

Tipo de artículo: Artículos de revisión

Temática: Ingeniería de software

Recibido: 3/6/2025 | Aceptado: 17/7/2025 | Publicado: 30/9/2025

Identificadores persistentes:

DOI: [10.48168/innosoft.s24.a292](https://doi.org/10.48168/innosoft.s24.a292)

ARK: [ark:/42411/s24.a292](https://nbn-resolving.org/ark:/42411/s24.a292)

PURL: [42411/s24.a292](https://purl.org/42411/s24.a292)

Revisión Sistemática de la Literatura de la Deuda de Requisitos: Sus Causas y Medición.

Systematic Review of the Debt Requirements Literature: Its Causes and Measurement.

Maria Janai Sanchez Hernández¹[\[0000-0002-3005-5235\]](#)^{*}, Maria Guadalupe Medina Barrera²[\[0000-0003-3074-0029\]](#), José Federico Ramírez Cruz³[\[0000-0002-4468-4171\]](#), Blanca Estela Pedroza-Méndez⁴[\[0000-0002-9819-635X\]](#)

¹Tecnológico Nacional de México / Instituto Tecnológico de Apizaco. Apizaco, México.
d96370620@apizaco.tecnm.mx

²Tecnológico Nacional de México / Instituto Tecnológico de Apizaco. Apizaco, México.
guadalupe.mb@apizaco.tecnm.mx

³Tecnológico Nacional de México / Instituto Tecnológico de Apizaco. Apizaco, México.
federico.rc@apizaco.tecnm.mx

⁴Tecnológico Nacional de México / Instituto Tecnológico de Apizaco. Apizaco, México.
blanca.pm@apizaco.tecnm.mx

*Autor para correspondencia: d96370620@apizaco.tecnm.mx

Resumen

La Deuda Técnica de Requisitos se define como la diferencia entre los requisitos inicialmente planteados y el producto final de software. Este estudio tuvo como objetivo realizar una Revisión Sistemática de la Literatura, basada en una metodología estructurada en tres fases: definición de un protocolo de búsqueda, selección de fuentes científicas relevantes y aplicación de criterios de inclusión y exclusión, seguido de la síntesis de la información recopilada. Se analizaron investigaciones de los últimos cinco años, considerando un total de trece artículos. Los resultados indican que las principales causas de la Deuda Técnica de Requisitos incluyen la falta de documentación formal, la presión por cumplir con plazos de entrega, la deficiente comunicación entre el cliente y el equipo de desarrollo, así como la ausencia de herramientas automatizadas que optimicen la trazabilidad de los requisitos, entre otros factores. Respecto a su medición, se han propuesto estrategias como el análisis costo-beneficio y la estimación de costos de rectificación; sin embargo, aún no se han validado en contextos reales, lo que limita su aplicabilidad práctica. En conclusión, la Deuda Técnica de Requisitos representa un desafío en la ingeniería de software, afectando directamente la calidad y el éxito de los proyectos. Este trabajo proporciona un panorama actualizado que puede servir como fundamento para futuras investigaciones en el área, con el fin de desarrollar estrategias más efectivas para su gestión y posible mitigación.

Palabras claves: Deuda de requisitos, deuda técnica de requerimientos, olores de requisitos, causas, medición.

Abstract

Technical Requirements Debt is defined as the difference between the initially stated requirements and the final software product. This study aimed to conduct a systematic literature review based on a methodology structured

in three phases: definition of a search protocol, selection of relevant scientific sources, and application of inclusion and exclusion criteria, followed by a synthesis of the collected information. Research from the last five years was analyzed, considering a total of thirteen articles. The results indicate that the main causes of Technical Requirements Debt include the lack of formal documentation, pressure to meet deadlines, poor communication between the client and the development team, as well as the absence of automated tools that optimize requirements traceability, among other factors. Regarding its measurement, strategies such as cost-benefit analysis and rectification cost estimation have been proposed; however, these have not yet been validated in real-world contexts, which limits their practical applicability. In conclusion, Technical Requirements Debt represents a challenge in software engineering, directly affecting project quality and success. This work provides an updated overview that can serve as a basis for future research in the area, with the goal of developing more effective strategies for its management and possible mitigation.

Keywords: *Debt of requirements, technical debt of requirements, requirement smells, causes, measurement.*

*

Introducción El concepto de deuda técnica - *Technical Debt* - (TD) fue introducido por Cunningham en 1993 [1] y posteriormente desarrollado por McConnell en 2008 [2]. Este término se refiere al costo futuro que implica tomar decisiones apresuradas en el desarrollo de software. Aunque un lanzamiento acelerado puede ser beneficioso inicialmente, las mejoras necesarias deben implementarse a tiempo; de lo contrario, la deuda acumulada puede derivar en problemas más complejos y costosos de resolver [3] y [4]. En la Tabla 1 se muestra la ampliación del concepto de TD realizada por [5] y [2]. Aquí, la TD es categorizada en cuatro tipos que considera las diferentes decisiones estratégicas que un equipo puede tomar de forma prudente o imprudente.

Tabla 1. Tipos de deuda técnica y sus decisiones estratégicas. Adaptada de Fowler [5].

	Prudente	Imprudente
Deliberado	“Debemos enviar ahora y asumir lo que venga después”.	“No tenemos tiempo para el diseño”.
Involuntario	“Ahora sabemos cómo debía haber sido”.	“¿Qué es diseño por capas?”.

Inicialmente, la deuda técnica se asociaba principalmente con el código fuente. Sin embargo, en [6] ampliaron la metáfora a otras áreas del ciclo de vida del software, incluyendo los requerimientos, pruebas y diseño.

