Revista Colombiana de Computación
Vol. 27, No. 1. Enero – Junio 2026, pp. 71-80
e-ISSN: 2539-2115, https://doi.org/10.29375/25392115.5072

REVISTA COLOMBIANA DE COMPUTACIÓN

Deuda técnica en el desarrollo de software habilitado por la IA y su impacto sociotécnico: Una revisión sistemática de la literatura

Technical debt in AI-enabled software development and its sociotechnical impact: A systematic literature review

María Guadalupe Medina Barrera[1], José Juan Hernández Mora[1]

[1] Tecnológico Nacional de México / Instituto Tecnológico de Apizaco, Tlaxcala, México.

guadalupe.mb@apizaco.tecnm.mx, juan.hm@apizaco.tecnm.mx

(Recibido: 17 Julio 2024; aceptado: 28 Junio 2026; publicado online: 30 Junio 2026)

Resumen. Los conjuntos de herramientas de Inteligencia Artificial – IA están siendo utilizados para desarrollar software innovador con gran rapidez, aunque seguir estos atajos podría generar Deuda Técnica de la IA – DTIA. Este tipo de deuda puede tener consecuencias tanto en lo técnico, como en lo social. Sin embargo, los desarrolladores no siempre son conscientes de si están incurriendo en DTIA durante sus proyectos. Peor aún, desconocen sus impactos sociotécnicos y cómo mitigarlos. Para reducir esta brecha, se desarrolló una Revisión Sistemática de la Literatura – RSL para averiguar cuáles son los tipos y prácticas de DTIA, así como cuáles son sus repercusiones sociotécnicas. Los resultados arrojaron variedad de tipos y antipatrones de DTIA, así como prácticas para identificarla y mitigarla. También, se encontró que existen daños a la calidad, seguridad y mantenibilidad del producto, además de retos como sesgos y discriminaciones. Esta información puede apoyar a científicos y practicantes para detectar y gestionar la deuda técnica y social en contextos de desarrollo de software habilitado por la IA.

Palabras clave: Deuda técnica, deuda sociotécnica, inteligencia artificial, software habilitado por la IA.

Abstract. Artificial Intelligence – AI frameworks are being used to develop innovative software quickly, although these shortcuts could bring with them Artificial Intelligence Technical Debt – AITD. AITD can be harmful from both technical and social perspectives. However, developers do not know whether their systems incur AITD. Even worse, they are unaware of its sociotechnical impacts and how to mitigate them. To reduce this gap, a Systematic Literature Review – SLR was developed to uncover which types and practices of AITD exist, as well as their sociotechnical implications. Results revealed a variety of AITD types and antipatterns, as well as identification and mitigation practices. Furthermore, the findings indicate that AITD has detrimental effects on the product's quality, security, and maintainability, while also introducing challenges such as biases and discrimination. This information can support scientists and practitioners to detect and manage technical and social debt in AI-enabled software development.

Keywords: Technical debt, sociotechnical debt, artificial intelligence, AI-enabled software.

1. Introducción

Las diversas herramientas de aprendizaje automático y aprendizaje profundo están poniendo una oferta amplia y accesible a los desarrolladores de software para construir rápidamente aplicaciones innovadoras. Es así como la Inteligencia Artificial – IA está siendo integrada en una gran cantidad de aplicaciones que impactan en la vida cotidiana. Pero la introducción de soluciones de software basadas en IA en lugares como las escuelas, los hogares y los hospitales exige que se garantice que estos productos beneficien a la sociedad en forma segura y preserven sus derechos a la justicia y a la equidad.

Luego entonces, seguir atajos durante la construcción de software dirigido por la IA puede conducir a la acumulación de Deuda Técnica de la Inteligencia Artificial – DTIA (Recupito y otros, 2024; Tang y otros, 2021); esto es, el esfuerzo adicional que se tiene que dedicar en etapas posteriores del desarrollo ocasionado por el empleo de soluciones subóptimas en la construcción de aplicaciones basadas en la IA. La DTIA puede darse de forma no intencional, o bien intencionada (Potdar & Shihab, 2014), por ejemplo, por razones estratégicas o competitivas para lanzar un producto o servicio a tiempo y con presupuesto.

Ahora bien, los impactos de la DTIA pueden ser diversos. Técnicamente, pueden dañarse la mantenibilidad y la seguridad de las aplicaciones (Recupito y otros, 2024). Socialmente, está la influencia invisible que puede tener la IA en las personas (Taddeo & Floridi, 2018) y las posibles implicaciones éticas (Petrozzino, 2021). Por ello, los desarrolladores deben adoptar oportunamente las acciones necesarias para atenuar los efectos adversos sobre

©2026 Universidad Autónoma de Bucaramanga. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC-SA 4.0) license (https://creativecommons.org/licenses/by-nc-sa/4.0/)

la calidad de los sistemas y sus implicaciones de índole social. De lo contrario, un tratamiento tardío puede resultar demasiado costoso, pudiendo llegar a la intratabilidad y desecho del producto. Más aún, es imperativo evaluar y atender las posibles implicaciones sociales y éticas para salvaguardar ante todo los derechos de los usuarios.

La DTIA es un fenómeno que ha suscitado un interés creciente en la comunidad científica. En respuesta a ello, diversos trabajos han realizado revisiones de la literatura para sintetizar el estado del conocimiento sobre este tema. Tales revisiones exploran desde la conceptualización de la deuda técnica (Thibodeau, 2025), la evolución e implicaciones de la DTIA en el proceso de ingeniería de software (Moreschini y otros, 2026), así como sus diferentes tipos y formas propuestas de medición (Sklavenitis & Kalles, 2025). No obstante, hasta donde alcanza nuestro conocimiento, las repercusiones sociotécnicas en el ámbito del desarrollo de software habilitado por la IA no han sido examinadas mediante una revisión de la literatura.

