

Especificación de Requisitos: Revisión Sistemática de Literatura

Ríos S. Gerardo, Torres S. Richard , León G. Israel, Solórzano J. Samantha and Collaguazo R. Gustavo

Universidad Nacional de Loja, Carrera de Ingeniería en Sistemas, Loja - Ecuador

jose.g.rios, richard.a.torres, israel.leon, samantha.solorzano, gustavo.collaguazo@unl.edu.ec

Abstract—Este documento presenta un estudio de mapeo y revisión sistemática de literatura referente al tema Especificación de Requisitos, con el fin de identificar y analizar los problemas que se presentan en este proceso, además de metodologías, estándares y herramientas existentes para resolverlos y comparar la influencia del LN y LC al momento de redactar los requisitos. Para desarrollar la revisión se utilizó la técnica propuesta por Bárbara Kitchenham, la misma que permitió seleccionar 30 artículos que presentan información relacionada con la presente revisión sistemática. Los artículos se seleccionaron desde el año 2015 en adelante; se encontró metodologías, estándares y herramientas para la especificación de requerimientos, además de los problemas que surgen durante el proceso.

Index Terms— Especificación de Requisitos, Ingeniería de requisitos, software, estándar, problema, método, herramienta.

I. INTRODUCCIÓN

En el desarrollo de sistemas la especificación de requisitos de software es la etapa más importante, donde se definen las restricciones y actividades que se deben llevar a cabo para satisfacer las necesidades del cliente, que a su vez puede determinar el éxito o fracaso del software en desarrollo [1]; en este proceso intervienen los clientes, ingenieros de requisitos y desarrolladores; como resultado de dicha etapa se obtiene un documento “Especificación de Requerimientos” (SRS), que servirá como base para las etapas posteriores del proceso de Ingeniería de Software.

Al ser una de las etapas críticas dentro de la ingeniería de requisitos, la intervención del factor humano muchas veces genera inconvenientes en la especificación, puesto que los stakeholders por lo general manejan diferentes campos o antecedentes [2]; incluso al manejar el LN y LC, pueden existir conflictos de ambigüedad, de acuerdo a [3] pueden generarse referencias cruzadas que en un futuro causarán conflictos en la implementación, estos y otros problemas se van identificando en los procesos posteriores a la especificación, generando mayor consumo de recursos como económicos, tiempo

y trabajo.

Existen diversas metodologías que ayudan a disminuir los problemas que se presentan durante el desarrollo de los requisitos de software, y además contribuyen a realizar el SRS, se adecuan a diferentes escenarios y maneras de trabajo de los stakeholders dependiendo del procedimiento y las necesidades del proyecto; un ejemplo de dichas herramientas son los Casos de Uso, Especificación de Descomposición, Plantillas parametrizadas de requerimientos, etc.

Agregado a las metodologías existentes, se han desarrollado diferentes estándares internacionales, como IEEE Std 830-2009 ISO/IEC/IEEE 29148:2011, ISO/IEC 15504 [4] [5] que buscan ayudar a los desarrolladores a construir documentos SRS bien organizados, que producirían a su vez software de calidad.

En busca de mejorar cada vez más el proceso de especificación de requerimientos, se ha optado por automatizar hasta cierto punto las actividades que se realizan, como la detección de ambigüedades o nominalizaciones [6]; para ello se han desarrollado diversas herramientas tales como DeNom, aToucan, Eclipse Xtext, etc.

Es por las razones expuestas en párrafos anteriores, que en este documento se presenta un estudio sistemático en donde se describe los diferentes estándares, metodologías y herramientas utilizadas en la especificación de requerimientos. También se abordan los problemas que surgen y la relación que existe entre el LN y LC en SRS. El documento se estructura en secciones. La sección 2 presenta la metodología para desarrollar la revisión sistemática. En la sección 3 se presentan los resultados obtenidos. En la sección 4 se presenta la discusión de las preguntas de investigación y en la sección 5 se define las conclusiones del presente artículo.

II. MÉTODO DE INVESTIGACIÓN

Las investigaciones deben estar basadas en evidencias que permitan identificar cómo ha sido abordado el objeto de investigación (Especificación de Requisitos) por otros autores.

Sin embargo, es necesario buscar y agregar evidencias usando estudios secundarios como son las RSL y estudios de mapeo sistemático [7]. Para esta investigación se utilizó el protocolo definido por [8] y se estructuró el documento en referencia a [9] [10] [11] [12].

A. Preguntas de Investigación

En base a la temática principal denominada “Especificación de requisitos” se planteó ocho preguntas de investigación clasificadas en preguntas para el mapeo sistemático (MQ) y la revisión sistemática (RQ):

- **MQ1:** ¿Cuántos estudios se publicaron a lo largo de los años en el área especificación de requisitos?
- **MQ2:** ¿Cuáles son las revistas y congresos que se han publicado en el área de especificación de requisitos?
- **MQ3:** ¿Quiénes son los autores más relevantes y activos en esta área?
- **RQ1:** ¿Cuáles son los estándares existentes para la especificación de requerimientos?
- **RQ2:** ¿Cuál es la diferencia entre la especificación de los requisitos de lenguaje natural y la especificación de los requisitos de lenguaje controlado?
- **RQ3:** ¿Qué metodologías se utilizan para la especificación de Requerimientos?
- **RQ4:** ¿Qué software o herramientas se utiliza para la especificación de requisitos?
- **RQ5:** ¿Cuáles son los problemas u obstáculos que se podrían presentar dentro del proceso de especificación de requerimientos?

B. Proceso de Búsqueda

De acuerdo a la metodología PICOC propuesta por [13] y en base a las preguntas de Investigación se determinaron un conjunto de términos para definir la cadena de búsqueda.

- Población (P): “Requirements Engineering”.
- Intervención (I): “Requirements Specification”.
- Comparación (C) : No Aplica
- Resultados (O): “Software”; “method”; “Standard”, “problems”, “Tool”.
- Contexto (C): Requirements Engineering.

C. Definición de los criterios de inclusión y exclusión

Para determinar los criterios de inclusión y exclusión que se aplicará en la selección de documentos, se ha tomado en cuenta el área de estudio al que pertenecen, año de publicación e idioma, y también las preguntas de investigación. Se definió con los operadores lógicos “AND” los criterios que el documento necesariamente tiene que cumplir y el operador lógico “OR” los criterios que los documentos podrían incluir o excluir.

