UNIVERSIDAD NACIONAL DE MISIONES
+INGENIO Revista de Ciencia Tecnología e Innovación
FACULTAD DE INGENIERÍA UNaM
Vol. 6 N° 2, julio-diciembre 2025, pag. 74-89
ISSN 2618-5520 on line
DOI - https://doi.org/10.36995/j.masingenio.2025.06.02.005

Desafíos y Oportunidades de Ingeniería de Requerimientos asistida por LLM y *Prompt Engineering*

Fernando Corinaldesi [1, 2], Hernán Ariel Domínguez [1, 2], Briant Alcides Gauna [3, 4, *]

[1] *Universidad Nacional de José C. Paz. Departamento de Economía, Producción e Innovación Tecnológica. Buenos Aires. Argentina*

[2] *Universidad Nacional de La Plata. Facultad de Informática. Buenos Aires, Argentina.*

[3] *Universidad Nacional de Misiones-CONICET. Facultad de Ingeniería. Instituto de Materiales de Misiones (IMAM) Grupo de Investigación y Desarrollo en Electrónica (GIDE). Misiones, Argentina.*

[4] *Universidad del Gran Rosario. Espacio de Diseño y Tecnologías. Rosario, Argentina.*

e-mails: fernando.corinaldesi@docentes.unpaz.edu.ar, dhariel2904@gmail.com, br.gauna@fio.unam.edu.ar

Resumen

En el campo del desarrollo de software, la ingeniería de requisitos y la interacción con modelos de lenguaje como ChatGPT están ganando cada vez más relevancia. Este estudio explora la intersección entre los requerimientos de software y las técnicas de *prompt engineering* utilizadas con ChatGPT. A través de un mapeo sistemático de la literatura, se identifican los principales desafíos y oportunidades en la aplicación de ChatGPT para la elicitación, análisis y gestión de requerimientos. Los resultados destacan la importancia de desarrollar *prompts* efectivos para obtener información precisa y relevante de ChatGPT en el contexto de la ingeniería de requisitos.

Palabras Clave *– Ingeniería de Software, Ingeniería de Requerimientos, ChatGPT, LLM, Ingeniería de Prompts, Desafíos, Beneficios, Oportunidades, Mapeo Sistemático*.

Abstract

In the field of software development, requirements engineering and interaction with language models such as ChatGPT are gaining increasing relevance. This study explores the intersection between software requirements and prompt engineering techniques used with ChatGPT. Through a systematic mapping of the literature, the main challenges and opportunities in applying ChatGPT for requirements elicitation, analysis, and management are identified. The results highlight the importance of developing effective prompts to obtain accurate and relevant information from ChatGPT in the context of requirements engineering.

Keywords *– Software Engineering, Requirements Engineering, ChatGPT, LLM, Prompt Engineering, Challenges, Benefits, Oportunities, Systematic Mapping.*

1. Introducción

En los últimos años, los avances en el procesamiento del lenguaje natural (NLP) han allanado el camino para aplicaciones transformadoras en diversos ámbitos [1], como la atención médica, las finanzas, el comercio electrónico y las redes sociales. Entre estos avances, los modelos de lenguaje de gran escala (*Large Language Models*, LLMs) han surgido como herramientas poderosas con el potencial de revolucionar las prácticas de ingeniería de software. Una de las áreas que está experimentando un cambio significativo es la ingeniería de requisitos, especialmente con la introducción de modelos de lenguaje como ChatGPT [2].

*Autor en correspondencia: br.gauna@fio.unam.edu.ar / / Revista +Ingenio: masingenio@fio.una.edu.ar

ChatGPT es un chatbot de IA construido utilizando los modelos de lenguaje de gran escala (LLMs) de OpenAI, como GPT-4 y versiones anteriores [3]. Ha establecido nuevos estándares (tareas, métricas, etc.) en inteligencia artificial al demostrar la capacidad de las máquinas para comprender las complejidades del lenguaje y la comunicación humana.

Por otro lado, los requisitos de Software pueden definirse como descripciones de cómo debería comportarse el sistema o de una propiedad o atributo del sistema. Estos se definen durante las primeras etapas del desarrollo de un sistema como una especificación de lo que debe implementarse [4]. El área de la ingeniería de software encargada de esta tarea se conoce como “Ingeniería de requisitos”.

La ingeniería de requisitos es un término relativamente nuevo que se ha inventado para abarcar todas las actividades involucradas en descubrir, documentar y mantener un conjunto de requisitos para un sistema basado en computadora. El uso del término "ingeniería" implica que se deben utilizar técnicas sistemáticas y repetibles para asegurar que los requisitos del sistema sean completos, consistentes, relevantes [4]. Sin embargo, en la práctica, la ingeniería de requisitos enfrenta diversos desafíos y problemas:

- Los requisitos no reflejan las necesidades reales del cliente para el sistema.
- Los requisitos son inconsistentes y/o están incompletos.
- Es costoso realizar cambios en los requisitos después de que se han acordado.
- Hay malentendidos entre los clientes, los que desarrollan los requisitos del sistema y los ingenieros de software que desarrollan o mantienen el sistema [4].

La introducción de tecnologías de IA, particularmente los modelos de lenguaje de gran escala como ChatGPT, ofrece nuevas oportunidades para abordar estos desafíos [2].

Sin embargo, para aprovechar eficazmente estas herramientas, es necesario dominar lo que se conoce como "*prompt*”, esto se debe a que la calidad de la salida generada por un modelo de lenguaje depende en gran medida del *prompt* que recibió. Los *prompts*, en este contexto, pueden ser, por ejemplo, una frase, una pregunta o una instrucción dada al modelo en lenguaje natural [5].

Dado este contexto, emerge como disciplina, la Ingeniería de *Prompts* (PE, por sus siglas en inglés), la cual es un proceso que se centra en crear, optimizar y refinar *prompts* para garantizar la relevancia y calidad de la salida, que abarca la generación de información que se alinea con el propósito previsto y es lingüísticamente sólida. El uso de PE puede fortalecer la calidad de lo que un modelo de lenguaje interpreta, así como cualquier salida generada [6]. Para los LLMs, la PE es esencial, ya que la profunda comprensión lingüística de los LLMs plantea un desafío en la formulación de *prompts* de alta calidad, lo que a su vez afecta la calidad de cualquier salida generada.