Para abordar este vacío, este trabajo presenta una Revisión Sistemática de la Literatura – RSL sobre los temas de interés relacionados con la DTIA, particularmente sus tipos y prácticas, así como las posibles repercusiones sociotécnicas. Los hallazgos pueden apoyar a los desarrolladores de software para prevenir impactos adversos en la construcción y uso de sistemas habilitados por la IA.

2 Metodología

Para realizar esta RSL sobre DTIA se desarrolló un proceso basado en las recomendaciones de (Kitchenham & Charters, 2007), considerando las etapas de planeación, desarrollo y resultados. La Figura 1 muestra los pasos llevados a cabo en cada etapa:

1. Definición de preguntas de investigación.
2. Definición de la estrategia de búsqueda.
3. Búsqueda y selección de los estudios.
4. Extracción de datos.
5. Síntesis de resultados.



Figura 1. Proceso ejecutado en la revisión sistemática de la literatura sobre DTIA.
Fuente: Adaptación basada en Kitchenham y Charters (2007).

2.1 Definición de preguntas de investigación

Para esta investigación se plantearon las siguientes preguntas:

- P1: ¿Cuáles son los tipos y prácticas de DTIA?
- P2: ¿Cuáles son las repercusiones sociotécnicas de la DTIA en el desarrollo de software basado en IA?

2.2 Definición de la estrategia de búsqueda

Para dar respuesta a las preguntas planteadas, se consultaron las bases de datos científicas *Springer Link*, *IEEE Xplore, Science Direct*, *ACM Digital Library* y *Google Scholar*. En ellas, se buscaron artículos publicados en los últimos ocho años con los siguientes términos:

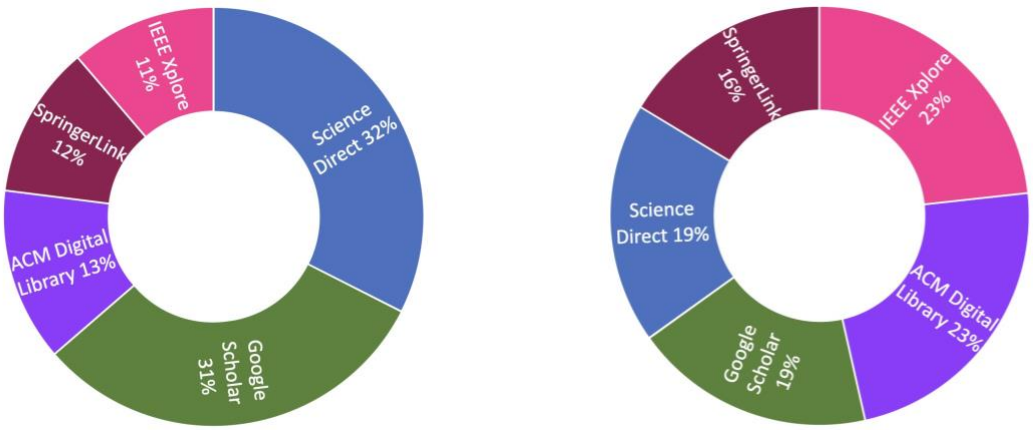
Para P1, *technical debt, artificial intelligence, machine learning, based, enabled, software, systems, applications.*

Para P2, se consideraron también los siguientes: *sociotechnical debt*, *social debt*, *ethical debt*.

Con los términos anteriores se construyó la siguiente cadena de búsqueda: *–technical debt OR social debt OR sociotechnical debt OR ethical debt– AND –artificial intelligence OR machine learning– AND –based OR enabled– AND –software OR systems OR applications–.*

2.3 Búsqueda y selección de estudios

La cadena de búsqueda anterior se introdujo en cada base de datos científica y se aplicaron filtros para seleccionar artículos de investigación y conferencias en las disciplinas de ingeniería de software, desarrollo de software, computación, aprendizaje automático e inteligencia artificial. Como resultado, se obtuvieron 561 estudios repartidos como se presenta en la Figura 2 a). Posteriormente, se eliminaron los estudios repetidos, así como los que no incluían en sus palabras clave, resumen o título, algunos de los términos considerados como deuda, inteligencia artificial o aprendizaje automático. A continuación, se realizó la lectura de cada estudio, identificando su objetivo, metodología y resultados; de esta manera, se eliminaron aquellas investigaciones que abordaban temáticas ajenas a la DTIA y sus repercusiones sociotécnicas. Finalmente, se obtuvieron 43 estudios distribuidos como se muestra en la Figura 2 b).



a) Encontrados **b)** Seleccionados

Figura 2. Estudios sobre DTIA en bases de datos científicas.
Fuente: Elaboración propia.

2.4 Extracción de datos

Una vez que se obtuvo el conjunto final de estudios para responder las preguntas P1 y P2, se extrajeron los siguientes datos: título del estudio, año de su publicación, revista o conferencia, autor(es), institución de adscripción, país e identificador de la publicación electrónica. Posteriormente, se identificaron los temas relevantes relacionados con la pregunta P1, considerando los aspectos prácticos de la DTIA y sus posibles variedades en las que puede darse. En cuanto a la pregunta P2, se identificaron los problemas de la DTIA, tanto en lo técnico al desarrollar software basado en IA como en lo social concerniente a su uso.

2.5 Síntesis de resultados

Después de concluir la extracción de datos, estos fueron agrupados de acuerdo con sus similitudes y diferencias, con lo que se reconocieron las temáticas sobresalientes. Esto permitió organizar y sintetizar la información como se describe en los resultados.