Uno de los tipos de deuda menos explorados es la Deuda Técnica de Requisitos - *Requirement Technical Debt* (RTD). Al respecto, [7] señala que “se ha prestado poca atención a los requisitos a lo largo del tiempo en el software: los requisitos a menudo están muy desincronizados con la implementación o no se utilizan en absoluto. Sin embargo, los requisitos son la validación definitiva del éxito del proyecto, ya que representan los

deseos de las partes interesadas para el sistema”.

Ahora bien, por su parte [8] identificaron que la deuda técnica puede clasificarse en diez tipos, cada uno con subtipos basados en sus causas. Entre ellos, la RTD que es de particular interés en este estudio por lo que se realiza una Revisión Sistemática de la Literatura (RSL) con el objetivo de identificar causas, así como analizar si hay propuestas de modelos de medición. Los resultados de esta RSL podrían apoyar a investigaciones enfocadas a la evaluación de su impacto y evolución durante el desarrollo de proyectos de software.

En la siguiente sección, se exponen los antecedentes de la RTD, proporcionando un marco que contextualiza el tema de RTD y permite entender su fundamento. A continuación, se desarrolla el proceso de la RSL, detallando los pasos clave, métodos empleados y su relevancia dentro del contexto de la investigación. Finalmente, en el apartado 4 se presentan los hallazgos y conclusiones de este trabajo de revisión, donde se analizan los resultados obtenidos y se proponen futuras áreas de investigación.

Deuda Técnica de Requisitos (RTD)

La fase de requisitos es una de las fases iniciales del ciclo de vida del desarrollo de software y se considera fundamental debido a su impacto en el éxito del producto final. En esta fase, se llevan a cabo actividades como la elicitación, análisis, negociación, documentación y validación de requisitos. Dado que es el punto en el que los usuarios expresan sus necesidades y se evalúa la viabilidad de desarrollo, existe la posibilidad de incurrir en RTD. En este sentido, la RTD se diferencia de otros tipos de deuda técnica debido a un factor clave: la participación del usuario. En [9], a diferencia de otros tipos de deuda técnica, en la RTD el usuario juega un papel fundamental en la definición de los requisitos y el bucle de retroalimentación, lo que hace que su participación sea especialmente crítica para el éxito del proyecto.

Según N. Ernst la RTD se define como “la distancia entre la solución óptima a un problema de requisitos y la solución real, con respecto a algún espacio de decisión” [7]. De manera similar, en [9] se explica que la RTD captura las consecuencias de decisiones subóptimas tomadas en relación con los requisitos, ya sea de forma deliberada, por razones estratégicas, o inadvertida, debido a cambios en el contexto. Estas decisiones pueden ocurrir durante la identificación, formalización e implementación de los requisitos.

En [10] amplían la definición de N. Ernst y proponen una clasificación de la RTD en tres tipos:

- Tipo 0: Se incurre en esta deuda cuando no se consideran adecuadamente las necesidades de los usuarios, a pesar de que estas han sido expresadas a través de los canales de comunicación establecidos.
- Tipo 1: Surge cuando el ingeniero de requisitos, analista de negocios o desarrollador formaliza las necesi-

dades de los usuarios utilizando “olores de requisitos” -*Requirement Smells*- (RS), es decir, formulaciones lingüísticas que violan estándares como ISO 29148. Esto puede generar requisitos ambíguos que, si no se corrigen, pueden impactar negativamente la implementación.

- Tipo 2: Se presenta cuando los desarrolladores implementan una solución basada en un conjunto de requisitos, pero estos requisitos cambian y la implementación no se ajusta a dichos cambios.

En este mismo sentido, [11], [12] señalan que, incluso en contextos ágiles de desarrollo de software, actividades como la documentación, el modelado, la priorización de los requisitos y la participación de los interesados son prácticas recomendadas. No obstante, estas actividades suelen llevarse a cabo de forma separada e informal, lo que dificulta la comprensión de necesidades, la intergración y el seguimiento. Cuando estas actividades se ejecutan de manera inadecuada, es inminente la aparición de fallos en el desarrollo de software, por lo tanto se incurre en la aparición y acumulación de deuda técnica, específicamente en forma de deuda de requisitos.

Revisión Sistemática de Literatura de la RTD

El objetivo principal de esta RSL es obtener un vistazo actualizado sobre el concepto de Deuda Técnica de Requisitos, desde 2018 hasta la fecha. Para llevar a cabo esto, se implementó la metodología propuesta por [13] que permite identificar, evaluar e interpretar la investigación actual en el ámbito de la Ingeniería de Software, la cual se sintetizó por [14]. De esta manera, esta RSL busca identificar estudios que aborden y clarifiquen las causas de RTD, así como conocer propuestas de modelos de medición o cuantificación.

Según el procedimiento establecido por [13] la RSL comienza con la definición y revisión de un protocolo que especifica las preguntas de investigación a responder y describe los métodos empleados en el desarrollo de la revisión. La RSL también establece estrategias de búsqueda que permiten evaluar la relevancia de la literatura en diversas bases de datos científicas, garantizando que los estudios seleccionados sean de alta calidad. Asimismo, se deben definir claramente los criterios de inclusión y exclusión para cerciorar que los estudios seleccionados contribuyan efectivamente a la esta RSL.

En este contexto los autores de [14] describen el proceso de la RSL en tres fases: Planificación, Conducción y Documentación, cada una con actividades clave específicas, las cuales se ilustran en la Figura 1 y se detallan en los siguientes apartados de este artículo.

