



Metodología para elaboración de requerimientos en aplicaciones con servicios web

Methodology for the elaboration of requirements in applications with web services

Patricia Páez Cárdenas¹ Christian David Arias Peralta² Luis Felipe Wanumen Silva³

Para citar este artículo: P. Páez, C. D. Arias y L. F. Wanumen, “Metodología para elaboración de requerimientos en aplicaciones con servicios web”. *Revista Vínculos: Ciencia, Tecnología y Sociedad*, vol 15, n° 2, julio-diciembre 2018, 160-174. DOI: <https://doi.org/10.14483/2322939X.13505>.

Recibido: 16-07-2018 / Aprobado: 28-07-2018

Resumen

Dependiendo de la técnica usada en la captura de requerimientos, se pueden obtener procesos innovadores e iterativos que verdaderamente sirvan de soporte a la construcción de aplicaciones que consumen servicios. Las metodologías ágiles sugieren la elicitación de requerimientos altamente cambiantes, mientras que las metodologías tradicionales sugieren detallar al máximo los requerimientos luego de hacer un proceso de priorización. En el presente artículo se analizan los enfoques de elicitación de las metodologías ágiles y tradicionales y se toman los aspectos de cada una de ellas que guardan concordancia con las características de las aplicaciones que consumen servicios, lo anterior con el fin de proponer un marco de trabajo para la elicitación de requerimientos respecto al caso específico de aplicaciones que utilizan servicios web. Al finalizar el artículo se realiza una validación en dos empresas que prestan servicios de desarrollo de *software*.

Palabras clave: análisis de requerimientos, desarrollo de *software* ágil, servicios web.

Abstract

Depending on the technique used in the capture of requirements, it is possible to obtain innovative and iterative processes that truly support the construction of applications that consume web services. In the present article the elicitation approaches of the agile and traditional software methodologies are analyzed and the aspects of the rules that guarantee the concordance with the characteristics of the applications that consume services are taken, with the purpose of proposing a framework of work for the elicitation of the information applied to the specific case of applications that use web services. At the end of the article, a validation will be carried out in two companies that provide software development services.

Keywords: agile software development, requirements analysis, web services.

1. Analista de desarrollo, Everis. Correo electrónico: duck1girl@gmail.com
2. Analista de requerimiento, GlobalHitss. Correo electrónico: christianariasperalta@gmail.com
3. Magíster Sistemas y Computación, Pontificia Universidad Javeriana. Correo electrónico: lwanumen@udistrital.edu.co

1. Introducción

El proceso de elicitación de requerimientos es la primera etapa de la ingeniería de requisitos donde se abstrae la comprensión del problema que se desea resolver con el producto de *software*, buscando así determinar y documentar los requisitos; constituye un proceso complejo en el que se presentan no solo problemas esenciales, sino accidentales [1], entre los que se pueden mencionar problemas de alcance, comprensión y volatilidad. La elicitación de requerimientos permite a los integrantes de un equipo de trabajo comprender los aspectos particulares del *software*, a fin de asegurar que su desarrollo posterior guarde relación y coherencia con las especificaciones iniciales del proyecto [2]. Existen muchas alternativas para la elicitación de requerimientos, sin embargo, algunas toman en cuenta el contexto en el que se envuelve el sistema y otras toman en cuenta el origen de los requerimientos; así, es importante por tanto proponer un *framework* [3] que permita recoger lo mejor de los enfoques de metodologías ágiles y tradicionales, adaptado a sistemas que consumen servicios.

Para realizar el proceso de elicitación se tomarán en cuenta los requerimientos no funcionales en términos de costo, valor, riesgo, tiempo de comercialización y calidad, además se verá cómo estos afectan directamente el desempeño de los requerimientos funcionales. El objetivo de la mayoría de las teorías de requerimientos es apoyar la construcción e implementación de requerimientos funcionales, sin embargo, a través de este artículo se enfatiza en cómo la forma de implementación de los requerimientos no funcionales afecta positiva o negativamente la consecución de los requerimientos funcionales.

La elicitación de requerimientos es un término que surge para revolucionar la forma tradicional en que se hace captura de requerimientos, dado que constituye el componente básico en el desarrollo de un proyecto de *software* y tiene alto impacto en las fases posteriores, pues que debe mitigar los problemas de definición, de alcance, de comprensión de la necesidad por parte del cliente y de cambios en

los requisitos durante el desarrollo del proyecto, de tal forma que con la elicitación se toman en cuenta aspectos cambiantes que provocan el dinamismo en la propuesta de requerimientos. En últimas, se está afirmando que hay requerimientos volátiles y otros que son levemente cambiantes [4]; en este sentido, la elicitación de requerimientos toma en cuenta la existencia de requerimientos cambiantes, sin embargo, cuando estos requerimientos cambian por razones propias del negocio, por razones de mercadeo o por cuestiones legales que afectan el tiempo de desarrollo de los sistemas, las metodologías de elicitación de requerimientos presentan impedimentos que deben ser solucionados [5].

Aquí es donde resulta imperativo tener en cuenta las necesidades de los involucrados en la especificación de requerimientos a fin de establecer los que sean comunes a la organización y que no entren en conflicto, máxime si se tiene en cuenta que estos requerimientos entrarán a ser implementados en sistemas sobre buses de servicios y deben ser expuestos a toda una organización de forma transversal o incluso expuestos a otras organizaciones.

2. Elicitación de requerimientos

La elicitación de requerimientos se puede hacer con un enfoque hacia las metodologías ágiles y con un enfoque hacia las metodologías tradicionales. En esta sección se muestran los dos posibles enfoques para realizar la elicitación de requerimientos.

A. Enfocado a metodologías ágiles

Analizando la literatura sobre metodologías ágiles, se puede apreciar la forma en que estas implementan fases de captura de requerimientos que distan un poco del mismo proceso sobre metodologías tradicionales, en el sentido de que se enfatiza en la elicitación de los aspectos más relevantes a nivel de negocio en las primeras etapas y de los aspectos más dependientes de la plataforma en las últimas etapas. Obviamente la evolución de las metodologías

ágiles altera y modifica la forma en que se realiza la captura de requerimientos [6], la elicitación de requerimientos enfocada a metodologías ágiles está en crecimiento y cada vez se esperan teóricos que aterricen aspectos tenidos en cuenta cuando surgen cambios de paradigmas en los desarrollos de *software*; en este sentido, uno de estos paradigmas es el desarrollo orientado a servicios, el cual ha revolucionado la forma en la que se elicitan requerimientos y exige que la elicitación tenga en cuenta no solo a los usuarios de la aplicación, sino a los usuarios que son afectados indirectamente por el desarrollo de tales sistemas. La Tabla 1 expone los diferentes enfoques producto del análisis realizado.

