



Especificación de Requisitos de Inteligencia Artificial en el Desarrollo de productos de Software

Artificial intelligence in primary and secondary education in Ecuador

Autores

✉ * **Pablo Fernando Ordóñez Ordóñez**



✉ **Yamilka Valeria Erazo Aleaga**



Universidad Nacional de Loja, Facultad de la energía, las industrias y los recursos naturales no renovable, Carrera de Computación, Ecuador, Loja.

Comó citar el artículo:

Ordóñez Ordóñez, P. F. & Erazo Aleaga, Y. V. (2024). Especificación de Requisitos de Inteligencia Artificial en el Desarrollo de productos de Software. *Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones*, 8(2), pp. 134–146. <https://doi.org/10.33936/isrtic.v8i2.7147>

Enviado: 30/11/2024
Aceptado: 19/12/2024
Publicado: 23/12/2024

Resumen

En el desarrollo de tecnologías actuales que implican inteligencia artificial (IA), la especificación de requisitos adquiere una importancia crítica, pues garantiza que los sistemas cumplan con las expectativas funcionales, éticas y de rendimiento desde las etapas iniciales del proceso. Sin embargo, la ausencia de guías claras para la especificación de requisitos de la IA de un sistema, pueden llevar a una especificación deficiente, afectando la calidad y efectividad del software. Por lo tanto, este artículo propone un procedimiento formal para integrar los requisitos de IA en el proceso de desarrollo de software, utilizando el estándar IEEE 830 como base para la especificación. Esto se desarrolló en 2 fases haciendo uso de diferentes metodologías: la Revisión Sistemática de la Literatura (RSL) de Bárbara Kitchenham y GPEI, respectivamente. La primera fase consistió en la realización de una RSL mediante Parsifal para descubrir y analizar las prácticas actuales para la integración de requisitos de IA en el proceso de desarrollo de software, así como para la especificación de dichos requisitos. La segunda fase adaptó el estándar IEEE 830 para la especificación de requisitos de IA mediante la metodología GPEI de interacción con IA generativa. Los resultados mostraron que, en su mayoría, los proyectos de software que integran IA no cuentan con especificaciones adecuadas para los requisitos de IA, y, como respuesta, se diseñó una plantilla basada en el estándar IEEE 830 para guiar este proceso, que incluye dos tablas para la especificación de requisitos de IA, desarrolladas con IA generativa y análisis empírico.

Palabras clave: IEEE 830; requisitos de IA; IA generativa; ERS; GPEI, ChatGPT.

Abstract

In the development of current technologies involving artificial intelligence (AI), requirements specification becomes critically important to ensure that systems meet functional, ethical and performance expectations from the early stages of the process. However, the absence of clear guidelines for the AI requirements specification of a system can lead to poor specification, affecting the quality and effectiveness of the software. Therefore, this paper proposes a formal procedure for integrating AI requirements into the software development process, using the IEEE 830 standard as the basis for the specification. This was developed in 2 phases, making use of different methodologies: Barbara Kitchenham's Systematic Literature Review (SLR) and GPEI, respectively. The first phase consisted of conducting an SLR using Parsifal to discover and analyse current practices for the integration of AI requirements into the software development process, as well as for the specification of these requirements. The second phase adapted the IEEE 830 standard for AI requirements specification using the GPEI methodology for generative AI interaction. The results showed that, for the most part, software projects integrating AI do not have adequate AI requirements specifications, and in response, a template based on the IEEE 830 standard was designed to guide this process, which includes two tables for AI requirements specification, developed with generative AI and empirical analysis.

Keywords: IEEE 830; AI Requirements; generative AI; SRS; GPEI.



1. Introducción

A partir de los últimos años, el campo de la inteligencia artificial (IA) ha experimentado un avance significativo, transformando el desarrollo de software en un área con mejoras en la automatización y análisis de procesos mediante sistemas. El uso de la IA para este tipo de sistemas, entre otras innovaciones, ha beneficiado a diversos sectores, pero también, ha introducido complejidades adicionales, especialmente en la fase de ingeniería de requisitos (Acevedo-Castiblanco et al., 2023). La investigación en la ingeniería de requisitos ha avanzado significativamente, abordando la especificación de estos desde múltiples perspectivas, y, sabiendo que esta es una etapa crucial en cualquier proyecto de desarrollo, pues define lo que el sistema debe cumplir para ser aceptado como producto final, cualquier ambigüedad o falta de información en la especificación de requisitos puede comprometer el desarrollo completo del proyecto (de las Puebas Encinas & Sánchez Escribano, 2019). Estudios previos han explorado y desarrollado soluciones de software logrando resultados relevantes para la ingeniería de requisitos, pero limitados a la formalización específica de requerimientos de IA en sus productos. Por ejemplo, Abad et al. (2017) propusieron un enfoque basado en la clasificación automática de requisitos mediante técnicas de aprendizaje supervisado, mientras que Ramírez Leal et al. (2015) se centraron en el desarrollo de una metodología que aplica técnicas de inteligencia artificial para optimizar la comunicación entre los stakeholders y los ingenieros de requisitos, para la mejora del proceso de captura de requisitos.

A pesar de estos avances, persisten limitaciones importantes. Cuando los proyectos de software incorporan IA, los requisitos relacionados con esta tecnología suelen no estar claramente especificados o, en muchos casos, se omiten por completo. La ausencia de guías específicas para la documentación de estos requisitos que especifican las necesidades a cumplir a la hora de desarrollar esta parte del software, ha derivado en dificultades en el proceso de desarrollo, llevando consigo a problemas de calidad y eficiencias en los sistemas de IA implementados (Guevara & Ponjuán, 2022).

De esto se determina que la creciente demanda de soluciones basadas en IA ha hecho urgente la necesidad de integrar adecuadamente los requisitos de esta tecnología en los proyectos de software, lo que se refiere a una especificación precisa y clara de estos requisitos, ya que es esencial para garantizar el éxito de desarrollo del software que utilice IA, mejorando tanto la precisión como la adaptabilidad de los sistemas resultantes (Llerena Ocaña et al., 2023). Es por esto que, para abordar este problema, este estudio propone un procedimiento formal para la integración de los requisitos de inteligencia artificial en el proceso de desarrollo de software,

con un enfoque particular en la especificación, donde, la pregunta de investigación que guía este trabajo es: ¿Cómo se puede establecer un procedimiento formal para integrar los requisitos de IA en el desarrollo de software, asegurando una adecuada ingeniería de requisitos?

Para lograrlo, no bastó con la aplicación de teorías generales sobre la ingeniería de requisitos, pues fue necesario un enfoque original y adaptado a la naturaleza específica de los proyectos de IA. Mediante una revisión sistemática de la literatura, los hallazgos encontrados en 32 estudios, de un total de 163 revisados, constatan la importancia de usar formalmente una especificación de requisitos de IA en los productos de software, ya que solo el 46,86% de estos estudios seleccionados han abordado parte del proceso de ingeniería de requisitos dentro de proyectos de desarrollo con IA o que utilizan la inteligencia artificial para agilizarlo. Tal es el caso de trabajos enfocados en la adaptación de metodologías ágiles o modelos de trabajo, en las que se pueden incluir SCRUM o MLOps (Machine Learning Operations), para la inclusión de revisiones y ajustes iterativos en cada etapa del desarrollo, asegurando que los requisitos evolucionen en respuesta a cambios en el entorno operativo o en las necesidades del usuario, asegurando que el sistema de IA se mantenga relevante y efectivo a lo largo de su ciclo de vida (Pei et al., 2022; Subramanya et al., 2022).

Según lo obtenido, Tiwari & Malik (2024), entre otros autores, mencionan que las aplicaciones que integran inteligencia artificial más usadas dentro del desarrollo de software son aquellas que integran esta tecnología para mejorar la eficiencia y la calidad en los procesos de diferentes ámbitos. Estas aplicaciones incluyen sistemas de gestión de proyectos, análisis predictivo, ciberseguridad, herramientas de análisis de requisitos, sistemas de atención al cliente, aplicaciones de análisis de datos que permiten la toma de decisiones informadas en tiempo real, herramientas para la mejora de la comunicación, como chatbots (Rodríguez Vargas et al., 2024), junto con muchas otras.

De la mano de esta información recolectada, algunos autores, como Pei et al. (2022), han mostrado como cada vez son más los campos de conocimiento en los que la inteligencia artificial se ve involucrada y tiene una demanda por su necesidad, incluyendo la salud, educación y sector legal como los más destacados, pero también mencionando la ciberseguridad, finanzas, transporte, manufactura, mecánica, agricultura y demás.