- **IC1:** Los Documentos deben estar relacionados con La Ingeniería de Requisitos. AND
- **IC2:** Los Documentos deben citar alguna metodología para la Especificación de Requerimientos. OR
- **IC3:** Los Documento debe describir algún estándar existente para la Especificación de Requisitos. OR
- **IC4:** Los Documentos deben detallar que problemas u obstáculos surgen durante el proceso de Especificación de Requerimientos. OR
- **IC5:** Los documentos incluyen el manejo de algún software orientado a la Especificación de requerimientos. OR
- **IC6:** Los Documentos deben estar escritos en inglés. AND
- **IC7:** El documento debe haber sido publicados del año 2015 en adelante AND

Los criterios de exclusión definidos para esta revisión son:

- **EC1:** La fecha de publicación de los documentos sean anteriores al año 2015, ya que se consideran desactualizados. OR
- **EC2:** Los Documentos sean incompletos e Imparciales. OR
- **EC3:** Los Documentos que no tengan relación dentro de la Ingeniería de Requisitos e Ingeniería del Software. OR
- **EC4:** El documento no debe estar escrito en otros idiomas que no sean inglés. OR

Para aplicar los criterios se realizó la lectura del título, abstract, palabras clave e introducción. Los documentos que aparecían en más de una biblioteca digital se descartaron con ayuda de la plataforma parsif.

D. Cadenas de Búsquedas

A través del método PICOC se definieron las palabras clave que apoyados en la sinonimia nos permiten realizar las combinaciones, haciendo uso de los operadores lógicos “AND” para conceptos complementarios, “OR” para conceptos similares; no se utilizó el operador “NOT”. Las búsquedas aplicadas a las bases de datos científicas seleccionadas son las siguientes:

IEEE Library:

- (“All Metadata”:“Requirements Engineering” AND “Requirements Specification” AND (“method” OR “Methodologies” OR “Software” OR “Standard” OR “Tool”))
- (“All Metadata”:“Requirements Engineering”) AND “Requirements Specification” AND “Requirements Specification Methodologies” OR “Requirements Specification Software” OR “Requirements Specification Standard” OR “Requirements Specification Tool” NOT Requirements Elicitation)

ScienceDirect:

- Keywords:((Requirements Engineering) AND (Requirements Specification) AND (method; OR Methodologies OR Software OR Standard OR Tool)) - Title, abstract, keywords((Requirements Engineering) OR (Requirements Specification)) - Article type (REV,FLA,EN,CH,CNF,DAT,EDI)

ACM Digital Library:

- AllField:((“Requirements Engineering”) AND (“Requirements Specification”) AND (“method” OR “Methodologies” OR “Software” OR “Standard” OR “Tool”))

E. Evaluación de Calidad

Una vez realizada la importación de los artículos y documentos encontrados mediante la búsqueda en las Bases de datos, es necesario evaluar la calidad de los mismos, para ello se definieron las siguientes preguntas que servirán como una pauta para calificar cada artículo.

- **QA1:** ¿El autor utiliza o describe un estándar para la especificación de requisitos?
- **QA2:** ¿El autor describe o usa alguna metodología o técnica de especificación de requisitos?
- **QA3:** ¿El autor realiza un análisis de los problemas pueden surgir dentro de la especificación de requerimientos?

- **QA4:** ¿El autor utiliza algún software o programa para la especificación de requisitos?

III. RESULTADOS

Para realizar esta RSL se hizo uso de la herramienta Parsifal que agiliza el proceso de una RSL, A continuación, se presentan las acciones que se llevaron a cabo para obtener los resultados:

1. Al ejecutar la cadena de búsqueda en las diferentes Bases de Datos se obtuvieron 935 artículos.
2. De los 935 documentos, se eliminaron 73 (7.81 %) artículos duplicados haciendo uso de la herramienta parsifal que identifica artículos iguales, dejando 862 (92.19 %) artículos para el análisis.
3. Los 862 artículos fueron analizados basándose en los títulos y resúmenes de cada uno. Para descartar los artículos que no son relevantes, se aplicó los criterios de exclusión e inclusión especificados en la herramienta parsifal, descartando así 770 (82.35 %). En la Tabla I se muestra en detalle la distribución de los artículos, la mayor parte proviene de la Base de Datos “Science Direct” con 374 artículos que representan el 40 % del total. Al final 92 fueron los documentos aceptados, que representan el 9.84 % del total.
4. Las preguntas planteadas en la sección II-D se especificaron en Parsifal y sirven para calificar los artículos seleccionados. Una vez calificados, finalmente se obtuvieron un total de 30 artículos, que representan el 3.21 % del total, estos se utilizaron para realizar la extracción de información relevante a esta RSL.

TABLE 1: ARTÍCULOS REVISADOS

Base de Datos	Artículos				
	Encon-trados	Dupli-cados	Revi-sados	Elimi-nados	Seleccio-nados
IEEE	283	4	279	233	33
Science Direct	374	17	357	343	27
ACM	278	52	226	194	32
Total	935	73	862	770	92

A. Informe del mapeo sistemático

MQ1 “¿Cuántos estudios se publicaron a lo largo de los años en la ingeniería de software: Especificación de requisitos?”

En el periodo 2015 – 2020 se han publicado 30 artículos. Se ilustra en la Fig.1 el número de artículos por año, siendo

el 2015 en donde más publicaciones hubieron relacionadas al tema, en comparación al 2016 que fue el año en el que menos aportaciones hubo.



Fig. 1: Publicaciones por año

MQ2 “¿Cuáles son las revistas y congresos que se han publicado en el área mencionada?”

En la Fig. 2, se detalla cuántos artículos se han publicado en conferencias y revistas, se observa que en todos los años, a excepción de 2020, han habido mayores contribuciones en las conferencias, dejando en segundo lugar las publicaciones hechas en revistas científicas; siendo libros y simposio los lugares de publicación menos utilizados.