B. Enfocado a metodologías tradicionales

En las metodologías tradicionales es necesario encauzar la elicitación de requerimientos hacia un entendimiento del entorno de los procesos de la organización, lo anterior con el fin de abstraer las necesidades principales que garanticen la construcción de un producto con calidad [13]. En la Tabla 2 se exponen algunos aspectos de la elicitación de requisitos en metodologías tradicionales.

3. Propuesta de framework desarrollada con la metodología

Una vez verificadas las ventajas y desventajas de la elicitación de requerimientos en las metodologías ágiles y tradicionales, se propone una metodología iterativa e incremental para la ingeniería de requerimientos con la cual es posible proponer mejoras en cada ciclo de *software* ejecutado y, además, se apoya en definiciones de autores mencionados anteriormente, esta se plantea en siete fases: requerimientos blandos, elicitación, análisis, validación semántica, especificación, validación y verificación de documentación.

En la Figura 1 se muestran las fases que tendría la propuesta metodológica, en caso de detectar algún error o fallo en una fase se podrá retornar a la fase anterior para mitigar el error detectado y poder avanzar hasta llegar a la fase de requerimientos blandos, en la cual ya concluyó el análisis de los datos recolectados del requerimiento. Cada fase mencionada anteriormente posee un conjunto de etapas, las cuales se puntualizan en la Figura 2.

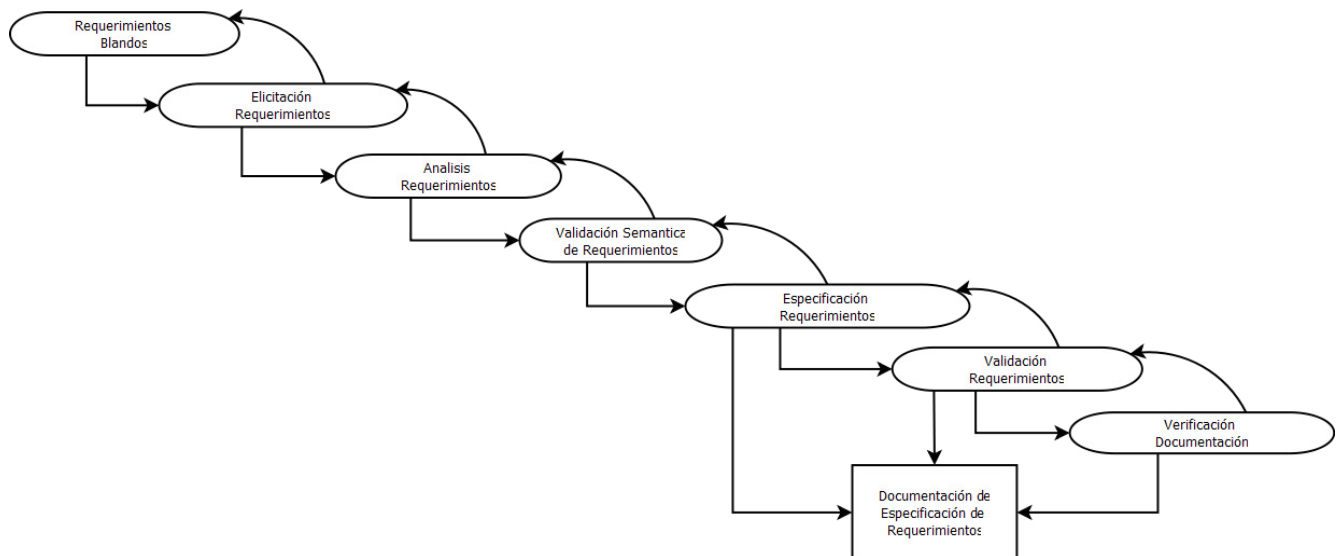


Figura 1. Fases de la metodología propuesta.

Fuente: elaboración propia.

Tabla 1. Enfoque de elicitación de requisitos a metodologías ágiles.