3 Resultados

La Figura 3 ilustra el número de estudios encontrados por año de publicación en esta RSL sobre DTIA. Adicionalmente, la Tabla 1 presenta las fuentes de donde provienen los estudios seleccionados. Se indica el nombre de la revista o conferencia y la cantidad de estudios correspondiente.

3.1 Tipos y prácticas de DTIA –P1

En la literatura sobre la DTIA, principalmente se estudian los tipos de DTIA, los antipatrones que la originan, las prácticas empleadas para desarrollar software dirigido por la IA, así como las que permiten la identificación y mitigación de la DTIA.

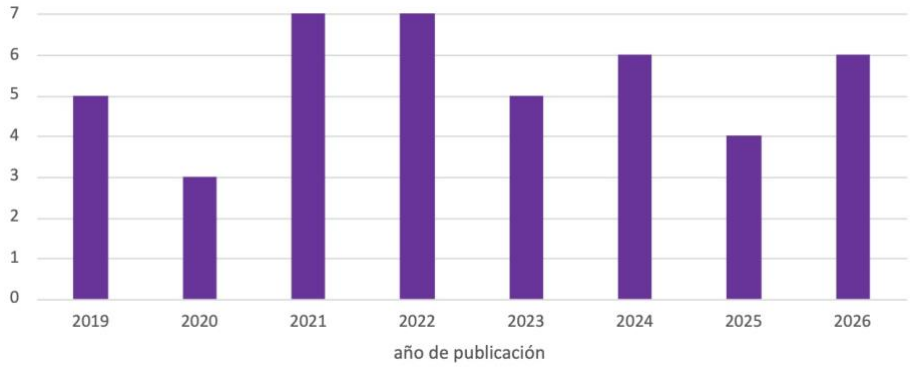


Figura 3. Estudios seleccionados por año de publicación
Fuente: Elaboración propia.

Tabla 1. Estudios seleccionados por fuente literaria

Fuente Literaria	Tipo	Cantidad
IEEE/ACM International Conference on Technical Debt	Conferencia	1
IEEE/ACM International Conference on Software Engineering	Conferencia	3
IEEE/ACM International Conference on AI Engineering – Software Engineering for AI	Conferencia	4
IEEE Conference	Conferencia	1
International Conference on Evaluation and Assessment in Software Engineering	Conferencia	1
ACM/IEEE International Symposium on Empirical Software Engineering and Measurement	Conferencia	1
2nd International Workshop on Requirements Engineering for AI-Powered Software	Conferencia	1
ACM Conference	Conferencia	3
ACIS	Conferencia	1
Journal of Systems and Software	Revista	5
ACM Transactions on Software Engineering and Methodology	Revista	4
SSRN	Revista	2
Empirical Software Engineering	Revista	2
Journal of Software Engineering Research and Development	Revista	1
Science	Revista	2
Nature Communications	Revista	1
Science and Engineering Ethics	Revista	1
Data Science and Management	Revista	1
Humanities and Social Sciences Communications	Revista	1
Harvard Business Review	Revista	1
Ethics and Information Technology	Revista	1
Applied Sciences	Revista	2
AI Ethics	Revista	1
Queue	Revista	1
Product-Focused Software Process Improvement LNCS	Capítulo	1

Fuente: Elaboración propia.

Desde la óptica de su intencionalidad, la Deuda Técnica – DT puede darse en forma deliberada o no. En el primer caso se le conoce como DT auto-admitida (Potdar & Shihab, 2014) y suele informarse manualmente por quien incurre en ella. Particularmente, entre todos los tipos de DT auto-admitida, la relacionada con el preprocesamiento de datos es la más frecuente, aunque también es la que menor esfuerzo requiere para tratarla (OBrien y otros, 2022). De cualquier forma, existe una diferencia de reporte entre la DT auto-admitida y la DT que se mide (Pavlič y otros, 2022), lo que deja ver que alguna DT podría no ser incluida en métricas y mediciones, por tanto, las acciones para su mitigación perderían efectividad.

Adicionalmente, la literatura extiende la clasificación de tipos de DT incidiendo a lo largo de las distintas fases del desarrollo. Se puede encontrar desde la DT de requerimientos (Melo y otros, 2022), datos, infraestructura, pipelines (Moreschini y otros, 2026), algoritmos (Simon y otros, 2026; Simon y otros, 2023), modelo, configuración, ética (Bogner y otros, 2021), justicia (de Souza Santos y otros, 2025), integración, *prompts* (Ehsani y otros, 2026; Aljohani & Do, 2025) y accesibilidad (Sklavenitis & Kalles, 2025). En cuanto a las primeras etapas del desarrollo, Alahdab y Çalıklı (2019) encontraron que existe DT oculta en los sistemas basados en IA que emerge en la fase de prototipado. Mientras tanto, Liu y otros (2020) exponen que "existe una cantidad significativa de DT en los conjuntos de herramientas de aprendizaje profundo, tales como deuda de diseño, documentación, pruebas, requerimientos, compatibilidad y algoritmos". En referencia a esta última, Simon y otros (2023) proponen investigar el impacto de la DT de algoritmos en las inconsistencias generadas por la IA y apuntan también a la necesidad de impulsar la creación de estándares para preservar la calidad del software impulsado por la IA. Cualquiera que sea la etapa del desarrollo, la DTIA puede llegar a ser tan dañina al grado de paralizar todo el ecosistema de trabajo, dando lugar a la aparición de nuevos tipos de deuda, incluso a nivel cognitivo (Storey, 2026). De aquí la importancia de detectar oportunamente todos los factores de riesgo que conlleva emplear herramientas de IA, IA generativa, aprendizaje automático y aprendizaje profundo durante el desarrollo de los sistemas.