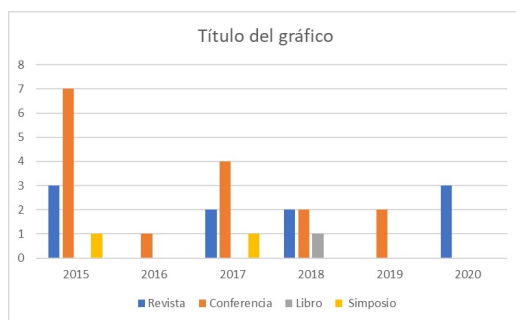


Fig. 2: Lugar y año de publicación de artículos

En la Tabla II se presentan los 30 artículos seleccionados donde se detalla la información, el título, el año de publicación y su fuente de publicación (nombre del congreso o revista) para identificar posibles escenarios de publicación, para próximas investigaciones.

TABLE 2: ARTÍCULOS EVALUADOS

Título	Año / Revista/ Congreso
From User Demand to Software Service: Using Machine Learning to Automate the Requirements Specification Process [14]	2017/ IEEE 25th International Requirements Engineering Conference Workshops
Detecting Non-Atomic Requirements in Software Requirements Specifications Using Classification Methods [2]	2019/ 1st International Conference on Cybernetics and Intelligent System (ICORIS)
Quality Assessment Method for Software Requirements Specifications Based on Document Characteristics and Its Structure [15]	2015/ 2nd International Conference on Trustworthy Systems and Their Applications
AToucan: An Automated Framework to Derive UML Analysis Models from Use Case Models [16]	2015/ ACM Transactions on Software Engineering and Methodology
Improvement in Requirement Specifications Using Petri Nets[17]	2017/International Conference on Computer, Electrical & Communication Engineering (ICCECE)
Process to enhance the quality of software requirement specification document [1]	2018/International Conference on Engineering and Emerging Technologies (ICEET)
Recommended practices for the specification of multi-agent systems requirements[5]	2017/ IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)
Towards an Ontology Pattern Language for Harmonizing Software Process Related ISO Standards [18]	2015/ SAC '15: Proceedings of the 30th Annual ACM Symposium on Applied Computing
The ontology-based approach to support the completeness and consistency of the requirements specification [19]	2015/ International Siberian Conference on Control and Communications (SIBCON)
Combining Xtext and OSLC for Integrated Model-Based Requirements Engineering [20]	2015/ 1st Euromicro Conference on Software Engineering and Advanced Applications

Enhancing requirements engineering for patient registry software systems with evidence-based components [21]	2017/ Journal of Biomedical Informatics	A requirement mining framework to support complex sub-systems suppliers[28]	2018/ Procedia CIRP
DeNom: a tool to find problematic nominalizations using NLP[6]	2015/IEEE Second International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)	A survey on the formalization of system requirements and their validation[29]	2020/ Array
Use case technique for requirements modeling in distributed development environments: A mapping study [22]	2015/ Latin American Computing Conference (CLEI)	Quality of software requirements specification in agile projects: A cross-case analysis of six companies[3]	2018/ Journal of Systems and Software
The Influence of Requirements in Software Model Development in an Industrial Environment [23]	2017/ ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)	A Design Method for Domain-Specific Models of Software Requirements Specification Based on Stakeholders' Concerns[30]	2018/ 25th Asia-Pacific Software Engineering Conference (APSEC)
Verifying goal-oriented specifications used in model-driven development processes[24]	2017/ Information Systems	The Parameterized Safety Requirements Templates[4]	2015/ IEEE/ACM 8th International Symposium on Software and Systems Traceability
Linguistic Patterns and Linguistic Styles for Requirements Specification (I): An Application Case with the Rigorous RSL/Business-Level Language[25]	2017/ EuroPLoP '17: Proceedings of the 22nd European Conference on Pattern Languages of Programs	Hidden fuzzy information: Requirement specification and measurement of project provider performance using the best worst method[31]	2020/ Fuzzy Sets and Systems
An Approach to Identify Use Case Scenarios from Textual Requirements Specification[26]	2019/ ISEC'19: Proceedings of the 12th Innovations on Software Engineering Conference (formerly known as India Software Engineering Conference)	Shared service recommendations from requirement specifications: A hybrid syntactic and semantic toolkit[32]	2015/ Information and Software Technology
Towards Mutation Analysis for Use Cases[27]	2016/ MODELS '16: Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems	Requirements specification for developers in agile projects: Evaluation by two industrial case studies[33]	2020/ Information and Software Technology
		4 - Requirements Specification of a Software Application[34]	2018/ Certifiable Software Applications 3
		A pattern-based method for refining and formalizing informal specifications in critical control systems[35]	2015/Journal of Innovation in Digital Ecosystems
		Variability Aspects at a Textual Requirements specification level[36]	2017/Requirements Engineering Conference Workshops (REW).

A simplified systematic literature review: Improving Software Requirements Specification quality with boilerplates [37]	2019/ Software Engineering (MySEC), Malaysian Conference
---	--

MQ3: ¿Quiénes son los autores más relevantes y activos en esta área?

Los autores más destacados, o que han participado en dos o más artículos son A.Vasconcelos C. Silva M. Goulão, A.R. Da Silva, O. Pastor, T. Yue; siendo A.R. Da Silva y O. Pastor, los autores más activos dentro del año 2017.

En los artículos [3] [33], los autores participantes son J. Medeiros A.Vasconcelos C. Silva M. Goulão, quienes publican dos artículos uno en el año 2018 y otro en 2020.

A.R. Da Silva en el año 2017 publica dos artículos [25] [36] en conferencias de especificación de requerimientos, ambos orientados específicamente al análisis del lenguaje, patrones y especificación.

[23] [24] Son publicados con un autor en común O. Pastor, el primero es presentado en el ESEM de 2017 y el segundo es publicado el mismo año en la revista Information Systems, dichos artículos tienen en común el enfoque de desarrollo de requerimientos a partir de modelos.

El autor T. Yue participó en los artículos [16] [27] publicados en 2015 y 2016 respectivamente, el primero en la revista ACM Transactions on Software Engineering and Methodology y el segundo durante la 19 conferencia 12th Innovations on Software Engineering.

B. Extracción de la Información

Las pautas para realizar la extracción de información están dadas por los criterios de selección de estudios. Se clasificó estándares, metodologías, herramientas y problemas que se llegan a dar durante la especificación de requisitos, para tener una visión clara de las preguntas RQ1, RQ3, RQ4 y RQ5 descritas en la Tabla III, Tabla IV, Tabla V y Tabla VI respectivamente. La pregunta RQ2 se describe más adelante en la tabla VII.