Estos mismos autores han abordado el tema de la especificación de requisitos en proyectos que incorporan inteligencia artificial, a pesar de que las tendencias a la especificación formal de estos requisitos son muy escasas

según lo hallado. Ellos, junto con una variedad de autores, han expuesto cómo existe una incertidumbre inherente a los modelos de aprendizaje automático, la necesidad de iteración continua en la definición de requisitos y la importancia de la colaboración entre diferentes stakeholders. Se han analizado también técnicas y metodologías que pueden ser utilizadas para mejorar la captura y gestión de requisitos, destacando la necesidad de un enfoque flexible que se adapte a la naturaleza dinámica de los proyectos de IA. La escasez de procedimientos o guías se vio claramente en la investigación, pues, ninguno de los autores ha establecido una práctica clara, y, aunque entre ellos, como Necula et al. (2024), han revisado técnicas de inteligencia artificial para aplicarlas en la ingeniería de requisitos, los estudios encontrados han basado su atención en usar la IA para la mejora del proceso de recolección y análisis de requisitos de software, más no se han enfocado en los requisitos de la IA como tal.

Es por todo esto que, este trabajo presenta una contribución novedosa al abordar una guía práctica basada en el estándar IEEE 830, que incluye tablas para la especificación de requisitos funcionales y no funcionales que contemple los aspectos únicos de la IA desde las primeras etapas del desarrollo. A diferencia de investigaciones previas relacionadas con la integración de IA en el proceso de desarrollo de software, como la propuesta conceptual de Sarco Calapiña & Toaquiza Pacheco (2022) que presentó un enfoque que explora cómo las metodologías tradicionales y ágiles pueden adaptarse para incorporar algoritmos de IA, dando una visión interesante de cómo se puede adaptar el ciclo de vida del desarrollo de software (SDLC) para proyectos que utilizan IA, o la taxonomía propuesta por Steidl et al. (2023) para describir el desarrollo continuo de IA en cuatro etapas, este estudio introduce una formalización práctica de la especificación de requisitos como una solución innovadora que no solo llena un vacío en la literatura, sino que también ofrece una herramienta práctica para los equipos de desarrollo que buscan integrar IA de manera eficiente y estandarizada en sus proyectos.

Bajo este contexto, este artículo se estructura de la siguiente manera para comprender cómo se desarrolló el trabajo presentado: la Sección 2 muestra los materiales y metodologías utilizadas; la Sección 3 muestra los resultados obtenidos y su discusión; finalmente, la Sección 4 muestra las conclusiones a las que se llegaron.

2. Materiales y Métodos

Se contextualiza el estado en el que se encuentra la especificación de requisitos de inteligencia artificial en los proyectos que desarrollan software, además del proceso de creación de una guía de especificación de requisitos basada en el estándar IEEE 830 diseñada para integrar los requisitos de IA. Para la investigación, se realizó una revisión sistemática de la literatura (RSL) siguiendo la metodología de Barbara Kitchenham y Charters. Para el establecimiento de la guía, se utilizó la metodología GPEI, propuesta por Juan David Velásquez, Carlos Franco y Lorena Cadavid en su estudio en 2023, haciendo uso de la

versión libre de ChatGPT 4.0.

Revisión Sistemática de la Literatura

Los pasos para la realización de la revisión Sistemática de la literatura establecida por Kitchenham y Charters, especifica el desarrollo de 3 etapas, de las cuales se detallarán las dos primeras: planificación de la revisión y realización de la revisión (Kitchenham & Charters, 2007).

A. Primera Etapa: Planificación de la Revisión

Dentro de la planificación, se establecieron parámetros sobre los cuales se guiaría la investigación para la recolección de la información, es decir, para obtener estudios relevantes que aporten a la investigación y así poder determinar qué avances y qué métodos están actualmente vigentes para la especificación de requisitos de IA dentro de proyectos de desarrollo de software.

Lo primero que se determinó fue el protocolo de revisión, lo que incluyó el establecimiento de objetivos, marco PICOC, palabras clave y sinónimos, y preguntas de investigación. Por su relevancia, a continuación, se indican las preguntas de investigación planteadas:

- **Pregunta 1:** ¿Qué tipos de aplicaciones en el campo del desarrollo de software están utilizando inteligencia artificial?
- **Pregunta 2:** ¿En qué campos del conocimiento se está utilizando software con inteligencia artificial?
- **Pregunta 3:** ¿Cómo se están especificando los requisitos del software en proyectos que incorporan inteligencia artificial?

Con el protocolo ya establecido, se diseñaron las cadenas de búsqueda, que en este caso fueron 4, y se diseñaron mediante el uso de las palabras clave y sinónimos de las mismas:

- (“artificial intelligence” OR “AI” OR “machine learning” OR “deep learning”) AND (“software requirements” OR “requirements engineering” OR “software development process” OR “software development life cycle”) AND (“formal procedure” OR “formal process” OR “formal method” OR “integration” OR “incorporation”) AND (planning OR reviewing OR conducting OR reporting)
- (“artificial intelligence” OR “AI” OR “machine learning” OR “deep learning”) AND (“software requirements” OR “requirements engineering” AND (“software development process” OR “software development life cycle”) AND (“formal procedure” OR “formal process” OR “formal method” OR “framework” OR “approach”) AND (“integration” OR “incorporation” OR “effective integration”)
- (“artificial intelligence” OR “AI” OR “machine learning” OR “deep learning” OR “intelligent systems”) AND (“software requirements” OR “requirements engineering” OR “requirements specification” OR “requirements elicitation” OR “requirements analysis”) AND (“software development process” OR “software development life cycle” OR sdhc OR “software engineering”) AND (“formal procedure” OR “formal process” OR “formal method” OR framework OR approach OR “formal integration” OR “effective integration” OR incorporation) AND (planning





OR reviewing OR conducting OR reporting OR “systematic review” OR “literature review”)

• (“Artificial Intelligence” AND “software development” AND (“machine Learning” OR “computer visión” OR “Natural language processing”) AND “EcuCiencia”)

Lo siguiente fue definir las fuentes en las que se introducirían las cadenas planteadas para la búsqueda de estudios, determinando así: Scopus, Web of Science y Google Scholar. Como último paso de la planificación, se establecieron criterios de inclusión y exclusión, que se observan en la **Tabla 1**.

Tabla 1. Criterios de inclusión y exclusión
Fuente: Los autores

Criterios de Inclusión	Criterios de Exclusión
Publicaciones en inglés y español	Publicaciones duplicadas
Estudios que aborden metodologías para la especificación, elicitación, análisis y validación de requisitos de software	Estudios que no se enfoquen en la integración de inteligencia artificial en el desarrollo de software
Estudios que traten sobre la integración de requisitos de inteligencia artificial en proyectos de desarrollo de software	Estudios que no proporcionen resultados empíricos o teóricos relevantes
Trabajos publicados desde 2019 hasta la fecha presente	Estudios no disponibles

B. Segunda Etapa: Realización de la Revisión

Para realizar la revisión, se identificó la investigación, refiriéndose esto a la búsqueda e importación de los estudios en las fuentes seleccionadas mediante las cadenas diseñadas. Seguidamente, se seleccionaron los estudios relevantes según los criterios de inclusión y exclusión establecidos.

Una vez seleccionados los estudios relevantes para la investigación, se hizo una evaluación de calidad de los mismos, esto significó puntuar cada estudio entre 0 y 4. Para asignar el puntaje, cada estudio se sometió a 4 preguntas de calidad que se contestaron según la información proporcionada en él. Las preguntas fueron:

- ¿El estudio menciona uno o más softwares/herramientas/ modelos asistidos por Inteligencia Artificial?
- ¿El estudio aborda el uso de software con Inteligencia Artificial en algún campo o área de conocimiento específica?
- ¿El estudio menciona o discute el uso de alguna norma o estándar para el establecimiento de requisitos en proyectos de

software que incorporan Inteligencia Artificial?

• ¿El estudio seleccionado aporta información relevante para la integración efectiva de requisitos de inteligencia artificial en el proceso de desarrollo de software?

• Para contestar a cada pregunta, se establecieron 3 respuestas: Sí, No y Parcialmente. A cada respuesta se le asignó un peso o puntaje: si el estudio contestaba con “Sí” a una pregunta, se le daba un puntaje de 1; si el estudio contestaba con “No” a una pregunta, no se le daba un puntaje de 0; finalmente, si el estudio contestaba con “Parcialmente” a una pregunta, se le daba un puntaje de 0,5. Luego de contestar a cada pregunta, las respuestas sumaban sus puntajes para obtener el puntaje final del estudio y saber qué tan relevante era para la investigación.

