naming forms are useless for generating consensus about such practices, since elements like work products, constraints, descriptions, and so on are used, and such elements interfere with practice recognition and usage. For this reason, in this paper we name a set of practices about non-functional requirements from their description in the state of the art. We also consider the elements required by good naming of practices. We hope this set of practices about non-functional requirements can be used as a starting point for detailed presentation by using graphical standards employed for practices.

**Keywords.** Non-Functional Requirements, Practices, Practice Naming.

## I. Introducción

En la Esencia de Semat (*Software Engineering Method and Theory; Object Management Group*) [1] se define una práctica en ingeniería de software como un enfoque repetitivo para hacer algo con un objetivo determinado. Esta definición implica la existencia de un conjunto de actividades que se organizan de acuerdo con reglas específicas [2] y que entregan buenos resultados con su uso.

Los requisitos no funcionales son restricciones en los servicios o funciones que ofrece el sistema, incluyendo restricciones de tiempo, restricciones en los procesos y estándares de desarrollo [3]. Algunos autores reconocen prácticas relacionadas con requisitos no funcionales [1]–[10], pero para nombrarlas emplean diferentes elementos: productos de trabajo y descripciones completas de funcionamiento [4], [5], restricciones [6] y nombres de métodos [7]. En otros casos emplean verbos junto con sustantivos [4], [6]–[10]. Barón [11] establece que una práctica bien nombrada en ingeniería de software debería contener un verbo nominalizado, un sustantivo y un adjetivo que identifique la forma en que se realiza la acción que se define con el verbo nominalizado. Barn [9] nombra una práctica con una estructura similar a la de Barón [11], utilizando un nombre de método, pero no aplicable al dominio de los requisitos no funcionales. Esta forma de nombramiento de las prácticas impide tener un consenso sobre cuáles son realmente las prácticas aplicables a los requisitos no funcionales y de qué forma se realizan.

Para solucionar estos problemas, en este artículo se propone el nombramiento a un conjunto de prácticas aplicables a requisitos no funcionales, a partir de una revisión de la estructura que proponen diferentes autores y tomando en cuenta la estructura de Barón [11]. Para ello, se recopila un conjunto de elementos de prácticas en diferentes artículos de la literatura, se evalúa su uso y se establece una nueva manera de nombramiento con base en la descripción que se suministra de la práctica. Este conjunto de elementos constitutivos y prácticas nombradas puede servir de base para una representación detallada que establezca los principales elementos que se involucran en las prácticas, con miras a una especificación precisa empleando un estándar, como por ejemplo el núcleo de la Esencia de Semat [1].

Este artículo se estructura de la siguiente manera: en la sección II se presenta el marco conceptual de esta propuesta, incluyendo los requisitos no funcionales, el concepto de práctica y los elementos para el nombramiento de la práctica en ingeniería de software; en la sección III se presentan los antecedentes para el nombramiento de prácticas relacionadas con requisitos no funcionales; en la sección IV se propone el nombramiento de un conjunto de prácticas sobre ese tema; finalmente, en la sección V se discuten las conclusiones y el trabajo futuro.

## II. Marco teórico

### A. Requisitos no funcionales

Sommerville [3] establece que los requisitos no funcionales corresponden a ciertas características de calidad de los sistemas de software, que se pueden agrupar de la siguiente manera:

- Requisitos del producto: usabilidad, eficiencia (rendimiento y espacio), confiabilidad y portabilidad.
- Requisitos organizacionales: entrega, implementación y estándares.
- Requisitos externos: interoperabilidad, requisitos éticos y requisitos legislativos (privacidad y seguridad).

### B. Práctica

En el núcleo de la Esencia de *Semat* [1] las prácticas sirven para conformar métodos, por lo cual es necesario que con la práctica se provea una forma sistemática y verificable de tratar algún aspecto del trabajo que se realiza. Por ello, es conveniente que las acciones repetibles que se puedan realizar en la práctica se definan adecuadamente y que el objetivo sea muy claro. Para Kemmis *et al.* [2] una práctica es una forma socialmente establecida de actividad humana cooperativa, en la cual arreglos característicos de acciones y actividades son comprensibles en términos de arreglos de ideas relevantes en discursos característicos. Estos arreglos se mantienen juntos para un proyecto distintivo.

Tanto el objetivo como el proyecto distintivo hacen que la práctica se imponga en el tiempo por el valor que ofrecen sus resultados para un grupo humano en particular. Eso hace que la práctica alcance su real valor y perdure.

