

Guías para mejorar la calidad de especificaciones de requerimientos en lenguaje natural

Nicolás Simón Ruffolo¹ [0009-0001-1926-4984], Giuliana Maltempo¹ [0009-0005-7441-5828], Juliana Ailen Delle Ville¹ [0009-0007-7888-7544], Carlos Carden¹, Elías Roa¹, Enzo Giandon¹, Gregorio Ponce¹, Juan Escudero¹, Juan Ignacio Glassman¹, Martina Hübler¹, Mauricio Rodríguez Ricon¹, Sebastián Díaz¹, Sofía Celano¹, Carlos Nusch¹, Gabriela Perez¹, Leandro Antonelli^{1,2} [0000-0003-1388-0337]

¹Lifia, Facultad de Informática, Universidad Nacional de La Plata.

² CAETI, Facultad de Tecnología Informática, Universidad Abierta Interamericana.
{nruffolo, lanto, jdeleville, gmaltempo, gperez, cnusch}@lifia.info.unlp.edu.ar
{sofiac2507, eliasroa12, emilianoandres2010, mauri030901, juanignacioglassman, juanescudero328, cardencarlos90}@gmail.com.ar
{Enzogiandon, martina.hubler}@alu.ing.unlp.edu.ar
gregorioponce@hotmail.com.ar

Abstract. La calidad de los requerimientos impacta directamente en el desarrollo del software. Una especificación ambigua o incompleta puede generar un costo de retrabajo para corregir código incorrecto de entre 100 y 200 veces el esfuerzo invertido en especificar requerimientos. El lenguaje natural es el más utilizado para especificar requerimientos, dado que los requerimientos son el medio de comunicación entre los usuarios y el equipo de desarrollo. Ahora bien, es común encontrar diferentes defectos en especificaciones de lenguaje natural, tales como ambigüedad, falta de precisión o redundancia. La detección de estos defectos es muy útil, dado que permite reducir el retrabajo de errores. Este artículo tiene por objetivo presentar una serie de guías para revisar ciertos aspectos de oraciones escritas en lenguaje natural con el fin de mejorar la calidad de las mismas. Si bien las guías pueden ser consideradas como una lista de chequeo para el ingeniero de requerimientos, este artículo muestra una herramienta que utiliza Inteligencia Artificial y que implementa servicios para que las revisiones puedan ser reutilizables por otras aplicaciones.

Keywords: Calidad en Especificaciones, Procesamiento de Lenguaje Natural, Especificaciones Semiestructuradas.

1 Introducción

La ingeniería de requerimientos constituye una etapa crítica en el desarrollo de software, ya que en ella se establecen las bases para el diseño, implementación y

validación del sistema. Los errores introducidos en esta etapa son costosos de corregir en fases posteriores del ciclo de vida, pudiendo crecer exponencialmente hasta 200 veces cuando son detectados tras la entrega del sistema [1]. En consecuencia, mejorar la calidad de las especificaciones desde su origen resulta esencial tanto para la calidad del producto final como para la eficiencia del proceso de desarrollo.

En la práctica, una parte significativa de las especificaciones se denota en lenguaje natural o en formatos semi-estructurados derivados de este. El lenguaje natural constituye un posible medio más accesible para clientes, usuarios y expertos del dominio; sin embargo, presenta varias debilidades conocidas, tales como ambigüedades, redundancias, inconsistencias y variaciones terminológicas. Estas características pueden dar lugar a diferentes interpretaciones y dificultar la consolidación del conocimiento del dominio [2].

Si bien, el lenguaje natural presenta ventajas, con respecto a la comunicación al momento de especificar, las debilidades pueden afectar a la calidad de la especificación. Por ello, resulta útil contar con guías que permitan verificar la calidad de las especificaciones en lenguaje natural y a partir de ellas, realizar una revisión sistemática de las mismas.

En los últimos años, los modelos de lenguaje de gran escala (LLMs) y las técnicas de Procesamiento de Lenguaje Natural (NLP) adquirieron relevancia en el análisis automático de textos. Mientras que los LLMs destacan por su capacidad para generar y comprender lenguaje natural, sus resultados no siempre son completamente determinísticos ni fácilmente auditables por lo que resulta necesario técnicas más confiables y auditables. Es por ello que resultan útiles los métodos basados en NLP. Este último enfoque pasó de simbólicos y estadísticos a modelos neuronales basados en arquitecturas Transformer, mejorando significativamente tareas como clasificación, extracción de información y análisis semántico, y consolidándose como una tecnología relevante para fortalecer la calidad de las especificaciones [3][4]. En particular, el análisis sintáctico constituye una de las técnicas fundamentales de NLP para la representación estructural de oraciones. Los árboles sintácticos de constituyentes y de dependencias permiten modelar gráficamente las relaciones gramaticales entre los elementos de una oración, identificando funciones como sujeto, predicado, objeto y modificadores. Estas representaciones se utilizan como base para tareas posteriores de interpretación semántica y extracción de información. Particularmente, los árboles de dependencias resultan adecuados para capturar relaciones binarias entre palabras, lo que facilita la identificación de acciones y entidades relevantes en textos técnicos, como las especificaciones de requerimientos [5]. No obstante, estas técnicas son de propósito general y requieren criterios de adaptación para su aplicación efectiva sobre textos semi-estructurados de dominio específico. En consecuencia, surge la necesidad de definir criterios y estrategias que permitan orientar la utilización de técnicas de NLP para la verificación automática de calidad en especificaciones escritas en lenguaje natural.