En el tema de los antipatrones, en un estudio empírico (Tang y otros, 2021) descubrieron nuevas categorías de DTIA específicas de los sistemas de aprendizaje automático y proponen los antipatrones correspondientes a los tipos de datos personalizados, código de extracción de características duplicado, código de modelo innecesario o duplicado, así como a la reutilización, comprensión y modificabilidad del código de modelo. Estos resultados complementan los antipatrones explorados por (Sculley y otros, 2015), entre los que se encuentran la erosión de límites, el entrelazamiento, los bucles de retroalimentación ocultos, consumidores no declarados, dependencias de datos, problemas de configuración, así como otros antipatrones a nivel de sistema. De estos últimos, Sculley y otros (2015) describen el código de pegamento –*glue code*–, junglas de pipelines, rutas de código experimentales, deuda de abstracción y varios olores comunes. Por otra parte, Shome y otros (2022) identificaron 14 problemas recurrentes en conjuntos de datos públicos, como signos tempranos de generación de DT en sistemas de aprendizaje automático y los agruparon en 4 categorías: valores redundantes, categóricos, faltantes y de cadena.

Es así como la ausencia de buenas prácticas permite que emerjan diversos antipatrones que generan DTIA. Entre los más frecuentes en la literatura están los relacionados con las deficiencias en los datos y los modelos (Bogner y otros, 2021). En este sentido, (Foidl y otros, 2022) estudiaron los olores de datos –*data smells*– que representan indicios de anomalías en los datos que alimentan los sistemas dirigidos por la IA y categorizaron 36 olores asociados a la credibilidad, comprensión y consistencia de los datos. Particularmente, este tipo de olores proliferan en esquemas de bases de datos. Además, (Foidl y otros, 2022) exponen las consecuencias de no detectarlos, entre las que se encuentran la generación de errores en inferencias, en el procesamiento y en las funcionalidades, hasta llegar a frenar la mantenibilidad a largo plazo. El impacto adverso de estos olores resulta especialmente preocupante cuando los sistemas se emplean en sectores críticos, como el de la salud (Foidl y otros, 2022). Los olores también emergen en el código –*code smells*– en formas muy variadas. Al respecto, (Zhang y otros, 2022) han logrado detectar hasta 22 clasificaciones. Por ello, estos catálogos de olores deben formar parte del aprendizaje de los estudiantes en carreras de desarrollo de software, permitiéndoles conocer y detectar con mayor facilidad los antipatrones propios de los ecosistemas de trabajo que utilizan herramientas de IA.

En cuanto a las prácticas empleadas, (Serban y otros, 2024) integraron un repertorio de prácticas de ingeniería para diseñar sistemas de software con componentes de aprendizaje automático y midieron su grado de adopción y sus resultados. Se halló que la adopción de prácticas es mayor cuando los equipos son más grandes, aunque tienden a descuidar los aspectos de confiabilidad; de cualquier forma, las prácticas más ampliamente adoptadas tienen pocos efectos relevantes. Más aún, (Martínez Fernández y otros, 2022) exponen que existe poco conocimiento sobre los enfoques de ingeniería de software para construir, operar y mantener sistemas basados en IA. Sobre todo, los desarrolladores independientes no cuentan con guías para construir software basado en IA y se enfrentan a tales proyectos sin supervisión o apoyo alguno. Al respecto, (Dolata y otros, 2024) exponen que los desarrolladores independientes que utilizan la IA generativa pueden no saber cómo tratar la imprevisibilidad de su producción, la aparición de alucinaciones y el esfuerzo inconsistente en los ciclos de prueba y error, así como las limitaciones de marcos de trabajo específicos de la IA.

De cualquier forma, los desarrolladores de IA emplean prácticas como revisiones manuales y refactorización para identificar y tratar la DTIA (Recupito y otros, 2024; Tang y otros, 2021), así como detectores de olores para

encontrar problemas sospechosos en los datos, en el código e incluso en los algoritmos (Shome y otros, 2022; Zhang y otros, 2022; Foidl y otros, 2022; Simon y otros, 2023) como formas de detección temprana para evitar la acumulación e intratabilidad. En cuanto a las prácticas de gestión de la DTIA, se adoptan cada vez más los marcos de trabajo como *Machine Learning Operations –MLOps* (Moreschini y otros, 2026). Estas prácticas se encaminan a favorecer la calidad, mantenibilidad y evolución de los sistemas.

No obstante, la literatura advierte sobre diversas cuestiones a las que se enfrentan los ecosistemas de desarrollo y que obstaculizan un acertado manejo de la DTIA. Liu y otros (2020) hallaron que 'los desarrolladores de herramientas de IA son conscientes de la DTIA' –p. 5–, en tanto que 'quienes construyen software típicamente no tienen conocimiento acerca de este tipo de deuda' –p. 3–. Se ha encontrado que la DTIA generada de forma consciente suele aparecer durante las primeras etapas del desarrollo (Bhatia y otros, 2025). Más aún, los desarrolladores de software basados en IA suelen realizar suposiciones (Yang y otros, 2021) que persisten al escribir código, requiriendo de gran esfuerzo para eliminarlas y provocando muchas veces una gestión tardía de una DT que invariablemente se ha acumulado. Por el contrario, existen deudas que los mismos desarrolladores detectan y eliminan con mayor frecuencia, como la deuda de requerimientos y de diseño; no obstante, suelen hacerlo bajo confrontaciones o acuerdos subóptimos (Liu y otros, 2021). Lamentablemente, este tipo de acuerdos ocurren también durante la implementación (Bhatia y otros, 2025). Por tanto, si no se logran pactar las mejores soluciones, es posible que persista la DTIA.

