International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 46 No. 2 Aug. 2025, pp. 146-152
© 2025 Innovative Space of Scientific Research Journals
http://www.ijias.issr-journals.org/

Aplicación de patrones de interacción en componentes sobre Storybook y su impacto en la deuda técnica en proyectos web

[Application of interaction patterns to Storybook components and their impact on technical debt in web projects]

Jonathan Delgado Guerrero[1-2], Yuliana León Bazan[3], and Christopher Crespo León[3]

[1]Facultad de Informática, Universidad Nacional de La Plata, La Plata, Buenos Aires, Argentina

[2]SENESCYT, Guayaquil, Guayas, Ecuador

[3]Facultad de Ciencias Matemáticas y Físicas, Universidad de Guayaquil, Guayaquil, Guayas, Ecuador

Copyright © 2025 ISSR Journals. This is an open access article distributed under the ***Creative Commons Attribution License***, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT: This study focuses on technical debt in web development, specifically how the early adoption of Storybook can mitigate this challenge. Our research used a mixed-methods approach, combining quantitative data on code quality and technical debt metrics with a qualitative case study involving three development teams. Storybook is a powerful tool for designing components with well-defined interaction patterns. By integrating these practices, it not only improves software quality and maintainability but also aids work in agile development environments. Technical debt often arises when teams create components with deficient interaction patterns; however, Storybook enables programmers to build components with consistent interaction patterns, which increases productivity and minimizes errors. In summary, the findings strongly suggest that integrating atomic design principles with interaction patterns and specialized tools like Storybook optimizes development processes, enhances technical debt management, and leads to the creation of more resilient and effective software.

KEYWORDS: Atomic design, interaction patterns, technical debt web projects.

RESUMEN: Este estudio se enfoca en la deuda técnica en el desarrollo web, centrándose en cómo la adopción temprana de Storybook puede mitigar este desafío. La investigación empleó un enfoque de métodos mixtos, combinando datos cuantitativos sobre la calidad del código y métricas de deuda técnica con un estudio de caso cualitativo en tres equipos de desarrollo. Storybook es una herramienta potente para diseñar componentes con patrones de interacción bien definidos. Al integrar estas prácticas, no solo se mejora la calidad y mantenibilidad del software, sino que también se ayuda al trabajo en entornos de desarrollo ágiles. La deuda técnica a menudo surge cuando los equipos desarrollan componentes con patrones de interacción deficientes; sin embargo, Storybook permite a los programadores crear componentes con patrones de interacción consistentes, lo que incrementa la productividad y minimiza los errores. En síntesis, las conclusiones sugieren firmemente que la integración de principios de diseño atómico con patrones de interacción y herramientas especializadas como Storybook optimiza los procesos de desarrollo, mejora la gestión de la deuda técnica y conduce a la creación de software más resiliente y efectivo.

PALABRAS-CLAVES: Diseño atómico, patrones de interacción, deuda técnica proyectos web.

Corresponding Author: Jonathan Delgado Guerrero

1 INTRODUCCIÓN

En el desarrollo actual de aplicaciones web, la deuda técnica se ha convertido en un desafío significativo para los equipos de desarrollo. Este término describe el costo acumulado por decisiones rápidas que, a largo plazo, dificultan el mantenimiento y escalabilidad del software. Históricamente, la deuda técnica surge al priorizar soluciones temporales sobre prácticas óptimas desde el inicio [1]. Este problema se agrava en entornos sin un enfoque estructurado para el diseño y creación de componentes, lo que aumenta la complejidad del código y reduce la calidad del producto final [2]. En este contexto, surge la pregunta de investigación: ¿puede la implementación temprana de herramientas como Storybook, que facilitan la creación de componentes con diseño atómico e integran patrones de interacción, reducir significativamente la deuda técnica en el desarrollo de aplicaciones web? Este trabajo explora esta cuestión, buscando conectar la adopción de metodologías de diseño atómico con la reducción de deuda técnica mediante la reutilización de componentes y una mejor colaboración entre desarrolladores. Un objetivo clave es evaluar la efectividad de Storybook como herramienta de desarrollo y como recurso para promover prácticas que aseguren la sostenibilidad del código a largo plazo, especialmente mediante la integración de patrones de interacción en componentes atómicos. Comprender cómo estas herramientas disminuyen la deuda técnica puede guiar a los desarrolladores hacia estrategias más efectivas [3], ofreciendo pautas prácticas para optimizar los procesos de desarrollo.