El uso de la ingeniería de *prompts* (PE) potencia las capacidades de los LLMs para resolver tareas de procesamiento del lenguaje natural (NLP) a un nivel más avanzado. Dado que el procesamiento del lenguaje natural ha sido, y sigue siendo, utilizado para tareas de ingeniería de requisitos (RE), mientras que el lenguaje natural sigue siendo un componente central de la ingeniería de requisitos, surgió la idea de utilizar la ingeniería de *prompts* para los LLMs en la ingeniería de requisitos. Sin embargo, para que los LLMs sean útiles dentro de la ingeniería de requisitos, deben adquirir conocimientos sobre las actividades de ingeniería de requisitos y aprender el contexto aplicado [7].

Este estudio se centra en explorar cómo ChatGPT y las técnicas de *prompt engineering* pueden aplicarse en el contexto de la ingeniería de requisitos. Buscamos identificar las oportunidades que estas tecnologías ofrecen para mejorar la elicitación, análisis y gestión de requerimientos, así como

los desafíos que presentan. Además, investigamos las mejores prácticas emergentes en el uso de estas tecnologías para la ingeniería de requisitos.

La relevancia de esta investigación se fundamenta en la creciente adopción de tecnologías de IA en el desarrollo de software y la necesidad de comprender cómo estas herramientas pueden mejorar los procesos existentes. Al explorar la intersección entre la ingeniería de requisitos y las capacidades de ChatGPT, este estudio busca contribuir al cuerpo de conocimiento en ambos campos y proporcionar orientación práctica para los profesionales del software.

2. Metodología Aplicada

Con el fin de cumplir con los objetivos planteados, este estudio adopta un enfoque de mapeo sistemático de la literatura [11]. Este tipo de estudio permite obtener una visión global del estado actual de la investigación en un campo específico, identificando, clasificando y organizando los trabajos existentes en categorías previamente definidas.

El procedimiento seguido comprende la definición de preguntas de investigación, la búsqueda sistemática de literatura, la aplicación de criterios de inclusión y exclusión, la selección de estudios relevantes y la posterior organización y síntesis de la información a partir de un esquema de clasificación basado en palabras clave.

2.1. Definición de Preguntas de Investigación

El primer paso del mapeo sistemático consistió en la definición de un conjunto de preguntas de investigación, las cuales orientaron tanto la estrategia de búsqueda como el posterior proceso de clasificación y análisis de los estudios seleccionados.

Las preguntas de investigación se enfocan en comprender cómo los modelos de lenguaje de gran escala (LLMs), como ChatGPT, pueden contribuir a distintas actividades de la ingeniería de requerimientos, particularmente en la creación, análisis y validación de requerimientos de software.

Las preguntas clave formuladas fueron:

(PC1) ¿Cómo puede ChatGPT ayudar en la creación de requerimientos de software?

(PC2) ¿Qué desafíos presenta el uso de *prompts* para obtener información clara sobre los requerimientos?

(PC3) ¿Cómo se pueden mejorar los *prompts* para obtener mejores respuestas sobre los requerimientos?

(PC4) ¿Qué tan precisa es la información que proporciona ChatGPT al usar *prompts* sobre requerimientos?

2.2. Estrategia de Búsqueda

Se llevó a cabo una búsqueda sistemática de artículos a través de la plataforma IEEE Xplore, con el objetivo de identificar estudios relevantes sobre el uso de modelos de lenguaje de gran escala y/o ChatGPT en la ingeniería de requerimientos. De manera complementaria, se realizó una búsqueda adicional en la base de datos ACM Digital Library, con el propósito de reducir posibles sesgos de publicación y ampliar la cobertura de trabajos relevantes en el ámbito de la ingeniería de software.

En ambos casos, los artículos recuperados fueron evaluados de acuerdo con los criterios de inclusión y exclusión definidos. Aquellos estudios que no cumplían con dichos criterios, ya sea por falta de relevancia directa con la temática o por limitaciones de acceso al texto completo, fueron descartados.

2.3. Criterios de Inclusión y Exclusión

Con el objetivo de asegurar la relevancia y coherencia del corpus analizado, se definieron criterios de inclusión orientados a seleccionar estudios directamente vinculados con el uso de modelos de lenguaje de gran escala en el contexto de la ingeniería de requerimientos.

Se incluyeron aquellos estudios (***criterios de inclusión***) que:

a) Abordan el uso de modelos de lenguaje de gran escala (LLMs), tales como ChatGPT, GPT-3 u otros modelos similares, en el ámbito de la ingeniería de software, con énfasis en actividades relacionadas con la creación, elicitación o análisis de requerimientos de software.

b) Analizan el uso de *prompts* o técnicas de ingeniería de *prompts* para interactuar con modelos de lenguaje con el fin de obtener o procesar información asociada a requerimientos de software.

c) Evalúan aspectos vinculados con la precisión, los desafíos o las oportunidades en la interacción con modelos como ChatGPT dentro del contexto de la ingeniería de requerimientos.

d) Corresponden a publicaciones académicas revisadas por pares, tales como artículos científicos presentados en conferencias, *journals* o *workshops* relevantes en ingeniería de software y procesamiento de lenguaje natural.

e) Han sido publicados en los últimos cinco años, con el fin de reflejar el estado actual de las tecnologías basadas en LLMs.

f) Se encuentran disponibles en idioma inglés o español.

g) Han sido recuperados a partir de las búsquedas realizadas en IEEE Xplore y ACM Digital Library.

De manera complementaria, se definieron ***criterios de exclusión*** para descartar estudios que no se ajustaran al foco del mapeo sistemático. Se excluyeron aquellos trabajos que:

a) No abordan el uso de modelos de lenguaje de gran escala en el contexto de la ingeniería de requerimientos, o que se centran exclusivamente en dominios distintos (por ejemplo, *marketing*, atención al cliente o educación).

b) Analizan técnicas de inteligencia artificial o procesamiento del lenguaje natural sin considerar el rol de los *prompts* o su aplicación en ingeniería de requerimientos.

c) Han sido publicados con anterioridad al período temporal definido.

d) Se encuentran redactados en idiomas distintos del inglés o español.

e) No cuentan con acceso al texto completo o cuyo resumen no incluye términos clave relacionados con ChatGPT, ingeniería de *prompts*, desafíos u oportunidades en ingeniería de requerimientos.

2.4. Proceso de Selección de Estudios

El proceso de selección de estudios se llevó a cabo en múltiples etapas secuenciales, con el objetivo de identificar un conjunto de trabajos relevantes y coherentes con el alcance del mapeo sistemático.