La literatura muestra que existe una expansión de las categorías tradicionales de deuda técnica cuando esta se genera en el contexto de la IA. Se observa el surgimiento de nuevas formas de deuda asociadas a los requerimientos, los modelos, la configuración, los algoritmos e incluso a aspectos éticos. Además, los antipatrones en sistemas de aprendizaje automático amplifican la acumulación de deuda. También, la literatura expone una brecha de conocimiento sobre la DTIA entre desarrolladores de IA y desarrolladores de software; esta situación puede ser adversa para lograr un adecuado y oportuno tratamiento de la DTIA. Ahora bien, si existen estrategias para identificar y mitigar la DTIA, no se cuenta con estándares y esquemas acreditados de formación de desarrolladores en este ámbito, lo que lleva a la necesidad de emplear enfoques reactivos, más que de prevención y control. Estos hallazgos revelan la complejidad y los riesgos que prevalecen durante el desarrollo de sistemas habilitados por la IA.

3.2 Repercusiones sociotécnicas –P2

En lo concerniente a las repercusiones de la DTIA, se indaga sobre su impacto tanto técnico como social al utilizar productos de software habilitados por IA.

Relativo al impacto técnico de la DTIA, los sistemas presentan tendencia a defectos, poca mantenibilidad, producción de dependencias, además de dificultades de acoplamiento y evolución (Recupito y otros, 2024). También, Recupito y otros (2024) hallaron que la DT de arquitectura y la DT de código tienen impactos múltiples en la calidad de los sistemas activados por IA, afectando su seguridad y mantenibilidad. En el ámbito de los sistemas de recomendación, los acuerdos subóptimos y compensaciones para eliminar la DT pueden tener efectos adversos en su rendimiento, escalabilidad y mantenibilidad a largo plazo (Moreschini y otros, 2024). Por ello, los investigadores están buscando mejorar la calidad de los sistemas basados en IA. Por ejemplo, Singh (2023) propone abordar un enfoque centrado en los datos para evaluar su calidad, preprocesamiento, aprendizaje por transferencia y aprendizaje automático en conjunto con la práctica de *DevOps*.

Considerando los temas éticos y de vulneración del bien social, la DTIA tiene un impacto social importante. Respecto de las consecuencias éticas del uso de la IA, (Petrozzino, 2021) encontró una variación entre quienes deciden incurrir en DT y quienes finalmente terminan pagándola. Cabe mencionar que, los sesgos y discriminaciones se revelan como los principales retos de la IA (Sartori & Theodorou, 2022; Nguyen y otros, 2026). De esto, se reportan casos en algoritmos de contratación (Bogen, 2019), reclutamiento (Chen, 2023) y cuidado de la salud (Obermeyer y otros, 2019). Por ejemplo: Bogen (2019) descubrió que la mayoría de los algoritmos de contratación se sesgan en forma preestablecida y expone que las disparidades deben tratarse en forma proactiva para garantizar la equidad. También, Chen (2023) encontró que existe sesgo algorítmico en el reclutamiento habilitado por IA, dando lugar a prácticas de contratación discriminatorias relacionadas con atributos personales como el género, la raza, el color de la piel y la personalidad. Referente al tema de salud, Obermeyer y otros (2019) mostraron que los algoritmos de predicción comerciales para ayudar a pacientes con necesidades de salud difíciles exhiben un sesgo racial significativo. También, los fallos de seguridad pueden dañar el bien social. Sobre esto, Rahman y Farhana (2021) encontraron errores relacionados con la seguridad en proyectos que hacen minería y almacenamiento de datos de COVID-19.

En este contexto, la literatura destaca la importancia de incorporar la perspectiva de las personas tanto en lo individual, como en su papel de actores sociales para enriquecer no solo el diseño y la implementación, sino también la usabilidad de los sistemas basados en IA (de Souza Santos y otros, 2024; Felstead y otros, 2023). Es así como, estas buenas prácticas contribuyen a mitigar los impactos sociales de la DTIA. En contribución, Floridi

y otros (2020) recomiendan preservar la ética, evitar intencionalidades negativas, respetar la autonomía, la explicabilidad y el sentido de justicia. Particularmente, Nguyen y otros (2026) exponen la importancia de las prácticas de gestión de requisitos de equidad en el ciclo del software para mitigar sesgos y la pérdida de confianza en los resultados emitidos por las aplicaciones basadas en la IA. Sotolani y otros (2026) destacan también las prácticas de integración de equidad, adopción de diseño inclusivo y fortalecimiento de la transparencia organizacional para reducir la deuda de equidad. Estas aportaciones son muestra del auge que está teniendo entre la comunidad científica, académica y empresarial sobre cómo la IA puede impactar positivamente a la sociedad (Tomašev y otros, 2020).

A partir de los estudios revisados, se puede percibir la DTIA como un fenómeno sociotécnico que tiene impacto no solo en la calidad de los sistemas, sino también en los resultados que estos producen para sus usuarios. Por una parte, las repercusiones técnicas de la DTIA podrían tener efectos sistémicos e incluso de deterioro del software. Por otra parte, se advierte sobre consecuencias éticas y sociales significativas para quienes no participaron en las decisiones técnicas durante el desarrollo, por ejemplo, los sesgos algorítmicos y los fallos que vulneran la seguridad y el bienestar. En consecuencia, el tratamiento y reducción de la DTIA no deben visualizarse solo como un tema de mejora tecnológica, sino también de garantía en la generación de beneficios con responsabilidad social.

4 Discusión

La presente RSL sobre DTIA permitió responder a los cuestionamientos sobre ¿cuáles son los tipos y prácticas de DTIA? –P1–, así como ¿cuáles son las repercusiones sociotécnicas de la DTIA en el desarrollo de software basado en IA? –P2–. La primera pregunta, P1, pretende averiguar si la DTIA se puede presentar de diversas formas y cuáles son estas variedades, además de conocer cuál es la práctica para tratar la DTIA durante la construcción de software basado en IA. La finalidad de la segunda pregunta P2 es investigar los impactos técnicos de la DTIA tanto en los procesos de desarrollo como en el producto final, así como también saber si existen impactos de tipo social al usar los productos de software habilitados por IA.