En este contexto, el presente trabajo propone una serie de guías de revisión orientadas a evaluar la calidad de una especificación en lenguaje natural. Asimismo, se presenta un prototipo basado en servicios que implementa mecanismos automáticos de verificación asociados a las guías; permitiendo su integración y su reutilización desde distintos sistemas de software.

El resto del documento se organiza de la siguiente manera. La sección 2 presenta una breve discusión de trabajos relacionados. La sección 3 describe cada una de las guías y justifica la pertinencia de estas. La sección 4 presenta el funcionamiento y desarrollo de la herramienta. Finalmente, la sección 5 muestra las conclusiones y los trabajos futuros.

2 Trabajos Relacionados

La calidad de las especificaciones en lenguaje natural ha sido ampliamente estudiada en el ámbito de los requerimientos. En estudios iniciales se identificaron problemas recurrentes, como la ambigüedad léxica, el uso de términos subjetivos y estructuras sintácticas complejas que dificultan la interpretación y verificabilidad de los requerimientos [6].

El estándar IEEE 830 y su evolución en ISO/IEC/IEEE 29148 establecieron criterios de calidad para especificaciones, incluyendo corrección, consistencia, completitud y verificabilidad. Estos estándares proporcionan lineamientos generales, aunque no ofrecen mecanismos automáticos de análisis [7].

Diversos enfoques buscan reducir las ambigüedades mediante el uso de lenguajes controlados, restringiendo la gramática y el vocabulario de la lengua general con el fin de mejorar la precisión y la formalidad [8]. Si bien estos enfoques reducen las ambigüedades, suelen imponer restricciones que pueden resultar artificiales o poco naturales para los usuarios finales.

Dentro de la comunidad de la ingeniería de requerimientos se han propuesto métodos sistemáticos para estructurar el conocimiento del dominio a partir de lenguaje natural. Entre ellos se destacan los siguientes: los enfoques basados en escenarios para capturar el comportamiento del sistema [9]; el Language Extended Lexicon (LEL), orientado a modelar el vocabulario del dominio [10]; las Kernel Sentences como mecanismo para representar hechos elementales del dominio [11]; técnicas de derivación de modelos conceptuales a partir de texto, como los métodos de extracción automática de entidades y relaciones [12].

Investigaciones recientes exploran la extracción automática de conocimiento mediante NLP, incluyendo la detección de entidades compuestas y relaciones semánticas enriquecidas. Estos enfoques permiten construir modelos conceptuales semiautomáticos y detectar inconsistencias terminológicas [13].

En los últimos años, los modelos basados en Transformers han demostrado un alto desempeño en tareas de representación semántica y similitud textual [14]. Sin embargo,

la mayoría de estos enfoques se proyecta a tareas generales de NLP y no están específicamente adaptados al análisis de especificaciones semi-estructuradas en contextos de ingeniería de requerimientos.

Finalmente, cabe mencionar que, a pesar de los avances del modelado conceptual y en técnicas automáticas de extracción de información, persiste una brecha entre los métodos teóricos de estructuración del conocimiento y las herramientas existentes que buscan proporcionar una retroalimentación directa sobre la calidad del texto producido por los autores de requerimientos [15].

3 Guías de revisión

En la presente sección se presentan las guías propuestas para verificar la calidad de las especificaciones. Con estas se busca intentar lograr una especificación más clara. Cada guía se enfoca en un aspecto de la lengua que resulta problemático. En particular son voz pasiva, frase negativa compleja, expresiones abstractas, tiempos verbales, repetición de palabras, puntuación, clichés, expresiones de opinión y percepción, y legibilidad.