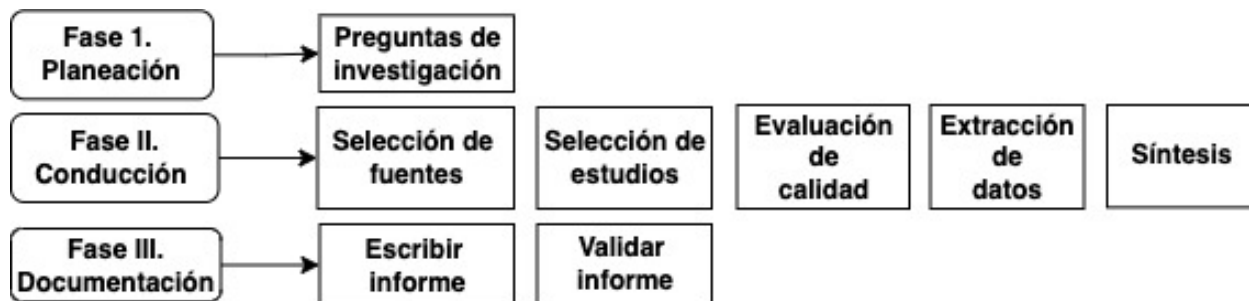


Figura 1. Proceso de RSL. Elaboración propia con base en Tebes et al. [14].

Planeación

Preguntas de investigación

El objetivo de esta Revisión Sistemática de la Literatura (RSL) es identificar y analizar de manera rigurosa si existen estudios que ofrezcan una definición clara y objetiva del concepto de “deuda técnica de requisitos”. En caso afirmativo, se pretende que dichos estudios contribuyan a responder las siguientes preguntas de investigación:

P1. ¿Cuáles son las causas que originan la deuda técnica de requisitos (RTD)?

P2. ¿Qué estrategias se han propuesto para la medición de la RTD?

Conducción

Selección de fuentes y cadena de búsqueda

El método de búsqueda utilizado en este estudio consistió en la selección de bibliotecas digitales relevantes para el área de conocimiento, previamente utilizadas y referenciadas en otras revisiones sistemáticas, como se detalla en la Tabla 2. Para ello, se eligieron cinco de las principales bibliotecas digitales que ofrecen artículos de acceso abierto.

Tabla 2. Bibliotecas digitales

Springer Link
Science Direct
IEEEExplore
ACM Digital Library
ArXiv

Para llevar a cabo el proceso de exploración en las bibliotecas digitales, fue necesario diseñar una cadena de búsqueda compuesta por los términos clave relevantes para este estudio, conectados mediante operadores lógicos. Dado que la mayoría de las publicaciones relevantes en el área se encuentran redactadas en inglés, la cadena de búsqueda se formuló exclusivamente en dicho idioma. La expresión final utilizada fue la siguiente:

(“Requirement debt” OR “Requirements Smells”) AND
(Features OR Causes OR Dimensions OR Factors OR Attributes) AND
(Measure OR Quantifying)

Extracción de datos

La búsqueda se llevó a cabo en cada una de las bibliotecas digitales seleccionadas, utilizando la cadena de búsqueda previamente definida. Los resultados preliminares obtenidos se presentan en la Figura 2. Como se puede observar, esta búsqueda inicial arrojó un total de 175 documentos. Durante esta primera etapa automatizada, se aplicaron filtros correspondientes al rango de fechas, la disciplina o área de conocimiento, y el tipo de acceso (limitado a publicaciones de acceso abierto).

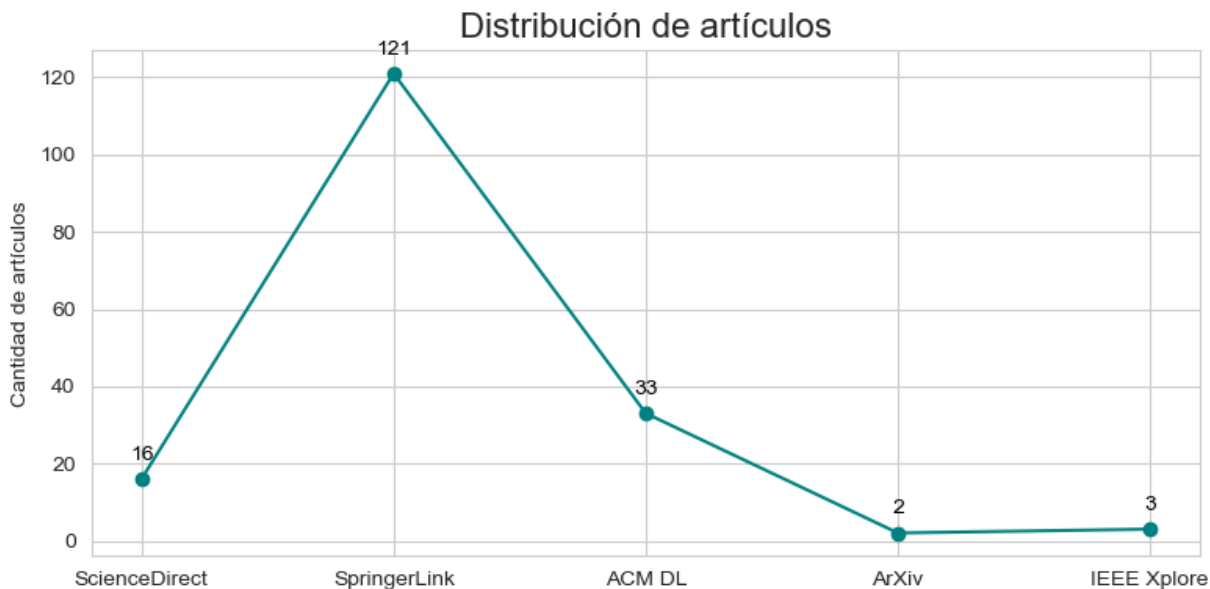


Figura 2. Distribución de artículos en primera búsqueda

La selección de los artículos se realizó mediante un enfoque de lectura en profundidad, comenzando por el título y continuando con el resumen, las palabras clave, la conclusión y, finalmente, el texto completo del artículo. Durante esta etapa, se aplicaron los criterios de inclusión y exclusión previamente establecidos. Se incluyeron aquellos estudios que abordaban las causas y los modelos de cuantificación de la deuda técnica de requisitos, así como aquellos que presentaban estudios de caso en los cuales se describía, directa o indirectamente, el concepto de RTD.