TABLE 3: ESTÁNDARES ENCONTRADOS EN LOS ARTÍCULOS

Estándar para la especificación de requisitos	Artículo ID/Referencia
ISO/IEC 15504-2 ISO/IEC 15504	[15], [18]
IEEE 830	[15], [27], [3], [30], [33]
ISO/IEC/IEEE 29148 ISO/IEC/IEEE 29148:2011	[18], [21], [3], [30], [4], [33]

En la tabla III se define los estándares para la especificación de requisitos que se han encontrado en más de dos documentos, con su respectiva referencia. De igual manera la tabla IV con las metodologías, la tabla V con las herramientas y en la Tabla VI con los problemas que pueden surgir durante la especificación de requisitos.

TABLE 4: METODOLOGÍAS ENCONTRADAS EN LOS ARTÍCULOS

Metodología para la especificación de requisitos	Artículo ID/Referencia
PNL (Técnicas de procesamiento del lenguaje natural)	[14], [26], [34], [35]
Ontologías de comportamiento	[14], [1], [18], [19], [6]
Ontologías de dominio	[14], [1], [18], [19], [20], [30]
Técnicas Gráficas (Casos de Uso).	[17], [22], [26], [3]
Especificaciones declarativas e imperativas	[23], [34]
MDA (Arquitectura dirigida por modelos)	[16], [24], [30]
SADT (Análisis Estructurado y Técnica de Diseño)	[17], [26], [32]
Modelados a través de patrones (PWUC, PWUD, PPP)	[18], [26], [20], [29], [4], [35]
Escritura con lenguaje natural controlado (CNL).	[20], [23], [26], [34]
MDD (Model-Driven Development)	[24], [4]
RSD (Requirements Specification for Developers)	[33], [34]
RSL (RAISE Specification Language)	[36], [34]

TABLE 5: HERRAMIENTAS ENCONTRADAS EN LOS ARTÍCULOS

Herramientas utilizadas para la especificación de requisitos	Artículo ID/Referencia
NuSMV	[20], [35]
EARS-CTRL (Syntax de Enfoque Fácil de Requisitos)	[29], [4]
MS Word	[3], [33]
CESAR	[4], [37]

TABLE 6: PROBLEMAS ENCONTRADOS EN LOS ARTÍCULOS

Problemas que dan en la especificación de requisitos.	Artículo ID/Referencia
Incompletitud e imprecisión. Ambigüedad y vaguedad Inexactitud e inconsistencia. (Usando lenguaje Natural)	[14], [17], [1], [37], [25], [26], [28], [29], [3], [15], [4], [31], [33], [34], [35]
Formalización de Requisitos	[16], [17], [36], [19]
Errores humanos Defectos en los Modelos	[24], [27]
Problemas de clasificación (ReferenciasCruzadas)	[28], [35]
Falta de trazabilidad	[31], [4], [32]

IV. DISCUSIÓN

A. Estándares para la especificación de requisitos.

Los artículos [15] [3] [30] [27] [33] [5] detallan el estándar IEEE 830, el cual es comúnmente usado en la industria, para modelar el proceso de desarrollo de SRS y ayudar a construir artefactos de alta calidad, a la fecha de esta revisión cuenta con 59958 vistas de texto completo en la IEEE Xplore, siendo el más popular de los estándares, en el mismo se describe un conjunto de 8 características de calidad para una buena SRS: corrección, falta de ambigüedad, integridad, coherencia, priorización, verificabilidad, modificabilidad y trazabilidad, no obstante, este estándar es reemplazado en el 2011 por el IEEE 29148 [30] mencionado en los artículos [18] [21] [3] [30] [4] [33] donde se introduce como nuevas características de calidad a la viabilidad, necesidad, la libre implementación y la singularidad, y se eliminan la priorización, la corrección y modificabilidad. Además, se distingue entre la especificación de requisitos de los interesados (StRS), especificación de requisitos del sistema (SyRS) y especificación de requisitos de software (SRS), se describen los métodos de obtención de requisitos de software y se proporcionan plantillas de especificación de requisitos [21]. En la práctica, la gran mayoría de procesos de especificación de requisitos no se pueden adherir en su totalidad al estándar IEEE 29148, en especial si se considera el desarrollo del uso de metodologías ágiles, donde la documentación es limitada, sin embargo, se ha logrado demostrar una cierta correlación entre el éxito de un proyecto y de requisitos de alta calidad [3] denotando su relevancia dentro del área.

En los artículos [15] [18] el estándar ISO/IEC 15504 proporciona recomendaciones sobre la evaluación de la calidad del proceso de desarrollo de software, como marco para evaluar la calidad del SRS.

En el artículo [18] se mencionan los estándares ISO/IEC 15288: 2008, ISO/IEC 12207:2008 y CMMI, que tratan acer-

ca de los procesos del ciclo de vida del sistema, los procesos del ciclo de vida del software y procesos del software de calidad, respectivamente, al juntarlos se puede evaluar todo el proceso del proyecto.

Entre otros estándares se identificó: El IEEE 1233, que es un estándar para la SRS anterior al IEEE 830 [15]; El IEEE STD 610.12-1990, una norma que define una lista de propiedades para la especificación en su conjunto y para ciertos requisitos, que se pueden utilizar para una gestión y un desarrollo de proyectos eficaces [19]; La ISO-26262, recomienda el uso de notaciones semiformales que disminuyan la ambigüedad [20]; Standard goal-oriented modeling, es el modelado estándar orientado a objetivos para una buena elección en el proceso de especificación de requisitos [20]; NASA-DID-P200, es el estándar de documentación de software de la NASA, tiene ventajas en cuanto a independencia, integración y precisión de los requisitos, pero no se sugiere para proyectos pequeños [30]; DI-IPSC-81433, definida por el departamento de defensa de los Estados Unidos [30] y El ISO 9126 define y describe una serie de características de calidad de un producto de software (características internas, externas y características de uso) [34].