La elección de estas guías se fundamenta en principios teóricos provenientes de la lingüística y la pragmática. En particular, el concepto de economía del lenguaje, derivado de la Ley del mínimo esfuerzo propuesta por George Kingsley Zipf, sostiene que los individuos tienden a minimizar el esfuerzo en la comunicación, lo que puede derivar en expresiones menos precisas o potencialmente ambiguas [16]. En este contexto, la eficiencia comunicativa no solo depende de la reducción del esfuerzo, sino también del cumplimiento de ciertas normas pragmáticas. Paul Grice establece, a través del Principio de Cooperación, que los intercambios comunicativos deben regirse por criterios de cantidad, cualidad, relación y modo, mensajes claros, relevantes y ordenados [17]. A su vez, la Teoría de la Relevancia de Dan Sperber y Deirdre Wilson plantea que la comunicación es exitosa cuando se logra maximizar los efectos cognitivos con el menor esfuerzo de procesamiento posible [18].

En el contexto de la ingeniería de requerimientos, estos principios adquieren especial relevancia, ya que la tendencia natural a reducir el esfuerzo puede afectar negativamente la claridad y precisión de las especificaciones. En consecuencia, y en línea con estos fundamentos teóricos, se proponen las siguientes guías orientadas a detectar y mejorar la calidad de los requerimientos, reduciendo la ambigüedad y logrando una comunicación más efectiva [19].

Las siguientes subsecciones, describen en detalle las guías propuestas. Para cada una, se incluye una breve descripción del aspecto lingüístico y cómo su uso podría afectar la calidad de los requerimientos. Cada guía ejemplifica una oración con problemas y una posible reformulación de la misma. Para los ejemplos se utilizará como dominio un sistema para gestionar el riego de un cultivo de tomate. En el mismo los

procesos son automatizados y otros dependen del usuario que se va a ejemplificar con reglas de negocio.

3.1 Voz pasiva

La voz pasiva implica una reorganización de las funciones sintácticas: el objeto directo de la oración activa pasa a desempeñar la función de sujeto paciente, mientras que el sujeto de la activa puede realizarse como complemento agente, generalmente introducido por la preposición "por". En textos narrativos, la voz pasiva suele emplearse cuando el agente de la acción es desconocido, irrelevante o evidente por el contexto (por ejemplo, "El banco fue robado").

Sin embargo, en especificaciones técnicas y documentos de requerimientos, la omisión del agente puede introducir ambigüedad o imprecisión respecto de la responsabilidad o comportamiento esperado del sistema. Dado que una especificación debe ser precisa, verificable y no ambigua, el uso de la voz pasiva puede dificultar la identificación clara de actores, componentes o responsabilidades. Por ello, se recomienda priorizar construcciones en voz activa que mencionen el sujeto responsable de la acción.

Por ejemplo: "El quintero inicia el riego" es una oración en voz activa, dado que "el quintero" es el sujeto de la acción y de la oración. Ahora bien "El riego es iniciado (omisión de sujeto)" es una oración en voz pasiva, dado que "El riego" es el sujeto de la oración, pero no es quien realiza la acción.

3.2 Frases negativas complejas

Una frase negativa compleja es aquella en la que coexisten múltiples elementos de negación dentro de una misma oración, generando cancelación o ambigüedad semántica.

En especificaciones técnicas, la claridad y la precisión semántica son requerimientos fundamentales. La presencia de negaciones múltiples puede afectar la comprensibilidad del texto, aumentar la carga cognitiva del lector y generar interpretaciones divergentes. En el contexto de la ingeniería de requerimientos, estas deficiencias pueden llegar a impactar en inconsistencias en la especificación, errores durante la implementación, mayores dificultades en los procesos de validación y verificación; además, puede incrementar los costos asociados a la sobrecarga de trabajo y la falta de alineación entre usuarios. En la versión original, la combinación de múltiples negaciones incrementa la complejidad interpretativa; la reformulación afirmativa mejora la claridad y precisión del requerimiento.

Por ejemplo: "El quintero nunca deja de iniciar el riego." "El quintero no puede no iniciar el riego". Una reformulación posible podría ser la siguiente: "El quintero inicia el riego." "El quintero no inicia el riego". A partir de los ejemplos anteriores se puede observar que, en la versión original, la combinación de múltiples negaciones

incrementa la complejidad interpretativa, mientras que en la reformulación sugerida mejora la claridad y precisión del requisito.

3.3 Palabras abstractas

Las palabras abstractas son términos principalmente sustantivos, adjetivos y verbos que refieren a cualidades, estados, procesos o conceptos generales que no poseen una referencia concreta y observable directa.

En especificaciones técnicas, este tipo de vocabulario puede introducir ambigüedad o falta de verificabilidad, ya que conceptos como “eficiencia”, “adecuado”, “óptimo” o “mejorar” pueden admitir múltiples interpretaciones dependiendo del contexto. Uno de los principios fundamentales en la redacción de requerimientos es la verificabilidad. La utilización de términos abstractos sin criterios explícitos de medición dificulta la validación objetiva del requisito y puede generar interpretaciones subjetivas entre las partes interesadas. Tales expresiones como “el sistema debe ser eficiente” o “la interfaz debe ser intuitiva” requieren definición adicional para transformarse en requerimientos medibles. En consecuencia, la identificación automática de palabras abstractas constituye una estrategia de apoyo para alertar al autor sobre posibles imprecisiones semánticas.