### C. Elementos para el nombramiento de la práctica en ingeniería de software

Para Barón [11] el nombre de una práctica en ingeniería de software se estructura con un verbo nominalizado, que indica lo que se hace con la práctica; un adjetivo, que indica cómo se hace y un sustantivo, que indica el objeto sobre el cual se aplica la práctica. Para la definición del nombre Barón [11] realiza un estudio de varios documentos relacionados con ingeniería de software, con el fin de determinar cuáles pueden ser los elementos que hacen parte de este nombramiento. De esta manera, define una taxonomía de verbos nominalizados, un conjunto de adjetivos candidatos (resultante de los diferentes modos que se definen en la literatura para realizar las acciones) y un conjunto de sustantivos (correspondientes, usualmente, a dimensiones fundamentales de la ingeniería de software o a sus derivados).

La taxonomía de verbos nominalizados incluye los siguientes [11]: análisis, modelado, implementación, prueba, despliegue, mantenimiento, verificación, validación, planificación, conformación, cooperación, coordinación, cierre, recolección, diseño, construcción, liberación, ope-

ración, aprobación, estimación, selección, certificación, control, suspensión, identificación, definición, integración, realización, habilitación, monitoreo, aceptación, organización, contratación, mejoramiento, aseguramiento, retiro, negociación, apoyo, entrenamiento, definición, comunicación, inspección, entrega, especificación, desarrollo, resolución, financiación, guía, valoración, entrenamiento, seguimiento, evaluación, revisión educación y priorización.

El conjunto de adjetivos (y sintagmas nominales o adjetivales funcionando como adjetivos) es el siguiente [11]: ágil, automatizado, autoorganizado, basado en actividades, basado en casos de uso, basado en componentes, basado en escenarios, basado en riesgos, basado en valor, bien organizado, causal, centrado en la arquitectura, colaborativo, compartido, concurrente, conjunto, continuo, cualitativo, cuantitativo, dirigido por pruebas, disciplinado, empírico, escalonado, estructurado, evolutivo, formal, funcional, global, incremental, integrado, interdisciplinario, iterativo, metódico, orientado al producto, orientado por objetos, personalizado, probado, progresivo, prospectivo, repetible, reutilizable, sistemático, sistémico, temprano, con valor compartido y visual.

Los sustantivos son los siguientes [11]: interesados, oportunidad, requisitos, sistema de software, equipo, representante de los interesados, necesidad, ítem de requisitos, elemento del sistema de software y miembro del equipo.

### III. Antecedentes

Algunos autores utilizan explícitamente la palabra práctica y la ejemplifican en el dominio de los requisitos no funcionales. Behutiye *et al.* [4] ejemplifican una práctica que llaman “historias de usuario” en el ámbito de la documentación de requisitos de alto nivel. En este caso, se trata de un producto de trabajo con el que se definen los requisitos, de modo que en la práctica no es clara la operación que se realiza ni la forma en que se hace. Se puede deducir que será algo que se realice sobre los requisitos con este producto de trabajo, pero no hay certeza sobre cuáles serán esas acciones.

Behutiye *et al.* [4] también emplean descripciones completas de funcionamiento para nombrar prácticas: “los requisitos no funcionales se documentan en épicas, características e historias de usuario, criterios de aceptación y definiciones de hecho, páginas wiki documentos de word con requisitos funcionales”. Esta práctica se nombra para el dominio de la documentación de los requisitos no funcionales. De forma similar, Mairiza y Zowghi [5] en el dominio del tratamiento de los requisitos no funcionales nombran así una práctica: “comúnmente, los desarrolladores de software no le dan mucha atención a los requisitos no funcionales”. En estos casos, si bien hay bastante información en relación con la

práctica, no se define explícitamente el objeto (porque incluso hay varios sustantivos) y tampoco es clara la manera en que se realiza la práctica (pues no hay adjetivos ni sintagmas nominales o adjetivales con función de adjetivo en las descripciones).

Kopczyńska *et al.* [6] en el dominio de la educación de requisitos utilizan para nombrar una buena práctica: “la restricción de tiempo (una hora y media con una hora para la educación)”. En este caso, aparecen dos verbos nominalizados pero ningún objeto, por lo cual es difícil saber de qué se trata la práctica; además, el adjetivo tampoco está explícito, lo que impide saber cómo se realiza la acción.

