Revista Conectividad
Enero – julio 2024 *ISSN:2806-5875*
pp. 93-108 *Correo: revista@ister.edu.ec*
Volumen 5, Número 1

Evaluando la calidad de las aplicaciones Low-Code: Un mapeo sistemático de la literatura

Assessing the Quality of Low-Code Applications: A Systematic Mapping Study

Miguel Botto-Tobar[1,2], Carlos Neil[1]

[1]Universidad Abierta Interamericana. Facultad de Tecnología Informática. Centro de Altos Estudios en Tecnología Informática. Buenos Aires, Argentina, miguelangel.bottotobar@alumnos.uai.edu.ar; carlos.neil@uai.edu.ar
[2] Universidad de Guayaquil, Guayaquil, Ecuador, miguel.bottot@ug.edu.ec

Autor para correspondencia: miguelangel.bottotobar@alumnos.uai.edu.ar

Fecha de recepción: julio 2023 ***Fecha de aceptación:*** diciembre 2023

RESUMEN

Las plataformas de desarrollo Low-Code buscan democratizar la creación de software al permitir que personas sin conocimientos técnicos profundos puedan construir aplicaciones. Esto ha reducido las solicitudes a los equipos de TI, permitiéndoles enfocarse en proyectos más complejos. Estas plataformas son altamente personalizables para satisfacer necesidades específicas, lo cual explica su creciente popularidad entre empresas que buscan optimizar sus procesos de desarrollo y mantenerse competitivas digitalmente. Sin embargo, la falta de calidad en las aplicaciones desarrolladas a través de estas plataformas es un desafío importante. En este contexto, para abordar la falta de calidad en las aplicaciones desarrolladas mediante plataformas Low-Code se realizó un mapeo sistemático de la literatura para conocer el estado actual sobre modelos, metodologías, guías y estándares de calidad, que permitan garantizar que estas aplicaciones cumplan con los requisitos y necesidades de los usuarios finales. Se identificaron 13 artículos y los resultados presentan que existen problemas basados en la falta de experiencia en desarrollo de software, limitaciones de las plataformas Low-Code, falta de pruebas exhaustivas, y dependencia de componentes, y se han realizado pruebas exhaustivas, utilizando contenedores; aprovechamiento de tecnologías como Blockchain y plataformas para el desarrollo de aplicaciones virtuales, y además técnicas de Deep Learning y Machine Learning como estrategias para garantizar la calidad. Sin embargo, es necesario la implementacion de algunas normas, estándares y directrices de desarrollo para garantizar la consistencia y la calidad en todas las aplicaciones desarrolladas a través de plataformas Low-Code.

Palabras clave: Low-Code; Aseguramiento de calidad; Desarrollo más rápido; Costos redu-cidos; Personalización.

ABSTRACT

Low-code development platforms seek to democratize software creation by allowing people without deep technical knowledge to build applications. This has reduced requests to IT teams,

allowing them to focus on more complex projects. These platforms are highly customizable to meet specific needs, which explains their growing popularity among companies looking to optimize their development processes and stay digitally competitive. However, the lack of quality in the applications developed through these platforms is a major challenge. In this context, to address the lack of quality in the applications developed through Low-Code platforms, a systematic mapping of the literature was carried out to know the current status of models, methodologies, guides and quality standards, which guarantee that these applications meet the requirements and needs of end users. 13 articles were identified and the results present that there are problems based on lack of experience in software development, limitations of Low-Code platforms, lack of exhaustive testing, and dependence on components, and extensive tests have been carried out, using containers; taking advantage of technologies such as Blockchain and platforms for the development of virtual applications, as well as Deep Learning and Machine Learning techniques as strategies to ensure quality. However, the implementation of some norms, standards, and development guidelines is necessary to ensure consistency and quality across all applications developed through Low-Code platforms.

Key words: Low-Code; Quality assurance; Faster development; Reduced costs; Customization.

INTRODUCCIÓN

Las plataformas de bajo código o también conocidas como Low-Code han surgido como soluciones prometedoras para simplificar el desarrollo de software y mejorar la productividad. Esto ha dado como resultado ciclos de desarrollo más rápidos, costos reducidos y mayor flexibilidad; y, permiten que los usuarios no técnicos participen en el proceso de desarrollo, lo que permite una mayor colaboración e innovación (Sahay et al., 2020). Además, estas plataformas no se limitan a industrias o casos de uso específicos, y se pueden utilizar para desarrollar una amplia gama de aplicaciones, desde aplicaciones móviles simples hasta soluciones empresariales complejas (Heuer et al., 2022). Esto ha llevado a un aumento significativo en la demanda de desarrolladores *I,* y ha abierto nuevas oportunidades para aquellos que buscan ingresar al campo.