Esta RSL se basó en las recomendaciones propuestas por (Kitchenham & Charters, 2007). La estrategia de búsqueda implementada hizo posible encontrar las investigaciones más relevantes para abordar las preguntas P1 y P2, las que vienen principalmente de *IEEE Xplore* y de *ACM Digital Library*, después de *Science Direct*, *Google Scholar* y *Springer Link*. En el proceso de selección, se detectó que una buena parte de los estudios encontrados tanto en *IEEE Xplore*, *ACM Digital Library* como en *Springer Link* estuvieron centrados en los temas indagados. Finalmente, la síntesis de resultados reveló cinco temáticas en cuanto a la pregunta P1: tipos de DTIA, antipatrones de DTIA, prácticas para identificar la DTIA, prácticas para mitigar y gestionar la DTIA, y prácticas para desarrollar software habilitado por la IA. En cuanto a la pregunta P2, se abordan tanto el impacto técnico asociado a la calidad, como el impacto social y ético de la DTIA.

Respecto de la pregunta P1, se identificó una amplia variedad de tipos de DTIA, lo que revela que puede presentarse en prácticamente todas las fases del desarrollo. Incluso puede darse de forma oculta o bien, ser auto-admitida. Además, se detectaron antipatrones específicos asociados con el uso del aprendizaje automático que, en consecuencia, generan DTIA. La mayoría de los antipatrones se encuentran asociados a deficiencias en los datos, los modelos y la codificación. Referente a las prácticas para tratar la DTIA, se encontró que los desarrolladores hacen uso de herramientas para encontrar sospechas de deuda relacionada con los datos, el código y los algoritmos, además de realizar revisiones manuales y refactorización. También se halló que existe una adopción creciente de nuevos paradigmas para gestionar la DTIA, como los *MLOps*. Aunque la adopción de tales prácticas es amplia, aún se descuidan aspectos como la confiabilidad de las aplicaciones (Serban y otros, 2024). Más aún, la literatura muestra un vacío en la existencia de guías específicas para desarrollar software habilitado por la IA. Esto último, advierte sobre la necesidad de contar con modelos y metodologías para construir, operar y mantener este tipo de productos innovadores.

Referente a la pregunta P2, se encontró que la DTIA afecta técnicamente la calidad y seguridad de las aplicaciones, además de su mantenibilidad a largo plazo y escalabilidad. Pero también, existen consecuencias éticas y sociales al utilizar productos con DTIA. Los principales retos son los sesgos y las discriminaciones. Por ello, los estudios proponen considerar a los usuarios desde las etapas de diseño, al mismo tiempo que también se aplican los lineamientos éticos que rigen el uso de la IA en las aplicaciones. Estas contribuciones son parte del creciente interés de la comunidad por incorporar buenas prácticas para producir aplicaciones libres de deuda tanto técnica como social.

Estos hallazgos revelan áreas de oportunidad que pueden constituir una agenda de investigación valiosa para lograr avances significativos en el estudio de la DTIA.

En primer término, la DTIA debe estudiarse y valorarse con perspectiva de responsabilidad sociotécnica que va más allá de tan solo mejorar la calidad de los productos de software y disminuir sus costos. En efecto, los resultados de esta RSL muestran que los efectos de la DTIA trascienden al ámbito social. Su influencia puede repercutir directamente en las personas con posibles impactos en sus derechos, bienestar y seguridad. A partir de

estos hallazgos, se identifican oportunidades de investigación orientadas al desarrollo de marcos integrales para la gestión de la DTIA abarcando aspectos tanto técnicos como organizacionales y de orden ético. En este sentido, futuras investigaciones pueden enfocarse en el diseño de métricas e indicadores de orden multidimensional que cuantifiquen simultáneamente impactos de orden técnico, social, ético y financiero.

En segundo término, trabajos futuros pueden enfocarse en proponer modelos de evaluación integrales que permitan explicar las relaciones entre los distintos tipos de DTIA. Esta RSL evidencia la gran cantidad de tipos de DTIA manifiestos durante los ciclos del *software* y que continúan diversificándose con la aparición de tecnologías disruptivas como la IA generativa. Adicionalmente, para continuar identificando nuevas formas de DTIA, es necesario estudiarlas sin seccionarlas para averiguar si existen fenómenos como interdependencias, efectos combinados o progresivos.

Finalmente, como lo exponen diversos autores en esta revisión, es necesario desarrollar modelos y metodologías específicas para construir software dirigido por la IA. Los esquemas como *DevSecOps* y *MLOps* han permitido avanzar en este sentido; sin embargo, debe avanzarse en su consolidación y diseño de normas para su certificación. Esto representaría un esfuerzo considerable para evolucionar hacia prácticas encaminadas a la prevención y el control adecuado de la DTIA.

Declaración de conflicto de intereses

Los autores declaran no tener conflictos de interés relacionados con la investigación, la autoría o la publicación de este artículo.