Criterios de calidad

Los criterios de inclusión y exclusión desempeñan un papel fundamental en la realización de una selección objetiva de estudios. Si estos criterios son demasiado amplios, pueden comprometer la calidad de la RSL y disminuir la confianza en los resultados obtenidos. La Tabla 3 presenta los criterios aplicados durante el proceso de búsqueda y selección de artículos.

Tabla 3. Criterios de inclusión y exclusión

Criterios de Inclusión	Publicaciones entre el año 2018- 2024. Publicaciones del área de computación e ingeniería de software. Publicaciones que incluyan los términos “deuda técnica de requisitos” o “deuda de requerimientos”
Criterios de Exclusión	Publicaciones de acceso restringido. Publicaciones que no contengan deuda de requisitos. Publicaciones sin descripción de metodología. Comentarios publicados en memorias u otros medios.

El Listado de artículos incluidos en esta revisión se presenta en la Tabla 4.

Tabla 4. Demografía de Publicaciones relevantes para la RSL

	Título	Año	Autores	Publicado
Art 1	23 shades of self-admitted technical debt: an empirical study on machine learning software	2022	[15]	ACM
Art 2	Towards Quantifying Requirements Technical Debt for Software Requirements concerning Veracity: A Perspective and Research Roadmap	2024	[16]	ArXiv
Art 3	Integrating Traceability within the IDE to Prevent Requirements Documentation Debt	2018	[17]	IEEE. 44th Euro-micro
Art 4	Quantifying Requirements Technical Debt: A Systematic Mapping Study and a Conceptual Model	2023	[9]	IEEE

	Título	Año	Autores	Publicado
Art 5	Towards a Holistic Definition of Requirements Debt	2019	[10]	IEEE
Art 6	A systematic literature review on Technical Deb prioritization: Strategies, processes, factors, and tools	2021	[18]	Science Direct
Art 7	Architectural design decisions that incur technical debt-An industrial case study	2021	[19]	Science Direct
Art 8	Identification and measurement of Requirements Technical Debt in software development: A systematic literature review	2022	[20]	Science Direct
Art 9	An initial theory to understand and manage requirements engineering debt in practice	2023	[21]	Science Direct
Art 10	Stakeholder influence on technical debt management in the public sector: An embedded case study	2022	[22]	Science Direct
Art 11	Modelling the quantification of requirements technical debt	2024	[23]	Springer
Art 12	Natural Language Requirements Testability Measurement Based on Requirement Smells	2024	[24]	Springer
Art 13	Technical Debt and Waste in Non-Functional Requirements Documentation: An Exploratory Study	2019	[25]	Springer

Extracción de datos y síntesis

Se llevó a cabo una síntesis de tipo cualitativo, en la cual los estudios seleccionados fueron agrupados según los dos ejes temáticos principales de esta RSL: causas y medición de la deuda técnica de requisitos. Posteriormente, se realizó un análisis comparativo de las distintas propuestas con el objetivo de identificar causas comunes de la RTD, así como enfoques y métodos utilizados para su cuantificación o medición.

Documentación

La redacción del informe se estructuró en función de cada pregunta de investigación diseñada en esta RSL. En la siguiente sección, se presentan los hallazgos obtenidos a partir del análisis de los estudios seleccionados.

Hallazgos

Causas de la RTD

P1. ¿Cuáles son las causas que originan la deuda técnica de requisitos (RTD)?

La deuda técnica de requisitos (RTD) puede originarse a partir de diversos factores. En su investigación, Melo et al. identifican 15 causas principales asociadas a la RTD, las cuales fueron extraídas de la revisión sistemática que llevó a cabo [20]. Además, el autor desarrolla una red que ilustra las relaciones entre dichas causas. Por ejemplo, se destaca que un proceso inadecuado de elicitación de requisitos se encuentra relacionado con una deficiente planificación de entrevistas, lo que impide que el cliente exprese con claridad sus necesidades. Esto da lugar a requisitos ambiguos, incompletos o con un bajo nivel de detalle. Asimismo, la ambigüedad de los requisitos se asocia con un estilo de redacción deficiente y problemas gramaticales, lo que dificulta su interpretación y comprensión.

De manera similar, en [21] se documentan las causas de la RTD surgidas a partir de entrevistas realizadas con profesionales del sector, agrupándolas en cuatro categorías: tiempo, producto, personas y artefactos. En cuanto a las causas relacionadas con el tiempo, estas están asociadas principalmente con los plazos de entrega del producto de software. Las causas vinculadas a las personas se deben, en gran medida, a la falta de conocimiento especializado para implementar ciertos requisitos por parte del equipo, así como a la mala comunicación entre los miembros del equipo y el cliente. Además, la falta de documentación adecuada de los requisitos genera una costosa reelaboración. En este mismo sentido, [23] presenta una recopilación de causas e indicadores de la RTD extraídos de un mapeo sistemático de literatura, destacando aspectos como la ambigüedad en la formalización de los requisitos, la entrega de requisitos incompletos, las soluciones subóptimas, la falta de trazabilidad entre

los requisitos y el código, la ausencia de documentos formales y las limitaciones de tiempo.

En la Tabla 4 se describen las principales causas identificadas en la literatura que pueden generar la RTD. Entre ellas se incluyen la ambigüedad (también conocida como “olores de requisitos”), los plazos de entrega, la falta de documentación formal, la ausencia de estrategias de verificación y validación de los requisitos, problemas de comunicación, la escasez de recursos, la experiencia del equipo y la falta de herramientas especializadas para llevar a cabo este proceso.