En una primera etapa, se identificaron los estudios potenciales a partir de las búsquedas realizadas en las bases de datos seleccionadas. En esta fase inicial, se recuperaron todos los trabajos que coincidían con la cadena de búsqueda definida.

En una segunda etapa, se realizó un filtrado preliminar basado en la lectura de los títulos y resúmenes, con el fin de descartar aquellos estudios que no cumplían con los criterios de inclusión establecidos o que resultaban claramente irrelevantes para el foco del estudio.

Los trabajos que superaron esta etapa fueron sometidos a una evaluación completa, mediante la revisión detallada de su contenido, incluyendo introducción, metodología, resultados y conclusiones, a fin de verificar su alineación con los objetivos del mapeo.

Finalmente, los estudios seleccionados fueron considerados para la extracción y síntesis de información relevante, permitiendo identificar desafíos, oportunidades, enfoques metodológicos y líneas de investigación emergentes en el uso de modelos de lenguaje de gran escala y técnicas de *prompt engineering* en la ingeniería de requerimientos.

La Fig. 1 resume de manera gráfica las principales etapas del proceso de selección y organización de los estudios incluidos en el mapeo sistemático.

La cadena de caracteres usada para realizar el filtro y búsqueda es:

("ChatGPT" OR "GPT-3" OR "Large Language Models" OR "LLMs") AND ("software requirements" OR "requirements engineering" OR "elicitation of requirements" OR "prompt engineering" OR "prompt design") AND ("accuracy" OR "precision" OR "challenges" OR "improvement")

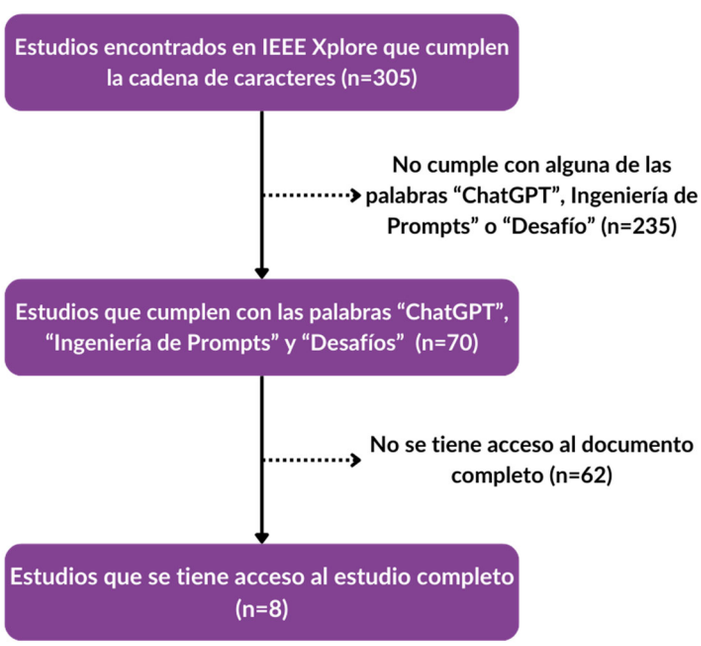

Fig. 1. Resumen del Mapeo Sistemático.

3. Estado de la Ingeniería de Requerimientos

A partir del análisis de los estudios seleccionados en el mapeo sistemático, se observa que la ingeniería de requerimientos (ER) es una disciplina clave dentro del desarrollo de software, centrada en definir y documentar las funcionalidades y comportamientos que un sistema debe cumplir para satisfacer las necesidades de sus usuarios.

3.1. Avances en la Automatización de la Ingeniería de Requisitos

Los estudios analizados en el mapeo sistemático evidencian que, en los últimos años, la automatización en la ingeniería de requerimientos ha ganado impulso, mediante el desarrollo de herramientas y técnicas orientadas a mejorar la calidad de los requisitos. Se han desarrollado herramientas y técnicas para mejorar la calidad de los requisitos mediante el uso de métodos como la validación formal, la verificación automática y los análisis de ambigüedad. Estos enfoques buscan minimizar los defectos antes de que los requisitos lleguen a las etapas de diseño y desarrollo.

La introducción de los modelos de lenguaje de gran escala, como GPT-3 y GPT-4, ha traído nuevas posibilidades al campo de la ingeniería de requisitos. Estos modelos, entrenados con grandes cantidades de datos textuales, son capaces de comprender y generar lenguaje natural, lo que los convierte en herramientas prometedoras para asistir en la detección de defectos en los documentos de requisitos.

Estudios recientes han evaluado empíricamente el desempeño de GPT-4 en tareas vinculadas al análisis estructural de artefactos asociados a requisitos, observándose que el modelo alcanza niveles cercanos a la corrección total en tareas formales basadas en reglas y estructuras lógicas predefinidas, aunque su rendimiento disminuye cuando se requiere interpretación semántica profunda o razonamiento contextual complejo [9]. Esta distinción resulta particularmente relevante en ingeniería de requerimientos, donde numerosos defectos surgen de ambigüedades conceptuales más que de errores sintácticos. Desde una perspectiva arquitectónica, además, la integración de modelos generativos en pipelines de análisis automático introduce nuevos puntos críticos de fallo, tales como segmentación inadecuada de documentos, deficiencias en el ranking semántico y persistencia de alucinaciones aun cuando la información relevante ha sido correctamente recuperada [10], lo que evidencia que la efectividad de la automatización no depende únicamente del modelo sino del diseño integral del sistema. En este contexto, también se han propuesto mecanismos de *structured requirement binding*, que incorporan identificadores estructurados y restricciones declarativas dentro del prompt para preservar la alineación semántica y facilitar la trazabilidad entre requisitos y artefactos generados [11], planteando una evolución hacia enfoques donde la trazabilidad se integra explícitamente en el proceso generativo.

3.2. Aplicación de IA en la Ingeniería de Requisitos

En relación con la precisión y confiabilidad de los modelos de lenguaje, uno de los ejes abordados en las preguntas de investigación, diversos estudios señalan que los LLMs, como ChatGPT, pueden ofrecer capacidades avanzadas para comprender el contexto y detectar defectos en los requerimientos. El uso de inteligencia artificial (IA) en la ingeniería de requisitos no es nuevo, pero los avances recientes en modelos de lenguaje han superado los enfoques anteriores. Tradicionalmente, las herramientas de procesamiento de lenguaje natural (NLP) se centraban en la detección de palabras clave o la implementación de reglas predefinidas para identificar posibles defectos en los documentos. Sin embargo, estos enfoques carecían de flexibilidad y no podían captar los matices del lenguaje natural. Investigaciones recientes han demostrado que los LLMs, como ChatGPT, pueden ofrecer una capacidad más avanzada para comprender el contexto y detectar defectos más sutiles.