Mirakhorli y Cleland-Huang [7] emplean “método de análisis de intercambio arquitectónico” como nombre para una práctica en el dominio de la creación y utilización de enlaces de trazabilidad de requisitos no funcionales. El problema con este nombre radica en el hecho de que los métodos usualmente son colecciones de varias prácticas, lo que no permite establecer de forma unívoca cuál sería la práctica que se está definiendo. Además, si bien se incluye el adjetivo “arquitectónico”, ese adjetivo parece ser aplicado al sustantivo “intercambio”, es decir, no expresa una forma de realizar la práctica sino un tipo de intercambio que se realiza.

En la literatura también se emplean nombres de práctica como “documentación de requisitos no funcionales” [4], “especificación de requisitos no funcionales” [6], “gestión de conocimiento arquitectónico” [7], “evaluación de la calidad de los requisitos no funcionales” [8], “educación de requisitos no funcionales” [9] y “confianza en conocimiento tácito” [10]. En todos estos casos se tiene, efectivamente, un verbo nominalizado y un objeto, que puede ser un sustantivo (“conocimiento”) o un sintagma nominal (“calidad de los requisitos”). Además, se tiene en algunos casos un adjetivo (“arquitectónico”) o un sintagma adjetival funcionando como adjetivo (“no funcionales”), pero nuevamente se califica al objeto y no al verbo nominalizado para expresar la manera de realizar la práctica.

Ahora, Barn [9] utiliza el nombre de práctica “ingeniería de requisitos orientada por metas”, que es el nombre de un método de ingeniería de software. En este caso sí aparece un sintagma nominal “orientada por metas” con función de adjetivo, pero no hay un verbo nominalizado que exprese acción.

Como se aprecia a partir de la revisión de la literatura, no se puede generar un consenso sobre el nombramiento de las prácticas relacionadas con requisitos no funcionales, porque los autores emplean siempre elementos diferentes para ese nombramiento (productos de trabajo, descripciones completas de funcionamiento, restricciones, nombres de métodos y conjuntos de verbos nominalizados y sintagmas nominales). Además, a pesar de que en ocasiones los nombres son muy de-

tallados, no se puede garantizar que se cumplan los diferentes elementos del nombramiento de las prácticas en ingeniería de software que propone Barón [11] y que suministran un punto de partida para una especificación más detallada de cada una de las prácticas.

#### IV. Nombramiento de las prácticas relacionadas con requisitos no funcionales

Como una manera de solucionar los problemas de la revisión de la literatura, en este artículo se propone el nombramiento de un conjunto de prácticas que se asocian con los requisitos no funcionales. Para ello, el primer paso es la determinación de los verbos nominalizados de los cuales se dispone en la literatura especializada. Además de los autores consultados para la revisión, se toman algunas propuestas en las cuales, si bien no se menciona explícitamente la palabra “práctica” ni se define el nombramiento y ejemplificación de prácticas, sí se detecta que puede haber presencia de verbos nominalizados en su contenido [12]–[35]. Posteriormente, se buscan los verbos nominalizados que usualmente acompañan el sintagma nominal “requisitos no funcionales” y se anota la presencia de cada verbo como se muestra en la Tabla 1. En dicha tabla se aprecian los autores en las columnas y los verbos nominalizados en las filas. En la intersección entre filas y columnas se coloca una X cuando hay presencia del verbo nominalizado en el trabajo que se anota. No se considera la cantidad de veces que se expresa un verbo nominalizado en alguna de las propuestas, pues con una mención que se haga del verbo ya es suficiente para comprobar el uso. En la columna final se totalizan los resultados.

Nótese que el valor máximo –presencia de los verbos “modelado”, “especificación” y “tratamiento”, con 15 presencias entre 31 propuestas– no es mayoritaria, pues cada una de las propuestas tiene un énfasis específico en el tratamiento de los requisitos no funcionales de la siguiente manera:

- Behutiye *et al.* [4] y Robiolo *et al.* [10] se centran en documentación.
- Mirakhorli y Cleland-Huang [7] se ocupan de asuntos relacionados con la trazabilidad.
- Montoya [8] se centra en la valoración y evaluación de la calidad de los requisitos no funcionales y para ello se vale de modelos.
- Aburub *et al.* [9], Chouali *et al.* [20] y Gnaho *et al.* [23] realizan mayoritariamente modelado.
- Buglione *et al.* [18] y Kassab *et al.* [27] se ocupan de la medición.
- Chopra *et al.* [19] y Dabbagh *et al.* [22] hacen énfasis en priorización.
- Cleland-Huang *et al.* [21] y Hussain *et al.* [25] trabajan en clasificación.