Lo siguiente fue extraer los datos más relevantes de cada estudio, por lo que se extrajeron los datos de: Autor(es), Año, Tipo de aplicación/herramienta/modelo del que trataba, tecnología de desarrollo, Campo o área de conocimiento, Estándar o norma de requisitos, Metodología, Tipo de documento, Trabajo desarrollado, el uso o no de formalidad de requerimientos, y si era o no novel.

Metodología GPEI

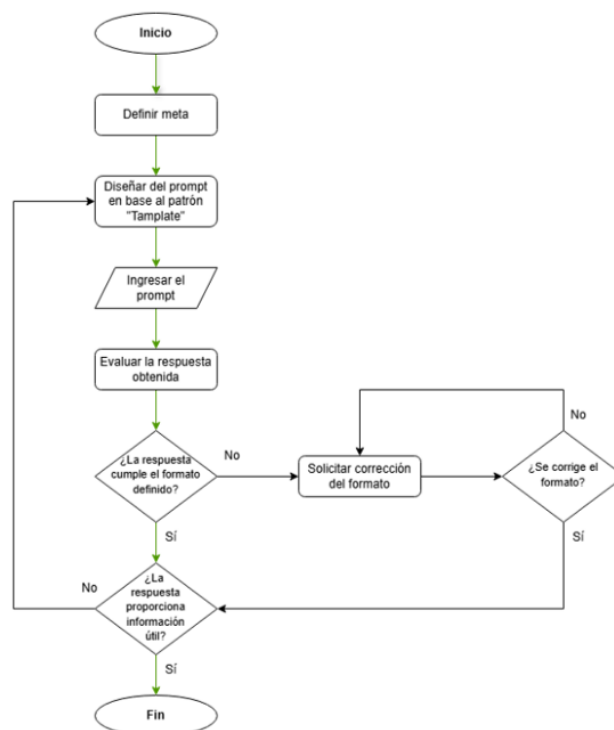


Figura 1. Proceso seguido para la creación de la tabla de especificación de requisitos en base a la metodología GPEI
Fuente: Los autores

La metodología GPEI debe su nombre a las siglas Goal-Prompt-Evaluate-Iterate (Velásquez Henao et al., 2023), por cada una de las etapas en la que consiste. En esta investigación, se aplicaron únicamente las tres primeras etapas, pues la cuarta, siendo opcional, no se vio necesaria. En la Figura 1 se puede observar un diagrama de procesos que representa el proceso que se estableció seguir en base a esta metodología, donde, las flechas marcadas en color verde, reflejan la secuencia seguida durante el desarrollo.

A continuación, se explica el proceso realizado en cada una de las etapas.

A. Definición de la meta

En la primera etapa se definió la meta de cumplir con el siguiente objetivo: “Crear el diseño de una tabla de especificación de requisitos para proyectos de software que incorporen inteligencia artificial, utilizando como base la tabla de especificación de requisitos del estándar IEEE 830”.

El objetivo fue el obtener un diseño de una única tabla, pues, aunque se esperó tener dos tablas diferentes, una para requisitos funcionales y otra para no funcionales, si la respuesta obtenida era aceptable, mediante análisis se añadirían las distinciones de cada tabla.

B. Diseño del Prompt

Se usó de uno de los patrones de diseño de prompts establecidos por Velásquez Henao et al.. El patrón seleccionado fue “Template”, dentro de la categoría “Output Customization”. La selección del patrón se hizo debido a que, este permitía al usuario definir el formato en el que deseaba la respuesta de la IA generativa. Por ende, el prompt diseñado fue el mostrado en la Figura 2:

Número de requisito	[Inserte aquí el texto] (Se refiere al ID del requisito, por ejemplo: RF01)
Nombre de requisito	[Inserte aquí el texto]
Tipo	☐ Requisito ☐ Restricción
Características	[Inserte aquí el texto] (Explicación básica del requisito, incluyendo de manera general lo que se espera de este requisito)
Descripción del requisito	[Inserte aquí el texto] (Explicación detallada y completa del requisito, incluyendo su funcionalidad y comportamiento esperado)
Fuente del requisito	[Inserte aquí el texto] (Especifica de dónde proviene el requisito, como usuarios, regulaciones o estándares industriales)
Requerimiento No Funcional (En caso de tratarse un requisito funcional)	[Inserte aquí el texto] (Se refiere a los requerimientos no funcionales con los que se relaciona el requisito)
Prioridad del requisito	☐ Alta/Eencial ☐ Media/Deseado ☐ Baja/ Opcional

En base a esta tabla del estándar IEEE 830, dime cómo podría modificarla para crear una plantilla que me sirva para especificar también requisitos de inteligencia artificial, tomando en cuenta que lo que quiero es integrar los requisitos de IA en el proceso de desarrollo de software. Debes mostrarme la tabla modificada en un formato igual que al de la imagen, es decir, en una tabla donde, a la derecha se muestren los parámetros de la tabla (los actuales y los añadidos), y a la izquierda se muestre el tipo de información que el usuario debe añadir para cada parámetro.

Figura 2. Prompt diseñado para la consulta en ChatGPT.
Fuente: Los autores

Este prompt permitió especificarle a la IA generativa, en este caso ChatGPT 4.0 versión libre, que como respuesta se requería una tabla de especificación de requisitos, basada en la tabla original del IEEE 830, que permitiera integrar parámetros cruciales relacionados con los requisitos de inteligencia artificial. Del mismo modo, se le pidió que estos parámetros se mostraran incluidos dentro de una tabla como la mostrada en la imagen.

C. Evaluación de la Respuesta

La respuesta obtenida por la IA se la puede observar en la Tabla 2.

Tabla 2. Respuesta de ChatGPT al prompt diseñado
Fuente: Los autores

Número de requisito	(Se refiere al ID del requisito, por ejemplo: RF01)
Nombre de requisito	
Tipo	☐ Requisito ☐ Restricción ☐ Requisito de IA ☐ Restricción de IA
Características	(Explicación básica del requisito de IA, incluyendo de manera general lo que se espera de este requisito)
Descripción del requisito	(Explicación detallada y completa del requisito de IA, incluyendo su funcionalidad, comportamiento esperado y métricas de rendimiento)
Fuente del requisito	(Especifica de dónde proviene el requisito de IA, como usuarios, regulaciones, estándares industriales o investigaciones académicas)
Requerimiento No Funcional	(Se refiere a los requisitos no funcionales con los que se relaciona el requisito de IA, como tiempos de respuesta, capacidad de procesamiento, interpretabilidad, etc.)
Prioridad del requisito	☐ Alta/Eencial ☐ Media/Deseado ☐ Baja/Opcional
Fuente de Datos	(Descripción de las fuentes de datos que se utilizarán para entrenar el modelo de IA, incluyendo calidad, cantidad, y representatividad de los datos)
Preprocesamiento de Datos	(Detalles sobre los procedimientos de preprocesamiento necesarios para los datos)
Métodos de Validación	(Descripción de los métodos de validación que se utilizarán para evaluar el modelo de IA)
Criterios de Éxito	(Especificación de los criterios de éxito y las métricas que deben cumplirse para considerar que el modelo de IA es adecuado para su uso)
Análisis de Sesgos	(Descripción de los posibles sesgos en los datos y en el modelo, y cómo se abordarán)

Esta respuesta se consideró aceptable en forma, ya que se apegó al formato especificado en el mensaje. Por otro lado, la respuesta fue tan completa como se esperaba, ya que incluyó los parámetros para los requisitos de inteligencia artificial en la tabla original, que sirvió como base para el diseño de las tablas de especificaciones que se desarrollarían más adelante. Como resultado, también fue tan precisa como se esperaba. En cuanto a la relevancia de la respuesta, se determinó que todos los parámetros enumerados eran relevantes y aplicables para especificar los requisitos de inteligencia artificial en los proyectos de software, ya que reflejan



la información que los proyectos deben documentar sobre la inteligencia artificial que se desarrollará. Por último, se observó que la respuesta no contenía elementos actualmente inexactos ni contextualmente inadecuados porque toda la información proporcionada se consideró precisa y dentro del contexto en el que se formuló la pregunta.