Estudios han demostrado que GPT-4 tiene una precisión considerable en la detección de requisitos incompletos, aunque aún enfrenta desafíos al identificar ambigüedades e inconsistencias [12]. A pesar de estas limitaciones, se ha demostrado que los LLMs pueden reducir el esfuerzo humano en la validación de requisitos al automatizar parte del proceso.

En relación con la **PC1**, estos estudios muestran que ChatGPT puede asistir en actividades de creación y validación de requerimientos, particularmente mediante la automatización parcial del análisis. Respecto de la **PC4**, vinculada con la precisión de la información generada por ChatGPT, estudios reportan que GPT-4 presenta resultados favorables en la detección de incompletitud, aunque con limitaciones en otros tipos de defectos.

En línea con estos hallazgos, estudios experimentales recientes evidencian que el desempeño de GPT-4 varía según la naturaleza de la tarea analizada: el modelo presenta mayor precisión en tareas estructurales basadas en reglas explícitas que en aquellas que requieren interpretación semántica profunda o razonamiento contextual complejo [9]. Esta distinción resulta especialmente relevante para la ingeniería de requerimientos, donde muchos defectos se originan en ambigüedades conceptuales más que en errores formales. Asimismo, la evaluación de la precisión en sistemas basados en LLM enfrenta desafíos metodológicos, dado que las métricas automáticas tradicionales no capturan adecuadamente la corrección factual ni la adecuación contextual de las respuestas, particularmente en arquitecturas que combinan recuperación y generación [13]. A ello se suma que la integración de LLM en pipelines complejos puede introducir puntos críticos de fallo —como problemas de segmentación de documentos, ranking semántico deficiente y persistencia de alucinaciones— que afectan el desempeño en escenarios reales [13]. En conjunto, estos resultados muestran que la precisión observada en entornos controlados debe analizarse con cautela cuando los modelos se aplican en contextos prácticos de ingeniería de requerimientos.

En conjunto, los hallazgos de esta sección permiten abordar las **PC1** y **PC4**, vinculadas con el aporte y la precisión de ChatGPT en la ingeniería de requerimientos.

4. *Prompt Engineering* como Disciplina Emergente

A partir del análisis de los estudios incluidos en el mapeo sistemático, se observa que la calidad de las respuestas de los LLMs, como GPT-4, depende en gran medida de los *prompts*, o las instrucciones que se les proporcionan. En este contexto, la disciplina conocida como *Prompt Engineering* ha emergido para optimizar la interacción con estos modelos. *Prompt Engineering* involucra el diseño cuidadoso de las instrucciones que se proporcionan al modelo para maximizar la precisión, relevancia y utilidad de las respuestas generadas.

Los estudios revisados evidencian que el uso de técnicas de *Prompt Engineering* puede mejorar la precisión de los modelos en la *elicitación* de requisitos, particularmente mediante la incorporación de ejemplos específicos o contextos detallados en los *prompts*. Esto reduce la posibilidad de respuestas incorrectas o incompletas por parte de los LLMs.

En este sentido, investigaciones recientes proponen ir más allá de la simple redacción cuidadosa del *prompt*, incorporando lo que se denomina *structured requirement binding*, donde los requisitos se integran explícitamente dentro del prompt mediante identificadores formales, restricciones funcionales y contexto regulatorio. Este enfoque busca transformar el requisito en una restricción estructurada que guía la generación y mejora la alineación entre la salida del modelo y la intención original del artefacto [11]. De esta manera, el *Prompt Engineering* deja de ser únicamente una práctica heurística y comienza a consolidarse como una estrategia sistemática orientada a garantizar trazabilidad y consistencia en entornos asistidos por LLM.

4.1. Limitaciones en el Uso de Modelos LLM

En relación con la **PC2**, orientada a identificar los desafíos asociados al uso de *prompts*, los estudios analizados señalan que, a pesar del potencial de ChatGPT y otros modelos similares en la ingeniería de requisitos, existen limitaciones importantes que deben ser consideradas. Los LLMs, en su configuración estándar, no poseen conocimiento de dominio especializado y no siempre pueden manejar casos complejos que requieren una profunda comprensión técnica [12]. La incapacidad de los modelos para realizar razonamientos complejos sobre información técnica o para cruzar referencias dentro de un documento extenso sigue siendo un área de mejora clave.

Además, los modelos de lenguaje, en entornos *zero-shot*, pueden generar respuestas imprecisas si los *prompts* no son suficientemente detallados o claros. El uso de técnicas como *few-shot learning*, en las que se proporcionan ejemplos dentro del *prompt*, puede mejorar la precisión de las respuestas generadas por los LLMs.

Desde una perspectiva más amplia, también se ha observado que los problemas asociados al uso de prompts no se limitan únicamente a su redacción, sino que pueden estar vinculados a fallas estructurales en el pipeline de generación. En sistemas que combinan recuperación y generación, se han identificado puntos críticos como segmentación inadecuada de documentos, deficiencias en el ranking semántico y persistencia de alucinaciones incluso cuando se recupera información relevante [10]. Estos factores evidencian que la calidad de las respuestas no depende exclusivamente del prompt en sí, sino del diseño integral del sistema que soporta la interacción con el modelo, lo que complejiza aún más los desafíos asociados a la PC2.

En relación con la **PC2**, orientada a identificar los desafíos en el uso de *prompts*, los estudios analizados señalan que la falta de contexto y especificidad afecta directamente la calidad de las respuestas generadas. Estas limitaciones ponen de manifiesto que la efectividad de los LLMs en ingeniería de requerimientos depende en gran medida del diseño adecuado de los prompts, constituyendo uno de los principales desafíos identificados en el mapeo.

4.2. Impacto en la Práctica de la Ingeniería de Requisitos

Los estudios incluidos en el mapeo sistemático indican que el impacto del uso de ChatGPT y técnicas de *Prompt Engineering* en la práctica de la ingeniería de requisitos ha sido significativo. Estas herramientas no solo ayudan a automatizar la detección de defectos en los requisitos, sino que también tienen el potencial de acelerar la documentación y la verificación de requisitos. Sin embargo, la adopción completa de estas herramientas requiere una integración más fluida con las plataformas de gestión de requisitos tradicionales, como Jira, Trello o Confluence.