2 REVISIÓN DE LA LITERATURA

La evolución del software ha impulsado la búsqueda de métodos que mejoren la calidad del código y reduzcan la deuda técnica. Storybook, una herramienta que permite crear y visualizar componentes aislados alineados con el diseño atómico, fomenta interfaces modulares, facilitando la reutilización y organización del desarrollo [4], [5]. Además, Storybook posibilita la integración de patrones de interacción en estos componentes, mejorando la consistencia y la experiencia del usuario. La deuda técnica, entendida como decisiones de diseño y programación que generan problemas a largo plazo pese a su utilidad inicial, es un reto constante. Implementar estas herramientas desde el inicio podría mitigarlo, justificando la necesidad de estudiar su efectividad. Estudios previos [6], [7], [8] han analizado cómo prácticas ágiles y diseño basado en componentes reducen la deuda técnica, aunque suelen centrarse en herramientas o metodologías específicas, no en la integración temprana de componentes atómicos con patrones de interacción. La literatura destaca que una estructura sólida de componentes genera código más limpio y mejora la colaboración entre diseñadores y desarrolladores, reduciendo malentendidos que incrementan la deuda técnica [9]. Otros trabajos abordan la deuda técnica desde la experiencia del usuario (UXDebt), proponiendo modelos para gestionarla [10]. Sin embargo, la implementación práctica de Storybook en diversos entornos y su impacto en patrones de interacción requieren mayor investigación, dado que la habilidad de los equipos para adoptar estas prácticas varía. El diseño modular, potenciado por herramientas como Storybook, permite desarrollar componentes reutilizables con comportamientos consistentes, reduciendo la complejidad del código [11]. Esto mejora la comunicación y asegura que las decisiones de diseño se trasladen eficazmente a la implementación técnica. No obstante, elegir la herramienta adecuada es crucial, ya que un mal uso puede aumentar la carga de trabajo [12]. Análisis de código muestran que estas herramientas mejoran la documentación y facilitan la integración de nuevas funcionalidades mediante componentes bien definidos. El diseño atómico, con énfasis en patrones de interacción, promueve consistencia visual y mantenibilidad, reduciendo la deuda técnica al desacoplar componentes en lugar de tratar las aplicaciones como bloques monolíticos [13].

3 METODOLOGÍA

La deuda técnica refleja decisiones a corto plazo que afectan la calidad y eficiencia del software a largo plazo. Herramientas como Storybook, que apoyan el diseño atómico e integran patrones de interacción, pueden abordar este problema. Este estudio investiga cómo la adopción temprana de Storybook reduce la deuda técnica en aplicaciones web, evaluando su efectividad en el desarrollo de componentes con patrones de interacción definidos y su impacto en la calidad y mantenibilidad del código. Se emplean análisis cuantitativos (métricas de deuda técnica) y cualitativos (estudio de caso sobre la experiencia de los desarrolladores), proporcionando una comprensión clara de la relación entre diseño atómico, patrones de interacción y gestión de la deuda técnica.

3.1 CONTEXTO DE APLICACIÓN

La empresa, ubicada en Ecuador y de tamaño mediano, promociona anuncios en medios físicos (revistas) y digitales (portal web), con una base de 12,000 clientes. Ofrece un software para gestionar estadísticas y datos de clientes, sin aplicaciones móviles. Cuenta con departamentos de Sistemas (3 desarrolladores y un líder) y Diseño (3 diseñadores y un jefe), responsables de bocetos y maquetas.

3.2 Recolección De Datos

Se analizó un requerimiento de complejidad media, diseñando componentes atómicos con patrones de interacción en Storybook durante dos semanas. SonarQube evaluó la calidad del código (duplicidad, bugs, code smells) [14], comparándolo con un desarrollo previo similar. Un SUS [6] midió la experiencia de los desarrolladores con Storybook.

3.3 Implementación De Patrones De Interacción En Storybook

El enfoque de Storybook basado en componentes permite a los desarrolladores encapsular patrones de interacción dentro de componentes reutilizables. Por ejemplo, un componente de botón con efecto de hover puede definirse con estados estandarizados (predeterminado, hover, activo, deshabilitado) y documentarse en Storybook. Esto asegura que todas las instancias del botón en el proyecto sigan el mismo patrón de interacción, reduciendo inconsistencias.