Como resultado, se obtuvo una base adecuada para diseñar las tablas de especificación de requisitos. Sin inconvenientes, este procedimiento se llevó a cabo solo una vez y no fue necesario repetirlo, y es por eso, que la 4ta fase de la metodología GPEI no se tomó en cuenta, pues no se necesitó de otra respuesta, y, por ende, no se necesitó repetir el proceso.

3. Resultados y Discusión

Se obtuvieron resultados tanto de la RSL como de la respuesta obtenida por ChatGPT a través del uso de la metodología GPEI. Por eso, a continuación, se muestran el análisis y síntesis de datos obtenidas de la RSL, así como la respuesta a las preguntas de investigación planteadas; luego, se expondrán las tablas de especificación de requisitos creadas, y, finalmente, se mostrarán el sitio web desarrollado con una plantilla dinámica.

Síntesis y análisis de datos de la RSL

Para el cálculo automático de los resultados estadísticos de los datos se utilizó Parsifal. El porcentaje de artículos encontrados por cada fuente fue el primer estadístico. El diagrama de sectores para estos porcentajes se muestra en la sección “a” de la Figura 3. Es posible observar que el 59,5% de los 163 artículos encontrados a través de las búsquedas fueron extraídos de Web of Science,

cuya fuente arrojó la mayor cantidad de resultados, mientras que el 39,3% fueron extraídos de Scopus, que se representa en rojo y tuvo el segundo mayor número de resultados. Debido a que solo se encontraron dos estudios, Google Scholar, representado en naranja, solo obtuvo el 1,2% de los resultados.

La siguiente estadística que se ofreció fue la cantidad de artículos que fueron encontrados y aceptados por cada fuente. En la sección “b” de la Figura 3 se observa el diagrama de barras, donde se muestran las cantidades de las tres fuentes. La cantidad de estudios recopilados de cada fuente se muestra en azul, mientras que la cantidad de estudios aceptados de esa fuente se muestra en rojo. Se extrajeron un total de 97 estudios de la Web of Science, pero solo se aceptaron 16. Se obtuvieron un total de 64 estudios de Scopus, de los cuales se aceptaron 14. Además, se extrajeron dos estudios de Google Scholar, ambos aceptados. En la sección de Referencias Bibliográficas, se pueden encontrar los 32 estudios aceptados y analizados.

La última estadística ofrecida fue la cantidad de artículos aceptados cada año. Esto se refiere a la cantidad de estudios publicados en cada uno de los seis años del intervalo de búsqueda (2019-2024). El diagrama de líneas generado para este análisis estadístico se muestra en la sección “c” de la Figura 3. Se puede observar que en 2019 se publicaron doce artículos, la cantidad aumentó a 20 en 2020 y luego a 28 en 2021, la cantidad disminuyó a 20 en 2022, coincidiendo con la cantidad reportada en 2020. El número de estudios aumentó a 28 en 2023, que fue el mismo que en 2021, y luego disminuyó a 20 en 2024, por lo que se puede demostrar que los resultados más altos encontrados se encontraron en las publicaciones publicadas en 2021 y 2023.

Preguntas de investigación RSL:

A. Pregunta 1: ¿Qué tipos de aplicaciones en el campo del desarrollo de software están utilizando inteligencia artificial?

Según la información obtenida de los estudios seleccionados, el uso de la inteligencia artificial en el desarrollo de software ha aumentado significativamente y ha dado lugar a un amplio abanico de aplicaciones. La aplicación de la inteligencia artificial en las diversas fases del desarrollo de software ha potenciado tanto la eficiencia como la calidad del producto final, enriqueciendo los modelos de requerimientos y mejorando la precisión de las estimaciones en proyectos de software (Devine et al., 2023; Laato et al., 2024). Por un lado, destaca el uso de herramientas de gestión asistidas por IA en el sector empresarial, que requieren sistemas que permitan a las empresas optimizar sus operaciones y apoyarlas en la toma de decisiones basadas en datos (Sarco Calpiña & Toaquiza Pacheco, 2022). Estas herramientas hacen uso de software de análisis predictivo, como

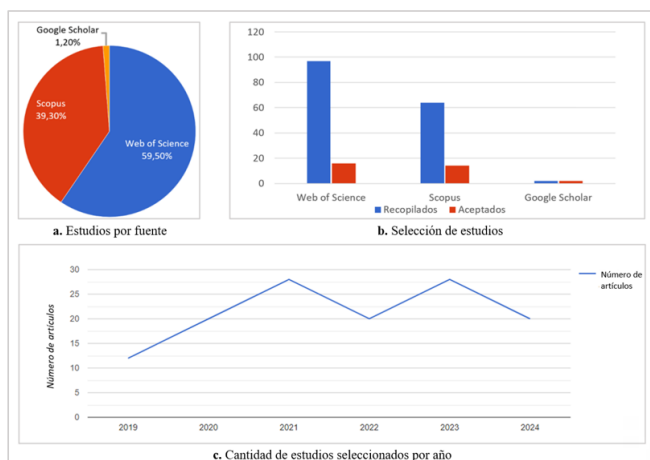


Figura 3. Bibliometría obtenida del análisis de datos.

Fuente: Los autores

es el caso de los sistemas de comercio electrónico que utilizan la inteligencia artificial para estimar precios y realizar predicciones más precisas (Barletta et al., 2023).

El software de análisis forense digital y el de monitorización de redes son dos tipos de aplicaciones en donde se está potenciando esta integración (Tiwari & Malik, 2024). En el ámbito de la ciberseguridad, este tipo de aplicaciones son utilizadas y necesarias para la detección de comportamientos anómalos en redes y sistemas, que supongan riesgos de seguridad (Subramanya et al., 2022). Adicionalmente, dentro del ámbito de la tecnología, se están desarrollando herramientas para el software de gestión, pero en esta ocasión se trata de dispositivos que utilizan IA con el fin de aumentar su eficiencia y automatización (Ranawana & Karunananda, 2021).

También se ha producido un cambio en las herramientas utilizadas en el proceso de desarrollo de software, como lo demuestra la proliferación de aplicaciones dedicadas al análisis de requisitos, modelado conceptual y extracción automatizada de requisitos (Abualhaija et al., 2020). Estas herramientas, que integran inteligencia artificial en sus operaciones, emplean técnicas como el procesamiento del lenguaje natural y el aprendizaje profundo para aumentar su eficiencia y precisión en estos procesos, en particular en la especificación de requisitos (Necula et al., 2024). En consonancia con esto, también se ha observado que se están desarrollando modelos de IA que categorizan el feedback de los usuarios para que el equipo de desarrollo pueda comprender mejor sus necesidades y expectativas como usuarios finales (Pei et al., 2022).

Sin embargo, fue posible destacar de la investigación que, a pesar de la existencia de diversos tipos de aplicaciones que representan avances significativos, la integración de la inteligencia artificial en el desarrollo de software aún enfrenta desafíos en términos de interpretabilidad, ética y confiabilidad (Villamizar et al., 2021).

B. Pregunta 2: ¿En qué campos del conocimiento se está utilizando software con inteligencia artificial?

Relacionado con la respuesta a la primera pregunta de investigación, el uso de software de inteligencia artificial se está dando en numerosos campos o áreas de conocimiento, donde se pueden encontrar las aplicaciones mencionadas anteriormente. Esta integración se ha extendido a una gran variedad de campos, demostrando su versatilidad y potencial de cambio en los sectores beneficiados. A continuación, se enumeran los campos que se encuentran con mayor frecuencia en las LSR que se realizan:

- El campo de la salud es digno de destacar. En este sentido, la IA se está utilizando para desarrollar herramientas de diagnóstico asistidas por computadora, como las que apoyan la detección temprana de la retinopatía diabética (Barletta et al., 2023; Pei et al., 2022). En este campo, este tipo de avances no solo mejoran la precisión del diagnóstico de los pacientes, sino que también tienen el potencial de aumentar el acceso a los servicios de salud en sectores con pocos recursos (Guizzardi et al., 2023).

- Como ya se ha comentado, el ámbito de la ciberseguridad se está incluyendo en estas implementaciones de software con IA. Existen sistemas diseñados para reforzar la defensa de otros sistemas ante ciberamenazas cada vez más sofisticadas (Tiwari & Malik, 2024), por ello, en este campo se utilizan software que detecta intrusiones, analiza la presencia de malware y predice vulnerabilidades del sistema, ya que utiliza la capacidad de la inteligencia artificial para procesar y analizar grandes volúmenes de datos en tiempo real para realizar detecciones (Devine et al., 2023).