Enfoque de elicitación	Ventajas	Desventajas
Ming Huo: compara el modelo de cascada con procesos ágiles para mostrar cómo estos logran cumplir los requerimientos de calidad de <i>software</i> bajo presión, en un ambiente inestable, y contribuye a obtener productos de <i>software</i> con garantía de calidad [7].	Tienen la capacidad de administrar requerimientos altamente cambiantes [7].	Cuando se trata de elicitar en sistemas altamente complejos, los mecanismos usados para el establecimiento de los requerimientos no permiten obtener todos los detalles necesarios y el nivel de granularidad de la elicitación no tiene el detalle deseado para tales sistemas.
Lan Cao: la evolución del entorno empresarial en el que operan las organizaciones pone en evidencia entornos cambiantes en cuanto a preferencias, tecnologías y presiones de tiempo, factores que vuelven inapropiados los requisitos previamente levantados [8].	Presenta un enfoque pragmático.	A la hora de elicitar requerimientos con alto detalle, y haciendo uso de metodologías ágiles, puede quedar corta la comunicación entre los analistas de requerimientos y los usuarios.
Amin: centra su interés en obtener pautas de ingeniería de requisitos de métodos ágiles que pueden ser considerados para el área de calidad [9].	La elicitación de requerimientos proporciona una ventaja competitiva en términos de calidad a las metodologías que hagan uso de esta técnica de requerimientos en aplicaciones que consumen servicios. La ER (elicitación de requerimientos) provee capacidad de verificación y validación a los sistemas <i>software</i> , siendo útil en actividades adicionales que pueden ser generadas en la gestión del cambio. La ER mejora la especificación de requerimientos no funcionales.	La elicitación de requerimientos requiere extraer las ideas principales para las distintas funcionalidades del sistema si se pretende que la ER efectivamente incremente la calidad en las metodologías ágiles. Si bien la ER ha sido probada en contextos como sistemas de gestión de cadena de suministros, es necesario hacer una incorporación de estas técnicas de captura y refinamiento de requerimientos en otros contextos antes de comprobar que globalmente es adaptable en cualquier contexto.
Deepti Mishra: desarrolla un enfoque de costo-valor mediante el cual clasifica los requisitos de mayor prioridad e identifica las limitaciones que pueden presentar en tiempo y presupuesto [10].	Funciona exitosamente en proyectos complejos como lo son las aplicaciones de <i>software</i> que consumen servicios. Ha sido implementado en organizaciones de mediano tamaño logrando un éxito importante y fortaleciendo la relación con el usuario final.	Cuando el equipo de ingeniería no tiene experiencia en la implantación de sistemas basados en elicitación de requerimientos, tiene muchos inconvenientes al optimizar y afinar el método usado en la elicitación y validarlo al interior de un proyecto de un sistema <i>software</i> .
Franek <i>et al.</i> : método de priorización de requisitos llamado <i>ranking</i> , basado en casos, el cual combina las preferencias de los interesados del proyecto con los requisitos que ordenan las aproximaciones calculadas a través de técnicas de aprendizaje automático [11].	Es incluyente con los procesos de análisis, validación y verificación de requerimientos, dando una importancia igual o mayor que en la implementación de metodologías tradicionales. La ER no solo permite tener un requerimiento, sino que permite analizar y validar la coherencia del requerimiento en el contexto del dominio donde se implementa un sistema <i>software</i> .	La documentación del proceso de elicitación de requerimientos puede ser escasa. Las técnicas al interior del proceso de ER son muy diversas y esto en ocasiones ocasiona que se descuiden los objetivos de un sistema <i>software</i> por analizar qué tipo de técnica se implementará para afinar el requerimiento.
Vithana: analiza la adaptabilidad de las empresas a las metodologías ágiles en el ámbito de la ingeniería de requisitos, demostrando que la labor es más efectiva cuando se logra un alto involucramiento del cliente en el proyecto [12].	Es usado ampliamente en proyectos que utilizan metodologías ágiles, dado que hace hincapié en la ingeniería de requisitos iterativos mediante las herramientas fundamentales de elicitación de requisitos como lo son la comunicación entre las partes interesadas, gestión de actualización y priorización de requisitos.	La gestión de elicitación de requisitos mediante métodos cualitativos en proyectos que desarrollan aplicaciones para el consumo de servicios web, funciona de manera más favorable en proveedores de <i>software offshore</i> . Requiere fortalecer el uso de técnicas tradicionales de elicitación de requisitos entre las que se encuentran creación de prototipos, desarrollo impulsado por prueba e interacción.

Fuente: elaboración propia.

Tabla 2. Enfoque de elicitación de requisitos a metodologías tradicionales.

Enfoque de elicitación	Ventajas	Desventajas
<p>Anna Perini <i>et al.</i>: Cuando no se realiza una priorización de requisitos asertiva, genera una toma de decisiones incorrecta y el producto final no cumplirá las expectativas, por esto, el autor analiza y compara entre una técnica nueva y una técnica híbrida de votación acumulativa jerárquica y <i>Fuzzy Expert Systemal</i> [14].</p>	<p>Permiten la priorización de requerimientos usando métodos de aprendizaje de máquina y, en este sentido, se toman en cuenta muchas variables para priorizar requerimientos.</p>	<p>El problema de este enfoque está dado por la ausencia de una variable de alto impacto antes de plantear el modelo de simulación de priorización de requerimientos.</p>
<p>Rahul Thakurta: propone un modelo de métricas de calidad que considera los atributos de calidad deseados en ISO / IEC en los primeros requisitos para que el producto cumpla con los criterios propuestos [15].</p>	<p>Es un método bastante interesante para priorizar requerimientos no funcionales basados en una lista de priorización.</p>	<p>El problema de este método consiste en el nivel de valor que le dan las personas a cada requerimiento en el sentido de que la priorización la hacen de forma subjetiva personas que muchas veces no tienen claros los impactos de un requerimiento no funcional sobre el sistema.</p>
<p>Naila Sharif <i>et al.</i>: se propuso una metodología que ayudara a elicitar y validar la obtención de requisitos usando técnicas de priorización, la cual busca ser una herramienta de ayuda al usuario para lograr un proceso de elicitación altamente efectivo [16].</p>	<p>La técnica para elicitar requerimientos propuesta por Sharif, posee las ventajas de la inteligencia computacional y de la ingeniería de <i>software</i>, en este sentido, une estos dos saberes para proveer una técnica de priorización de gran calidad. Entre las ventajas de usar lógica difusa en la elicitación de requerimientos se encuentra la posibilidad de incluir un conjunto de reglas que se usan para controlar procesos y, en este caso, el sistema <i>software</i>; así, a partir de estas reglas más adelante se podrían incluso plantear acciones de control para verificar si el requerimiento se cumple y en qué grado.</p> <p>El saber en qué grado se cumple un requerimiento es otra de las ventajas fuertes de incorporar lógica difusa en la elicitación de requerimientos propuesta por Sharif.</p>	<p>En los casos donde la priorización de requerimientos se pueda obtener mediante modelos matemáticos, el uso de la técnica de Sharif muy probablemente arrojaría peores resultados.</p> <p>El problema de la técnica radica en el tiempo que requiere el algoritmo para el aprendizaje, en el sentido de que, si este tiempo no es lo suficientemente grande, es posible que el sistema arroje resultados no coherentes o incluso arroje resultados que entorpezcan el proceso de elicitación de requerimientos en las aplicaciones que consumen servicios.</p>
<p>Nupul Kukreja <i>et al.</i>: el autor plantea una técnica o modelo de priorización de requisitos que contenga factores y priorización de los requisitos, ya sea para un desarrollo de <i>software</i> tradicional o un desarrollo de <i>software</i> ágil [17].</p>	<p>El método propuesto por Kukreja es una buena alternativa cuando se tiene definido un requerimiento ideal, dado que el método busca menor distancia con este.</p> <p>Es muy útil cuando se tiene claro un requerimiento ideal, pero se sabe de antemano que el cumplimiento de este requerimiento ideal no es posible de elicitar debido a alguna restricción de negocio, lo cual obviamente generaría que se conozca de antemano la similitud a nivel de negocio de los diferentes requerimientos con respecto a un requerimiento ideal.</p>	<p>El problema inherente al uso del método de Kukreja radica en la dificultad de encontrar requerimientos ideales base con los cuales generar los otros requerimientos. En esta dirección, si los requerimientos iniciales base están mal seleccionados, entonces el resto de requerimientos derivados calculados con este método van a estar desviados de los requerimientos ideales.</p>
<p>Rubaida Easmin <i>et al.</i>: plantea un esquema de priorización de requisitos que crea inicialmente órdenes parciales a partir de los aportes de las partes interesadas y luego las utiliza en la implementación de una técnica de búsqueda binaria [18].</p>	<p>Toma en cuenta las retroalimentaciones de cada una de las personas involucradas en el planteamiento de requerimientos.</p>	<p>El problema fundamental de este método cuando se aplica a la elaboración de requerimientos en aplicaciones que consumen servicios, se presenta cuando los profesionales de las tecnologías de la información no cuentan con experiencia en implementación de requerimientos sobre sistemas basados en arquitecturas orientadas a servicios; en este sentido, las retroalimentaciones que proveen durante el proceso de elicitación no son fuertes y obviamente harían que el proceso de establecimiento de requerimientos no provea su mayor efectividad por estar basado este método en las retroalimentaciones de las personas que han implementado dichos requerimientos.</p>