Las plataformas Low-Code han ganado una atención significativa en los últimos años, debido a su potencial para simplificar el desarrollo de software y mejorar la productividad. Sin embargo, la falta de calidad ha sido un gran desafío para los desarrolladores y las organizaciones (Al Alamin et al., 2021). Por ejemplo: la falta de recursos para realizar pruebas exhaustivas y garantizar la calidad del software desarrollado (Khorram et al., 2020); la ausencia de estándares abiertos ha dificultado evaluar, investigar y contribuir a su escalabilidad (Galhardo & Silva, 2022). Además,

la ausencia de "polinización cruzada" entre desarrolladores Low-Code e investigadores de ingeniería basada en modelos (Sanchis et al., 2020).

En consecuencia, es crucial investigar este problema, debido a que se han vuelto cada vez más populares debido a su capacidad para permitir el desarrollo de aplicaciones software con un mínimo de código o nulo (Sahay et al., 2020). Y, esto ha derivado a la falta de estándares abiertos para la evaluación, la investigación y la contribución a su escalabilidad (Galhardo & Silva, 2022). Este problema está presente en el software desarrollado por desarrolladores no profesionales, debido que estas plataformas se han convertido en una herramienta importante para la transformación digital en varias industrias (Sanchis et al., 2020; Talesra & Nagaraja, 2021), y se han desarrollado rápidamente, por lo que plataformas maduras de desarrollo Low-Code actualmente juegan un papel importante en el cumplimiento de las necesidades comerciales de las empresas. La falta de calidad en las plataformas Low-Code ha sido abordado por varios estudios, tales como: Ihirwe *et al.*, enfatizan la importancia de evaluar la calidad de las plataformas de ingeniería Low-Code y basadas en modelos para facilitar el proceso de ingeniería de los sistemas de Internet de las cosas (IoT) (Ihirwe et al., 2022). Además, Galhardo y Silva identificaron el problema de las plataformas Low-Code que siguen diferentes enfoques con lenguajes propietarios, lo cual es un desafío cuando los clientes necesitan migrar a otras tecnologías o tienen la intención de definir la especificación de sus aplicaciones de forma legible y compatible con la plataforma. manera independiente (Galhardo & Silva, 2022). Por lo tanto, se espera que el aseguramiento de calidad a través de pruebas sólidas ayude a identificar defectos y garantizar que el código de producción sea sólido en diversas condiciones de uso.

TRABAJOS RELACIONADOS

Low-Code es una tendencia en la industria del desarrollo de software que busca proporcionar a aquellos sin capacitación específica en computación herramientas fáciles de usar para crear aplicaciones o programas para satisfacer una necesidad comercial particular. En el mercado, múltiples plataformas permiten a los usuarios desarrollar soluciones de software con algún código o, en algunos casos, sin necesidad de codificación. Este artículo explora los orígenes de esta tendencia y cómo ha evolucionado, y su adopción en el desarrollo de software (Vera et al., 2022).

Khorram et al. exploran las pruebas de software Low-Code, un enfoque de desarrollo que permite crear aplicaciones por medio de interfaces gráficas y componentes predefinidos en lugar de código tradicional. El estudio identifica varios desafíos en las pruebas Low-Code, como la falta de herramientas y marcos de trabajo específicos, dado que las pruebas deben adaptarse al uso de componentes predefinidos. Además, garantizar la calidad del software requiere pruebas exhaustivas debido a la complejidad de las aplicaciones. No obstante, este enfoque también ofrece oportunidades para automatizar pruebas con herramientas de generación e inteligencia artificial, acelerando los procesos y mejorando la eficiencia. Asimismo, facilita la colaboración entre equipos de desarrollo y pruebas, mejorando la calidad del software (Khorram et al., 2020).

Pin García et al. realizaron una revisión sistemática sobre el uso de la inteligencia artificial para asegurar la calidad del software. Analizaron su aplicación e impacto en diferentes etapas del ciclo de vida, incluyendo la detección de errores, la mejora de la eficiencia y la automatización de pruebas. Los resultados muestran que la inteligencia artificial ha demostrado ser una herramienta valiosa para garantizar la calidad del software (Pin García et al., 2020). Yan destaca que las plataformas de desarrollo de bajo o nulo código facilitan y automatizan la creación de aplicaciones sin necesidad de conocimientos profundos de programación. Además, permiten a las organizaciones desarrollar rápidamente aplicaciones personalizadas, acelerando el proceso y adaptándose a requisitos cambiantes del mercado. Así mismo, señala que este tipo de programación es una parte emergente de la ingeniería de software asistida por computadora; y presenta un análisis del estado actual de investigación de estas plataformas y una comparación entre las existentes (Yan, 2021). Soundberg et al., analizan cómo la inteligencia artificial sin código (*no-code AI*) puede democratizar el acceso y uso de IA, especialmente en empresas. El artículo destaca cómo el aprendizaje automático está transformando relaciones con clientes, diseño de productos y gestión de recursos humanos en grandes compañías. Sin embargo, se ha demostrado que las PYMES adoptan lentamente estas innovaciones, exponiéndose a perder competitividad (Sundberg & Holmström, 2023).