- También se está utilizando en el sector energético, concretamente en el mercado electrónico, para ayudar a la predicción de precios y la optimización de la distribución de energía (Devine et al., 2023). De esta forma, mediante el uso de algoritmos de aprendizaje automático, se analizan los patrones históricos de consumo y producción, así como los factores externos relacionados con la demanda del mercado y el clima, lo que conduce a una gestión más eficaz de los recursos energéticos y a una mejor planificación a largo plazo (Laato et al., 2024).

- Otra industria que está incorporando inteligencia artificial es la manufacturera y la automotriz, en concreto en las áreas de software de control de calidad, optimización de procesos y diseño asistido por computadora (Abualhaija et al., 2020). Esto significa que mediante el uso de software con inteligencia artificial (IA) se pueden analizar datos en tiempo real para identificar fallas en la producción, mejorar la eficiencia operativa y facilitar el diseño de productos innovadores (Ahmad et al., 2023). Con estas aplicaciones, la automatización inteligente está avanzando en este campo y ayudando a la evolución del sector de la Industria 4.0. También juegan un papel importante en la toma de decisiones basadas en datos (Asaad & Avksentieva, 2023).

- Otra área de conocimiento en la que se están utilizando software de inteligencia artificial es en el ámbito jurídico, que ha sufrido una transformación con la introducción de estos sistemas, ya que se utilizan para la extracción automatizada de datos jurídicos y el análisis de documentos legales (Sleimi et al., 2021). Con estas herramientas, el ámbito jurídico se vuelve más eficiente para la investigación jurídica y, al mismo tiempo, facilita el acceso a información pertinente en grandes colecciones de documentos legales (Abualhaija et al., 2020; Abdullah et al., 2021).

- El ámbito educativo también se incluye en estos campos. El software habilitado con IA se está utilizando para adaptar las experiencias de aprendizaje a las necesidades de los estudiantes, analizar su desempeño y agilizar los procedimientos administrativos (Franch, 2021). Cabe destacar que, tras la pandemia de Covid-19, las herramientas de IA han experimentado un aumento significativo debido al aumento de las necesidades específicas de los estudiantes durante el tiempo de clase en entornos de aprendizaje en línea, lo que provocó un aumento en la demanda de sistemas más sofisticados para este propósito (Georgievski, 2023; Ranawana & Karunananda, 2021; Rahman et al., 2023).



C. Pregunta 3: ¿Cómo se están especificando los requisitos del software en proyectos que incorporan inteligencia artificial?

Debido a la naturaleza dinámica e impredecible de la inteligencia artificial, la especificación de requisitos en proyectos de desarrollo de software que involucran IA ha generado desafíos únicos porque los métodos utilizados para definir y documentar los requisitos varían enormemente (Sarco Calpiña & Toaquiza Pacheco, 2022). Si bien este proceso ha experimentado varios avances, es fundamental señalar que no existe una norma o una directriz ampliamente aceptada para esta especificación de requisitos de IA en proyectos que la utilizan. La información encontrada en los estudios seleccionados muestra que los proyectos y organizaciones de desarrollo tienen diferentes niveles de formalidad para especificar los requisitos, lo que deja espacio para establecer estándares, procedimientos o lineamientos que permitan mejorar las prácticas en esta materia y asegurar una mayor consistencia y calidad (Ahmed et al., 2022; Albán & Franco Ortega, 2024). Como resultado, en respuesta a los desafíos, se están adoptando diversas metodologías y enfoques para trabajar en la captura y gestión de los requisitos.

Existe una clara tendencia en el uso de metodologías ágiles, como SCRUM y AUP (Agile Unified Process), para proyectos de inteligencia artificial. Con el objetivo de adaptarse a la evolución de los sistemas de IA en función de su exposición a más datos y escenarios de uso, esta tendencia se está desarrollando cambiando las metodologías para incluir ciclos de retrofeedback más frecuentes y una mayor flexibilidad en la definición de requisitos. Este enfoque se traduce en una especificación de requisitos más flexible e iterativa, algo que se vislumbra esencial en el desarrollo de este tipo de sistemas (Rivero et al., 2020; Pei et al., 2022).

De manera similar, los ciclos de desarrollo de software se están desarrollando específicamente para sistemas que incorporan aprendizaje automático. Estos modelos de ciclo de vida intentan incorporar las características únicas del desarrollo de sistemas de IA, que incluyen aspectos como la gestión y preparación de grandes conjuntos de datos, la elección y modificación de modelos y la evaluación del rendimiento del sistema en diversos contextos (Althar & Samanta, 2021; Núñez et al., 2024).

Otra innovación en este campo que está ganando cada vez más popularidad es el uso de ontologías, mediante las cuales se utilizan técnicas basadas en ellas para el diseño de requisitos esotéricos para la IA en sistemas en desarrollo. El objetivo de este método es definir formal y estructuralmente los requisitos, que son un componente crítico de los sistemas de inteligencia artificial debido a su capacidad de tomar decisiones por sí mismos y las consecuencias que se derivan de ello. Estas técnicas basadas

en ontologías permiten una mejor consideración y representación de los requisitos estéticos de la IA que deben ser incorporados en el diseño y desarrollo del sistema (Franch et al., 2020; Shafiq et al., 2021).

Además, existen técnicas de procesamiento de lenguaje natural y de aprendizaje automático que se utilizan para extraer y clasificar automáticamente los requisitos. Estas técnicas, consideradas como una herramienta útil, permiten el análisis de grandes volúmenes de texto de los que se pueden extraer y categorizar requisitos potenciales. Estos volúmenes de texto son archivos que pueden incluir documentación como archivos de proyectos anteriores, comentarios de usuarios, especificaciones técnicas, etc. Aunque esta técnica no se centra en el proceso de especificación de requisitos, se desarrolla en este ámbito para identificar requisitos que de otro modo serían pasados por alto por los altos estándares de forma tradicional (Morales-Ramirez et al., 2019; Perez-Verdejo et al., 2020; Anish et al., 2019).

El uso de modelos de trabajo MLOps (Machine Learning Operations) ha ganado popularidad como una forma de incorporar consideraciones específicas de inteligencia artificial a lo largo del ciclo de desarrollo de software porque, además de abordar los requisitos iniciales, estos modelos también manejan la gestión continua de los mismos a medida que el sistema evoluciona y se delega a producción (Subramanya et al., 2022; Li et al., 2020).

Por último, cabe destacar que recientemente se ha puesto el foco en las especificaciones de requisitos, que en este caso no funcionan dentro de los proyectos de desarrollo de software, incluida la IA. Aquí, hay un enfoque específico en áreas como la robustez, la equidad y la interpretabilidad de los sistemas o modelos. Esta especificación de prácticas ha sido posible gracias a la creación de manuales de buenas prácticas para el desarrollo de software asistido por inteligencia artificial, que tienen como objetivo proporcionar pautas y estándares sobre cómo los equipos deben abordar estos aspectos (Albán & Franco Ortega, 2024; Ijaz et al., 2019).

Tablas de especificación de requisitos creadas

A. Tabla de especificación de requisitos funcionales

La tabla de requisitos funcionales se diseñó a partir de la Tabla 2, ya que sirvió como base para su desarrollo. A estos parámetros base se agregaron dos métodos manuales más: Flujo Normal y Flujo Alternativo. En consecuencia, el diseño final de la tabla cuenta con 16 parámetros, nueve más que la tabla que se muestra en el papel. El diseño de la tabla tal como se incorporó al estándar se muestra en la Figura 4.

La tabla desarrollada para requisitos funcionales con integración

de IA ofrece una estructura más detallada y ajustada a las necesidades específicas de los sistemas de IA que la considerada en la sección tradicional IEEE 830 sobre requisitos específicos. La sección “3.2 Requisitos funcionales” de la plantilla estándar original se vuelve innecesaria cuando se integran todos los parámetros mostrados en la Figura 4. Esta redundancia que surgiría de mantener tanto la sección tradicional de requisitos funcionales como la nueva tabla se eliminó con la implementación de una tabla tan específica que ya no era necesario tener una sección que replicara la misma información que se establecía en las tablas generadas para cada requisito funcional. Al combinar todos los datos en una sola tabla, se esperaba evitar la duplicación de datos dentro del mismo documento de especificación de solicitud y facilitar la gestión y revisión de las solicitudes.