B. Diferencia entre Lenguaje Natural y Lenguaje Controlado en la Especificación de requisitos.

El LN se utiliza a menudo porque tiene riqueza comunicativa, flexibilidad y se puede jugar con las expresiones produciendo metáforas o ambigüedades, la ventaja del LN es que otorga la facilidad de comprensión para el Cliente, pero para la especificación de requerimientos que es un proceso en el que se necesita precisión y exactitud, no resulta idóneo.

La diferencia existente entre LN y LC en los aspectos tales como ambigüedad, facilidad de comprensión para el cliente, el programador y errores en requerimientos está dada por [17] y se ilustra en la tabla VII.

TABLE 7: COMPARACIÓN ENTRE LN Y LC

Criterios	Lenguaje Natural (LN)	Lenguaje Controlado (LC)
Ambigüedad	Muy Alto	MuyBajo
Facilidadde comprensión para el Cliente	Muy Alto	Muy bajo
Facilidadde comprensión para el programador	Muy Bajo	Muy Alto
Errores en Requerimientos	Muy Alto	Muy Bajo

Según una encuesta realizada en [1], el 30 % de las cuestiones de calidad están relacionadas con la ambigüedad, el 22 % con la conformidad y el restante 48 % con la exhaustividad, facilidad de comprensión y la coherencia. Según [37] el LN

es bastante propenso a la ambigüedad y sin hacer cumplir las restricciones, puede resultar difícil analizar los requisitos, sin embargo, puede ser mucho más entendible para los stakeholders y esto es importante, pues como menciona [20] los requisitos generalmente se utilizan para la comunicación con las partes interesadas que tienen antecedentes diferentes (por ejemplo, clientes, marketing, otros departamentos de la organización). Esto significa que los requisitos deben formularse de manera que puedan ser fácilmente entendidos por personas provenientes de diferentes disciplinas.

El desarrollo de enfoques es útil para eliminar los problemas de LN del documento, convirtiendo los requisitos al LC, para reducir significativamente la ambigüedad [1]. El objetivo de Lenguaje Controlado es escribir requisitos sintácticamente correctos y uniformes [20]. Como en [18] que hace uso de una ontología para definir un 'estándar' para el dominio del proyecto, dicha ontología elaborada a partir de las entradas y relaciones base es alimentada por los patrones de modelado que permiten expresar los requisitos de LN a LC.

A partir de los procesos de especificación de requisitos con usuarios entrenados, se aprenden ontologías conductuales que luego se utilizan para apoyar el proceso de especificación de requisitos para usuarios no entrenados que por lo general hacen uso del LN para determinar estos requisitos [14].

En 2014, T. Kuhn [4] elaboró una encuesta que describía cien requisitos en LC en inglés. De todos los lenguajes analizados por Kuhn, sólo la herramienta Attempto podría haber contribuido a la definición de las plantillas de requisitos de seguridad parametrizados, debido a su apoyo a la elaboración de frases sustantivas complejas, plurales, referencias anafóricas, cláusulas subordinadas, de modalidad y preguntas.

El lenguaje natural controlado debe cumplir con las mejores prácticas descritas en el estado de la técnica y en las recomendaciones de la reglamentación [4], como las de la norma ISO 29148, que, entre otras cosas, recomienda:

- “Es mejor evitar el uso del término ‘debe’, debido a una posible interpretación errónea como requisito”.
- “Usar declaraciones positivas y evitar requisitos negativos como ‘no deberá’ ”.
- “Usar voz activa: evitar el uso de voz pasiva, como ‘deberá poder seleccionar’ ”.

C. Metodologías para la especificación de requisitos.

Las ontologías de dominio presente en los artículos [14] [1] [18] [19] [20] [30], modelado de patrones tratada en los artículos [18] [26] [20] [29] [4] [35], ontologías de comportamiento mencionada en los artículos [14] [1] [18] [19] [6],

Técnicas de procesamiento del lenguaje natural de los artículos [14] [26] [34] [35] y técnicas gráficas de los artículos [17] [22] [26] [3], son las metodologías que más resaltan en los artículos estudiados, las cuales representan un 22.73 % del total de metodologías encontradas, esto puede indicar que sean las que mejor se adaptan a los problemas, fáciles de usar o completas. Sin embargo, el otro 77.27 % de metodologías que si bien, no son las más populares, puede llegar a ser buena idea considerarlas para algún proyecto específico.

Las metodologías a continuación, tienen como eje central técnicas lingüísticas. Ontologías de Dominio [14] [1] [18] [19] [6] se asegura de dar un significado más profundo a un término, mientras que, Métodos de Clasificación [2] se centra en clasificar requisitos en documentos en función de declaraciones atómicas. Las Especificaciones Declarativas e Imperativas [23] [34] centran su trabajo en escribir una especificación textual de todas las características del sistema y por otro lado la Escritura con Lenguaje Controlado [20] [23] [26] [34] se enfoca en crear documentos consistentes, legibles y traducibles. Finalmente OCL [36] es un Lenguaje de Expresión que permite describir operaciones de restricción sobre modelos Orientados a Objetos. De igual forma existen metodologías que se centran en el uso de herramientas externas para el tratamiento de los requisitos, como PNL (Técnicas de procesamiento del lenguaje natural) [14] [26] [34] [35] en donde se pretende que una máquina entienda lo que expresa una persona en lenguaje natural, por otro lado, Validación de Métricas [15] hace uso de herramientas especializadas y métodos efectivos de análisis con el fin de evaluar la calidad de especificación.

Las metodologías Gráficas, agilizan la comprensión y orden de los documentos. Las Técnicas Gráficas [17] [22] [26] [3] que muestran las actividades y tareas del usuario de manera sencilla y concisa, mientras que Técnicas de especificación de Descomposición [3] consiste en una estructura de niveles, donde los elementos se descomponen en sub elementos, esta metodología es parecida a SADT [17] [26] [32] cuyo eje es una descripción jerárquica de funciones y gráficos de estados. MMD (Model-Driven-Development) [24] [4] trabaja con modelos para plantear un elemento entre la propuesta tradicional y las plataformas de objetos; Restricted Use Case Modelling [27] se enfoca en el modelado de casos de uso con el objetivo de restringir la manera en la que los stakeholders puedan documentar los casos de uso.