- Matsumoto *et al.* [29] se ocupan de asuntos relacionados con la verificación.
- Riojas *et al.* [31] y Zou *et al.* [35] se enfocan en asuntos de comprensión.
- Shah y Patel [32] se centran en especificación.
- Song *et al.* [33] trabajan en trazabilidad.

En la Tabla 1 basta con la mención de los diferentes verbos nominalizados para determinar su presencia en cada una de las propuestas, pero el énfasis de cada propuesta es el que permite determinar a cuál de las prácticas de requisitos no funcionales se refieren. De los verbos nominalizados que se identifican en la Tabla 1, 15 se encuentran también en el listado de Barón [11] y uno adicional se encuentra con una variación. Barón [11] tiene el verbo nominalizado “operación” y se identificó en la literatura “operacionalización”. Habría que profundizar también en la diferencia entre “modelado” y “representación”, pues parecen ser similares en la literatura, lo cual explicaría por qué Barón [11] solo considera “modelado”. Los verbos nominalizados restantes bien podrían incrementar la base de Barón [11], pues el tema de los requisitos no funcionales también hace parte de la ingeniería de software.

En relación con el adjetivo (o sintagma nominal o adjetival con función de adjetivo para calificar el verbo nominalizado), es necesario hacer una revisión caso por caso para realizar el nombramiento definitivo de la práctica a la que se refiere la propuesta. A continuación, se discuten algunos de esos casos.

La denominación “historias de usuario” para una práctica que abordan Behutiye *et al.* [4] no es adecuada a la luz de la estructura de Barón. Sin embargo, el hecho de que trabajen en el dominio de la documentación de requisitos de alto nivel deja claro que, en lugar de ser un dominio, ahí hay más información sobre la orientación de la práctica. Las historias de usuario se usan para ese tipo de documentación, por lo que el nombramiento definitivo de la práctica debería ser “documentación de requisitos de alto nivel basada en historias de usuario”, siguiendo la estructura de algunos de los sintagmas nominales que define Barón [11] para que funcione como adjetivo. Ahora, al revisar la otra práctica que Behutiye *et al.* [4] denominan inicialmente “los requisitos no funcionales se documentan en épicas, características e historias de usuario, criterios de aceptación y definiciones de hecho, páginas wiki documentos de word con requisitos funcionales”, parece claro que también se trata de una práctica de documentación de requisitos no funcionales, pero esta vez aparece un conjunto de productos de trabajo que se fundamentan en descripciones textuales. Por ello, ambas prácticas se pueden agrupar en una práctica cuyo nombramiento final es “documentación de requisitos de alto nivel basada en descripciones textuales”.

**Tabla 1.** Presencia de verbos nominalizados en diferentes propuestas que se refieren a requisitos no funcionales. Fuente: elaboración propia.

Verbo normalizado	Autores																																		
	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]	[32]	[33]	[34]	[35]	Total			
Modelado	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	15	
Especialización	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	15	
Tratamiento	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	15	
Identificación	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	14	
Educación	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	14	
Clasificación	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	12	
Análisis	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	10	
Consideración	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	10	
Gestión	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	10	
Satisfacción	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	9	
Representación	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	9	
Evaluación	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	8	
Verificación	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	7	
Comprensión	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	7	
Documentación	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	6	
Trazabilidad	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	6	
Implementación	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	5	
Priorización	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	5	
Integración	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	5	
Medición	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	4	
Mantenimiento	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	2	
Operacionalización	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	2	
Validación	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	2	
Detección	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	
Valoración	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	
Realización	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	
Prueba	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	
Uso	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	
Manejo	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	
Formulación	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	
Garantía	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	

En el caso de la práctica que Mairiza y Zowghi [5] denominan “comúnmente, los desarrolladores de software no le dan mucha atención a los requisitos no funcionales”, no hay manera de ligar el nombre con alguno de los verbos nominalizados que se identifican en la Tabla 1. Además, en la redacción misma se nota más una alusión a un problema que a una operación repetible que entrega buenos resultados.