ORCID ID

María Guadalupe Medina Barrera https://orcid.org/0000-0003-3074-0029
José Juan Hernández Mora https://orcid.org/0000-0003-2878-7290

Referencias

Alahdab, M., & Çalıklı, G. (2019). Empirical Analysis of Hidden Technical Debt Patterns in Machine Learning Software. *International Conference on Product-Focused Software Process Improvement. 11915*, págs. 195–202. Springer. https://doi.org/10.1007/978-3-030-35333-9_14

Aljohani, A., & Do, H. (2025). PromptDebt: A Comprehensive Study of Technical Debt Across LLM Projects. *29th International Conference on Evaluation and Assessment in Software Engineering EASE'25* (págs. 371-382). ACM. https://doi.org/10.1145/3756681.3756976

Bhatia, A., Khomh, F., Adams, B., & Hassan, A. (2025). An Empirical Study of Self-Admitted Technical Debt in Machine Learning Software. *ACM Transactions on Software Engineering and Methodology (TOSEM), 35*(8), 1- 41. https://doi.org/10.1145/3785001

Bogen, M. (2019). All the Ways Hiring Algorithms Can Introduce Bias. *Harvard Business Review*. Obtenido de https://hbr.org/2019/05/all-the-ways-hiring-algorithms-can-introduce-bias

Bogner, J., Verdecchia, R., & Gerostathopoulos, I. (2021). Characterizing Technical Debt and Antipatterns in AI-Based Systems: A Systematic Mapping Study. *2021 IEEE/ACM International Conference on Technical Debt (TechDebt)*, (págs. 64-73). https://doi.org/10.1109/TechDebt52882.2021.00016

Chen, Z. (2023). Ethics and discrimination in artificial intelligence-enabled recruitment practices. *Humanities and Social Sciences Communications, 10*, 567. https://doi.org/10.1057/s41599-023-02079-x

de Souza Santos, R., de Lima, L., Baldassarre, M. T., & Spínola, R. (2024). Preliminary insights on industry practices for addressing fairness debt. *18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement ESEM'24* (págs. 566-571). ACM. https://doi.org/10.1145/3674805.3695406

de Souza Santos, R., Fronchetti, F., Freire, S., & Spinola, R. (2025). Software Fairness Debt: Building a Research Agenda for Addressing Bias in AI Systems. *ACM Transactions on Software Engineering and Methodology, 34*(5), 1-21. https://doi.org/10.1145/3709357

Dolata, M., Lange, N., & Schwabe, G. (2024). Development in times of hype: How freelancers explore Generative AI? *IEEE/ACM 46th International Conference on Software Engineering (ICSE '24)* (págs. 1-13). New York: Association for Computing Machinery. https://doi.org/10.1145/3597503.3639111

Ehsani, R., Rawal, S., Cai, Y., & Chatterjee, P. (2026). Faster Code, Deeper Debt? A Multivocal Literature Review on Technical Debt and Its Early Signs in LLM-Assisted Software Development. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 1-38. https://doi.org/10.1145/3820165

Felstead, C., Stockdale, R., & Scheepers, H. (2023). A dignity perspective on the potential harm of AI technologies: The case of Robodebt. *ACIS 2023 Proceedings.* Obtenido de https://aisel.aisnet.org/acis2023/100

Floridi, L., Cowls, J., King, T., & Taddeo, M. (2020). How to Design AI for Social Good: Seven Essential Factors. *Science and Engineering Ethics, 26*, 1771-1796. https://doi.org/10.1007/s11948-020-00213-5

Foidl, H., Felderer, M., & Ramler, R. (2022). Data Smells: Categories, Causes and Consequences, and Detection of Suspicious Data in AI-based Systems. *1st International Conference on AI Engineering – Software Engineering for AI (CAIN)* (págs. 229-239). Association for Computing Machinery. https://doi.org/10.1145/3522664.3528590

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering, version 2.3.* Durham University . EBSE Technical Report EBSE-2007-01. Obtenido de https://ebse.webspace.durham.ac.uk/ebse-bibliography/guidelines-for-performing-systematic-literature-reviews-in-software-engineering/

Liu, J., Huang, Q., Xia, X., Lo, D., & Li, S. (2020). Is Using Deep Learning Frameworks Free? Characterizing Technical Debt in Deep Learning Frameworks. *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society* (págs. 1-10). Seoul: Association for Computing Machinery. https://doi.org/10.1145/3377815.3381377

Liu, J., Huang, Q., Xia, X., Shihab, E., Lo, D., & Li, S. (2021). An exploratory study on the introduction and removal of different types of technical debt in deep learning frameworks. *Empirical Software Engineering, 26*. https://doi.org/10.1007/s10664-020-09917-5

Martínez Fernández, S., Bogner, J., Franch, X., Oriol, M., Siebert, J., Trendowicz, A., . . . Wagner, S. (2022). Software engineering for AI-based systems: a survey. *ACM Transactions on Software Engineering and Methodology (TOSEM), 31*(2), 1-59. https://doi.org/10.1145/3487043

Melo, A., Fagundes, R., Lenarduzzi, V., & Barbosa Santos, W. (2022). Identification and measurement of Requirements Technical Debt in software development: A systematic literature review. *Journal of Systems and Software, 194*, 111483. https://doi.org/10.1016/j.jss.2022.111483

Moreschini, S., Arvanitou, E. M., Kanidou, E. P., Nikolaidis, N., Su, R., Ampatzoglou, A., . . . Lenarduzzi, V. (2026). The Evolution of Technical Debt from DevOps to Generative AI: A multivocal literature review. *The Journal of Systems and Software, 231*, 112599. https://doi.org/10.1016/j.jss.2025.112599

Moreschini, S., Lenarduzzi, V., & Coba, L. (2024). Towards a Technical Debt for AI-based Recommender System. *TechDebt '24: Proceedings of the 7th ACM/IEEE International Conference on Technical Debt.* Association for Computing Machinery. https://doi.org/10.1145/3644384.3648574

Nguyen, T., Boufaied, C., & Santos de Souza, R. (2026). A Gray Literature Study on Fairness Requirements in AI-enabled Software Engineering. *Proceedings of the 2nd International Workshop on Requirements Engineering for AI-Powered SoftwarE, RAISE'26* (págs. 1-9). Association for Computing Machinery. https://doi.org/10.1145/3786174.3788351

Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science, 366*(6464), 447-453. https://doi.org/10.1126/science.aax2342