Desarrollamos una biblioteca de patrones de interacción en Storybook, que incluye:

- Patrón de Validación de Formularios: Asegura un manejo consistente de errores y retroalimentación al usuario en los campos de entrada.
- Patrón de Interacción de Modales: Estandariza el comportamiento de los diálogos modales, incluyendo animaciones de abrir/cerrar y características de accesibilidad.
- Patrón de Menú de Navegación: Implementa un menú responsivo y plegable con estilo y comportamiento consistentes.

Cada patrón fue documentado con guías de uso, consideraciones de accesibilidad y casos de prueba, mejorando la colaboración entre desarrolladores y la reutilización de componentes.

La implementación de patrones de interacción en Storybook ofrece varios beneficios clave:

- Consistencia: Los patrones estandarizados minimizan las variaciones en el comportamiento de los componentes, lo que a su vez reduce los "code smells" (malas prácticas de código).
- Reutilización: Los componentes modulares pueden ser reutilizados en distintos proyectos, disminuyendo la duplicación de código.
- Capacidad de prueba: Al estar aislados, los componentes son más fáciles de probar, lo que mejora la cobertura del código y reduce los defectos.
- Colaboración: La documentación visual de Storybook fomenta una mejor comunicación entre desarrolladores y diseñadores.

3.4 Diseño Experimental

Comparamos dos versiones del proyecto:

- Línea Base: Desarrollada sin la implementación de Storybook, dependiendo de una creación de componentes ad-hoc (según la necesidad del momento).
- Mejorada con Storybook: Esta versión incorporó Storybook junto con sus patrones de interacción.

Las métricas se recopilaron durante un periodo de tres meses, y la retroalimentación cualitativa se obtuvo a través de entrevistas estructuradas y encuestas SUS (System Usability Scale).

3.5 Métricas De Deuda Técnica Previas

Las métricas de deuda técnica fueron tomadas desde SonarQube enlazado a la rama de desarrollo de una funcionalidad con id feature-6530, como consta en la Imagen 1 [15]. Este desarrollo fue el último previo y que consta de características de complejidad, estimación de tiempo y número de líneas similares (4000) a las de la propuesta que hace uso de storybook. En la misma se evidencian 12 issues (categorizados como code smells), 15.2% de duplicidad de código en comparación con el código de la rama master. También se visualiza una métrica de cobertura de tests que no supera la establecida por el proyecto (70%).

3.6 EVALUACIÓN DE USABILIDAD DE STORYBOOK

Se propone la creación de un SUS (System Usability Scale) específico para evaluar el impacto de la adopción temprana de Storybook en la evolución de la deuda técnica en proyectos web [16]. Los ítems están directamente relacionados con cómo Storybook ha influido en la acumulación o reducción de la deuda técnica; la perspectiva del desarrollador sobre el uso de esta herramienta. En la Tabla 1 se detallan las preguntas de evaluación propuestas.

Tabla 1. Preguntas SUS para evaluar usabilidad de Storybook

		Total desacuerdo				Muy de acuerdo
		1	2	3	4	5
1	Storybook me ayudó a mantener un código más limpio y consistente a lo largo del proyecto.					
2	Gracias a Storybook, fue más fácil identificar y corregir problemas de diseño antes de que se convirtieran en una deuda técnica mayor.					
3	Considero que Storybook ha reducido significativamente la deuda técnica acumulada en este proyecto.					
4	Storybook me permitió desarrollar componentes de manera más aislada, lo que facilitó su mantenimiento y redujo la deuda técnica.					
5	La adopción temprana de Storybook hizo que fuera más fácil realizar cambios en la interfaz de usuario sin introducir nueva deuda técnica.					
6	Storybook me ayudó a comprender mejor la estructura del proyecto y a detectar posibles áreas de deuda técnica.					
7	Creo que Storybook es una herramienta valiosa para prevenir la acumulación de deuda técnica a largo plazo.					
8	Storybook facilitó la colaboración entre diseñadores y desarrolladores, lo que redujo el riesgo de introducir deuda técnica por malentendidos.					
9	Gracias a Storybook, pude entregar funcionalidades más rápido y con menor deuda técnica.					
10	Considero que Storybook es una herramienta esencial para cualquier proyecto que busque minimizar la deuda técnica.					