Como se muestra en la Figura 4 el diseño de la tabla de requerimientos funcionales incluye dos parámetros de la tabla conocida como “especificación de casos de uso”: Flujo Normal y Flujo Alternativo. Al agregar estos parámetros, se indicaron y establecieron dentro de la misma tabla las tareas específicas que debe realizar cada requerimiento funcional, eliminando así la necesidad de brindar detalles cruciales de cada requerimiento en una sección separada y facilitando la comprensión de cada requerimiento al consultar una sola sección. Como resultado, tener una sección después y agregar nuevamente estos requerimientos en una sección diferente no solo sería redundante, sino que también se determinaría como un proceso ineficiente.

Número de requisito	[Inserte aquí el texto] (Se refiere al ID del requisito, por ejemplo: RF01)		
Nombre de requisito	[Inserte aquí el texto]		
Tipo	☐ Requisito de Software ☐ Restricción de Software ☐ Requisito de IA ☐ Restricción de IA		
Características	[Inserte aquí el texto] (Explicación básica del requisito, incluyendo de manera general lo que se espera de este requisito)		
Descripción del requisito	[Inserte aquí el texto] (Explicación detallada y completa del requisito, incluyendo su funcionalidad y comportamiento esperado)		
Fuente del requisito	[Inserte aquí el texto] (Especifica de dónde proviene el requisito, como usuarios, regulaciones o estándares industriales)		
Requerimiento No Funcional	[Inserte aquí el texto] (Se refiere a los requerimientos no funcionales con los que se relaciona el requisito)		
Prioridad del requisito	☐ Alta/Eencial ☐ Media/Deseado ☐ Baja/ Opcional		
Flujo Normal	Usuario: [Inserte aquí el texto] (Pasos detallados que describen el flujo normal de eventos dados por el usuario)	Sistema: [Inserte aquí el texto] (Pasos detallados que describen el flujo normal de eventos dados por el sistema)	
Flujo Alternativo	[Inserte aquí el texto] (Pasos detallados que describen las variaciones posibles del flujo normal) (El número de la acción alterna se compone del número de la acción normal de la cual sale el flujo alterno, más una letra en orden alfabético, por ejemplo: 1a, 1b, etc.)		
Fuente de Datos	[Inserte aquí el texto] (Descripción de las fuentes de datos que se utilizarán para entrenar el modelo de IA, incluyendo calidad, cantidad, y representatividad de los datos)		
Preprocesamiento de Datos	[Inserte aquí el texto] (Detalles sobre los procedimientos de preprocesamiento necesarios para los datos)		
Métodos de Validación	[Inserte aquí el texto] (Descripción de los métodos de validación que se utilizarán para evaluar el modelo de IA)		
Criterios de Éxito	[Inserte aquí el texto] (Especificación de los criterios de éxito y las métricas que deben cumplirse para considerar que el modelo de IA es adecuado para su uso)		
Análisis de Sesgos	[Inserte aquí el texto] (Descripción de los posibles sesgos en los datos y en el modelo, y cómo se abordarán)		
Impacto Ético	[Inserte aquí el texto] (Evaluación del impacto ético del uso del modelo de IA y medidas para mitigar cualquier efecto negativo)		

Figura 4. Tabla de especificación de requisitos funcionales
Fuente: Los autores

B. Tabla de Especificación de requisitos No Funcionales

Aun cuando fue diseñada de manera similar a la tabla anterior, esta tuvo varios cambios realizados específicamente para

especificar los requerimientos no funcionales. La tabla de requerimientos no funcionales fue diseñada con 14 parámetros, como se muestra en la Figura 5. De manera similar al diseño anterior, esta tabla fue incluida en la plantilla completa junto con una breve descripción de lo que se espera especificar dentro de cada uno de los parámetros de la misma. Un aspecto que se tomó en consideración en este diseño fue que, en el parámetro “Categoría”, se agregaron las clasificaciones de requerimientos no funcionales que se establecieron dentro de IEEE 830 porque la sección de la plantilla que se destinaba a esta clasificación, quedó omitida.

Esta tabla contiene toda la información pertinente, eliminando así la necesidad de repetir los detalles en una sección separada. Al concentrar toda la información en una sola tabla, se esperaba evitar la duplicación y facilitar la gestión y revisión de las solicitudes de una manera más efectiva. En otras palabras, se le dio la oportunidad al usuario de la plantilla de obtener toda la información necesaria y significativa sobre cada solicitud no funcional en una sección sin tener que consultarla en secciones separadas, con algunos datos en una sección y otros datos en otra. De esta manera, se determinó que la tabla diseñada optimizaría el proceso de revisión y gestión de las solicitudes no funcionales, y, es por esto, que la sección “3.3 Requisitos no funcionales” de la plantilla original del estándar IEEE 830, fue eliminada.

Al momento de eliminar esta sección se observó que el objetivo principal de la misma era clasificar los requerimientos no funcionales en sus seis categorías: confiabilidad, seguridad, disponibilidad, mantenibilidad y portabilidad. Al agregar las categorías de requerimientos no funcionales a la tabla diseñada, no sólo se optimizó el proceso de gestión, sino que también se determinó que la sección “3.3 Requisitos no funcionales” quedaría obsoleta e innecesaria, por lo que se consideró conveniente su eliminación.

Número de requisito	[Inserte aquí el texto] (Se refiere al ID del requisito, por ejemplo: RF01)		
Nombre de requisito	[Inserte aquí el texto]		
Tipo	☐ Requisito de Software ☐ Restricción de Software ☐ Requisito de IA ☐ Restricción de IA		
Características	[Inserte aquí el texto] (Explicación básica del requisito, incluyendo de manera general lo que se espera de este requisito)		
Descripción del requisito	[Inserte aquí el texto] (Explicación detallada y completa del requisito, incluyendo su funcionalidad y comportamiento esperado)		
Fuente del requisito	[Inserte aquí el texto] (Especifica de dónde proviene el requisito, como usuarios, regulaciones o estándares industriales)		
Categoría	☐ Rendimiento ☐ Seguridad ☐ Fiabilidad ☐ Disponibilidad ☐ Mantenibilidad ☐ Portabilidad		
Prioridad del requisito	☐ Alta/Eencial ☐ Media/Deseado ☐ Baja/ Opcional		
Fuente de Datos	[Inserte aquí el texto] (Descripción de las fuentes de datos que se utilizarán para entrenar el modelo de IA, incluyendo calidad, cantidad, y representatividad de los datos)		
Preprocesamiento de Datos	[Inserte aquí el texto] (Detalles sobre los procedimientos de preprocesamiento necesarios para los datos)		
Métodos de Validación	[Inserte aquí el texto] (Descripción de los métodos de validación que se utilizarán para evaluar el modelo de IA)		
Criterios de Éxito	[Inserte aquí el texto] (Especificación de los criterios de éxito y las métricas que deben cumplirse para considerar que el modelo de IA es adecuado para su uso)		
Análisis de Sesgos	[Inserte aquí el texto] (Descripción de los posibles sesgos en los datos y en el modelo, y cómo se abordarán)		
Impacto Ético	[Inserte aquí el texto] (Evaluación del impacto ético del uso del modelo de IA y medidas para mitigar cualquier efecto negativo)		

Figura 5. Tabla de especificación de requisitos no funcionales diseñada.
Fuente: Los autores

Los resultados obtenidos tras la realización de la RSL y el diseño de las tablas de especificación de requisitos, se explican a detalle en una web informativa desarrollada en el sitio web de la Carrera de Computación de la Universidad Nacional de Loja. La web se denominada “SRS-AI”, se desarrolló utilizando cuatro secciones principales: la primera sección expuso información respecto al estándar IEEE 830; la segunda sección explicó de qué trata la integración de IA al estándar IEEE 830; la tercera sección mostró la información respecto al presente trabajo realizado explicando las tablas de especificación de requisitos diseñadas, la eliminación de las secciones específicas del estándar IEEE 830; y por último, la cuarta sección expuso el equipo de investigación a cargo del proyecto. En el siguiente enlace, se puede acceder a la web: <https://computacion.unl.edu.ec/srs-ai>.

En la tercera sección de la página informativa, llamada “Plantilla de integración de requisitos de inteligencia artificial”, se añadió el enlace a una plantilla web dinámica donde el usuario puede desarrollar el documento completo de especificación de requisitos con integración de requisitos de inteligencia artificial; en la Figura 6, se puede ver el inicio de la aplicación, que, con más detalle, puede observarse en el siguiente enlace: <https://computacion.unl.edu.ec/srs/>.

Figura 6. Plantilla web “SRS-AI”.