Chaudhary et al. proponen desarrollar aplicaciones de Internet de las Cosas (IoT) con plataformas de bajo código para realizar análisis en el borde de la red. Su enfoque distribuye la carga de análisis de datos entre la nube central y el borde, buscando mejorar la escalabilidad y eficiencia del sistema

(Chaudhary et al., 2022). Martínez et al. diseñaron y desarrollaron una arquitectura IoT utilizando la plataforma FIWARE. Esto les permitió evaluar la calidad de aplicaciones IoT creadas con dicha plataforma y proporcionar recomendaciones para mejorarla (Martínez et al., 2018).

Por otro lado, Sahay et al. analizan y comparan plataformas de desarrollo de bajo código, modelando sus variabilidades y similitudes, para brindar soporte a su comprensión. Para lograr esto, los autores utilizan diagramas de características, una notación habitual en análisis de dominio, con el objetivo principal de estudiar las plataformas existentes (Sahay et al., 2020). Martins et al. proporcionan una visión general de cómo desarrollar una aplicación de bajo código con OutSystems. El artículo describe en detalle el proceso, explicando pasos clave como diseño de interfaz de usuario, lógica de negocio e integración con bases de datos y servicios externos. Incluyen ejemplos e ilustraciones de cada paso. También abordan desafíos comunes como gestión de complejidad, optimización de rendimiento y seguridad, ofreciendo consejos y buenas prácticas para superarlos (Martins et al., 2020). Waszkowski presenta Aurea BPM, una plataforma de bajo código que automatiza procesos empresariales en manufactura. La plataforma desarrollar y desplegar rápidamente aplicaciones sin necesidad de código personalizado, facilitando la automatización y mejorando la eficiencia. Sin embargo, reconoce desafíos asociados como garantizar seguridad de datos e integridad de procesos, y la necesidad de capacitar a los usuarios en el uso de la plataforma (Waszkowski, 2019). ALSAADI et al. investigaron los factores que influyen en la adopción de plataformas de desarrollo de bajo código, a través de una encuesta a distintos usuarios. El objetivo fue identificar aquellos elementos que afectan su utilización. Los resultados revelaron que la experiencia previa en desarrollo, la facilidad de uso, la disponibilidad de recursos de apoyo, la satisfacción con las plataformas y la percepción de beneficios, son factores clave que impactan en la decisión de adoptar y utilizar estas herramientas. Estos hallazgos pueden guiar a desarrolladores y organizaciones interesadas, para comprender mejor qué considerar al elegirlas (ALSAADI et al., 2021).

METODOLOGÍA

Se ha realizado un mapeo sistemático de la literatura considerando las pautas que se proporcionan en trabajos como los (Budgen, D.; Turner, M.; Brereton, P.; Kitchenham, 2008; Keele University, 2007; Petersen, K., Feldt, R., Mujtaba, S., Mattsson, 2008). Un mapeo sistemático es un medio para categorizar y resumir la información existente sobre una pregunta de investigación de manera imparcial. El estudio se realizó en tres etapas: Planificación, Realización y Reporte. Las actividades relacionadas con las etapas de planificación y realización de nuestro estudio de mapeo sistemático se describen en las siguientes subsecciones y la etapa de informe se presenta en la Sección Resultados.

Etapa de Planificación

En esta etapa, se han realizado las siguientes actividades con el fin de establecer un protocolo de revisión: (1) Establecimiento de la pregunta de investigación; (2) Definición de la estrategia de búsqueda, (3) Selección de estudios primarios, (4) Evaluación de la calidad, (5) Definición de la estrategia de extracción de datos y (6) Selección de métodos de síntesis. Cada uno de ellos se explica en detalle a continuación.

Pregunta de investigación

El objetivo de este estudio es examinar el uso actual de las estrategias para abordar la falta de calidad en las aplicaciones desarrolladas mediante plataformas Low-Code desde el punto de vista de la siguiente pregunta de investigación: *¿Cuáles son las características y desafíos específicos de las plataformas de desarrollo Low-Code desde la perspectiva de los profesionales y desarrolladores de software?* Dado que nuestra pregunta de investigación es demasiado amplia, se ha descompuesto en sub-preguntas más detalladas: **RQ1.** ¿Cuáles son los principales factores que contribuyen a la falta de calidad en las aplicaciones Low-Code? **RQ2.** ¿Qué métodos y herramientas existen para evaluar la calidad de las aplicaciones Low-Code? **RQ3.** ¿Qué estrategias se han implementado para mejorar la calidad de las aplicaciones Low-Code?.