En cuanto a la ambigüedad, los estudios reportados en Art2, Art4, Art6, Art8 y Art12 coinciden en que la ambigüedad de los requisitos es una causa clave de la RTD. Un requisito que no utiliza un lenguaje claro y preciso puede dar lugar a malinterpretaciones de las necesidades, como lo menciona en [16]: “Una desviación menor de los requisitos es estrictamente un fallo que puede causar daños fundamentales al producto de software definitivo”. La ambigüedad puede analizarse a través de la presencia o ausencia de sustantivos, verbos, adverbios y adjetivos dentro de la definición de los requisitos.

Por otro lado, la causa relacionada con el tiempo es mencionada en los estudios Art3, Art6, Art7, Art9, Art10 y Art11, desde la perspectiva de que los plazos limitados frecuentemente obligan a tomar decisiones apresuradas (intencionadas), resultando en la entrega de productos de software que no cumplen completamente con las especificaciones, o en productos que cumplen parcialmente pero no son efectivos ni eficientes. Esto, a su vez, contribuye a la presencia de RTD en los proyectos. Esta variable está estrechamente vinculada con la priorización de los requisitos, como se describe en los estudios Art2, Art8 y Art11, que señalan que a menudo se seleccionan requisitos que no aportan valor inmediato al cliente.

Además, los trabajos Art3, Art4, Art5, Art8, Art9, Art10, Art11, Art12 y Art13 destacan que la falta de un documento formal que describa las necesidades del cliente o usuario, junto con una redacción deficiente, dificulta la comprensión de los requisitos, lo que impacta negativamente en la aparición de RTD.

De igual manera, las propuestas presentadas en Art10, Art11 y Art12 abordan el concepto de verificación/validación de los requisitos. Se considera que un requisito que pasa por una fase de revisión alcanza un nivel de confianza, mientras que la ausencia de verificación aumenta la probabilidad de que el proyecto enfrente RTD en las etapas posteriores del ciclo de vida del software.

Finalmente, los estudios Art1, Art2, Art10 y Art11 subrayan que los cambios (volatilidad) en los requisitos o en el contexto del proyecto pueden ser problemáticos, ya que pueden dar lugar a una falta de trazabilidad entre los requisitos y el código, tal como lo menciona Art3.

En relación con las causas, la comunicación, ya sea entre los miembros del equipo, entre diferentes áreas o

con el cliente, juega un papel crucial. Si los canales de comunicación no son adecuados, la información puede volverse deficiente, lo que a su vez genera una falta de comprensión de los requisitos, tal como se señala en los estudios Art2, Art8 y Art11. Además, la complejidad de los requisitos es otro factor identificado como causa de RTD, según Art2 y Art8.

Una causa que es mencionada exclusivamente en el estudio Art7 es la limitación de recursos. Este factor podría considerarse relevante, ya que, en ausencia de recursos suficientes para abordar la complejidad del proyecto, es común que se prioricen actividades más económicas de desarrollar, dejando de lado aquellas que aportan mayor valor al usuario o cliente.

Asimismo, la falta de conocimientos o experiencia del equipo frente a un proyecto determinado se considera una causa significativa de la RTD, como se menciona en Art11 y Art13. De manera similar, la ausencia de herramientas automatizadas especializadas para la definición de requisitos también se identifica como una causa relevante de la RTD, según Art3, Art12 y Art13.

Tabla 5. Causas de la RTD identificadas en la literatura

Núm. de Artículo	Causas identificadas
Art 1	Cambios en las especificaciones de requisitos, intenciones poco claras y actualizaciones en cascada. Requisitos funcionales que requieren cambios futuros Requisitos No funcionales, que no cumplen con los estándares (rapidez, memoria, seguridad, etc)
Art 2	Inadecuada recopilación/ambigüedad Priorización de requisitos Implementación parcial Complejidad de los requisitos Variabilidad contextual
Art 3	Requerimientos insuficientes o incompletos (casos de uso, historias de usuario, SRS) baja calidad Requisitos obsoletos Trazabilidad de los requisitos Comunicación entre los departamentos Limitación de tiempo (falta de comprensión) Falta de comunicación con los clientes Uso de aplicaciones distintas para documentar
Art 4	Falta de entendimiento de las necesidades de los clientes Ambigüedades Olores de requisitos Requisitos incompletos
Art 5	Documentación ineficiente de requerimientos (olores de requisitos)
Art 6	Tiempo empresarial Falta de claridad de los requisitos Prácticas de refactorización insuficientes

Núm. de Artículo	Causas identificadas
Art 7	Limitación de tiempo Limitación de recursos Baja comprensión de los requisitos
Art 8	Bajo nivel de documentación Requerimientos ambiguos No definición de requerimientos no funcionales Requerimientos vagos o incompletos Falta de comunicación con el cliente Atajos y soluciones alternativas Presiones de calendario No reflejar las necesidades del cliente Falta de experiencia Inadecuada priorización de requisitos Mala redacción Entrevistas no planeadas Falta de guiones Requisitos complejos Mala elicitación
Art 9	Tiempo Producto Personas Artefactos
Art 10	Calendario (tiempo) Mala documentación Falta de verificación de requisitos Cambios de contexto
Art 11	Las ineficiencias en la identificación y estimación de los requisitos Ineficiencias en la especificación de los requisitos u olores de los requisitos Brecha de comunicación entre los miembros del equipo Fechas de entrega (limitaciones de tiempo) Volatilidad de los requisitos Falta de comunicación con los clientes Falta de pruebas de requisitos Falta de experiencia y conocimientos Falta de documentación
Art 12	Requerimientos olorosos (escritos con lenguaje subjetivo, ambigüedades) Interpretación errónea de los requisitos dado el lenguaje ambiguo y subjetivo Longitud de los requisitos Falta de comprobación de los requisitos Falta de metodologías para documentar los requisitos.
Art 13	La falta de documentación de los requisitos No funcionales (mantenibilidad, fiabilidad, usabilidad y rendimiento) Tamaño del proyecto Tipo de proyecto Falta de conocimiento para documentar este tipo de requisitos Falta de herramientas

En la Figura 3 se presenta la distribución de las principales causas identificadas en cada investigación, destacando que las causas más frecuentemente citadas son: la ambigüedad en la redacción de los requisitos (conocida como “olors de requisitos”), los problemas relacionados con el tiempo, la ausencia de un documento formal de requisitos, la volatilidad de los requisitos y los problemas de comunicación.