Por ejemplo, la siguiente frase contiene una imprecisión: “El sistema debe mejorar la eficiencia del riego del cultivo.”. En esta oración no se especifica qué significa “mejorar la eficiencia”, ni bajo qué criterio se evalúa dicha mejora, por lo que la regla de negocio resulta abstracta. Una posible reformulación puede ser: ““El sistema debe activar el riego cuando la humedad del suelo sea inferior al 30% y mantenerlo durante 15 minutos por sector.”. Esta versión resulta más clara y precisa ya que define condiciones observables y medibles dentro del dominio. A partir de este ejemplo, se puede observar que los términos “mejorar” y “eficiencia” presentan un grado de abstracción que requiere una definición cuantificable para asegurar la verificabilidad del requisito.

3.4 Tiempos verbales

En las especificaciones, suele preferirse el uso del tiempo presente o del futuro con valor prescriptivo, dado que los requerimientos describen comportamientos esperados y no acciones ya realizadas. El uso de tiempos pasados puede indicar descripciones narrativas, antecedentes históricos o ambigüedad respecto al carácter normativo del enunciado.

La redacción de requerimientos debe expresar comportamientos obligatorios o esperados del sistema, manteniendo claridad temporal y coherencia normativa. En el contexto de un sistema de riego para cultivo de tomate, el empleo de tiempos verbales en pasado puede generar confusión sobre si el requisito describe una funcionalidad ya ejecutada, un evento previo o una condición histórica. Esta ambigüedad afecta la

verificabilidad y puede dificultar la trazabilidad entre especificación, implementación y pruebas. Por ello, se recomienda evitar tiempos pasados en la formulación de requerimientos, salvo en secciones descriptivas o contextuales claramente delimitadas.

Por ejemplo: “El sistema activó el riego cuando la humedad del suelo fue inferior al 30%.”. Una posible reformulación puede ser: “El sistema activa el riego cuando la humedad del suelo es inferior al 30%.” En el primer caso, el uso del pretérito introduce una referencia temporal pasada que no resulta adecuada para la especificación de un comportamiento requerido. En cambio, la reformulación, en tiempo presente expresa de manera clara y normativa el comportamiento esperado del sistema dentro del dominio.

3.5 Repetición de palabras

En el contexto de especificaciones técnicas, la reiteración léxica puede afectar la claridad, la concisión y la calidad estilística del texto. Se considera repetición cuando una misma palabra aparece más de una vez en una misma estructura oracional sin que dicha repetición cumpla una función técnica deliberada.

La redacción de requerimientos debe ser precisa y concisa. La repetición no intencional puede introducir redundancia, aumentar la carga cognitiva del lector y afectar la legibilidad del documento. Si bien en algunos casos la reiteración puede ser necesaria para evitar ambigüedad referencial, en otros constituye un problema de estilo que puede resolverse mediante reformulación o sustitución léxica controlada. Detectar estas repeticiones permite asistir al autor en la mejora del texto sin alterar el significado técnico del requerimiento.

Por ejemplo: “El sistema activa la activación del riego en cada sector del cultivo.” En esta oración se observa la repetición en “activa” y “activación”, lo cual afecta la claridad del enunciado. Una posible reformulación podría ser: “El sistema activa el riego en cada sector del cultivo.” En esta versión se elimina la redundancia, manteniendo el significado original y mejorando la claridad y legibilidad del requisito.

3.6 Puntuación inusual

En especificaciones técnicas, la correcta utilización de la puntuación es esencial para garantizar claridad sintáctica y evitar interpretaciones erróneas. Errores formales en este nivel pueden afectar la legibilidad del texto y alterar su significado.

La puntuación cumple una función estructural en la delimitación de proposiciones, enumeraciones y condiciones. La ausencia de signos de apertura o cierre, el uso excesivo de símbolos o inconsistencias en el empleo de mayúsculas y minúsculas pueden generar ambigüedad o dificultar la comprensión del requerimiento. En documentos técnicos, donde la precisión lingüística es crítica, estas inconsistencias constituyen defectos de redacción que deben ser detectados y corregidos. La

identificación automática de patrones de puntuación inusual permite asistir al autor en la mejora formal del texto sin intervenir en su contenido semántico.