También existen metodologías que hacen uso de técnicas, documentos, códigos y recursos externos al proyecto, como SBSE [14] que consiste en el uso de técnicas pertenecientes a otras ingenierías para solucionar problemas cuyas condiciones puedan ser llevadas a cabo por estas, en cambio MDA considera los modelos previos de códigos que se usan como base para el desarrollo del proyecto nuevo y Parameterized

Requirements Templates [4] se basa en el uso de plantillas, lo que permite al ingeniero observar con cierta perspectiva los requerimientos. Finalmente la metodología RSL (RAISE Specification Language) [25] [34] consiste en métodos para lidiar con inconvenientes y problemas que puedan surgir durante la elaboración de software.

D. Problemas durante el proceso de la especificación de requisitos.

Los autores de los artículos [14] [17] [1] [37] [25] [26] [28] [29] [3] [15] [4] [31] [33] [34] [35] concuerdan en que el uso de lenguaje natural para la especificación de requisitos es muy común y practico, pero esto puede acarrear varios problemas debido a que el lenguaje natural carece de organización, dando lugar a inconsistencias e inexactitud, incompletitud e imprecisión, además de ambigüedades y vaguedad en las especificaciones. Por otro lado el artículo [2] propone que los malentendidos con las partes interesadas con respecto a las especificaciones es un error común, dado por diferentes antecedentes o campos de la ciencia en los que trabajan los stakeholders.

En los artículos [16] [17] [36] [19] se da a conocer la dificultad que representa la formalización de requisitos es decir, la transformación de requisitos en lenguaje natural a una especificación estructurada y recalca la importancia que tiene realizar la formalización, para evitar los problemas que se pueden presentar utilizando lenguaje natural; pero en el artículo [20] se menciona que el uso de anotaciones formales puede ser una tarea bastante engorrosa, porque los ingenieros también provienen de otras disciplinas en las que no están capacitados para aplicar estas notaciones generando así problemas de usabilidad.

[37] [23] [34] Establecen que la falta de verificación y validación de requisitos provoca una mala especificación y complica la gestión de los mismos, dificultando su análisis e interpretación; otro problema que se presenta en [31] [4] [32] es la falta de trazabilidad de los requisitos generando conflictos internos para las demás fases de un proyecto. Otros problemas que se pueden encontrar son: el no uso de la misma semántica propuesto por [18] provocando problemas de interoperabilidad. El artículo [6] presenta la problemática de las nominalizaciones dentro de la especificación de requisitos, dividiendo los fallos de requisitos en 3 categorías: supresión, generalización y distorsión; también en los artículos [24] [27] se presenta que los errores humanos son un problema que puede provocar defectos en los modelos de casos de uso generando inconsistencias en proyectos que utilizan metodologías de modelado. Para terminar en los artículos [28] [35] se define como problema la mala clasificación de requisitos provocando referencias cruzadas, y finalmente en el artículo [33] se determina que un equipo de trabajo poco pro-

ductivo genera especificaciones breves, incompletas, vagas y ambigüas.

E. Herramientas para la especificación de requisitos.

Dentro del análisis de la utilidad y funcionalidad de las herramientas, se identificó seis elementos recurrentes, los cuales se propone como parámetros para comparar las herramientas; una vez determinados los problemas durante la especificación de requisitos en el punto D. Se dará una calificación a cada parámetro según el problema que ayuda a resolver; como se describe en la tabla VIII.

En la tabla IX se especifican las herramientas y los parámetros que cumplen respectivamente, a fin de identificar cuál de ellas es la más completa dentro de las presentadas en los diferentes artículos.

TABLE 8: PARAMETROS PARA EVALUAR HERRAMIENTAS

ID	Parámetro	Problema	Valor
PE1	Clasificación-Detección de problemas	Inconsistencias Inexactitud Incompletitud Nominalización	5
PE2	Redacción de requisitos	Dificultad en la formalización	5
PE3	Lenguaje de Especificación	Ambigüedades Vaguedad	4
PE4	Establecer relaciones entre los datos	Rastreabilidad Conflictos Referencias cruzadas	4
PE5	Apoyo en la realización de diagramas UML	Inconsistencia en casos de uso	3
PE6	Plantillas o etiquetas	Problemas de interoperabilidad	2
PE7	Evaluar Aceptación	Falta de verificación y validación	1

Como se puede apreciar en la tabla IX, las herramientas REACT [14], Weka [28], WordNet [32], RUBRIC [20], Parser 1 [26] han obtenido una calificación igual o superior a 10, tanto REACT, Weka y WordNet presentan soluciones optimizadas a los problemas más comunes, sin embargo, dejan de lado la resolución de cualquier otro. Dichas herramientas se centran en la detección de errores y la redacción o recomendación de requisitos, siendo [14] [18] las que extraen requisitos semi-estructurados, a diferencia de [32] que, si bien ayuda en la redacción mediante el análisis de oraciones, no estructura propuestas de requerimientos, cabe recalcar que [14] [28] identifican errores muy diferentes, el primero descarta expresiones irrelevantes o ambiguas y el segundo más técni-

TABLE 9: MATRIZ EVALUACIÓN DE ARTÍCULOS

SW	PE 1	PE 2	PE 3	PE 4	PE 5	PE 6	PE 7	Tot
REACT	X	X						10
Weka	X	X						10
DeNom	X		X					9
Femmer	X	X						5
WordNet	X	X						10
RUBRIC			X				X	11
ConQAT	X							5
CESAR	X					X		7
EARS	X					X		7
Redmine			X					4
Protegè		X		X	X			7
ParserI		X			X	X		10
AToucan				X	X			12
Compass			X				X	6

camente se orienta a determinar referencias cruzadas. Siendo finalmente Weka la que resalta demostrando una precisión del 91.2% en su identificación tanto de los requerimientos semiestructurados, como en referencias cruzadas [28].

La herramienta con mayor calificación es AToucan [16] que contempla mayor utilidad en la resolución de problemas, dicha herramienta apoya en la realización de diagramas UML, de manera que se puedan establecer relaciones y secuencia entre los elementos y sean de utilidad para los analistas al momento de especificar requisitos, fue probada en diferentes casos de estudio para diagramas de casos de uso y secuencia, dando como resultado requisitos más completos y mejor estructurados, en comparación esta la herramienta RUBRIC [20] que hace uso de Nusmv, Eclipse y Xtext, apoyando en la redacción de requisitos, sin ofrecer una propuesta, sin embargo al hacer uso de Nusmv facilita la redacción de los requisitos al lenguaje formal, que representa su mayor ventaja; además de ayudar en la verificación de los requisitos redactados, siendo este último el problema menos relevante que contempla; finalmente se encuentra [26] que a diferencia de las anteriores genera varias etiquetas, interpreta frases de especificación de requisitos y utilizando las reglas del lenguaje NLP, rellena varios elementos de los casos de uso, resolviendo un problema más relevante como lo es el apoyo en diagramas UML en comparación a la evaluación de aceptación.