Fuente: Los autores

Esta aplicación se desarrolló con todas las secciones que componen la guía de especificación de requerimientos, incluyendo campos para agregar texto, botones y tablas que permiten al usuario completar el documento según sus necesidades. Además, se agregó un botón para exportar el documento en formato PDF. También se agregó un botón flotante de acción (FAB) para que el usuario tuviera dos opciones más: Limpiar Plantilla y Cargar, por ejemplo: la opción “Limpiar Plantilla” permite al usuario eliminar toda la información agregada a la página; la opción “Cargar ejemplo” permitiría al usuario visualizar la hoja completa con información de un proyecto hipotético para poder visualizar qué tipo de información debe contener cada sección.

Utilidad percibida

En la Figura 7 se observa la evaluación obtenida de una encuesta realizada a los estudiantes de la Carrera de Computación de la UNL, resultando un promedio de 3,8 de utilidad que se considera alta en la escala del 1 al 5. Las preguntas fueron las siguientes:

P1. ¿Qué tan útil consideras la plantilla para estructurar, organizar y documentar requisitos funcionales y no funcionales? Ya sea de software o de IA.

P2. ¿Qué tanto consideras que el uso de esta plantilla contribuye a definir de manera clara y precisa los requisitos específicos de IA, asegurando su trazabilidad y adecuación a lo largo de todo el proceso de desarrollo?

P3. ¿Qué tan fácil te resultó entender y completar los campos de la plantilla (incluyendo los específicos para IA)?

P4. ¿Qué tan útil consideras que sería esta plantilla para estructurar y documentar requisitos en un proyecto real que incluya sistemas de IA?

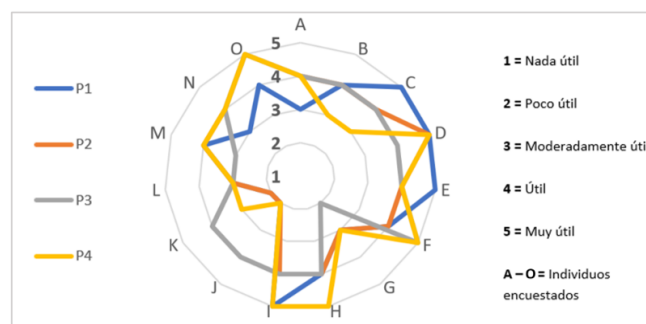


Figura 7. Evaluación de la utilidad percibida.

Fuente: Los autores

Asimismo, a la interrogante: ¿Alguna vez habías especificado requisitos de inteligencia artificial, ya sea de manera general o mediante un método formal? el 86.7% de los encuestados respondió que no había desarrollado anteriormente ningún tipo de especificación de requisitos de inteligencia artificial, esto demuestra que no se están usando estructuras formales en la especificación de requisitos en el proceso de desarrollo de software.

4. Conclusiones

Mediante la RSL realizada se diagnosticó un vacío de procedimientos formales y se usó el estándar IEEE 830 en conjunto con la metodología GPEI para solventar esta necesidad,

utilizando tablas de especificación de requisitos funcionales y no funcionales que incluyeran parámetros encargados de especificar las particularidades de la IA.

La eliminación de redundancias dentro de la guía planteada, fue un factor clave para la consolidación de la información relevante respecto a cada requisito, independientemente de su categoría, permitiendo optimizar su documentación, además de promover una mejor comunicación entre los clientes y partes interesadas, con el equipo encargado del desarrollo, lo que permite una mejor alineación con el cumplimiento de los objetivos de cada proyecto.

La creación de una web informativa y plantilla para la implementación de esta guía, refuerza la aplicabilidad práctica del procedimiento propuesto, estas herramientas no solo facilitan la adopción de estos procesos en entornos reales, sino que también contribuyen a la difusión de mejores prácticas en la industria que pueden mejorar la calidad de los sistemas desarrollados, reduciendo los riesgos asociados a especificaciones deficientes.

La principal limitación del estándar IEEE 830 es su independencia, ya que su enfoque rígido y detallado dificulta la integración con otros marcos y estándares más modernos, lo que limita su adaptación a metodologías ágiles y enfoques iterativos. Este obstáculo puede ser superado mediante el uso de normas más recientes como ISO/IEC 29148, que ofrece mayor flexibilidad y se integra eficazmente con otros estándares ISO, permitiendo una gestión más dinámica y colaborativa de los requisitos. Esto abre la puerta a trabajos futuros que favorezcan un enfoque más holístico, adaptable y alineado con las prácticas de desarrollo de software actuales.

Finalmente, el uso práctico de esta propuesta de procedimiento formal de especificación de requisitos para software con IA con los estudiantes en un entorno real se percibió una utilidad de 3.8/5 categorizándose como alta.

Contribución de los autores

Pablo Fernando Ordóñez Ordóñez: Conceptualización, Administración del proyecto, Análisis formal, Supervisión, Redacción- revisión y edición del artículo, Recursos. **Yamilka Valeria Erazo Aleaga:** Investigación, Análisis formal, Metodología, Software, Validación, Visualización, Redacción-borrador original del artículo, Redacción- revisión y edición del artículo.

Conflictos de interés

Los autores declaran no tener ningún conflicto de interés.

Referencias bibliográficas

Abad, Z. S. H., Karras, O., Ghazi, P., Glinz, M., Ruhe, G., & Schneider, K. (2017). What Works Better? A Study of Classifying Requirements. *Proceedings - 2017 IEEE 25th*

International Requirements Engineering Conference, RE 2017, 496–501. <https://doi.org/10.1109/RE.2017.36>

Abdullah, R. W., Ahmad, S., Asmai, S. A., Lee, S. W., & Zain, Z. M. (2021). Research Efforts and Challenges in Crowd-based Requirements Engineering: A Review. *International Journal of Advanced Computer Science and Applications*, 12(9), 395–402. <https://doi.org/10.14569/IJACSA.2021.0120945>

Abualhaija, S., Arora, C., Sabetzadeh, M., Briand, L. C., & Traynor, M. (2020). Automated demarcation of requirements in textual specifications: a machine learning-based approach. *Empirical Software Engineering*, 25(6), 5454–5497. <https://doi.org/10.1007/s10664-020-09864-1>

Acevedo-Castiblanco, J.-A., Suárez-Barón, M.-J., & González-Sanabriz, J.-S. (2023). Categorización e integración de columnas de opinión y contenido de páginas web aplicando técnicas de Procesamiento de Lenguaje Natural Categorization and Integration of Opinion Columns Content in Web Pages Applying Natural Language Processing Techniques. *Cómo citar. Artículo de Investigación Ingeniería y Competitividad*, 25(3), 25. <https://doi.org/10.25100/iyc.v25i3.13220>

Ahmad, K., Abdelrazek, M., Arora, C., Grundy, J., & Bano, M. (2023). Requirements Elicitation and Modelling of Artificial Intelligence Systems: An Empirical Study. <http://arxiv.org/abs/2302.06034>

Ahmed, S., Ahmed, A., & Eisty, N. U. (2022). Automatic Transformation of Natural to Unified Modeling Language: A Systematic Review. <https://doi.org/10.1109/SERA54885.2022.9806783>

Albán, H., & Franco Ortega, A. O. (2024). Manual de buenas prácticas para el desarrollo de software apoyado en inteligencia artificial para estudiantes de la Carrera de Software de la Universidad Estatal de Bolívar, período 2023-2024.