Ahora, para la práctica que Kopczyńska *et al.* [6], en el dominio de la educación de requisitos, denominan “la restricción de tiempo (una hora y media con una hora para la educación)”, se menciona directamente el verbo nominalizado “educación” al interior de la restricción y se menciona también el tiempo que se asigna al proceso durante un taller en la industria. Si bien esto indicaría un adjetivo como “temporizado”, otros adjetivos que propone Barón [11] como metódico o bien organizado se adaptan a este tipo de práctica y son más generales para describir la idea. En este caso, el nombramiento propuesto es “educación metódica de requisitos”. La otra práctica que Kopczyńska *et al.* [6] nombran “especificación de requisitos no funcionales” tiene dos de los elementos que define Barón [11] como necesarios para el nombramiento de la práctica: el verbo nominalizado y el sintagma nominal sobre el que se aplica. Con el fin de determinar el adjetivo (o sintagma nominal con función de adjetivo) se requiere explorar más la documentación para establecer con qué mecanismo se realiza la especificación o a qué paradigma se refiere. Según Kopczyńska *et al.* [6], “uno de los enfoques que apuntan a mejorar la práctica de la especificación de requisitos no funcionales es el uso de plantillas” y más adelante emplean el sintagma nominal “basado en plantillas” para referirse específicamente a este enfoque. Así, el nombramiento para esta práctica se define como “especificación de requisitos no funcionales basada en plantillas”.

En cuanto a la práctica que Mirakhorli y Cleland-Huang [7] nombran como “método de análisis de intercambio arquitectónico”, al revisar la documentación al respecto se encuentra que, efectivamente, es el nombre de un método y que tiene el siguiente proceso:

- Se usa un árbol de utilidad para generar escenarios concretos de calidad, que luego se transforman en requisitos no funcionales específicos.
- El sistema se descompone en nodos intermedios que representan objetivos de calidad, como confiabilidad, rendimiento y seguridad y los nodos hoja de bajo nivel representan escenarios de calidad.
- Se documentan implícitamente las relaciones de calidad entre escenarios de calidad y los elementos arquitectónicos en que se realizan.
- Cada requisito no funcional se mapea a su correspondiente decisión arquitectónica.

De la descripción se pueden deducir al menos dos prácticas denominadas “documentación de elementos arquitectónicos basada en escenarios” y “mapeo sistemático de requisitos no funcionales”.

La práctica que Mirakhorli y Cleland-Huang [7] denominan “gestión de conocimiento arquitectónico” es empleada dentro del dominio de un conjunto de herramientas que posibilitan esa gestión. Por ello, se selecciona el adjetivo que menciona Barón [11] para esos casos. El nombramiento definitivo de la práctica es “gestión automatizada de conocimiento arquitectónico”.

Montoya [8] emplea el núcleo de la Esencia de *Semat* [1] para representar gráficamente una práctica que denomina “evaluación de la calidad de los requisitos no funcionales”. En este caso, el verbo nominalizado es “evaluación” y el sintagma nominal es “calidad de los requisitos no funcionales”. Aunque el sintagma nominal es extenso, no se suministra una idea clara de cómo se realiza la evaluación, por lo cual se requiere un adjetivo (o un sintagma nominal con funciones de adjetivo) para nombrar la práctica. Al revisar la documentación, Montoya [8] basa sus prácticas en la representación del núcleo de la Esencia de *Semat* y en esquemas preconceptuales (ambos son modelos). Por ello, se selecciona nuevamente una de las estructuras que propone Barón [11] y se establece el nombramiento definitivo de la práctica como evaluación de la calidad de los requisitos no funcionales basada en modelos.

Barn [9] literalmente establece que la práctica “educación de requisitos no funcionales” se realiza “sistemáticamente”, por lo cual el nombramiento de esa práctica se establece como “educación sistemática de requisitos no funcionales”.

Finalmente, en el caso de la práctica “confianza en el conocimiento tácito” [10], si bien tiene dos de los elementos que reconoce Barón [11] para el nombramiento de la práctica, pues “confianza” es uno de los verbos nominalizados que se reconocen en la Tabla 1 y “conocimiento tácito” es un sintagma nominal sobre el que se aplica el verbo nominalizado, no existe en la documentación mayor información sobre esa práctica en particular. Sin embargo, indagando más al respecto, Brockmann y Anthony [36] establecen que, en general, se puede predecir que una alta confianza en el conocimiento tácito se correlaciona positivamente con el nivel de experiencia que un tomador de decisiones tiene en un campo específico. Por ello, un adjetivo como “experimentada” o “con base en la experiencia” es más adecuado para esta práctica. Por ello, el nombramiento definitivo es “confianza en conocimiento tácito con base en la experiencia”.

El conjunto definitivo de las prácticas con su nombramiento revisado se propone en la Tabla 2.