Fuente: elaboración propia.

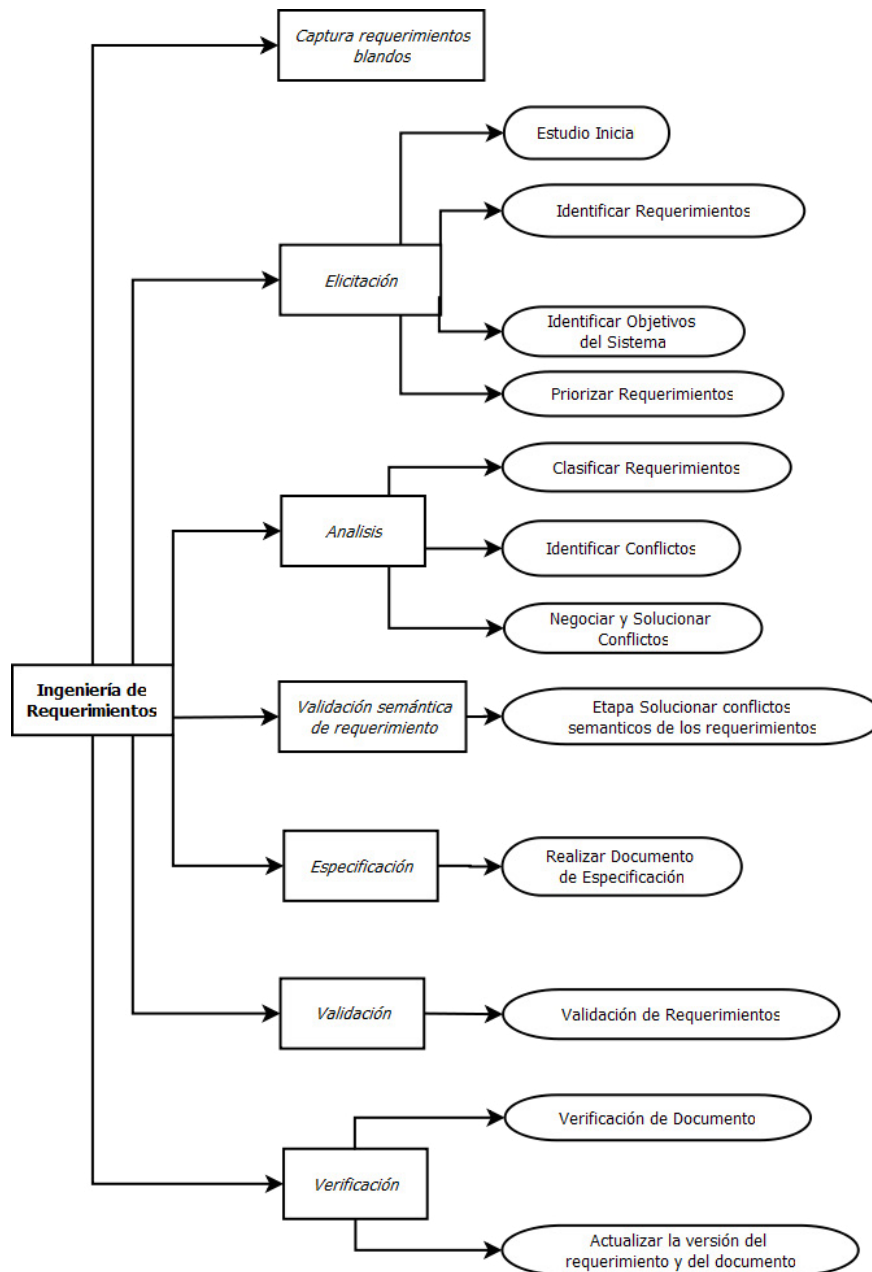


Figura 2. Metodología propuesta.

Fuente: elaboración propia.

3.1. Fase de captura de requerimientos blandos

Los requerimientos blandos son importantes para comprender las metas internas de los diferentes actores y el conocerlos permite explorar las intenciones que hay detrás de los requerimientos formales [19]. La anterior razón motivó la propuesta de esta fase a

la que se llamó “captura de requerimientos blandos”, esta captura debe tener en cuenta las facilidades de desarrollo, el tiempo estimado, la reutilización de las interfaces de programación de aplicaciones (API, por sus siglas en inglés), la conexión del servicio, el soporte al servicio entre otros factores. En la Tabla 3 se muestran algunas metas blandas perseguidas

por los desarrolladores contra las metas blandas perseguidas por el usuario final; se puede apreciar cómo hay una rivalidad en los requerimientos que cada uno como actor del sistema propone.

3.2. Fase de elicitación

En esta fase se tiene como objetivo principal presentar los procesos de la organización para identificar las necesidades de los involucrados del sistema a desarrollar de alguna fuente de información brindada por el cliente, entre las que se encuentran entrevistas, observación de procesos, cuestionarios, lluvias de ideas y talleres.