4.3. Herramientas y Frameworks que Integran IA Generativa

Desde una perspectiva aplicada, los estudios revisados describen diversas herramientas y *frameworks* [11] que comienzan a integrar IA generativa para apoyar la captura y gestión de requerimientos. GitHub *Copilot*, por ejemplo, utiliza modelos de lenguaje para sugerir código y documentación en tiempo real, lo que podría extenderse a la generación automática de especificaciones técnicas basadas en requisitos definidos en lenguaje natural. Otras herramientas, como Microsoft Azure DevOps, también están explorando la integración de IA para mejorar la colaboración y la documentación en el desarrollo de software.

En esta línea, propuestas recientes orientadas a la integración de LLM en entornos de desarrollo sugieren incorporar mecanismos de trazabilidad directamente dentro del flujo de generación,

incluyendo la inclusión automática de metadatos estructurados, identificadores de requisito y validación posterior mediante recuperación semántica [11]. Este enfoque permite no solo generar artefactos asistidos por IA, sino también mantener vínculos explícitos entre requisitos y salidas producidas por el modelo, facilitando su integración con herramientas tradicionales de gestión y control de versiones, y fortaleciendo la confiabilidad de la adopción práctica en contextos organizacionales.

En conjunto, los aspectos analizados en esta sección permiten dar respuesta a la **PC2**, vinculada con los desafíos del uso de *prompts*, y a la **PC3**, relacionada con las estrategias para mejorar la calidad de las respuestas generadas por los LLMs.

5. Desafíos y Beneficios del uso de ChatGPT en la Ingeniería de Requisitos

A partir de la síntesis de los estudios incluidos en el mapeo sistemático, se identifican una serie de beneficios y desafíos asociados al uso de ChatGPT en la ingeniería de requerimientos.

5.1. Beneficios

Mejora en la detección de defectos en requisitos

En relación con la **PC1**, orientada a analizar cómo ChatGPT puede asistir en la creación y análisis de requerimientos, los estudios revisados indican una mejora en la detección de defectos en requisitos. Los modelos de lenguaje como ChatGPT han demostrado ser útiles para detectar defectos comunes en los requisitos de software. En particular, pueden identificar ambigüedades cuando un requerimiento se puede interpretar de múltiples formas, lo que podría llevar a confusión durante la implementación. De manera similar, ChatGPT puede señalar inconsistencias cuando dos o más requisitos son contradictorios o incoherentes. Además, puede ayudar a detectar la incompletitud de los requisitos, es decir, cuando faltan detalles críticos, como rangos numéricos o especificaciones técnicas.

Capacidad para analizar grandes cantidades de documentación rápidamente

ChatGPT permite a los equipos de ingeniería de software procesar grandes volúmenes de requisitos de manera eficiente. En lugar de depender de revisiones manuales extensas, un modelo de IA puede realizar análisis rápidos y ofrecer sugerencias o correcciones inmediatas. Esto es especialmente útil en proyectos a gran escala, donde los requisitos a menudo abarcan cientos de páginas. Este beneficio refuerza el potencial de ChatGPT como herramienta de apoyo en actividades tempranas de la ingeniería de requerimientos, en línea con la **PC1**.

Apoyo en la automatización de tareas repetitivas

ChatGPT puede generar automáticamente secciones de documentos de requisitos basándose en ejemplos previos o especificaciones generales. Esto incluye la creación de plantillas para especificaciones de sistema, guías de usuario, o incluso la generación de escenarios de prueba. La automatización de estas tareas permite a los equipos concentrarse en tareas más complejas, como la toma de decisiones técnicas. La automatización de estas tareas constituye uno de los principales aportes identificados en la literatura analizada respecto del uso de ChatGPT en ingeniería de requerimientos (**PC1**).

Evidencia experimental reciente respalda estos beneficios al mostrar que modelos como GPT-4 pueden desempeñarse de manera consistente en tareas estructurales asociadas al análisis de artefactos

técnicos, alcanzando niveles elevados de corrección cuando se aplican reglas formales y patrones bien definidos [9]. Estos resultados sugieren que los LLM pueden actuar como asistentes eficaces en tareas de revisión preliminar, detección de omisiones estructurales y validación inicial de consistencia, reduciendo la carga cognitiva del ingeniero y acelerando ciclos tempranos de análisis de requisitos.

5.2. Desafíos

En relación con la **PC2**, centrada en los desafíos del uso de *prompts* para obtener información clara sobre los requerimientos, los estudios analizados destacan diversas limitaciones asociadas a los modelos de lenguaje.

Limitaciones de los modelos (falsos positivos, falta de entendimiento de dominio específico): A pesar de los avances, los modelos de lenguaje como ChatGPT no son infalibles [14]. Pueden generar falsos positivos, identificando defectos que en realidad no existen. Esto sucede porque carecen del conocimiento profundo de dominio necesario para comprender completamente los matices de ciertos requisitos técnicos. Los modelos tienden a basarse únicamente en patrones de lenguaje y no tienen una verdadera comprensión del contexto. Este aspecto pone de manifiesto que el diseño inadecuado de *prompts* constituye uno de los desafíos centrales identificados en el mapeo (**PC2**).

Dependencia de la calidad de los *prompts* (buen diseño de *prompts*): Los resultados que genera ChatGPT dependen en gran medida de la calidad de los *prompts* que se le proporcionan. Si los *prompts* son vagos o poco claros, las respuestas también lo serán. Esto introduce una nueva competencia en los equipos de ingeniería de software: la necesidad de diseñar *prompts* eficaces para obtener resultados útiles.

Falta de integración fluida en herramientas tradicionales de gestión de requisitos: Actualmente, muchas herramientas de gestión de requisitos no están diseñadas para integrarse fácilmente con modelos de lenguaje generativos como ChatGPT. Esto significa que los equipos deben trabajar en múltiples plataformas, lo que puede reducir la eficiencia y aumentar la posibilidad de errores. Esta limitación refuerza la necesidad de considerar la integración de los LLMs en el ecosistema de herramientas existentes, como uno de los desafíos recurrentes señalados en la literatura (**PC2**).