Por ejemplo: “El sistema debe activar el riego cuando la humedad del suelo sea menor al 30%, y cuando la temperatura supere los 28°C.” En esta oración se observa una coma duplicada que introduce ruido sintáctico y afecta la legibilidad del enunciado. Una posible reformulación podría ser: “El sistema debe activar el riego cuando la humedad del suelo sea menor al 30% y cuando la temperatura supere los 28°C.” En esta versión se corrige el uso de la puntuación, eliminando la redundancia y mejorando la claridad del requisito sin modificar su significado.

3.7 Clichés

Se entiende por cliché a una expresión lingüística repetitiva o estereotipada que, debido a su uso frecuente y genérico, aporta poca información específica al requerimiento. En el contexto de la ingeniería de requerimientos, los clichés suelen manifestarse mediante formulaciones amplias o frases de uso común que aparentan describir una necesidad del sistema, pero que no establecen criterios concretos ni verificables.

Las especificaciones técnicas deben priorizar la claridad, precisión y verificabilidad. Sin embargo, es frecuente encontrar expresiones como “de fácil utilización”, “amigable para el usuario”, “de alta calidad” o “con tecnología de última generación”, las cuales pueden resultar insuficientes desde el punto de vista técnico si no se acompañan de definiciones objetivas. Este tipo de formulaciones puede dar lugar a interpretaciones diferentes entre analistas, desarrolladores y usuarios del sistema. La detección de clichés permite identificar descripciones potencialmente vagas y promover su reformulación mediante criterios específicos y observables.

Por ejemplo: El sistema debe realizar un riego inteligente del cultivo.” En esta oración, la expresión “riego inteligente” constituye un cliché, ya que no especifica qué criterios utiliza el sistema para determinar el momento, duración o intensidad del riego. Una posible reformulación podría ser: “El sistema debe activar el riego cuando la humedad del suelo sea inferior al 30% y detenerlo al alcanzar el 45%.

3.8 Opinión y percepción

Se consideran expresiones de opinión aquellas que manifiestan creencias, juicios o valoraciones subjetivas (por ejemplo, “creer”, “considerar”, “opinar”, “parecer”), mientras que las expresiones de percepción refieren a procesos sensoriales o cognitivos (por ejemplo, “ver”, “notar”, “percibir”, “sentir”). En el contexto de especificaciones técnicas, este tipo de construcciones puede introducir subjetividad y afectar la precisión del requisito.

Los requerimientos deben formularse de manera objetiva y verificable, evitando depender de interpretaciones personales o estados cognitivos de los actores

involucrados. En un sistema de riego para cultivo de tomate, expresiones que apelan a opiniones o percepciones pueden generar ambigüedad, ya que no describen comportamientos observables del sistema sino valoraciones subjetivas del quintero u otros usuarios.

Por ejemplo: “El quintero percibe que el riego es suficiente” o “se considera que el sistema riega de manera adecuada” no establecen criterios medibles ni condiciones objetivas de validación. La identificación de este tipo de patrones permite advertir posibles desviaciones respecto de una redacción técnica formal. La siguiente regla de negocio contiene un verbo de percepción: “El quintero percibe que el nivel de humedad del suelo es adecuado.” En esta oración, el uso de “percibe” introduce un criterio subjetivo que no puede ser verificado de manera objetiva. Una posible reformulación podría ser: “El sistema debe indicar que el nivel de humedad del suelo es adecuado cuando se encuentre entre el 40% y el 60%.” En esta versión se reemplaza la percepción subjetiva por un criterio cuantificable y observable, mejorando la precisión y verificabilidad del requisito.

3.9 Legibilidad

La facilidad de lectura de un texto mediante el cálculo del índice de Fernández-Huerta es uno de los métodos de legibilidad más utilizados para el idioma español. Este índice constituye una adaptación de métricas clásicas de legibilidad y permite estimar el grado de complejidad lingüística de un texto en función de variables cuantificables, como la longitud de las oraciones y la cantidad de sílabas por palabra.

La claridad en la redacción de requerimientos no solo depende de la precisión semántica, sino también de su comprensibilidad general. En el contexto de un sistema de riego para cultivo de tomate, la presencia de oraciones extensas, múltiples subordinadas o estructuras complejas puede dificultar la interpretación de las reglas de negocio y aumentar el riesgo de errores en etapas posteriores del desarrollo. El índice de Fernández-Huerta proporciona un indicador cuantitativo que permite estimar qué tan fácil o difícil resulta comprender un texto escrito, ofreciendo además una clasificación aproximada del perfil de lector recomendado. La incorporación de esta métrica permite complementar las revisiones cualitativas con un criterio objetivo de evaluación.