3.2.1. Etapa 1: realizar el estudio inicial

En el *framework* propuesto se propone un estudio al que se llama estudio inicial, el cual intenta manifestar cuál es el problema que se quiere resolver con el requerimiento para así poder identificar los términos del sistema que se va a desarrollar. El insumo es conocer el modelo principal del negocio y un diccionario de términos utilizados, es importante que el equipo comprenda la necesidad real de la

organización, así se evita realizar una recolección de requerimientos imprecisa y será posible brindar las estimaciones correctas; adicional a ello, se detectarán los diferentes usuarios, características y los roles que cada uno de estos desempeña dentro de la organización. Para realizar esta etapa es recomendable empezar las entrevistas o consultas con los líderes funcionales, quienes poseen el dominio del problema y cuentan con una visión de los procesos, para así poder continuar con los usuarios finales que son los que ayudarán con información detallada del entorno organizacional. En la Tabla 4 se enumeran los recursos que pueden ser utilizados en este proceso.

3.2.2. Etapa 2: identificar los objetivos del sistema

A continuación se procede a conocer los objetivos de negocio que tiene la organización y las necesidades para poder lograrlos, estas necesidades deberán ser documentadas en las historias de usuario, de manera que expresen lo que se espera que haga la aplicación, así como los resultados que se esperan lograr a corto, mediano y largo plazo. Las herramientas que ayudarán en esta etapa se indican en la Tabla 5.

Tabla 3. Metas blandas perseguidas.

Metas blandas perseguidas por los desarrolladores	Metas blandas perseguidas por el usuario final
El servicio desarrollado sea lo más fácil de implementar.	El servicio consumido funcione con la mínima cantidad de parámetros.
El tiempo estimado para el desarrollo del servicio sea lo más amplio posible.	El tiempo estimado para el desarrollo del servicio sea lo más pronto posible.
El servicio se pueda desarrollar basado en la mayor cantidad de API posibles, lo cual añade un problema de costos.	El servicio sea lo más barato posible y no le interesa si el <i>software</i> se hace o no basado en API existentes.
La conexión al servicio corra por cuenta del cliente.	El servicio funcione con alta disponibilidad y el cliente no tenga responsabilidades por el mal uso que se haga del servicio.
Los desarrolladores quieren que después de la entrega del servicio las adecuaciones, las mejoras y los controles de cambio los haga el cliente.	El usuario final quiere que se le brinde soporte posventa, con alta disponibilidad y soporte a los controles de cambio.
Si surgen cambios de estándares en el ambiente, estos cambios se puedan hacer siempre y cuando existan otros contratos y estos deben ser asumidos por el cliente.	El cliente quiere que el servicio sea actualizado por parte del desarrollador, así el primero no haya visualizado los posibles cambios tecnológicos que pudieran presentarse.

Fuente: elaboración propia.

Tabla 4. Técnicas/herramientas para el desarrollo del estudio inicial.

Técnicas/herramientas	Entradas	Salidas
<ul style="list-style-type: none"> • Investigación bibliográfica para tener visión del negocio. • Técnicas de Elicitación: entrevistas, cuestionarios, etc. • Plantillas como la de datos de los participantes. 	Información recolectada.	<ul style="list-style-type: none"> • Introducción del problema. • Participantes: cliente, desarrolladores, usuarios del sistema.

Fuente: elaboración propia.

Tabla 5. Técnicas/herramientas para identificar los objetivos del sistema.

Técnicas/herramientas	Entradas	Salidas
<ul style="list-style-type: none"> • Técnicas de elicitación: entrevistas, cuestionarios, lluvia de ideas, talleres. • Uso de plantillas para objetivos. 	Información recolectada.	<ul style="list-style-type: none"> • Objetivos del sistema. • Requerimientos. • Restricciones. • Alcance del proyecto. • Participantes del proyecto.

Fuente: elaboración propia.

3.2.3. Etapa 3: identificar requerimientos

Con la información adquirida en las etapas anteriores, se avanza a identificar los requerimientos de la aplicación mediante la utilización de técnicas indicadas en la Tabla 6, a través de las cuales el analista de requerimientos construirá los requerimientos de la aplicación siguiendo de las necesidades que exprese el usuario. Es posible también facilitar la identificación de requerimientos expresando los objetivos en conjuntos de características de la aplicación.

3.2.4. Etapa 4: priorizar los requerimientos

Esta etapa utiliza las herramientas indicadas en la Tabla 7 para establecer un orden de los requerimientos en función de la necesidad e importancia que le dé el usuario, de esta forma se busca lograr una organización en los procesos involucrados en la construcción del producto de software; a los requerimientos se le asignarán etiquetas de calificaciones, según lo que el usuario exponga: alta, media, baja, o por definir.

3.3. Fase de análisis

3.3.1. Etapa 5: clasificar los requerimientos

Se propone la Tablas 8, la Tabla 9 y la Tabla 10 para abordar la etapa de clasificación, se tendrán en cuenta los requerimientos relacionados con la experiencia usuario, *hardware*, *software* y comunicación, se procederá después con los requerimientos funcionales, donde se precisa que “un requerimiento funcional es aquel que agrupa las operaciones del proceso de la información que realiza la aplicación, tales ítem como: el almacenamiento de la información, generación de informes, estadísticas, operaciones, etc.” [20]. Posterior a ello se manejarán los requerimientos no funcionales, los cuales encierran los atributos del sistema como son: rendimiento, disponibilidad, seguridad, fiabilidad, adaptabilidad, funcionalidad, entrega, implementación, etc.

Finalmente se procederá con los diagramas de casos de uso y la descripción de cada requerimiento.

Tabla 8. Requerimientos de interfaz.

Requerimientos comunes de los interfaces			
Usuario	Hardware	Software	Comunicación

Fuente: elaboración propia.

Tabla 9. Requerimientos funcionales.

Requerimientos funcionales	
RF1	
RF2	
.....	
RF.n	

Fuente: elaboración propia.

Finalmente se procederá con los diagramas de casos de uso y la descripción de cada requerimiento.

3.3.2. Etapa 6: identificación de conflictos

Se identificarán los problemas que existen entre cada uno de los requerimientos de la aplicación a través de los atributos mencionados en la Tabla 11. Cada uno de los requerimientos se registra en la matriz de la Tabla 11 para determinar que cada uno cumpla los ítems establecidos en dicha matriz,

de no ser así se considera que hay un conflicto. Se realizará un inventario de los conflictos encontrados para hallar la solución a cada uno mediante las técnicas indicadas en la Tabla 12.