Estudios recientes han identificado múltiples puntos críticos en la ingeniería de sistemas basados en *Retrieval-Augmented Generation* (RAG). En particular, se describen fallos recurrentes como la ausencia de contenido relevante en la base de conocimiento, problemas en la segmentación de documentos (*chunking*), deficiencias en el ranking semántico y la persistencia de alucinaciones incluso cuando la información adecuada ha sido recuperada [13]. Asimismo, se ha evidenciado que la evolución de reglas de negocio y la actualización incompleta del conocimiento constituyen desafíos adicionales, ya que cuando la base histórica no refleja nuevas regulaciones el modelo puede generar respuestas incorrectas pese a que la arquitectura técnica funcione adecuadamente; además, la correcta contextualización depende de múltiples factores específicos que deben ser considerados explícitamente [15]. A ello se suma que la evaluación de la calidad en estas arquitecturas presenta dificultades metodológicas, dado que las métricas automáticas tradicionales no siempre capturan la adecuación contextual ni la corrección factual, lo que puede conducir a una sobreestimación de la confiabilidad del sistema [13]. En conjunto, estos hallazgos demuestran que la precisión en entornos de ingeniería de requerimientos no depende únicamente del modelo o del *prompt*, sino del diseño integral del *pipeline* de recuperación, actualización del conocimiento y validación humana.

En conjunto, los beneficios y desafíos identificados en esta sección permiten dar respuesta a la **PC1**, vinculada con el aporte de ChatGPT en la ingeniería de requerimientos, y a la **PC2**, relacionada con las dificultades y limitaciones en el uso de *prompts*.

6. Técnicas y Estrategias de *Prompt Engineering* para mejorar la Calidad de las Respuestas

En respuesta a la **PC3**, orientada a identificar cómo mejorar la calidad de las respuestas obtenidas mediante prompts, el análisis de los estudios incluidos en el mapeo sistemático propone diversas técnicas de *Prompt Engineering*. El *Prompt Engineering* juega un papel crucial en la optimización de las interacciones con modelos de lenguaje como ChatGPT. El diseño adecuado de un *prompt* (instrucción o consulta dada al modelo) puede marcar una gran diferencia en la calidad y relevancia de las respuestas generadas. Para garantizar que los resultados sean precisos y alineados con las necesidades de la ingeniería de requisitos, es esencial aplicar técnicas y estrategias efectivas al formular los *prompts* [10]. A continuación, se exploran algunos enfoques clave para mejorar la calidad de las respuestas generadas por modelos de lenguaje.

6.1. Tipos de Prompts Efectivos

Los estudios revisados coinciden en que los prompts efectivos son aquellos que son claros, específicos y ofrecen el contexto necesario para guiar al modelo hacia la respuesta deseada. Una de las características más importantes de estos *prompts* es la capacidad de eliminar la ambigüedad, asegurándose de que el modelo entienda claramente la tarea solicitada [16]. Existen varios tipos de *prompts* que pueden ser especialmente útiles en el contexto de la ingeniería de requisitos:

Prompts de Contexto Amplio:

Estos *prompts* proporcionan al modelo un marco contextual más completo, lo que le permite generar respuestas más precisas y relevantes al tener en cuenta información adicional. Por ejemplo, al preguntar sobre un conjunto de requisitos, un *prompt* como "En este documento de requisitos, ¿hay alguna inconsistencia entre la sección de seguridad y la de autenticación?" permite que el modelo considere más elementos del contexto general del documento antes de ofrecer una respuesta.

Prompts Condicionales:

Este tipo de *prompt* se utiliza cuando se desea que el modelo evalúe una condición específica antes de ofrecer una respuesta. Los *prompts* condicionales son útiles cuando se necesita una verificación basada en una acción o evento. Por ejemplo, un *prompt* como "Si el sistema no recibe una respuesta en 5 segundos, ¿qué debería ocurrir?" le permite al modelo procesar esa condición antes de sugerir un comportamiento o resultado.

Prompts de Verificación:

Los *prompts* de verificación son particularmente útiles para asegurar que ciertos aspectos críticos estén presentes en un documento o conjunto de requisitos. Permiten confirmar que se han incluido todos los elementos necesarios. Un ejemplo de este tipo de *prompt* sería: "Confirma si todos los requisitos de accesibilidad están correctamente especificados en este documento.". Esto orienta al modelo a buscar y confirmar la presencia de elementos clave en el texto.

En el contexto de la ingeniería de requisitos, la claridad, especificidad y amplitud de los *prompts* resultan esenciales para obtener resultados que realmente aporten valor. Estos enfoques permiten que

el modelo se concentre en las áreas más relevantes del texto y reduzcan el riesgo de generar respuestas ambiguas o incompletas [12].

Más allá de la claridad y especificidad textual, investigaciones recientes proponen estructurar los prompts incorporando identificadores formales de requisitos, restricciones explícitas y metadatos de trazabilidad, con el objetivo de guiar la generación de manera más controlada. Este enfoque, denominado *structured requirement binding*, integra dentro del *prompt* referencias directas a requisitos específicos (por ejemplo, mediante etiquetas estructuradas) y establece restricciones declarativas que reducen la variabilidad de salida y mejoran la alineación semántica entre el requisito y el artefacto generado [11]. De este modo, el *Prompt Engineering* evoluciona desde una práctica basada en heurísticas lingüísticas hacia una estrategia más sistemática orientada a garantizar consistencia, auditabilidad y estabilidad en entornos de ingeniería de requerimientos.

Estos tipos de *prompts* representan estrategias recurrentes identificadas en la literatura analizada para mejorar la calidad de las respuestas generadas por los LLMs (**PC3**).

6.2. Estrategias de Few-shot y Zero-shot

Dentro del conjunto de técnicas relevadas en el mapeo sistemático, se destacan dos estrategias comunes de *Prompt Engineering* para interactuar con modelos de lenguaje: *few-shot* y *zero-shot*. Estas estrategias permiten optimizar el rendimiento del modelo según el contexto y la información disponible:

Estrategia *Zero*-shot:

En la estrategia *zero-shot*, se le pide al modelo que realice una tarea sin haber recibido ejemplos específicos con anterioridad. Esto es útil cuando el modelo ya cuenta con un conocimiento suficientemente amplio del dominio o cuando se busca una interpretación general de los requisitos. En este caso, la calidad del *prompt* es clave, ya que no se le proporcionan ejemplos adicionales para contextualizar la tarea. Por ejemplo, un *prompt* del tipo "Detecta inconsistencias en este conjunto de requisitos" es una tarea *zero-shot* donde el modelo deberá interpretar la tarea sin ejemplos previos.