**Tabla 2.** Nombramiento definitivo de las prácticas relacionadas con requisitos no funcionales. Fuente: elaboración propia.

Autor	Nombramiento inicial	Nombramiento definitivo
[4]	Historias de usuario.	Documentación de requisitos de alto nivel basada en historias de usuario.
[4]	Los requisitos no funcionales se documentan en épicas, características e historias de usuario, criterios de aceptación y definiciones de hecho, páginas wiki documentos de word con requisitos funcionales.	Documentación de requisitos de alto nivel basada en descripciones textuales.
[5]	Comúnmente, los desarrolladores de software no le dan mucha atención a los requisitos no funcionales.	Nota: este enunciado no constituye una práctica.
[6]	La restricción de tiempo (una hora y media con una hora para la educación)	Educación metódica de requisitos
[6]	Especificación de requisitos no funcionales	Especificación de requisitos no funcionales basada en plantillas
[7]	Método de análisis de intercambio arquitectónico	<ul style="list-style-type: none"><li>• Documentación de elementos arquitectónicos basada en escenarios.</li><li>• Mapeo sistemático de requisitos no funcionales.</li></ul>
[7]	Gestión de conocimiento arquitectónico	Gestión automatizada de conocimiento arquitectónico
[8]	Evaluación de la calidad de los requisitos no funcionales	Evaluación de la calidad de los requisitos no funcionales basada en modelos
[9]	Educación de requisitos no funcionales	Educación sistemática de requisitos no funcionales
[10]	Confianza en el conocimiento tácito	Confianza en conocimiento tácito con base en la experiencia

## V. Conclusiones y trabajo futuro

En este artículo se propuso una manera de nombramiento de las prácticas sobre requisitos no funcionales tomando como punto de partida la forma en que los autores lo hacen con respecto a las prácticas y, posteriormente, realizando un análisis sobre la información consignada en la literatura especializada sobre requisitos no funcionales. El análisis se realizó retomando la estructura de nombramiento de prácticas propuesta por Barón [11], que incluye un verbo nominalizado que expresa la operación repetitiva que se realiza con la práctica, un adjetivo (o un sintagma nominal o adjetival con funciones de adjetivo) para expresar el enfoque con el que se realiza la práctica y un sintagma nominal que representa el objeto sobre el que se aplica la práctica.

Bajo esta óptica, se identificaron 31 verbos nominalizados de común ocurrencia para este tipo de requisitos, entre los cuales los más usados fueron: modelado, especificación, tratamiento, identificación, educación, clasificación, análisis, consideración y gestión. Más de la mitad hacía parte de los verbos nominalizados que identificó Barón [11].

Posteriormente, se determinó un conjunto de adjetivos (incluyendo sintagmas nominales y adjetivales) para las prácticas de requisitos no funcionales identificadas en la literatura especializada; el conjunto definitivo fue el siguiente: basado en historias de usuario (*user-story-based*), basado en descripciones textuales (*text-description-based*), metódico (*methodical*), basado en plantillas (*template-based*), basado en escenarios (*scenario-based*), sistemático (*systematic*), automatizado (*au-*

*tomated*), basado en modelos (*model-based*), con base en la experiencia (*experience-based*).

La información sobre cuál adjetivo o sintagma utilizar para el enfoque se basó en la revisión cuidadosa de las referencias, puesto que allí se consigna la manera específica en que se realizan las operaciones.

Dado que los requisitos no funcionales constituyen uno de los temas que hace parte de la ingeniería de software (que es la base para el trabajo de Barón [11]), se puede concluir que los verbos nominalizados adicionales que se identificaron en este artículo pueden complementar la base de verbos nominalizados de Barón [11], ya que conservan la misma estructura y función de dicha base. Igual se puede establecer sobre los adjetivos y sintagmas nominales y adjetivales identificados.

Para el nombramiento de otras prácticas relacionadas con requisitos no funcionales se puede seguir el mismo procedimiento de este artículo.

Las líneas de trabajo futuro que se derivan de esta propuesta son las siguientes:

- Incremento de la base de verbos nominalizados: no se espera que el conjunto de verbos nominalizados que se suministra sea completo, pues pueden surgir más operaciones no incluidas en otras propuestas de investigación. Sin embargo, como en el caso de Barón [11], es posible que ya se esté acotando el número de verbos nominalizados que se usan para el nombramiento de prácticas.
- Elaboración de un inventario de prácticas relativas a los requisitos no funcionales: como se apreció en la revisión, hay mucha confusión con el nombramiento de las prácticas en requisitos no funcionales, pero con la estructura que se suministra en este artículo se puede

proceder a identificar y realizar un nombramiento adecuado para esas prácticas, con el fin de eliminar prácticas repetidas sobre este tema.