Las herramientas que obtuvieron una calificación de 7 o menos [31] [4] [37] [29] [19] [33], las cuales resuelven problemas menos relevantes o a su vez no contemplan los suficientes; en el caso de Cesar [4] resulta en un lenguaje de especificación obsoleto que brinda plantillas para la detección de errores, pero no permite un seguimiento para la re-

solución, siendo su utilidad actualmente muy baja, a partir de esta herramienta surge EARS [29] [4], que mejora estas deficiencias, contemplando sólo dos de los problemas identificados, y siendo uno de ellos de menor incidencia.

Femmer y ConQAT [31] ambas a partir de glosarios se centran únicamente en la detección de palabras ambiguas, siendo su alcance limitado a la fase de identificación y por lo tanto haciéndola una herramienta limitada para el proceso de especificación.

En [19] se hace uso de la herramienta Protegè, un editor de ontología, permite organizar los requerimientos según una clasificación de clases y atributos, de manera que se puedan establecer relaciones opcionales entre estas clases y sus atributos para realizar la especificación, además dicha herramienta tiene la opción para describir las especificaciones con la estructura de casos de uso en lenguaje natural.

Compass es una herramienta usada en el artículo [29] implementa un enfoque diferente y admite varios análisis. En lugar de extender la semántica del lenguaje SLIM, la herramienta utiliza extensiones de modelo, como modelos de error, además de inyecciones de fallas, que le permiten inyectar fallas automáticamente en el modelo nominal.

V. CONCLUSIONES

Se detecta cómo el problema más común en la especificación de requisitos la ambigüedad, inconsistencia, incompletitud, inexactitud y vaguedad que se da en las especificaciones realizadas en lenguaje natural, si bien este lenguaje tiene muchas ventajas como: ser bastante práctico y entendible por todos los stakeholders, son varios los autores que concuerdan que para evitar este tipo de problemas se debe realizar una formalización de los requisitos utilizando lenguaje controlado con notaciones formales. Algunas de las soluciones propuestas son el uso de metodologías y estándares como el IEEE 830 y el IEEE 129148, los cuales han evolucionado conjuntamente con las tecnologías renovándose en función del contexto, planteando mejoras con el fin de obtener un producto de calidad. También el uso de ontologías tanto de comportamiento como de dominio, modelado a través de patrones y técnicas gráficas brindan mayor facilidad y flexibilidad para todas las partes interesadas dentro del proceso, esto porque definen glosarios, plantillas y medios que ayudan a moderar los problemas como la ambigüedad e inexactitud.

La mayoría de autores han hecho uso de diferentes herramientas, EARS provee mejoras con respecto a las plantillas, que permiten identificar y rastrear de mejor manera los elementos problemáticos; además existen prototipos de herramientas muy prometedoras, como es el caso de AToucan, RUBRIC, Parser I, las cuales, al contemplar mayores problemas relevantes a solucionar, se vuelven herramientas más completas.

REFERENCIAS

- [1] Syed Waqas Ali, Qazi Arbab Ahmed, and Imran Shafi. Process to enhance the quality of software requirement specification document. *2018 International Conference on Engineering and Emerging Technologies, ICEET 2018*, 2018-Janua:1–6, 2018.
- [2] Fahrizal Halim and Daniel Siahaan. Detecting Non-Atomic Requirements in Software Requirements Specifications Using Classification Methods. *2019 1st International Conference on Cybernetics and Intelligent System, ICORIS 2019*, 1(August):269–273, 2019.
- [3] Juliana Medeiros, Alexandre Vasconcelos, Carla Silva, and Miguel Goulão. Quality of software requirements specification in agile projects: A cross-case analysis of six companies. *Journal of Systems and Software*, 142:171–194, 2018.
- [4] Pablo Oliveira Antonino, Mario Trapp, Paulo Barbosa, and Luana Sousa. The Parameterized Safety Requirements Templates. *Proceedings - 2015 IEEE/ACM 8th International Symposium on Software and Systems Traceability, SST 2015*, pages 29–35, 2015.
- [5] Khaled Slhoub, Marco Carvalho, and Walter Bond. Recommended practices for the specification of multi-agent systems requirements. *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2017*, 2018-Janua:179–185, 2017.
- [6] Mathias Ländhäußer, Sven J. Körner, Walter F. Tichy, Jan Keim, and Jennifer Krisch. DeNom: A tool to find problematic nominalizations using NLP. *2nd International Workshop on Artificial Intelligence for Requirements Engineering, AIRE 2015 - Proceedings*, pages 9–16, 2015.
- [7] Barbara A Kitchenham, David Budgen, and O Pearl Brereton. Using mapping studies as the basis for further research—a participant-observer case study. *Information and Software Technology*, 53(6):638–651, 2011.
- [8] Pearl Brereton, Barbara A Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, 80(4):571–583, 2007.
- [9] Isatou Hydara, Abu Bakar Md Sultan, Hazura Zulzalil, and Novia Admodisastro. Current state of research on cross-site scripting (XSS)—A systematic literature review. *Information and Software Technology*, 58:170–186, 2015.
- [10] Frâncila Weidt Neiva, José Maria N David, Regina Braga, and Fernanda Campos. Towards pragmatic interoperability to support collaboration: A systematic review and mapping of the literature. *Information and Software Technology*, 72:137–150, 2016.
- [11] R. Figueroa Diaz y F. Ajila Zaquinaula J. Iñiguez Bagnas, R. Guamán Quinché. Revisión Sistemática de Literatura: Inyección SQL en Aplicaciones web Systematic Literature Review: SQL Injection in Web Applications. *Latin American Journal of Computing Faculty of Systems Engineering Escuela Politécnica Nacional*, 3:8, 2016.
- [12] Pablo Vicente Torres-Carrión, Carina Soledad González-González, Silvana Aciar, and Germanía Rodríguez-Morales. Methodology for systematic literature review applied to engineering and education. In *2018 IEEE Global Engineering Education Conference (EDUCON)*, pages 1364–1373, 2018.
- [13] Mark Petticrew and Helen Roberts. *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons, 2008.
- [14] Larijn Van Rooijen, Frederik Simon Bäumer, Marie Christin Platenius, Michaela Geierhos, Heiko Hamann, and Gregor Engels. From user demand to software service: Using machine learning to automate the requirements specification process. *Proceedings - 2017 IEEE 25th International Requirements Engineering Conference Workshops, REW 2017*, pages 379–385, 2017.
- [15] Patra Thitisathienkul and Nakornthip Prompoon. Quality Assessment Method for Software Requirements Specifications Based on Document Characteristics and Its Structure. *Proceedings - 2nd International Conference on Trustworthy Systems and Their Applications, TSA 2015*, pages 51–60, 2015.
- [16] Tao Yue, Lionel C. Briand, and Yvan Labiche. aToucan: An automated framework to derive UML analysis models from use case models. *ACM Transactions on Software Engineering and Methodology*, 24(3), 2015.
- [17] A. Keshav Bharadwaj and V. K. Agrawal. Improvement in requirement specifications using petri nets. *2017 International Conference on Computer, Electrical and Communication Engineering, ICCECE 2017*, pages 1–8, 2018.
- [18] Fabiano B. Ruy, Ricardo A. Falbo, Monalessa P. Barcellos, and Giancarlo Guizzardi. Towards an ontology