Estrategia Few-shot:

En contraste, la estrategia *few-shot* consiste en proporcionar uno o más ejemplos específicos que guíen al modelo en la tarea que debe realizar. Esto puede ser particularmente útil en situaciones más complejas o técnicas, donde el modelo puede beneficiarse de ejemplos que aclaren el formato o la lógica requerida para su respuesta. Por ejemplo, al analizar requisitos de seguridad, se pueden incluir ejemplos que muestren cómo deben formularse esos requisitos antes de pedirle al modelo que genere más. Esta estrategia mejora significativamente la precisión, especialmente cuando se trata de tareas complejas o contextos técnicos específicos.

Ambas estrategias permiten adaptar el modelo según el grado de especificidad o generalidad que se requiere en las respuestas, lo que mejora la efectividad en la generación de información relevante.

Evidencia experimental reciente respalda la utilidad de estas estrategias al demostrar que la incorporación de estructuras adicionales en el *prompt*, como reglas explícitas o razonamiento paso a paso, puede mejorar el desempeño del modelo en tareas formales. En evaluaciones controladas, el uso de enfoques estructurados permitió a GPT-4 alcanzar mayores niveles de corrección en comparación con configuraciones más simples, especialmente en tareas basadas en reglas definidas [9]. Estos resultados sugieren que la estrategia *few-shot*, combinada con indicaciones

estructuradas, resulta particularmente efectiva cuando la tarea requiere precisión técnica o cumplimiento de patrones específicos.

La comparación entre estrategias *few-shot* y *zero-shot* permite identificar patrones de mejora en la precisión y relevancia de las respuestas, constituyendo un aporte directo a la **PC3**.

6.3. Consideraciones Técnicas en el Diseño de Prompts

A partir de la síntesis de los estudios analizados, se identifican diversos factores técnicos que influyen en la calidad de los *prompts* y, en consecuencia, en las respuestas generadas por los modelos de lenguaje. Algunas de las consideraciones más importantes incluyen:

Longitud del *Prompt*:

Los prompts más largos tienden a proporcionar más contexto al modelo, lo que generalmente conduce a respuestas más detalladas y precisas. Sin embargo, es crucial equilibrar la cantidad de información proporcionada para evitar abrumar al modelo con detalles innecesarios. Por ejemplo, en el caso de requerimientos complejos, es preferible estructurar el *prompt* de manera que proporcione el contexto suficiente sin incluir información redundante.

Especificidad del Lenguaje:

El uso de un lenguaje técnico claro y preciso es fundamental para mejorar la exactitud de las respuestas. En la ingeniería de requisitos, términos vagos o ambigüedades en el lenguaje pueden llevar a respuestas incorrectas o irrelevantes. Un prompt específico que use términos técnicos claros permite que el modelo comprenda mejor la tarea y produzca una salida más alineada con las expectativas.

Formato Estructurado:

La manera en que se estructura un *prompt* puede influir en cómo el modelo procesa la información. Formular preguntas de manera lógica, o utilizar listas numeradas, ayuda al modelo a interpretar mejor la tarea solicitada. Por ejemplo, al solicitar que el modelo evalúe diferentes secciones de un documento de requisitos, un *prompt* estructurado en secciones o pasos específicos facilita la comprensión y mejora la precisión en las respuestas.

Además de estos factores, estudios recientes destacan que la estabilidad de las respuestas puede verse afectada por la naturaleza estocástica del modelo, incluso cuando el *prompt* se mantiene constante. Se ha observado que la generación puede variar entre ejecuciones sucesivas, lo que introduce desafíos en términos de reproducibilidad y consistencia en entornos críticos [17]. En este contexto, se recomienda complementar el diseño estructurado del *prompt* con mecanismos de control, como la inclusión de restricciones explícitas y metadatos de trazabilidad, para reducir la variabilidad y mejorar la alineación entre requisito y salida generada.

Estas consideraciones técnicas sintetizan los principales factores identificados en la literatura para optimizar el diseño de prompts en contextos de ingeniería de requerimientos (**PC3**).

6.4. Evaluación del Impacto del Prompt Engineering en la Calidad de los Textos Generados

Los estudios incluidos en el mapeo sistemático destacan la importancia de evaluar la efectividad del *Prompt Engineering* mediante criterios claros que permitan analizar el rendimiento del modelo en términos de precisión, relevancia y coherencia [16]. Algunas métricas clave incluyen:

Precisión:

Este criterio evalúa cuántas de las respuestas generadas por el modelo son correctas en comparación con los estándares o expectativas predefinidas. En el contexto de la ingeniería de requisitos, la precisión se refiere a la capacidad del modelo para identificar y resolver defectos en los requisitos, como ambigüedades o inconsistencias.

Relevancia:

La relevancia mide si las respuestas del modelo abordan específicamente las necesidades planteadas en el *prompt*. Un modelo que ofrece respuestas relevantes es capaz de concentrarse en las áreas más importantes del texto o de la tarea, evitando respuestas fuera de contexto.

Calidad Lingüística:

Es importante también evaluar la calidad lingüística de las respuestas generadas. Esto incluye verificar que las respuestas sean claras, libres de errores y adecuadas para su uso en un contexto práctico. La coherencia gramatical y la estructura lógica de las respuestas son aspectos cruciales que determinan la calidad general de los textos generados.

Realizar experimentos que comparen los resultados obtenidos mediante diferentes estrategias de *prompting* puede ayudar a identificar cuáles son más efectivas [18,19] en diferentes contextos. Variaciones en el diseño de los *prompts*, como el uso de ejemplos en estrategias *few-shot* o la incorporación de más contexto en *prompts* largos, pueden afectar significativamente la calidad de las respuestas y proporcionar información valiosa sobre cómo mejorar el rendimiento del modelo.

En conjunto, las técnicas y estrategias analizadas en esta sección permiten dar respuesta a la **PC3**, evidenciando cómo el diseño adecuado de *prompts* contribuye a mejorar la calidad y precisión de las respuestas generadas por los LLMs en la ingeniería de requerimientos. No obstante, investigaciones recientes advierten que la evaluación de sistemas basados en generación aumentada por recuperación presenta limitaciones cuando se utilizan únicamente métricas automáticas tradicionales, dado que estas no siempre capturan la corrección factual ni la adecuación contextual de las respuestas [13]. En consecuencia, se propone complementar las métricas cuantitativas con evaluación humana experta y análisis cualitativo, especialmente en dominios técnicos como la ingeniería de requerimientos, donde la validez semántica y la coherencia con el dominio son factores críticos.

7. Conclusión

A partir del mapeo sistemático realizado, los resultados obtenidos indican que el uso de ChatGPT y técnicas de *Prompt Engineering* en la ingeniería de requerimientos puede mejorar significativamente la detección de defectos y la generación de documentación. Pero a pesar de su potencial, los LLM pueden introducir sesgos o perpetuar errores derivados de los datos con los que fueron entrenados, lo que requiere una supervisión cuidadosa durante el proceso de ingeniería de requisitos.