- Representación gráfica de las prácticas de requisitos no funcionales: Montoya [8] utiliza el núcleo de la Esencia de *Semat* [1] para identificar y graficar los elementos fundamentales de dos prácticas de requisitos no funcionales. De igual manera se puede proceder con las restantes que se consignan en la Tabla II con su nombramiento adecuado. Lo anterior con el fin de revisar la especificación detallada de cada una de esas prácticas y luego proceder a mapear los diferentes elementos que allí aparezcan en constructos ya definidos para todas las prácticas, como las actividades, las competencias, los roles y las fases. En tal ocasión se incluirían elementos que puedan permitir la comparación, como los alfas y los espacios de actividad, que constituyen el núcleo de la Esencia de *Semat* [1].

## VI. Agradecimientos

Este artículo se financió con recursos provenientes del proyecto de investigación “Modelo para la evaluación de requisitos no funcionales de un sistema de software”, con código Hermes 35194, financiado por Colciencias y la Universidad Nacional de Colombia, sede Medellín.

## Referencias

- [1] Object Management Group, “Essence–Kernel and Language for Software Engineering Methods Version 1.2,” 2018. [Online]. Available: <https://www.omg.org/spec/Essence/1.2/PDF>.
- [2] S. Kemmis, J. Wilkinson, C. Edwards-Groves, I. Hardy, P. Grootenboer, and I. Hardy, *Changing practices, changing education*. Singapur: Springer, 2014.
- [3] I. Sommerville, *Software engineering*, 8th ed. Boston: Pearson Education, 2007.
- [4] W. Behutiye, P. Karhapää, D. Costal, M. Oivo, and X. Franch, “Non-functional requirements documentation in agile software development: Challenges and solution proposal,” *Lecture Notes in Computer Science*, vol. 10611, pp. 515–522, 2017.
- [5] D. Mairiza and D. Zowghi, “Constructing a catalogue of conflicts among non-functional requirements,” in *Evaluation of novel approaches to software engineering ENASE*, 2010, Atenas, Grecia, pp. 31–44.
- [6] S. Kopczyńska, J. Nawrocki, and M. Ochodek, “An empirical study on catalog of non-functional requirement templates: Usefulness and maintenance issues,” *Information and software technology*, vol. 103, pp. 75–91, 2018.
- [7] M. Mirakhorli and J. Cleland-Huang, “Tracing non-functional requirements”. in *Software and systems traceability*, J. Cleland-Huang *et al.* Eds. Londres: Springer-Verlag, 2012.
- [8] Y. Montoya, “Evaluation model for non-functional requirements of a software product,” Tesis de Maestría, Fac. Ing., Universidad Nacional de Colombia, Medellín, 2016.
- [9] B. Barn, “Do you own a Volkswagen? Values as non-functional requirements,” *Lecture Notes in Computer Science*, vol. 9856, pp. 151–162, 2016.
- [10] G. Robiolo, E. Scott, S. Matalonga, and M. Felderer, “Technical debt and waste in non-functional requirements documentation: An exploratory study,” *Lecture Notes in Computer Science*, vol. 11915, pp. 220–235, 2019.
- [11] A. Barón, “Modelo para la definición unificada de la práctica como constructo teórico en ingeniería de software,” Tesis de Doctorado, Fac. Ing., Universidad Nacional de Colombia, Medellín, 2019.
- [12] F. Aburub, M. Odeh, and I. Beeson, “Modelling non-functional requirements of business processes,” *Information and software technology*, vol. 103, pp. 75–91, 2018.
- [13] K. Adams, *Non-functional requirements in systems analysis and design*. New York: Springer, 2015.
- [14] D. Akoumianakis and C. Stephanidis, “Using non-functional requirements as design drivers for universal access,” *Lecture Notes in Computer Science*, vol. 3041, pp. 141–155, 2005.
- [15] D. Ameller, X. Burgués, D. Costal, C. Farré, and X. Franch, “Non-functional requirements in model-driven development of service-oriented architectures,” *Science of computer programming*, vol. 168, pp. 18–37, 2018.
- [16] D. Ameller and X. Franch, “How do software architects consider non-functional requirements: A survey,” *Lecture Notes in Computer Science*, vol. 6182, pp. 276–277, 2010.
- [17] A. Berztiss, “Non-functional requirements in the design of software,” *Lecture Notes in Computer Science*, vol. 750, pp. 375–386, 1994.
- [18] L. Buglione, O. Ormandjieva, and M. Daneva, “Using PSU for early prediction of COSMIC size of functional and non-functional requirements,” *Lecture Notes in Computer Science*, vol. 5338, pp. 352–361, 2008.
- [19] R. Chopra, V. Gupta, and D. Chauhan, “Experimentation on accuracy of non functional requirement prioritization approaches for different complexity projects,” *Perspectives in science*, vol. 8, pp. 79–82, 2016.
- [20] S. Chouali, A. Hammad, and H. Mountassir, “Assembling components using SysML with non-functional requirements,” *Electronic notes in theoretical computer science*, vol. 295, pp. 31–47, 2013.