OBrien, D., Biswas, S., Imtiaz, S., Abdalkareem, R., Shihab, E., & Rajan, H. (2022). 23 shades of self-admitted technical debt: an empirical study on machine learning software. *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (págs. 734-746). Association for Computing Machinery. https://doi.org/10.1145/3540250.3549088

Pavlič, L., Hliš, T., Heričko, M., & Beranič, T. (2022). The Gap between the Admitted and the Measured Technical Debt: An Empirical Study. *Applied Sciences, 12*(15), 7482. https://doi.org/10.3390/app12157482

Petrozzino, C. (2021). Who pays for ethical debt in AI? *AI Ethics, 1*, 205-208. https://doi.org/10.1007/s43681-020-00030-3

Potdar, A., & Shihab, E. (2014). An Exploratory Study on Self-Admitted Technical Debt. *2014 IEEE International Conference on Software Maintenance and Evolution* (págs. 91-100). IEEE. https://doi.org/10.1109/ICSME.2014.31

Rahman, A., & Farhana, E. (2021). An Empirical Study of Bugs in COVID-19 Software Projects. *Journal of Software Engineering Research and Development, 9*(3), 3:1 – 3:19. https://doi.org/10.5753/jserd.2021.827

Recupito, G., Pecorelli, F., Catolino, G., Lenarduzzi, V., Taibi, D., Di Nucci, D., & Palomba, F. (2024). Technical debt in AI-enabled systems: On the prevalence, severity, impact, and management strategies

for code and architecture. *Journal of Systems and Software, 216*, 112151. https://doi.org/10.1016/j.jss.2024.112151

Sartori, L., & Theodorou, A. (2022). A sociotechnical perspective for the future of AI: narratives, inequalities, and human control. *Ethics and Information Technology, 24*(4). https://doi.org/10.1007/s10676-022-09624-3

Sculley, D., Holt, G., Golovin, D., Eugene Davydov, Phillips, T., Ebner, D., . . . Dennison, D. (2015). Hidden Technical Debt in Machine Learning Systems. *NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems* (págs. 2503-2511). MIT Press. Obtenido de https://dl.acm.org/doi/10.5555/2969442.2969519

Serban, A., van der Blom, K., Hoos, H., & Visser, J. (2024). Software engineering practices for machine learning — Adoption, effects, and team assessment. *Journal of Systems and Software, 209*, 111907. https://doi.org/10.1016/j.jss.2023.111907

Shome, A., Cruz, L., & Deursen, A. (2022). Data Smells in Public Datasets. *CAIN '22: Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI* (págs. 205-216). Association for Computing Machinery. https://doi.org/10.1145/3522664.3528621

Simon, E. I.-O., Hettiarachchi, C., Potanin, A., Suominen, H., & Fard, F. (2026). Automated detection of algorithm debt in deep learning frameworks: an empirical study. *Empirical Software Engineering, 31*, 66. https://doi.org/10.1007/s10664-026-10807-5

Simon, E. I.-O., Vidoni, M., & Fard, F. H. (2023). Algorithm Debt: Challenges and Future Paths. *2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN)*, (págs. 90-91). https://doi.org/10.1109/CAIN58948.2023.00020

Singh, P. (2023). Systematic review of data-centric approaches in artificial intelligence and machine learning. *Data Science and Management, 6*(3), 144-157. https://doi.org/10.1016/j.dsm.2023.06.001

Sklavenitis, D., & Kalles, D. (2025). A Scoping Review and Assessment Framework for Technical Debt in the Development and Operation of AI/ML Competition Platforms. *Applied Sciences, 15*, 7165. https://doi.org/10.3390/app15137165

Sotolani, R., Freire, S., Fronchetti, F., De Souza Santos, R., & Spinola, R. (2026). Exposing hidden bias: A study of fairness debt in gray literature. *Journal of Systems and Software, 238*, 112866. https://doi.org/10.1016/j.jss.2026.112866

Storey, M. A. (2026). From Technical Debt to Cognitive and Intent Debt. *ACM Queue, 24*. https://doi.org/10.1145/3807966

Taddeo, M., & Floridi, L. (2018). How AI can be a force for good. *Science, 361*, 751-752. https://doi.org/10.1126/science.aat5991

Tang, Y., Khatchadourian, R., Bagherzadeh, M., Singh, R., Stewart, A., & Raja, A. (2021). An Empirical Study of Refactorings and Technical Debt in Machine Learning Systems. En Y. Tang, R. Khatchadourian, M. Bagherzadeh, R. Singh, A. Stewart, & A. Raja (Ed.), *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, (págs. 238-250). https://doi.org/10.1109/ICSE43902.2021.00033

Thibodeau, S. (2025). Reconceptualizing Technical Debt as an Organizational Security, Safety, and Sociotechnical Risk. *Safety and Security Advances, 2*(1), 37-52. https://doi.org/10.61093/ssa.2(1).37-52.2026

Tomašev, N., Cornebise, J., Hutter, F., Mohamed, S., Picciariello, A., Connelly, B., . . . Clopath, C. (2020). AI for social good: unlocking the opportunity for positive impact. *Nature Communications, 11*. https://doi.org/10.1038/s41467-020-15871-z

Yang, C., Liang, P., Fu, L., & Li, Z. (2021). Self-Claimed Assumptions in Deep Learning Frameworks: An Exploratory Study. *EASE '21: Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineerin* (págs. 139 - 148). Association for Computing Machinery. https://doi.org/10.1145/3463274.3463333

Zhang, H., Cruz, L., & Deursen, A. (2022). Code Smells for Machine Learning Applications. *CAIN '22: Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI* (págs. 217-228). Association for Computing Machinery. https://doi.org/10.1145/3522664.3528620