En relación con las Preguntas Clave planteadas, los hallazgos del mapeo permiten sintetizar los principales aportes, desafíos y limitaciones del uso de ChatGPT en la ingeniería de requerimientos. En este sentido, los resultados permiten dar respuesta a la **PC1**, evidenciando el potencial de ChatGPT como herramienta de apoyo en la creación, análisis y validación de requerimientos; a la **PC2**, identificando los principales desafíos asociados al diseño de *prompts*; y a la **PC3**, destacando técnicas y estrategias de *Prompt Engineering* orientadas a mejorar la calidad de las respuestas generadas.

Asimismo, los estudios analizados permiten abordar la **PC4**, mostrando que, si bien los modelos como GPT-4 presentan niveles de precisión adecuados en determinadas tareas, aún exhiben limitaciones en la detección de ambigüedades e inconsistencias complejas.

En conjunto, este mapeo sistemático ofrece una visión estructurada del estado actual de la investigación sobre el uso de LLMs y *Prompt Engineering* en la ingeniería de requerimientos, aportando una base para futuras investigaciones y aplicaciones prácticas en el área.

Referencias

[1] T. Mahbub, D. Dghaym, A. Shankarnarayanan, T. Syed, S. Shapsough and I. Zualkernan, "Can GPT-4 aid in detecting ambiguities, inconsistencies, and incompleteness in requirements analysis? A comprehensive case study.," in IEEE Access, doi: 10.1109/ACCESS.2024.3464242.

[2] Marques, N., Silva, R. R., & Bernardino, J. (2024). Using ChatGPT in Software Requirements Engineering: A Comprehensive Review. Future Internet, 16(6), 180. https://doi.org/10.3390/fi16060180

[3] Marr, B. A Short History of ChatGPT: How We Got to Where We Are Today. Available online: https://www.forbes.com/sites/bernardmarr/2023/05/19/a-short-history-of-chatgpt-how-we-got-to-where-we-are-today/?sh=454c1d75674f (accessed on 1 March 2024).

[4] Sommerville, I., & Sawyer, P. (1997). Requirements engineering: a good practice guide. John Wiley & Sons, Inc.

[5] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," ACM Computing Surveys, vol. 55, no. 9, pp. 1–35, 2023.

[6] Arvidsson, S. (2023). Prompt engineering guidelines for LLMs in Requirements Engineering. University of Gothenburg/Department of Computer Science and Engineering.

[7] J. White, Q. Fu, S. Hays, et al., "A prompt pattern catalog to enhance prompt engineering with chatgpt," arXiv preprint arXiv:2302.11382, 2023.

[8] Guido Tebes, Denis Peppino, Pablo Becker y Luis Olsina, "Proceso para Revisión Sistemática de Literatura y Mapeo Sistemático".

[9] Kimya Khakzad Shahandashti, Mithila Sivakumar, Mohammad Mahdi Mohajer, Alvine Boaye Belle, Song Wang, and Timothy Lethbridge. 2024. Assessing the Impact of GPT-4 Turbo in Generating Defeaters for Assurance Cases. In Proceedings of the 2024 IEEE/ACM First International Conference on AI Foundation Models and Software Engineering (FORGE '24). Association for Computing Machinery, New York, NY, USA, 52–56. https://doi.org/10.1145/3650105.3652291

[10] Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven Failure Points When Engineering a Retrieval Augmented Generation System. In Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI (CAIN '24). Association for Computing Machinery, New York, NY, USA, 194–199. https://doi.org/10.1145/3644815.3644945

[11] Fei Wang, XueFeng Xi, ZhiMing Cui, Huan Dai, and XinYu Wang. 2025. Embedding Traceability in Large Language Model Code Generation: Towards Trustworthy AI-Augmented Software Engineering. In Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering (FSE Companion '25). Association for Computing Machinery, New York, NY, USA, 1760–1763. https://doi.org/10.1145/3696630.3730569.

[12] Frank P.-W. Lo, Jianing Qiu, Zeyu Wang, Junhong Chen, Bo Xiao, Wu Yuan, Senior Member, IEEE, Stamatia Giannarou, Gary Frost and Benny Lo, Senior Member, IEEE, "Dietary Assessment with Multimodal ChatGPT: A Systematic Analysis". Acceso desde IEEE Xplore 10.1109/JBHI.2024.3417280:

[13] Souvick Das, Novarun Deb, Nabendu Chaki, and Agostino Cortesi. 2025. A Multi-Agent RAG Framework for Regulatory Compliance Checking of Software Requirements. ACM Trans. Softw. Eng. Methodol. Just Accepted (December 2025). https://doi.org/10.1145/3785472

[14] Denis Donadel, Francesco Marchiori, Luca Pajola, Mauro Conti, Spritz Matter Srl;. Can LLMs Understand Computer Networks? Towards a Virtual System Administrator", revisado en IEEE Xplore, DOI: 10.1109/LCN60385.2024.10639641.

[15] Tor Sporsem and Rasmus Ulfsnes. 2025. Towards Requirements Engineering for RAG Systems. In Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering (EASE '25). Association for Computing Machinery, New York, NY, USA, 782–788. https://doi.org/10.1145/3756681.3757040

[16] C. Arora, J. Grundy, and M. Abdelrazek, "Advancing requirements engineering through generative ai: Assessing the role of llms," arXiv preprint arXiv:2310.13976, 2023.

[17] Jiawei Shao, Jingwen Tong, Qiong Wu, Wei Guo, Zijian Li, Zehong Lin, Jun Zhang. "WirelessLLM: Empowering Large Language Models Towards Wireless Intelligence". Acceso desde IEEE Xplore, DOI: 10.23919/JCIN.2024.10582827

[18] Daeseung Park1, Gi-taek An2, Chayapol Kamyod3, and Cheong Ghil Kim1,∗ "A Study on Performance Improvement of Prompt Engineering for Generative AI with a Large Language Model" Consultado en IEEE Xplore,, DOI: 10.13052/jwe1540-9589.2285.

[19] Toufique Ahmed, Kunal Suresh Pai, Premkumar Devanbu, Earl T. Barr, "Automatic Semantic Augmentation of Language Model Prompts (for Code Summarization)". 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE), revisado en IEEE Xplore: https://ieeexplore.ieee.org/document/10548617