Por ejemplo: “El sistema deberá activar el riego en aquellos sectores del cultivo donde, considerando las mediciones registradas por los sensores de humedad del suelo y las condiciones ambientales actuales, se determine que los niveles no son adecuados para el desarrollo óptimo de las plantas.” Una posible reformulación podría ser: “El sistema debe activar el riego cuando la humedad del suelo sea inferior al 30%.” En el primer caso, la longitud de la oración y la presencia de múltiples cláusulas subordinadas incrementan la dificultad de lectura. En cambio, la reformulación propone una estructura más simple y directa, lo que mejora la legibilidad y facilita la comprensión del requisito.

4 Herramienta

Esta sección describe la herramienta que implementa servicios que verifican la coherencia del texto con las guías propuestas. El resto de la sección se organiza de la siguiente manera. Inicialmente se describe la arquitectura y luego se detalla una descripción de cómo es su funcionamiento.

El servicio que se implementa es el que revisa si se cumple o no la guía para mejorar la calidad de las especificaciones, disminuyendo la ambigüedad y favoreciendo la precisión.

4.1 Arquitectura y diseño

En la figura 1 se tiene un resumen del funcionamiento de la arquitectura de microservicios que se utiliza para implementar las nueve guías mencionadas. En la misma se puede observar como el cliente interactúa con un API Gateways, y este último es responsable de realizar las diferentes llamadas a los servicios requeridos por el usuario.

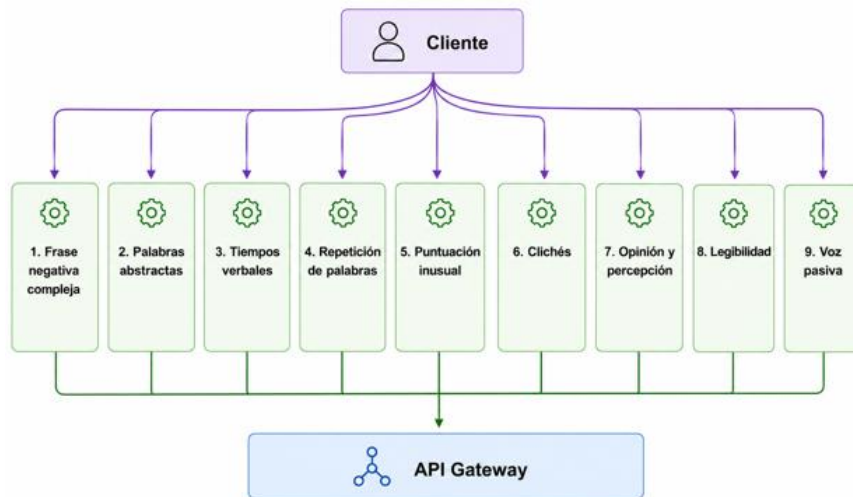


Fig. 1. Diagrama de arquitecturas

La herramienta fue implementada como una aplicación web basada en una arquitectura de microservicios. Su desarrollo se realizó utilizando Python y el framework Django para la construcción de los servicios y la lógica de negocio. La interfaz de usuario fue desarrollada mediante tecnologías web estándar, incluyendo HTML. Asimismo, para las tareas de procesamiento de lenguaje natural requeridas por las distintas guías, se empleó la biblioteca SpaCy [20], la cual proporciona funcionalidades para el análisis y procesamiento de texto.

4.2 Funcionamiento

El usuario tiene a disposición un formulario en donde ingresa un texto que es analizado mediante la invocación de distintos microservicios, los cuales procesan la información y generan recomendaciones orientadas a mejorar la calidad del contenido. La herramienta recibe como entrada el texto del usuario ingresado. El mismo será analizado por los diferentes microservicios seleccionados por el usuario los cuales devolverán una sugerencia. El usuario a partir de la sugerencia podría mejorar la especificación ingresada realizando los cambios necesarios.

Los microservicios implementados para analizar los inputs son los siguientes: (i) Frase negativa compleja. Este se encarga de identificar oraciones que contengan una doble negación. (ii) Palabras abstractas. Identifica sustantivos, adjetivos y verbos abstractos. Para ello se combinan reglas morfológicas, con análisis de similitud semántica basado en lemas y vectores de palabras. (iii) Tiempos verbales. El mismo realiza un análisis morfológico basado en un conjunto de patrones para capturar construcciones verbales compuestas y perífrasis frecuentes. (iv) Repetición de palabras. Se encarga de analizar el texto considerando el singular/plural de sustantivos e ignorar palabras frecuentes (artículos, preposiciones, conjunciones y pronombres) para detectar repeticiones innecesarias. (v) Puntuación inusual. Identifica errores en el uso de puntuación y mayúsculas, como signos de apertura/cierres faltantes, repetición excesiva de signos, uso incorrecto de mayúsculas/minúsculas o espaciado inadecuado. (vi) Clichés. Identifican clichés utilizando un glosario en combinación de lemas y patrones, de forma de identificar frases que contengan el cliché. (vii) Opinión y percepción. Identifica patrones que utilizan verbos que expresan opinión, (creencias o valoraciones) o percepción (captación sensorial o cognitiva. (viii) Legibilidad. Analiza la facilidad de lectura de un texto utilizando el índice de Fernández Huerta [21], uno de los métodos más usados para el idioma español. Además, indica el perfil de lector recomendado. (ix) Voz pasiva. Identifica construcciones en voz pasiva mediante patrones sintácticos y luego las convierte a voz activa. Las implementaciones de todos los servicios mencionado se encuentran en el repositorio de GitHub [22].