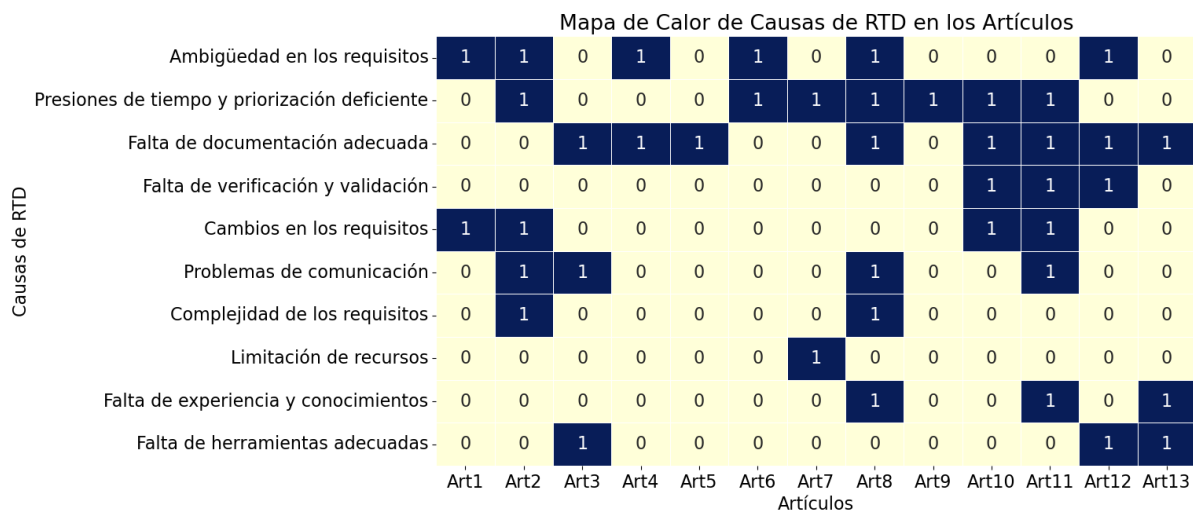


Figura 3. Distribución de causas de RTD encontrada en artículos

Estrategias de medición de la RTD

P2. ¿Qué estrategias se han propuesto para la medición de la RTD?

El proceso de medición se entiende como la actividad de analizar y cuantificar los costos y esfuerzos necesarios para gestionar adecuadamente la deuda técnica [8], [9]. En este sentido, es crucial contar con estrategias de medición y cuantificación que permitan evaluar el impacto de la deuda técnica y facilitar una toma de decisiones más efectiva.

Una propuesta relevante de medición es la presentada en [16], los autores introducen el concepto de “Veracidad de requisitos”, una categoría específica vinculada a la confianza, autenticidad y demostrabilidad. Los requisitos de veracidad pueden ser tanto funcionales como no funcionales. Para los requisitos funcionales, se debe verificar que los datos integrados al software sean correctos. En el caso de los requisitos no funcionales, se debe evaluar la fiabilidad, disponibilidad y rendimiento, es decir, la manera en que los datos son accedidos, almacenados, mantenidos y recuperados, como parte de sus atributos de calidad. Según esta perspectiva, la RTD debe cuantificarse considerando tanto los requisitos funcionales como no funcionales, dependiendo del contexto de desarrollo.

La propuesta de Perera et al. [16] se alinea con los conceptos del Método de Análisis Costo-Beneficio (CBAM), que implica identificar los requisitos con deuda, evaluar el costo de su corrección, valorar los beneficios esperados

de dicha corrección y calcular cuáles aportan mayor valor con menor esfuerzo, priorizándolos en consecuencia. Además, se toma en cuenta la Teoría Moderna de la Cartera (MPT), que organiza los requisitos en términos de su valor y el riesgo asociado a su implementación, seleccionando aquellos que minimicen el riesgo y maximicen el beneficio. La estimación de la deuda de requisitos de veracidad puede influir significativamente en la selección de la arquitectura que se implementará.

Además, Perera et al. [9], [23] identificaron 57 conceptos relacionados con la cuantificación de la RTD, los cuales fueron clasificados en cuatro categorías: proceso o tiempo, costo, beneficio y probabilidad. Estos conceptos fueron agrupados en un total de 14 elementos, los cuales se representaron en un mapa relacional diseñado para modelar los aspectos clave de la cuantificación de la RTD.