pattern language for harmonizing software process related ISO standards. *Proceedings of the ACM Symposium on Applied Computing*, 13-17-April:388–395, 2015.

- [19] Tatiana Avdeenko and Natalia Pustovalova. The ontology-based approach to support the completeness and consistency of the requirements specification. *2015 International Siberian Conference on Control and Communications, SIBCON 2015 - Proceedings*, 2015.
- [20] Nadja Marko, Andrea Leitner, Beate Herbst, and Alfred Wallner. Combining Xtext and OSLC for Integrated Model-Based Requirements Engineering. *Proceedings - 41st Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2015*, pages 143–150, 2015.
- [21] Doris Lindoerfer and Ulrich Mansmann. Enhancing requirements engineering for patient registry software systems with evidence-based components. *Journal of Biomedical Informatics*, 71:147–153, 2017.
- [22] Viviana Alferillo, María Inés Lund, and Gerardo Maturro. Use case technique for requirements modeling in distributed development environments: A mapping study. *Proceedings - 2015 41st Latin American Computing Conference, CLEI 2015*, 2015.
- [23] Jorge Echeverria, Francisca Perez, Jose Ignacio Panach, Carlos Cetina, and Oscar Pastor. The Influence of Requirements in Software Model Development in an Industrial Environment. *International Symposium on Empirical Software Engineering and Measurement*, 2017-Novem:277–286, 2017.
- [24] Giovanni Giachetti, Beatriz Marín, Lidia López, Xavier Franch, and Oscar Pastor. Verifying goal-oriented specifications used in model-driven development processes. *Information Systems*, 64:41–62, 2017.
- [25] Alberto Rodrigues Da Silva. Linguistic patterns and linguistic styles for requirements specification (I): An application case with the rigorous rsl/business-level language. *ACM International Conference Proceeding Series*, Part F1320(I):1–27, 2017.
- [26] Saurabh Tiwari, Deepti Ameta, and Asim Banerjee. An approach to identify use case scenarios from textual requirements specification. *ACM International Conference Proceeding Series*, 2019.
- [27] Huihui Zhang, Tao Yue, Shaikat Ali, and Chao Liu. Towards mutation analysis for use cases. *Proceedings - 19th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS 2016*, pages 363–373, 2016.
- [28] Romain Pinquie, Philippe Véron, Frédéric Segonds, and Nicolas Croué. A requirement mining framework to support complex sub-systems suppliers. *Procedia CIRP*, 70:410–415, 2018.
- [29] Konstantinos Mokos and Panagiotis Katsaros. A survey on the formalisation of system requirements and their validation. *Array*, 7(January):100030, 2020.
- [30] Akiyuki Takoshima and Mikio Aoyama. A Design Method for Domain-Specific Models of Software Requirements Specification Based on Stakeholders’ Concerns. *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, 2018-Decem:542–550, 2018.
- [31] Mehdi Rajabi Asadabadi, Elizabeth Chang, Ofer Zwi-kael, Morteza Saberi, and Keiran Sharpe. Hidden fuzzy information: Requirement specification and measurement of project provider performance using the best worst method. *Fuzzy Sets and Systems*, 383:127–145, 2020.
- [32] M. Brian Blake, Iman Saleh, Yi Wei, Ian D. Schlesinger, Alexander Yale-Loehr, and Xuanzhe Liu. Shared service recommendations from requirement specifications: A hybrid syntactic and semantic toolkit. *Information and Software Technology*, 57(1):392–404, 2015.
- [33] Juliana Medeiros, Alexandre Vasconcelos, Carla Silva, and Miguel Goulão. Requirements specification for developers in agile projects: Evaluation by two industrial case studies. *Information and Software Technology*, 117:106194, 2020.
- [34] Jean-Louis Boulanger. Requirements Specification of a Software Application. *Certifiable Software Applications 3*, pages 43–73, 2018.
- [35] Mohamed Ghazel, Jing Yang, and El-Miloudi El-Koursi. A pattern-based method for refining and formalizing informal specifications in critical control systems. *Journal of Innovation in Digital Ecosystems*, 2(1-2):32–44, 2015.
- [36] Alberto Rodrigues Da Silva, João Fernandes, and Sofia Azevedo. Variability aspects at a textual requirements specification level. *Proceedings - 2017 IEEE 25th International Requirements Engineering Conference Workshops, REW 2017*, pages 240–247, 2017.

- [37] Umairah Anuar, Sabrina Ahmad, and Nurul A. Emran. A simplified systematic literature review: Improving Software Requirements Specification quality with boilerplates. *2015 9th Malaysian Software Engineering Conference, MySEC 2015*, pages 99–105, 2016.