3.4. Fase de validación semántica de requerimientos

3.4.1. Etapa 7: solucionar conflictos semánticos de los requerimientos

Se realizará un ejercicio de trazabilidad hacia atrás, que consiste en relacionar cada requerimiento con el responsable de este. El analista gestionará el proceso de negociación con los miembros del equipo por cada conflicto encontrado y hallará una solución que satisfaga a los involucrados, en caso de no encontrar una solución, se utilizará la técnica de resolución por decreto como se indica en la Tabla 13 y se recomienda utilizarlo como último recurso. El producto de esta etapa es la creación de un nuevo requerimiento que quedará documentado como solución del requerimiento inicial y permitirá realizar el debido seguimiento a través de un registro histórico en la matriz modelo indicada en la Tabla 14.

Tabla 10. Requerimientos no funcionales.

Requerimientos no funcionales					
Rendimiento	Seguridad	Disponibilidad	Comunicación	Mantenibilidad	Portabilidad

Fuente: elaboración propia.

Tabla 11. Matriz de comprobación de características.

Requerimiento	No ambiguo, completo y correcto	Medible y verificable	Alcanzable y realista	Consistente (no contradictorio)	Que no oculte a otro requerimiento
R1					
R2					
R3					
.....					
Rn					

Fuente: elaboración propia.

Tabla 12. Herramientas para identificar conflictos.

Técnicas/herramientas	Entradas	Salidas
Tabla de comparación de características de requerimientos. Experiencia del analista	Documento con requerimientos clasificados	Documento conflictos

Fuente: elaboración propia.

Tabla 13. Herramientas para solucionar y negociar conflictos.

Técnicas/herramientas	Entrada	Salida
Trazabilidad hacia atrás Resolución por decreto	Tabla de conflictos	Tabla de solución de conflictos o nuevos requerimientos

Fuente: elaboración propia.

Tabla 14. Conflictos y soluciones.

Requerimiento	Motivo del conflicto	Solución

Fuente: elaboración propia.

3.5. Fase de especificación

3.5.1. Etapa 8: realizar documento de especificación

Se procede con la documentación de la especificación de requerimientos de la aplicación, el analista detallará de forma objetiva y clara los requerimientos, incluyendo puntos como funcionalidad, rendimiento, restricciones y atributos que se relacionarán con algunos requerimientos no funcionales registrados y, a su vez, con alguna interfaz externa, si aplica.

3.6. Fase de validación

La finalidad de esta fase es medir la calidad sobre los requerimientos, identificar oportunidades de mejora, conflictos entre estos, omisiones de necesidades del usuario, ambigüedades en la documentación, entre otros.

3.6.1. Etapa 9: validación de requerimientos (funcionales y no funcionales)

Se realiza un análisis exhaustivo que permita asegurar que se cubrieron a cabalidad las necesidades

del usuario y, de no ser así, exponer las falencias que no permitan llegar a tal fin para disiparlas de manera consensuada con el cliente.

Para realizar la validación de requerimiento se pueda realizar una tabla en la cual se almacene la información de los que están listados en la documentación de la especificación del requerimiento, también se puede aplicar la técnica de *checklist*, la cual se puede componer de criterios como:

- Correcta: si la aplicación alcanza cada uno de los requerimientos solicitados.
- No ambiguo: el requerimiento contiene una única interpretación.
- Consistencia: los requerimientos planteados están alineados con la documentación del proyecto.
- Categorización: cada requerimiento debe tener un peso y tamaño de importancia y prioridad en el proyecto basado en el nivel de complejidad.
- Verificable: comprobar que la aplicación cumple con los requerimientos planteados.
- Modificable: cambios en la estructura de los requerimientos pueden aplicarse con facilidad y de forma consistente, manteniendo la estructura.

- Trazable: que el requerimiento permita llevar la documentación de mejoras aplicadas o que se vayan a desarrollar a futuro.

A continuación, la Tabla 15 expone las técnicas que pueden ser utilizadas en esta etapa.

3.7. Fase de verificación

3.7.1. Etapa 10: verificación de documento

La documentación deberá ser legible para todas las partes interesadas, es decir, debe ser claro para todos los roles que intervienen en el desarrollo de proyecto, incluyendo el usuario final.

3.7.2. Etapa 11: actualizar la versión del requerimiento y del documento

Para esta etapa se debe manejar una carpeta en la cual se almacenarán las diferentes versiones de un requerimiento y, a su vez, tener los respectivos controles de cambios y registros (*logs*) de las mejoras de la documentación de los requerimientos del usuario.

4. Resultados

A continuación se presentan los resultados del proceso de aplicación de esta teoría, con el fin de realizar la validación del *framework* propuesto para la elicitación de requerimientos en aplicaciones que consumen servicios web.

En el proceso de validación se dio a conocer el *framework* a una muestra de personal que interviene en proyectos de software, entre los roles que desempeñan se encuentran analistas y líderes en las áreas

de requerimientos, pruebas, gestión de proyectos y desarrollo de software.

4.1. Cuestionario-encuesta

La encuesta es una de las herramientas planteadas en el modelo TAM (*Technology Acceptance Model*) para la medición del empleo del *framework* propuesto en este artículo, lo anterior en relación con la elicitación de requisitos de aplicaciones que consumen servicios web. El cuestionario consta de ocho preguntas, cuatro de ellas para evaluar percepción de utilidad, dos para evaluar la percepción de la facilidad de uso y dos para evaluar la actitud hacia el uso. A continuación se indican las preguntas seleccionadas para medir cada una de las variables a evaluar.

4.1.1. Variable 1. Utilidad percibida

- ¿Usar el nuevo *framework* de elaboración de requerimientos le parece útil?
- ¿La implementación del *framework* propuesto permitirá reducir la ambigüedad de los requisitos?
- ¿Utilizar el *framework* propuesto ayudará a plasmar correctamente la necesidad del usuario en la documentación de los requisitos?
- ¿El uso de la metodología planteada ayudará a mejorar la especificación de requerimientos para proyectos futuros en su organización?

4.1.2. Variable 2. Facilidad de uso

- 5. ¿Le pareció fácil aprender a utilizar el *framework* de ER?

Tabla 15. Técnicas/herramientas validar requerimientos.