- [21] J. Cleland-Huang, R. Settini, X. Zou, and P. Solc, "Automated classification of non-functional requirements," *Requirements engineering*, vol. 12, pp. 103–120, 2007.
- [22] M. Dabbagh, S. Lee, and R. Parizi, "Functional and non-functional requirements prioritization: empirical evaluation of IPA, AHP-based, and HAM-based approaches," *Soft computing*, vol. 20, 4497–4520, 2016.
- [23] C. Gnaho, F. Semmak, and R. Laleau, "Modeling the impact of non-functional requirements on functional requirements," *Lecture Notes in Computer Science*, vol. 8697, pp. 59–67, 2014.
- [24] K. Gómez, C. Baron, P. Esteban, C. Gutiérrez, and L. Laredo, "How to find non-functional requirements in system developments," *IFAC PapersOnLine*, vol. 51, no. 11, pp. 1573–1578, 2018.
- [25] I. Hussain, L. Kosseim, and O. Ormandjieva, "Using linguistic knowledge to classify non-functional requirements in SRS documents," *Lecture Notes in Computer Science*, vol. 5039, pp. 287–298, 2008.
- [26] S. Izukura, S. Hosono, H. Sakaki, E. Numata, K. Kimita, and Y. Shimomura, "Bridging non-functional requirements and IT service design," *Procedia CIRP*, vol. 30, pp. 24–29, 2015.
- [27] M. Kassab, O. Ormandjieva, M. Daneva, and A. Abran, "Non-functional requirements size measurement method (NFSM) with COSMIC-FFP," *Lecture Notes in Computer Science*, vol. 4895, pp. 168–182, 2008.
- [28] T. Marew, J. Lee, and D. Bae, "Tactics based approach for integrating non-functional requirements in object-oriented analysis and design," *The journal of systems and software*, vol. 82, pp. 1642–1656, 2009.
- [29] Y. Matsumoto, S. Shirai, and A. Ohnishi, "A method for verifying non-functional requirements," *Procedia computer science*, vol. 112, pp. 157–166, 2017.
- [30] E. Poort, N. Martens, I. VandeWeerd, and H. Van Vliet, "How architects see non-functional requirements: Beware of modifiability," *Lecture Notes in Computer Science*, vol. 7195, pp. 37–51, 2012.
- [31] M. Riojas, S. Lysecky and J. Rozenblit, "Understanding non-functional requirements for pre-college engineering technologies". In *Innovative technologies in management and science*, R. Klempos and J. Nikodem Eds. Londres: Springer-Verlag, 2015.
- [32] T. Shah and S. Patel, "A novel approach for specifying functional and non-functional requirements using RDS (Requirement Description Schema)," *Procedia computer science*, vol. 79, pp. 852–860, 2016.
- [33] S. Song, Y. Kim, S. Park and S. Park, "A non-functional requirements traceability management method based on architectural patterns," In *Computers, networks, systems, & industrial engineering*, R. Lee, Ed. Berlin: Springer-Verlag, 2015.
- [34] C. Theaker and J. Whitworth, "Designing with non-functional requirements," In *Software quality and productivity*, M. Lee *et al.* Eds. Dordrecht: Springer, 1995.
- [35] J. Zou, L. Xu, M. Yang, X. Zhang, and D. Yang, "Towards comprehending the non-functional requirements through developers' eyes: An exploration of stack overflow using topic analysis," *Procedia computer science*, vol. 112, pp. 157–166, 2017.
- [36] E. Brockmann and W. Anthony, "The influence of tacit knowledge and collective mind on strategic planning," *Journal of managerial issues*, vol. 10, no. 2, pp. 204–222, 1998.