Althar, R. R., & Samanta, D. (2021). The realist approach for evaluation of computational intelligence in software engineering. *Innovations in Systems and Software Engineering*, 17(1), 17–27. <https://doi.org/10.1007/s11334-020-00383-2>

Anish, P. R., Sainani, A., Ahmed, A., & Ghaisas, S. (2019). Implementation-centric classification of business rules from documents. *Proceedings - 2019 IEEE 27th International Requirements Engineering Conference Workshops, REW 2019*, 227–233. <https://doi.org/10.1109/REW.2019.00047>

Asaad, J., & Avksentieva, E. (2023). Review of ways to apply machine learning methods in software engineering. *E3S Web of Conferences*, 449. <https://doi.org/10.1051/e3s-conf/202344907018>



- Barletta, V. S., Caivano, D., Gigante, D., & Ragone, A. (2023). A Rapid Review of Responsible AI frameworks: How to guide the development of ethical AI. *ACM International Conference Proceeding Series*, 358–367. <https://doi.org/10.1145/3593434.3593478>
- de las Puelas Encinas, G., & Sánchez Escribano, M. (2019). Definición de Requisitos Funcionales bajo Especificación IEEE para un Sistema de Ingeniería. Universidad Politécnica de Madrid.
- Devine, P., Koh, Y. S., & Blincoe, K. (2023). Evaluating software user feedback classifier performance on unseen apps, datasets, and metadata. *Empirical Software Engineering*, 28(2). <https://doi.org/10.1007/s10664-022-10254-y>
- Franch, X. (2021). Data-Driven Requirements Engineering: A Guided Tour. *Communications in Computer and Information Science*, 1375, 83–105. https://doi.org/10.1007/978-3-030-70006-5_4
- Franch, X., Seyff, N., Oriol, M., Fricker, S., Groher, I., Vierhauser, M., & Wimmer, M. (2020). Towards Integrating Data-Driven Requirements Engineering into the Software Development Process: A Vision Paper. In N. Madhavji, L. Pasquale, A. Ferrari, & S. Gnesi (Eds.), *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 12045, pp. 135–142). Springer International Publishing. <https://doi.org/10.1007/978-3-030-44429-7>
- Georgievski, I. (2023). Conceptualising Software Development Lifecycle for Engineering AI Planning Systems. *Proceedings - 2023 IEEE/ACM 2nd International Conference on AI Engineering - Software Engineering for AI, CAIN 2023*, 88–89. <https://doi.org/10.1109/CAIN58948.2023.00019>
- Guevara, J. D. C., & Ponjuán, D. T. (2022). Methodology proposal for specifying records management requirements from requirements engineering. *Investigacion Bibliotecologica*, 36(91), 33–48. <https://doi.org/10.22201/iib-24488321xe.2022.91.58555>
- Guizzardi, R., Amaral, G., Guizzardi, G., & Mylopoulos, J. (2023). An ontology-based approach to engineering ethicality requirements. *Software and Systems Modeling*, 22(6), 1897–1923. <https://doi.org/10.1007/s10270-023-01115-3>
- Ijaz, K. B., Inayat, I., & Allah Bukhsh, F. (2019). Non-functional Requirements Prioritization: A Systematic Literature Review. *Proceedings - 45th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2019*, 379–386. <https://doi.org/10.1109/SEAA.2019.00064>
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. <https://www.researchgate.net/publication/302924724>
- Laato, S., Mäntymäki, M., Minkinen, M., & Dennehy, D. (2024). Machine Learning System Development in Information Systems Development Praxis. *Communications of the Association for Information Systems*, 54. <https://doi.org/10.17705/1cais.05406>
- Li, M., Shi, L., Yang, Y., & Wang, Q. (2020). A Deep Multi-task Learning Approach for Requirements Discovery and Annotation from Open Forum. *Proceedings - 2020 35th IEEE/ACM International Conference on Automated Software Engineering, ASE 2020*, 336–348. <https://doi.org/10.1145/3324884.3416627>
- Llerena Ocaña, L. A., Viscaino Naranjo, F. A., Culque Topanta, W. V., & Baño Naranjo, F. P. (2023). Estudio neutrosófico sobre la afectación de la inteligencia artificial en el desarrollo de software. *Neutrosophic Computing and Machine Learning*, 29, 295–304.
- Morales-Ramirez, I., Kifetew, F. M., & Perini, A. (2019). Speech-acts based analysis for requirements discovery from online discussions. *Information Systems*, 86, 94–112. <https://doi.org/10.1016/j.is.2018.08.003>
- Necula, S. C., Dumitriu, F., & Greavu-Şerban, V. (2024). A Systematic Literature Review on Using Natural Language Processing in Software Requirements Engineering. In *Electronics (Switzerland)* (Vol. 13, Issue 11). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/electronics13112055>
- Núñez, A. G., Granda, M. F., Saquicela, V., & Parra, O. (2024). Machine Learning-Enhanced Requirements Engineering: A Systematic Literature Review. *International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE - Proceedings*, 521–528. <https://doi.org/10.5220/0012688100003687>
- Pei, Z., Liu, L., Wang, C., & Wang, J. (2022). Requirements Engineering for Machine Learning: A Review and Reflection. 166–175. <https://doi.org/10.1109/REW56159.2022.00039>
- Perez-Verdejo, J. M., Sanchez-Garcia, A. J., & Ocharan-Hernandez, J. O. (2020). A systematic literature review on machine learning for automated requirements classification.

- Proceedings - 2020 8th Edition of the International Conference in Software Engineering Research and Innovation, CONISOFT 2020, 21–28. <https://doi.org/10.1109/CONISOFT50191.2020.00014>
- Rahman, S., Ahmed, F., & Nayebi, M. (2023). Mining Reddit Data to Elicit Students' Requirements During COVID-19 Pandemic. Proceedings - 31st IEEE International Requirements Engineering Conference Workshops, REW 2023, 76–84. <https://doi.org/10.1109/REW57809.2023.00021>
- Ramírez Leal, J. C., Giraldo Orozco, W. J., & Hernández, R. A. (2015). Aproximación metodológica para la formalización de técnicas en Ingeniería de Requisitos [Methodological approach to formalization of techniques on requirements engineers]. Apropriación de Técnicas de Elicitación de Conocimiento y de Comunicación Personalizadas Para Potenciar El Proceso de Ingeniería de Requisitos, 32, 165–181.
- Ranawana, R., & Karunananda, A. S. (2021). An Agile Software Development Life Cycle Model for Machine Learning Application Development. 5th SLAAI - International Conference on Artificial Intelligence and 17th Annual Sessions, SLAAI-ICAI 2021. <https://doi.org/10.1109/SLAAI-ICAI54477.2021.9664736>
- Rivero, L., Diniz, J., Silva, G., Borralho, G., Braz Junior, G., Pava, A., Alves, E., & Oliveira, M. (2020, December 1). Deployment of a Machine Learning System for Predicting Lawsuits against Power Companies: Lessons Learned from an Agile Testing Experience for Improving Software Quality. ACM International Conference Proceeding Series. <https://doi.org/10.1145/3439961.3439991>
- Rodriguez Vargas, A. J., Falconí Peláez, S. V., Mazón Olivo, B. E., & Tusa Jumbo, E. A. (2024). Integración de Inteligencia Artificial en Telemedicina: Desarrollo y Evaluación de Chatbot Especializado en Enfermedades Virales. Informática y Sistemas Revista de Tecnologías de La Informática y Las Comunicaciones, 8(2), 60–69. <https://doi.org/10.33936/isrtic.v8i2.6840>
- Sarco Calapiña, W., & Toaquiza Pacheco, D. V. (2022). Aplicación de la inteligencia artificial en el desarrollo de software.
- Sarco Calpiña, W. M., & Toaquiza Pacheco, D. V. (2022). Aplicación de la inteligencia artificial en el desarrollo de software.
- Shafiq, S., Mashkoor, A., Mayr-Dorn, C., & Egyed, A. (2021). NL-P4IP: Natural Language Processing-based Recommendation Approach for Issues Prioritization. Proceedings - 2021 47th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2021, 99–108. <https://doi.org/10.1109/SEAA53835.2021.00022>
- Sleimi, A., Sannier, N., Sabetzadeh, M., Briand, L., Ceci, M., & Dann, J. (2021). An automated framework for the extraction of semantic legal metadata from legal texts. Empirical Software Engineering, 26(3). <https://doi.org/10.1007/s10664-020-09933-5>
- Steidl, M., Felderer, M., & Ramler, R. (2023). The pipeline for the continuous development of artificial intelligence models—Current state of research and practice. Journal of Systems and Software, 199. <https://doi.org/10.1016/j.jss.2023.111615>
- Subramanya, R., Sierla, S., & Vyatkin, V. (2022). From DevOps to MLOps: Overview and Application to Electricity Market Forecasting. Applied Sciences (Switzerland), 12(19). <https://doi.org/10.3390/app12199851>
- Tiwari, S., & Malik, A. (2024). Harnessing the Power of Artificial Intelligence in Software Engineering for the Design and Optimization of Cyber-physical Systems. <https://doi.org/10.1201/9781003474111-10>
- Velásquez Henao, J. D., Franco Cardona, C. J., & Cadavid Higuaita, L. (2023). Prompt Engineering: a methodology for optimizing interactions with AI-Language Models in the field of engineering. DYNA, 90(230), 9–17. <https://doi.org/10.15446/dyna.v90n230.111700>
- Villamizar, H., Escovedo, T., & Kalinowski, M. (2021). Requirements Engineering for Machine Learning: A Systematic Mapping Study. Proceedings - 2021 47th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2021, 29–36. <https://doi.org/10.1109/SEAA53835.2021.00013>