Técnicas/herramientas	Entradas	Salidas
<ul style="list-style-type: none"> • Revisión de requerimientos. • Entrevistas. • Revisiones en grupo. 	Documento de especificación de requerimientos.	<ul style="list-style-type: none"> • Lista de problemas. • Lista de acciones.

Fuente: elaboración propia.

- 6. ¿Encontró fácil especificar requerimientos aplicando la metodología propuesta?

4.1.3. Variable 3. Actitud hacia el uso

- 7. ¿Usar el *framework* de ER es una buena opción para maximizar la eficacia de la elicitación de requerimientos?
- 8. ¿Le agrada la idea de usar el *framework* de ER?

La escala de calificación utilizada en la escala psicométrica tipo Likert indicada en la Tabla 16, la cual permite que el usuario encuestado especifique el nivel de acuerdo o desacuerdo con una declaración (punto o pregunta).

Tabla 16. Escala de calificación utilizada.

Valor	Item
1	Muy en desacuerdo.
2	En desacuerdo.
3	Indeciso.
4	De acuerdo.
5	Totalmente de acuerdo.

Fuente: elaboración propia.

4.1.4. Formato de encuesta

En la Figura 3 se plantea el modelo de formulario utilizado en la implementación de la encuesta.

Figura 3. Formato de encuesta utilizada.

Encuesta de validación					
<i>Objetivo: evaluar el nivel de aceptación respecto a la implementación del framework para elicitación de requisitos propuesto.</i>					
<i>Nombre:</i>					
<i>Título profesional:</i>					
<i>Cargo:</i>					
<i>Proyecto:</i>					
Califique las preguntas de 1 a 5, donde 5 representa calificación positiva y 1 representa calificación negativa.					
Variable externa: utilidad percibida					
Pregunta	1	2	3	4	5
1. ¿Usar el nuevo <i>framework</i> de elaboración de requerimientos le parece útil?					
2. ¿La implementación del <i>framework</i> propuesto permitirá reducir la ambigüedad de los requisitos?					
3. ¿Utilizar el <i>framework</i> propuesto ayudará a plasmar correctamente la necesidad del usuario en la documentación de los requisitos?					
4. ¿El uso de la metodología planteada ayudará a mejorar la especificación de requerimientos para proyectos futuros en su organización?					
Variable externa: facilidad de uso					
Pregunta	1	2	3	4	5
5. ¿Le pareció fácil aprender a utilizar el <i>framework</i> de ER?					
6. ¿Encontró fácil especificar requerimientos aplicando la metodología propuesta?					
Variable externa: actitud hacia el uso					
Pregunta	1	2	3	4	5
7. ¿Usar el <i>framework</i> de ER es una buena opción para maximizar la eficacia de la elicitación de requerimientos?					
8. ¿Le agrada la idea de usar el <i>framework</i> de ER?					
Conclusiones y sugerencias					

Fuente: elaboración propia.

4.1.5. Tabulación de la encuesta

La Figura 4 reúne los resultados consolidados después de la tabulación de la encuesta.

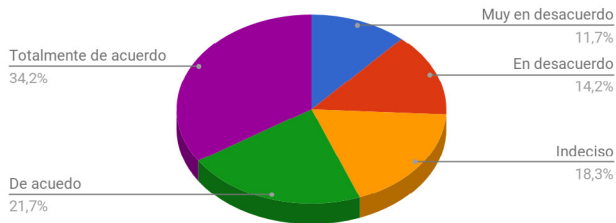


Figura 4. Gráfico de resultados de la encuesta.
Fuente: elaboración propia.

Se realiza una tabulación de la encuesta por variable de apreciación, la Figura 5, la Figura 6 y la Figura 7 permiten apreciar la percepción de la muestra poblacional respecto a la implementación del *framework*, el resultado es favorable.



Figura 5. Gráfico de resultados de utilidad percibida de la encuesta.
Fuente: elaboración propia.



Figura 6. Gráfico de resultados de facilidad de uso de la encuesta.
Fuente: elaboración propia.

Actitud hacia el uso

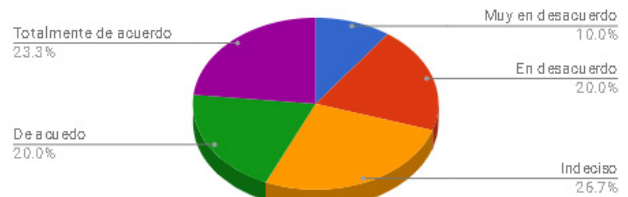


Figura 7. Gráfico de resultados de actitud hacia el uso de la encuesta.
Fuente: elaboración propia.

4.1.6. Resultados finales

Se realiza el cálculo de quintiles para establecer la posición del resultado final, la descripción de cada quintil se indica en la Tabla 17.

Tabla 17. Quintiles establecidos para la tabulación de resultados.

Quintiles	Puntaje total
Q1	Entre 8 y 14.4.
Q2	Mayor a 14.4 y menor a 20.8.
Q3	Mayor a 20.8 y menor a 27.2.
Q4	Mayor a 27.2 y menor a 33.6.
Q5	Mayor a 33.6 y menor a 40.

Fuente: elaboración propia.

Se obtienen los siguientes datos estadísticos indicados en la Tabla 18, donde se indica un valor de aceptación positivo del 70.5%, gracias a un puntaje total de 28.2.

Tabla 18. Resultados estadísticos de la encuesta.

Datos estadísticos	
Mediana	4
Moda	5
Mínimo	8
Máximo	40
Puntaje Total	28.2
Porcentaje de aceptación	70.50%
Quintil	Q4

Fuente: elaboración propia.

Ubicando el puntaje final obtenido —28.2— se observa que está en una posición positiva en el gráfico radial de la Figura 8, es decir que se obtuvo un resultado favorable en el sondeo realizado con un total de quince encuestados.