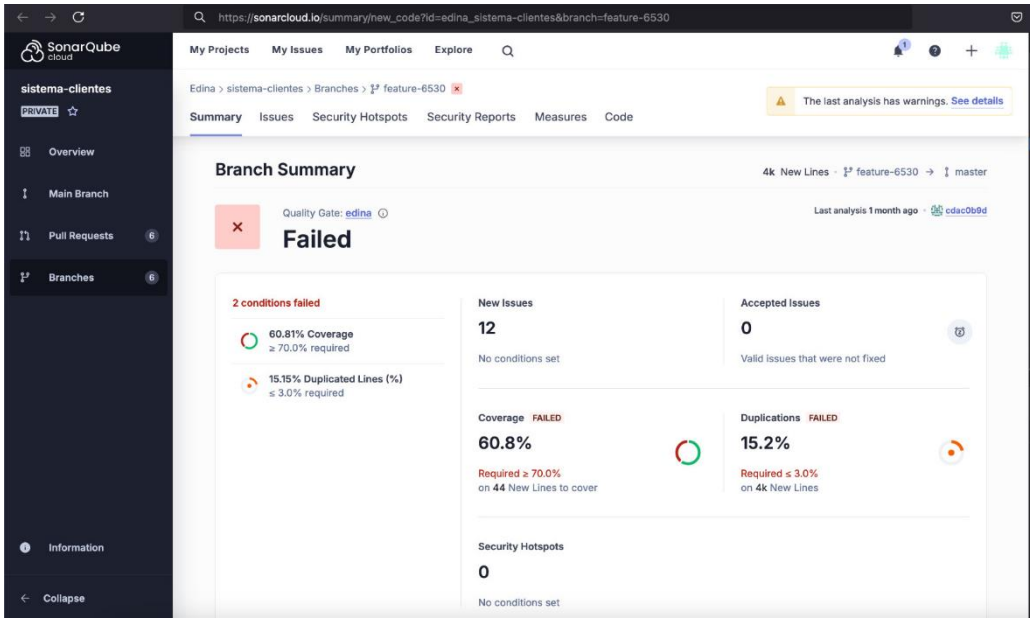

Fig. 1. Deuda técnica de la rama feature-6530 usando SonarQube (sin storybook)

4 RESULTADOS

El uso temprano de Storybook, con su enfoque en componentes atómicos y patrones de interacción, mejora significativamente la calidad del código al facilitar la creación y prueba de componentes consistentes. Se observaron menos errores y una mejor colaboración entre diseño y desarrollo, destacando la importancia de un diseño sistemático para evitar la deuda técnica. Comparado con desarrollos sin Storybook, se redujeron los problemas de código y la duplicidad, gracias a la

reutilización de componentes con patrones de interacción definidos. Usar herramientas como Storybook, que ayuda a crear componentes con un diseño atómico, se ha mostrado como una buena solución para estos problemas desde el inicio del desarrollo. Los resultados de esta investigación muestran que usar Storybook de forma temprana mejora mucho la calidad del código, ya que hace más fácil la creación y prueba de componentes de forma ordenada y clara. También se notó menos errores de implementación y mejor colaboración entre equipos de diseño y desarrollo, lo que refuerza que planificar y diseñar bien es crucial para evitar la deuda técnica en software, como se evidencia en la Imagen 2. La investigación destaca que fortalecer la colaboración entre diseñadores y desarrolladores, junto con un enfoque sistemático hacia el diseño y la documentación, puede ser clave para que proyectos de software ambiciosos tengan éxito.