En la figura 2 se tiene un ejemplo concreto de un input de una oración ingresada. En esta oración se puede observar que la herramienta realiza un análisis microservicios y detecta errores. Estos se los muestra en el cuadro de texto inferior a modo de sugerencia.

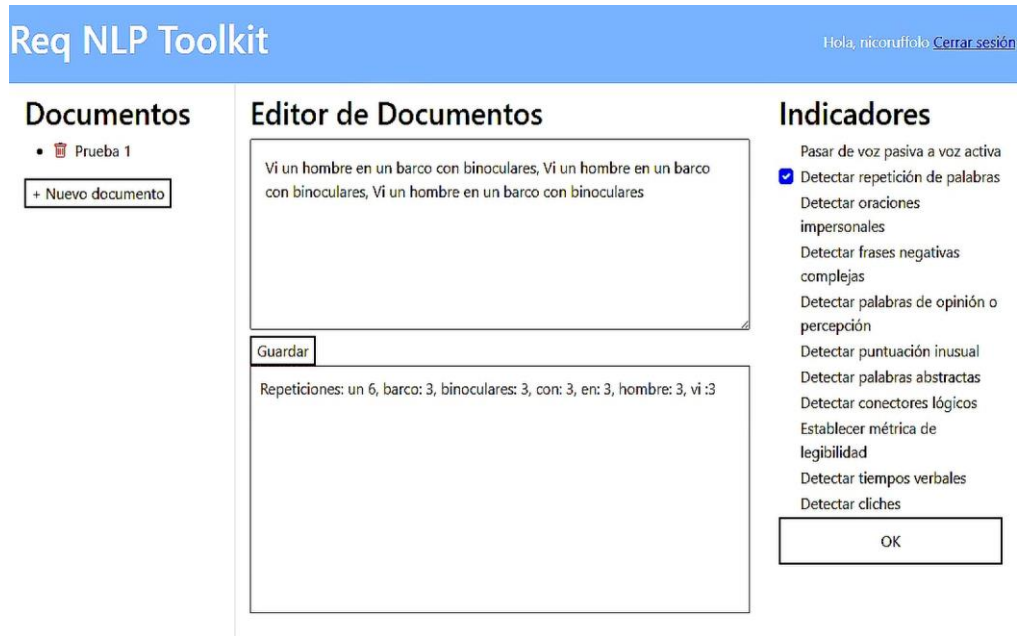


Fig. 2. Interfaz de la herramienta en funcionamiento.

5 Conclusiones

Este trabajo presentó un conjunto de 9 guías para revisar la calidad de las especificaciones escritas en lenguaje natural. Estas abordan diferentes problemas como ambigüedad, vaguedad, inconsistencias terminológicas, uso de voz pasiva, negaciones, repeticiones léxicas, puntuación inusual, palabras abstractas, tiempos verbales ambiguos, clichés, expresiones de opinión o percepción y niveles inadecuados de legibilidad. Además, se presenta un prototipo de herramienta que hace uso de servicios que implementan dichas guías.

El enfoque contribuye a mejorar la claridad, verificabilidad y mantenibilidad de los requerimientos [23], incorporando tantos análisis cualitativos basados en patrones lingüísticos como métricas cuantitativas, entre ellas el índice de legibilidad de Fernández-Huerta [21]. La arquitectura desarrollada permite analizar los textos sin modificarlos, garantizando un proceso no intrusivo y fácilmente integrable en distintos entornos de desarrollo.

Se presentan diversas fortalezas como la aplicabilidad directa en entornos reales de ingeniería de requerimientos. Independencia de herramientas específicas, facilitando su adopción en diversos contextos tecnológicos. Integración con enfoques basados en Léxico Extendido del Lenguaje (LEL) y modelado conceptual. Arquitectura modular y extensible que permite incorporar nuevas guías sin rediseñar el sistema. Utilidad en

contextos modernos como la generación automática de texto mediante inteligencia artificial, donde la revisión sistemática de calidad resulta especialmente relevante.