- a) Costo de implementación de requisitos: Las necesidades del usuario se capturan en requisitos que generan un costo, el cual corresponde al costo total de la implementación de uno o más requisitos formalizados.
- b) Modelado de artefactos: Esto incluye aspectos como los “olores de requisitos”, requisitos insuficientes, incompletos o desfasados.
- c) Costo de rectificación de la RTD: Se modela el proceso de rectificación de la RTD, el cual puede incurrir en un costo dependiendo de la etapa en que se realice. Este costo también está asociado con la atención de necesidades descuidadas.
- d) Interés y probabilidad de la RTD: Se considera el costo asociado a la aclaración de un requisito ambiguo, la realización de entrevistas adicionales, o la resolución de desajustes. Además, se evalúa la probabilidad de que un requisito no tenga impacto.
- e) Modelado de componentes de interés: Esto incluye el costo adicional relacionado con la realización de entrevistas adicionales con los usuarios si las necesidades capturadas están incompletas.
- f) Beneficio de la rectificación: Se evalúa tanto el beneficio de rectificar la RTD como el beneficio de abordar la RTD a corto plazo.

Otra propuesta de medición es la presentada en [10], quienes proponen estrategias para cuantificar los tres tipos de deuda técnica que ellos definen de la siguiente manera:

- Tipo 0: Cuantificar el costo de implementar necesidades no atendidas. Esto debe considerar el interés relacionado con la etapa de desarrollo actual, ya que, a medida que se avanza en el desarrollo, el costo de abordar estas necesidades aumentará.
- Tipo 1: Cuantificar el costo de corregir los “olores de requisitos”. En este caso, se debe tener en cuenta el impacto que esta corrección tendrá en otras actividades del proyecto.
- Tipo 2: Cuantificar la diferencia entre la implementación actual y los “olores de requisitos”. Esto implica

comparar la implementación existente con los posibles cambios, considerando tanto el costo principal como el interés involucrado. La solución propuesta debe ser la mejor posible para optimizar los resultados.

Por su parte, Melo et al. [20] identificaron 16 estrategias para identificar y medir la RTD, que incluyen la revisión de los requisitos con el cliente, la comunicación, la revisión entre pares, el análisis de costo-beneficio, enfoques de cuantificación, el enfoque de vecinos más cercanos, diagramas de causa y efecto, acciones de prevención, mapas de pagos, identificación a través de normas ISO/IEC/IEEE 29148:2018 [26] y plantillas de documentación. Además, se identificaron varias métricas que pueden ser utilizadas para la medición, tales como:

- Principal: Esfuerzo requerido para completar una tarea que no ha sido atendida.
- Interés: Se refiere a la penalización que se deberá pagar, es decir, el trabajo adicional necesario para alcanzar la completitud o corrección de los requisitos.

En [20] también consideran el enfoque de decisión si-if, que establece que la deuda debe ser pagada cuando el costo principal es más bajo que el interés. Otro enfoque propuesto es el cuándo-when, que determina cuándo debe pagarse la deuda, evaluando si es conveniente hacerlo en el periodo actual o si puede esperar.

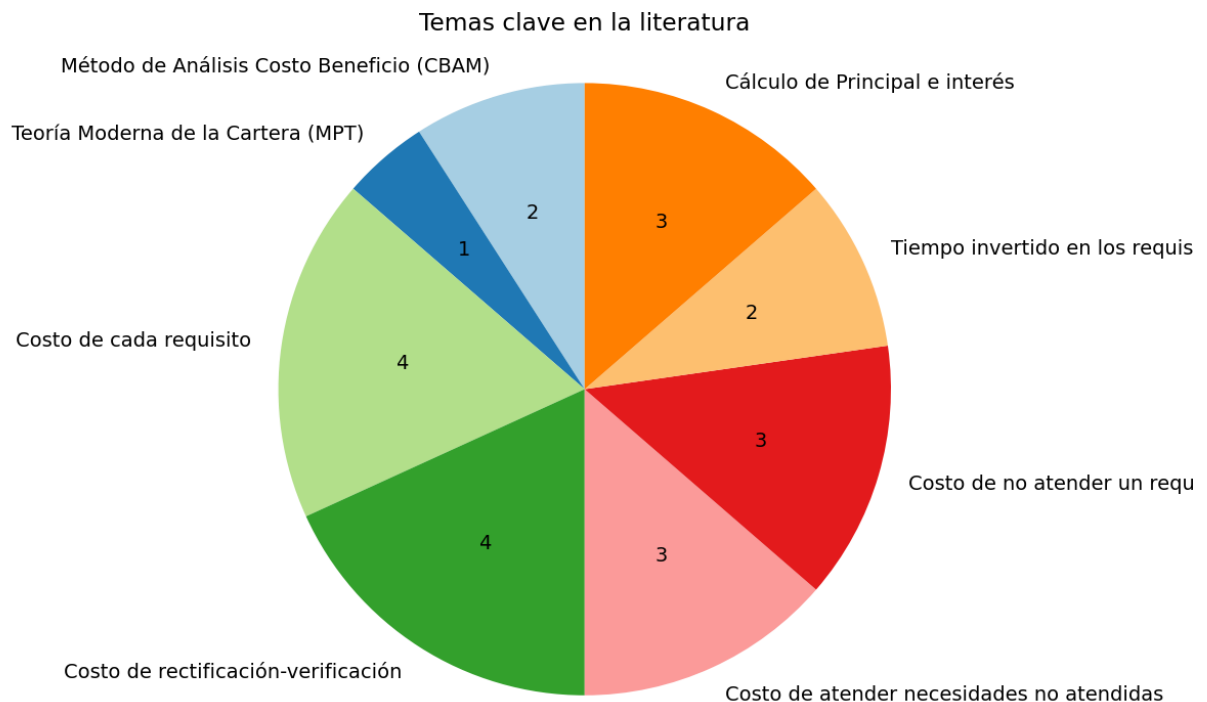


Figura 4. Temas identificados en la literatura

Discusión

Esta Revisión Sistemática de Literatura (RSL) nos permitió investigar el estado actual en cuanto a las causas de la deuda de requisitos y las propuestas de medición realizadas en los últimos cinco años. Se revisaron 13 artículos que se consideraron relevantes para responder las preguntas de investigación planteadas inicialmente.