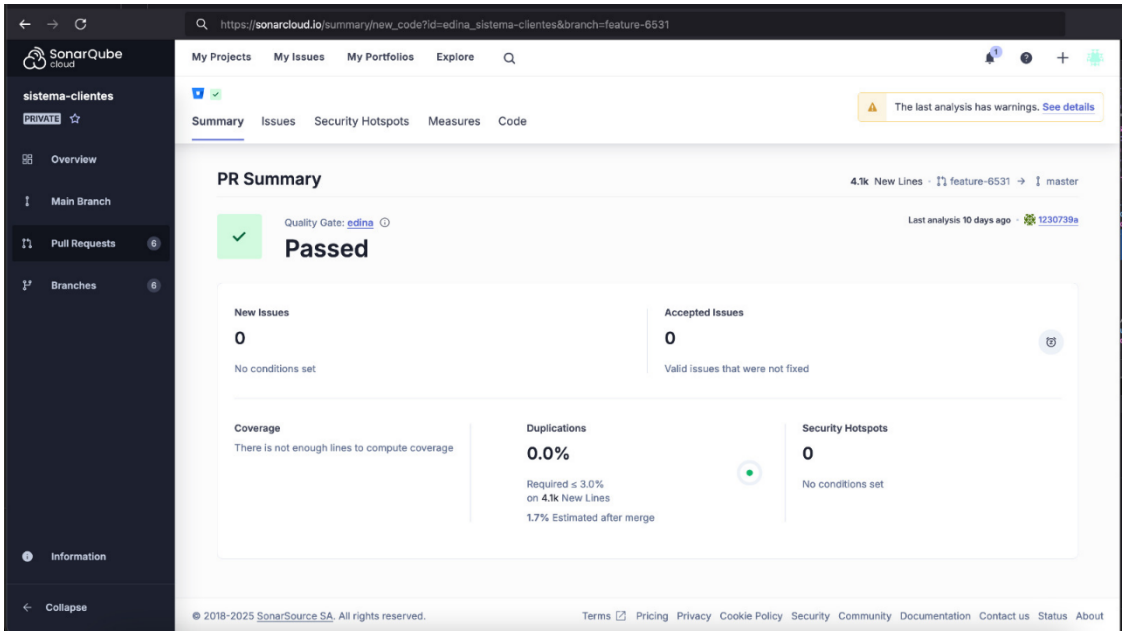

Fig. 2. Deuda técnica de la rama feature-6531 usando SonarQube (con storybook)

Si realizamos una comparación con los valores previos obtenidos del desarrollo sin la aplicación de storybook se evidencia una mejora en los números de issues y la duplicidad del código, gracias a la reutilización de componentes que posibilita el uso de Storybook complementado con el diseño atómico de los componentes visuales. En la Tabla 2 se resumen estos valores obtenidos.

Tabla 2. comparativa de deuda técnica entre feature-6530 y 6531

Característica	# Issues (Olores de código)	% Cobertura de código	% Duplicación de código	Descripción
feature-6530	12	60.8%	15.20%	Esta característica presenta un número moderado de problemas de código (code smell), lo que podría indicar áreas de mejora en la calidad del código. La cobertura de código es algo baja, sugiriendo que una parte significativa del código no está cubierta por pruebas unitarias. Además, hay un porcentaje considerable de código duplicado, lo que podría indicar oportunidades para refactorizar y mejorar la mantenibilidad del código.
feature-6531	0	70%	0%	Esta característica presenta un estado de código muy bueno. No se han detectado problemas de código, la cobertura de código es alta, lo que indica una buena confianza en el código, y no hay duplicaciones, lo que sugiere un código bien estructurado y mantenido.

La evaluación de la usabilidad de Storybook a través de la escala SUS arrojó una puntuación de 87.5, lo que indica una alta satisfacción por parte de los 3 desarrolladores. Los resultados muestran un alto nivel de acuerdo en cuanto a la capacidad de Storybook para mantener un código limpio y consistente, así como para reducir la deuda técnica. En particular, los ítems relacionados con la colaboración entre equipos y la agilidad en el desarrollo obtuvieron las puntuaciones más altas. Estos hallazgos sugieren que Storybook es una herramienta valiosa para mejorar la calidad del código y la eficiencia en proyectos similares. Se recomienda explorar la implementación de Storybook en otros proyectos para obtener beneficios similares.

5 DISCUSIÓN

Este estudio demuestra que la implementación temprana de Storybook, enfocada en patrones de interacción en componentes atómicos, reduce la deuda técnica en proyectos web. Mejora la calidad del código, disminuye errores y fortalece la colaboración entre equipos. Sin embargo, se limita a un caso específico con un equipo mediano, y no aborda rendimiento ni escalabilidad. Futuras investigaciones deberían explorar diversos contextos, el impacto de los patrones de interacción en la experiencia del usuario y la combinación de Storybook con otras herramientas.