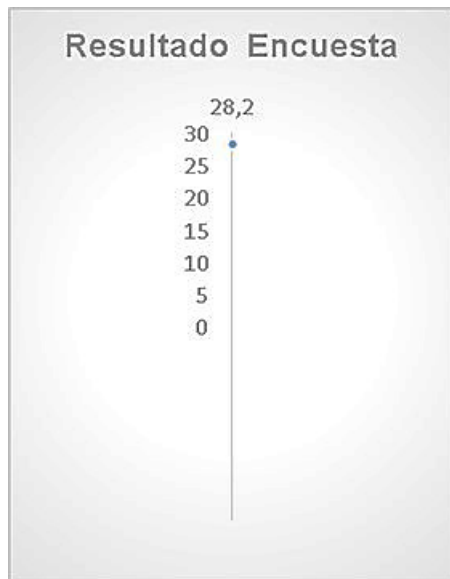


Figura 8. Gráfico radial de resultados generales de la encuesta.

Fuente: elaboración propia.

5. Conclusiones

Se propone una metodología para la elaboración de requerimientos en aplicaciones que consumen servicios a partir de pautas de distintos autores, con el cual se cubren los aspectos básicos de ingeniería de requerimientos, también aspectos de más relevancia como los requisitos blandos del usuario, para descifrar las necesidades del cliente de una manera más profunda; además, se propone una validación exhaustiva de los requisitos con la cual facilitará a las partes interesadas en el desarrollo de proyectos de *software*, diseñar aplicaciones con un alto grado de fidelidad a los objetivos propuestos por los usuarios para el producto final solicitado.

La encuesta realizada con la metodología de aceptación TAM permitió evidenciar que los usuarios potenciales del *framework* consideran que es una buena opción para implementarlo en el desarrollo

de los proyectos de *software*, teniendo en cuenta los resultados de las variables de utilidad, percibida y facilidad de uso. La actitud hacia el uso sugiere la utilización de herramientas amigables para socializar el *framework* con los usuarios. En general se percibe un nivel de aceptación favorable para la utilización de la metodología planteada en este artículo por parte de los roles principales involucrados en la fabricación de aplicaciones de *software* que consumen servicios.

Referencias

- [1] M. Serna, "Analysis and selection to requirements elicitation techniques", in 7th Colombian Computing Congress (CCC), Medellín, October, 2012, <https://doi.org/10.1109/ColombianCC.2012.6398001>
- [2] M. Muqem y M. Beg, "Validation of requirement elicitation framework using finite state machine", In International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, July, 2014, <https://doi.org/10.1109/ICCICCT.2014.6993145>
- [3] C. Maines, B. Zhou, S. Tang y Q. Shi, "Towards a Framework for the Extension and Visualisation of Cyber Security Requirements in Modelling Languages", In 10th International Conference on Developments in eSystems Engineering (DeSE), Paris, June, 2017.
- [4] S. Nazir, "Why Quality? Iso 9126 Software Quality Metrics (Functionality) Support By Uml Suite", *ACM SIGSOFT Software Engineering Notes*, vol. 30, n° 2, 2005.
- [5] Elsevier, "A decade of agile methodologies: towards explaining agile software development", *The Journal of Systems and Software*, vol. 85, n° 6, 2012.
- [6] Z. Sultan, Rabiya, S. Nazir y S. Asim, "Analytical Review On Test Cases Prioritization Techniques: An Empirical Study", *International Journal Of Advanced Computer Science And Applications (IJACSA)*, vol. 8, n° 2, 2017.

- [7] M. Huo, J. Verner, L. Zhu y M. A. Babar, "Software quality and agile methods", In 28th Annual International Computer Software and Applications Conference, Hong Kong, October, 2004.
- [8] L. Cao and B. Ramesh, "Agile requirements engineering practices: an empirical study", *IEEE Software*, vol. 25, n° 1, pp. 60-67, 2008, <https://doi.org/10.1109/MS.2008.1>
- [9] A. Eberlein y J. C. Sampaio, "Agile requirements definition: a view from requirements engineering", In Proceedings of the International Workshop on Requirement Engineering, 2002.
- [10] J. Karlsson y K. Ryan, "Prioritizing Requirements Using A Cost-Value Approach", *IEEE Software*, vol. 14, n° 5, pp. 67-74, 1997, <https://doi.org/10.1109/52.605933>
- [11] A. Perini, A. Susi y P. Avesani, "A machine learning approach to software requirements prioritization", *IEEE Transactions on Software Engineering*, vol. 39, n° 4, 2013, <https://doi.org/10.1109/TSE.2012.52>
- [12] V. N. Vithana, "Scrum requirements engineering practices and challenges in offshore software development", *International Journal of Computer Applications*, vol. 116, n° 22, 2015.
- [13] C. Arias, G. Vilanova, M. Miranda, J. Fontana, S. Rivadeneira y D. Cruz, "Construcción de modelos de requerimientos a partir de modelos de procesos de negocio". [En línea]. Disponible en: https://www.researchgate.net/publication/233529267_Construccion_de_Modelos_de_Requerimientos_a_partir_de_Modelos_de_Procesos_de_Negocio
- [14] N. Sharif, K. Zafar y W. Zyad, "Optimization of requirement prioritization using computational intelligence technique", In International Conference On Robotics And Emerging Allied Technologies In Engineering (ICREATE), Islamabad, april, 2014, <https://doi.org/10.1109/ICREATE.2014.6828370>
- [15] F. L. Butt, S. N. Bhatti, S. Sarwar, A. M. Jadi y A. Saboor, "Optimized order of software testing techniques in agile process – A systematic approach", *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, n° 1, 2017.
- [16] S. N. Bhatti, M. Usman y A. A. Jadi, "Validation to the requirement elicitation framework via metrics", *ACM SIGSOFT Software Engineering Notes*, vol. 40, n° 5, September 2015, <https://doi.org/10.1145/2815021.2815031>
- [17] A. R. Asghar, A. Shah, A. Tabassum y S. N. Bhatti, "The impact of analytical assessment of requirements prioritization models: an empirical study", *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, n° 2, 2017.
- [18] R. Thakurta, "A framework for prioritization of quality requirements for inclusion in a software project", *Software Quality Journal*, vol. 21, n° 4, pp. 573-597, 2013, <https://doi.org/10.1007/s11219-012-9188-5>
- [19] A. Gross y J. Doerr, "What do software architects expect from requirements specifications? results of initial explorative studies", In First IEEE International Workshop on the Twin Peaks of Requirements and Architecture (TwinPeaks), Chicago, September, 2012, <https://doi.org/10.1109/TwinPeaks.2012.6344560>
- [20] I. Sommerville y M. Alfonso, "Ingeniería del software". Madrid: Pearson Educación, 2005.