Conclusiones y trabajos futuros

Las decisiones tomadas para alcanzar los objetivos dentro de una organización enfocada en el desarrollo de software recaen sobre los miembros del equipo y pueden tener un impacto significativo en el producto, el equipo, el cliente y la empresa. Esta RSL ha permitido identificar que el tema de RTD es un área de oportunidad para profundizar en la investigación.

*

Agradecimientos Agradecemos al Tecnológico Nacional de México / Instituto Tecnológico de Apizaco, así como al Consejo Nacional de Humanidades, Ciencias y Tecnologías, por el apoyo brindado para la realización de este trabajo.

*

Contribución de Autoría Maria Janai Sanchez Hernández: [Conceptualización](#), [Investigación](#), [Metodología](#), [Redacción - borrador original](#). Maria Guadalupe Medina Barrera: [Conceptualización](#), [Supervisión](#), [Escritura](#), [revisión y edición](#). José Federico Ramírez Cruz: [Metodología](#), [Análisis formal](#). Blanca Estela Pedroza-Méndez: [Validación](#), [Curación de datos](#).

Referencias

- [1] W. Cunningham, “The WyCash portfolio management system,” *ACM SIGPLAN OOPS Messenger*, vol. 4, no. 2, pp. 29–30, 1993.
- [2] S. McConnell, “Managing Technical Debt,” 2008, available: www.construx.com/whitepapers.
- [3] Z. S. H. Abad and G. Ruhe, “Using real options to manage Technical Debt in Requirements Engineering,” in *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, 2015, pp. 230–235.
- [4] C. Berenguer *et al.*, “Technical Debt is not Only about Code and We Need to be Aware about It,” in *XX Brazilian Symposium on Software Quality*, 2021, pp. 1–12.
- [5] F. Martin, “Technical Debt Quadrant,” 2025, available: <https://martinfowler.com/bliki/TechnicalDebtQuadrant.html>.
- [6] N. Brown *et al.*, “Managing technical debt in software-reliant systems,” in *Proceedings of the FSE/SDP workshop on Future of software engineering research*, 2010, pp. 47–52.
- [7] N. A. Ernst, “On the role of requirements in understanding and managing technical debt,” in *2012 3rd International Workshop on Managing Technical Debt (MTD)*, 2012, pp. 61–64.
- [8] Z. Li, P. Avgeriou, and P. Liang, “A systematic mapping study on technical debt and its management,” *Journal of Systems and Software*, vol. 101, pp. 193–220, 2015.
- [9] J. Perera, E. Tempero, Y. C. Tu, and K. Blincoe, “Quantifying Requirements Technical Debt: A Systematic Mapping Study and a Conceptual Model,” in *2023 IEEE 31st International Requirements Engineering Conference (RE)*, 2023, pp. 123–133.

- [10] V. Lenarduzzi and D. Fucci, “Towards a Holistic Definition of Requirements Debt,” in *International Symposium on Empirical Software Engineering and Measurement*, 2019.
- [11] V. Bonfim and F. Benitti, “Requirements debt: causes, consequences, and mitigating practices,” 2022, pp. 13–18.
- [12] V. D. Bonfim and F. B. V. Benitti, “Ontored: Requirements debt ontology,” 2024.
- [13] B. Kitchenham *et al.*, “Systematic literature reviews in software engineering – A systematic literature review,” *Inf Softw Technol*, vol. 51, no. 1, pp. 7–15, 2009.
- [14] G. Tebes *et al.*, “Proceso para Revisión Sistemática de Literatura y Mapeo Sistemático,” 2020.
- [15] D. O'Brien *et al.*, “23 shades of self-admitted technical debt: an empirical study on machine learning software,” in *Proceedings of the 30th ACM Joint European Software Engineering Conference*, 2022, pp. 734–746.
- [16] J. Perera *et al.*, “Towards Quantifying Requirements Technical Debt for Software Requirements concerning Veracity,” 2024, arXiv: 2407.00391.
- [17] S. Charalampidou *et al.*, “Integrating traceability within the IDE to prevent requirements documentation debt,” in *44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2018, pp. 421–428.
- [18] V. Lenarduzzi *et al.*, “A systematic literature review on Technical Debt prioritization: Strategies, processes, factors, and tools,” *Journal of Systems and Software*, vol. 171, 2021.
- [19] M. Soliman, P. Avgeriou, and Y. Li, “Architectural design decisions that incur technical debt — An industrial case study,” *Inf Softw Technol*, vol. 139, 2021.
- [20] A. Melo *et al.*, “Identification and measurement of Requirements Technical Debt in software development: A systematic literature review,” *Journal of Systems and Software*, vol. 194, p. 111483, 2022.
- [21] J. Frattini *et al.*, “An initial theory to understand and manage requirements engineering debt in practice,” *Inf Softw Technol*, vol. 159, p. 107201, 2023.
- [22] M. E. Nielsen and C. Madsen, “Stakeholder influence on technical debt management in the public sector: An embedded case study,” *Gov Inf Q*, vol. 39, no. 3, 2022.
- [23] J. Perera *et al.*, “Modelling the quantification of requirements technical debt,” *Requir Eng*, vol. 29, no. 4, pp. 421–458, 2024.

- [24] M. Zakeri-Nasrabadi and S. Parsa, “Natural Language Requirements Testability Measurement Based on Requirement Smells,” 2024, arXiv: 2403.17479.
- [25] G. Robiolo *et al.*, “Technical Debt and Waste in Non-Functional Requirements Documentation: An Exploratory Study,” 2019, arXiv: 1909.12716.
- [26] ISO/IEC/IEEE, “Systems and software engineering – Life cycle processes – Requirements engineering,” 2025, available: <https://standards.ieee.org/ieee/29148/6937/>.