No obstante, la propuesta presenta ciertas limitaciones. Asimismo, el enfoque actual depende del idioma español y requiere adaptación para su aplicación en otros contextos lingüísticos. Finalmente, algunas detecciones basadas en patrones pueden requerir ajustes para dominios específicos, donde determinados términos o estructuras podrían ser válidos técnicamente.

Como líneas de trabajo futuro pensamos ampliar la evaluación empírica mediante estudios controlados con muestras más amplias y pertenecientes a diversos dominios. Asimismo, se prevé integrar técnicas de aprendizaje automático que permitan generar sugerencias automáticas de reformulación, extender el soporte a múltiples idiomas e incorporar mecanismos de detección semántica que reduzcan falsos positivos y mejoren la identificación de ambigüedades profundas. Adicionalmente, se plantea realizar una evaluación experimental más exhaustiva, mediante la incorporación de datasets específicos, métricas de desempeño y procesos de validación empírica. Finalmente se propone analizar la efectividad de las guías planteadas.

Referencias

1. Boehm, B. W.: Software Engineering Economics. Prentice-Hall, Englewood Cliffs, NJ (1981).
2. Sommerville, I.: Software Engineering. Pearson Education (2015).
3. Jurafsky, D., Martin, J. H.: Speech and Language Processing. Prentice Hall (2009).
4. Huang, X., Ruan, W., Huang, W., Jin, G., Dong, Y., Wu, C., Bensalem, S., Mu, R., Qi, Y., Zhao, X., Cai, K., Zhang, Y., Wu, S., Xu, P., Wu, D., Freitas, A., Mustafa, M. A.: A Survey of Safety and Trustworthiness of Large Language Models through the Lens of Verification and Validation. Artificial Intelligence Review, 57(175), Springer (2024).
5. Manning, C. D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999).
6. Shah, U. S., Jinwala, D. C.: Resolving Ambiguities in Natural Language Software Requirements: A Comprehensive Survey. ACM SIGSOFT Software Engineering Notes (2015).
7. IEEE Computer Society: IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications. IEEE (1998).
8. Aerospace and Defence Industries Association of Europe (ASD): ASD-STE100: Simplified Technical English – Standard for Technical Documentation (Issue 9). Brussels (2025).
9. McGraw, K. L., Harbison, K.: User-Centered Requirements: The Scenario-Based Engineering Process. CRC Press (1997).
10. Leite, J. C. S. D. P., Franco, A. P. M: A strategy for conceptual model acquisition. En Requirements Engineering. Proceedings of IEEE International Symposium on (pp.

243-246). Río de Janeiro, Brasil: Departamento de Informática, Pontificia Universidade Católica do Rio de Janeiro (1993).

11. Antonelli, L., Fernández, A., Ruffolo, N., Sansone, E., Torres, D.: A Collaborative Approach to Specify Kernel Sentences Using Natural Language. Workshop on Requirements Engineering (WER) (2022)
12. Maltempo, G., Dioguardi, F., Fernández, A., Torres, D., Tanevitch, L.: Multi-Word Entity Extraction and Rich Relationship Identification to Derive Conceptual Models from Natural Language Specifications. Workshop on Requirements Engineering (WER) (2024).
13. Dos Santos, C. R., Maldonado, J. C.: Analyze of Verbs in Requirements with Natural Language Processing to Enhance Comprehension. Workshop on Requirements Engineering (WER) (2004).
14. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics (2019).
15. Méndez Fernández, D., Wagner, S., Lochmann, K., Baumann, A., de Carne, H.: Field Study on Requirements Engineering: Investigation of Artefacts, Project Parameters, and Execution Strategies. *Information and Software Technology*, 54(2), pp. 162-178 (2012).
16. Zipf, G. K.: *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley, Cambridge, MA (1949).
17. Grice, H. P.: *Logic and Conversation*. In: Cole, P., Morgan, J. L. (eds.) *Syntax and Semantics*. Academic Press, New York, NY (1975).
18. Sperber, D., Wilson, D.: *Relevance: Communication and Cognition*. 2nd edn. Blackwell Publishers, Oxford, UK (1995).
19. Oo, K. H., Nordin, A., Ismail, A. R., Sulaiman, S.: An Analysis of Ambiguity Detection Techniques for Software Requirements Specification (SRS). *International Journal of Engineering & Technology* (2018).
20. Vasiliev, Y.: *Natural Language Processing with Python and SpaCy: A Practical Introduction*. No Starch Press (2020).
21. Fernández-Huerta, J.: Medidas sencillas de lecturabilidad. *Consigna*, 214, pp. 29-32 (1959).
22. <https://github.com/nicoruffolo/T-cnicas-de-NLP-y-AI-aplicadas-a-textos-semiestructurados>.
23. Wiegers, K., Beatty, J.: *Software Requirements*. 3rd ed., Microsoft Press (2013).