6 CONCLUSIONES

El presente trabajo ha ayudado a entender el impacto de usar herramientas como Storybook desde el inicio en la reducción de la deuda técnica en el desarrollo de aplicaciones web. Se han tratado temas importantes como el diseño atómico, la mejora del código y el trabajo en equipo. Al responder al problema de investigación, se ha evidenciado que usar Storybook no solo ayuda a manejar mejor los componentes de software, sino que también reduce la aparición de errores y la deuda técnica a lo largo del desarrollo. Sin ser resultados concluyentes, para el futuro se sugiere que se realicen más investigaciones sobre la efectividad de Storybook en diferentes contextos y tamaños de proyectos, así como la integración de nuevas tecnologías que complementen su uso. También se recomienda investigar la experiencia del usuario y el efecto en los tiempos de entrega de proyectos cuando Storybook se implementa a gran escala.

REFERENCIAS

[1] L. F. S. Vásquez y M. J. C. P. Arriaga, «Defectos de diseño y su impacto en la deuda técnica», *AVANCES DE INVESTIGACIÓN EN INGENIERÍA APLICADA,* p. 11.

[2] L. A. Rosser y J. H. Norton, «A systems perspective on technical debt», en *2021 IEEE Aerospace Conference (50100),* IEEE, 2021, pp. 1-10.

[3] M. Siavvas, D. Tsoukalas, M. Jankovic, D. Kehagias, y D. Tzovaras, «Technical debt as an indicator of software security risk: a machine learning approach for software development enterprises», *Enterprise Information Systems,* vol. 16, n.º 5, p. 1824017, 2022.

[4] S. Handal, C. Koo, y O. Adelakun, «Design systems implementation for digital transformation: a case study», en *International Conference on Information Technology & Systems,* Springer, 2022, pp. 42-51.

[5] T. Nummela, «Using Storybook. js for component-driven design system web development», 2024.

[6] C. Hernández-Nieto, X. Sánchez, y P. Salinas, «A Mobile First Approach for the Development of a Sustainability Game», *Procedia Computer Science,* vol. 75, pp. 182-185, ene. 2015, doi: 10.1016/j.procs.2015.12.236.

[7] H. Saeeda, M. O. Ahmad, y T. Gustavsson, «Exploring Process Debt in Large-Scale Agile Software Development For Secure Telecom Solutions», en *Proceedings of the 7th ACM/IEEE International Conference on Technical Debt,* 2024, pp. 11-20.

[8] G. Soares *et al.,* «Investigating how agile software practitioners repay technical debt in software projects», en *Proceedings of the XXI Brazilian Symposium on Software Quality,* 2022, pp. 1-10.

[9] A. Chiddarwar, «TOOLS AND TECHNIQUES FOR MANAGING TECHNICAL DEBT», 2024.

[10] L. Denuzière, E. Rodriguez, y A. Granicz, «Piglets to the rescue: Declarative User Interface Specification with Pluggable View Models», en *Proceedings of the 25th symposium on Implementation and Application of Functional Languages,* en IFL '13. New York, NY, USA: Association for Computing Machinery, ago. 2013, pp. 105-115. doi: 10.1145/2620678.2620689.

[11] R. Lanciaux, Modern Front-end Architecture: Optimize Your Front-end Development with Components, Storybook, and Mise en Place Philosophy. Springer, 2021.

[12] D. A. d'Aragona, F. Pecorelli, M. T. Baldassarre, D. Taibi, y V. Lenarduzzi, «Technical debt diffuseness in the apache ecosystem: A differentiated replication», en *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER),* IEEE, 2023, pp. 825-833.

[13] B. Frost, *Atomic Design*. Brad Frost Web, 2016. [En línea]. Disponible en: https://books.google.com.ec/books?id=1e92vgAACAAJ.

[14] G. A. Campbell y P. P. Papapetrou, *SonarQube in Action,* 1st ed. USA: Manning Publications Co., 2013.

[15] D. Pina, A. Goldman, y C. Seaman, «Sonarlizer xplorer: a tool to mine github projects and identify technical debt items using sonarqube», en *Proceedings of the International Conference on Technical Debt,* 2022, pp. 71-75.

[16] P. Vlachogianni y N. Tselios, «Perceived usability evaluation of educational technology using the System Usability Scale (SUS): A systematic review», *Journal of Research on Technology in Education,* vol. 54, n.º 3, pp. 392-409, 2022.