Estrategia de búsqueda

Primero se buscó estudios primarios en: IEEExplore, ACM DL, Springer Link y Science Direct. Sin embargo, los resultados contenían múltiples inconsistencias, por ejemplo, en IEEEXplore, al

agregar un OR a la consulta redujo el número de resultados. Aunque los términos de búsqueda se presentaron en sus resúmenes, ACM DL y Science Direct omitieron algunos documentos. Por lo tanto, se optó por Google Académico ya que proporciona amplia cobertura de diferentes fuentes electrónicas (Martín-Martín et al., 2021), y como recomendado por la Universidad de Keele (Keele University, 2007) y se llevó a cabo por Landman et al. (Landman et al., 2017) y Botto-Tobar et al., (Botto-Tobar et al., 2022).

Para realizar la búsqueda automática de las bibliotecas digitales seleccionadas, se utilizó la cadena de búsqueda (ver **Tabla 1**). Se llevó a cabo en julio de 2023, y el período revisado incluyó estudios publicados desde 2018 a 2023 (inclusive).

Selección de estudios primarios

Cada estudio fue evaluado para decidir su inclusión o no (considerando el título, resumen y palabras clave). Se incluyeron los estudios que cumplieron con al menos uno de los siguientes criterios de inclusión: (1) Trabajos que presentaran estrategias para abordar la falta de calidad en las aplicaciones desarrolladas mediante plataformas Low-Code. (2) Artículos que presenten ejemplos o estudios empíricos (por ejemplo, casos de estudio, experimentos) sobre estrategias para abordar la falta de calidad en las aplicaciones desarrolladas mediante plataformas Low-Code.

Tabla 1. Cadena de búsqueda aplicada.

Concepto o vértice	Términos alternativos o sinónimos	
Plataformas Low-Code	*("Low-Code" OR "low code")*	AND
Proceso de desarrollo	*("mejores prácticas" OR enfoques)*	AND
Competencias y capacitación	*(competencias OR capacidades) AND capacitación*	

Se excluyeron los estudios que cumplieron al menos uno de los siguientes criterios de exclusión:

- Artículos introductorios para números especiales, libros o talleres.
- Informes duplicados del mismo estudio en diferentes fuentes.
- Trabajos con menos de cinco páginas.
- Trabajos no escritos en español.

Síntesis. Se aplicaron métodos de síntesis tanto cuantitativos como cualitativos. La síntesis cuantitativa se basó en:

- Contar los estudios primarios que se clasifican en cada respuesta de nuestras sub-preguntas de investigación.
- Contar el número de artículos encontrados en cada fuente bibliográfica por año.

La síntesis cualitativa se basa en la inclusión de varios estudios representativos para cada criterio considerando los resultados de la evaluación de la calidad.

Etapa de Realización

La aplicación del protocolo de revisión resultó en los siguientes resultados preliminares (ver Tabla 2): Por lo tanto, se seleccionaron un total de 13 trabajos de investigación de acuerdo con los criterios de inclusión.

Tabla 1. Resultados de la etapa de realización.

Fuente	Estudios Potenciales	Estudios Seleccionados
Google Académico	73	13

RESULTADOS

Los resultados generales, que se basan en el recuento de los estudios primarios que se clasifican en cada una de las respuestas a nuestras sub-preguntas de investigación, se presentan en la Tabla 3. Los documentos incluidos que se citan en esta sección como [SXX] se refieren al Apéndice A.

Tabla 2. Resultados del mapeo sistemático.

Sub-preguntas de investigación	Resultados	
	# Estudios	Porcentaje (%)
¿Cuáles son los principales factores que contribuyen a la falta de calidad en las aplicaciones Low-Code?	3	23%
¿Qué métodos y herramientas existen para evaluar la calidad de las aplicaciones Low-Code?	3	23%
¿Qué estrategias pueden ser implementadas para mejorar la calidad de las aplicaciones Low-Code?	7	54%

RQ1. ¿Cuáles son los principales factores que contribuyen a la falta de calidad en las aplicaciones Low-Code?

Santos et al., destacó la falta de experiencia en desarrollo de software, las limitaciones de las plataformas Low-Code, falta de pruebas exhaustivas, y además la dependencia de componentes predefinidos (Santos, 2023). Vaello Peláez menciona que el análisis estático del código generado por la plataforma Low-Code implica revisar el código fuente para identificar posibles problemas de calidad, como código redundante, falta de modularidad o violaciones de buenas prácticas de programación (Vaello Peláez, 2023).

Por otro lado, las limitaciones del análisis dinámico puro en la generación de modelos para aplicaciones de código bajo pueden resultar en modelos incompletos, lo que puede afectar la calidad del código generado. Y la falta de conocimiento de las mejores prácticas de desarrollo de aplicaciones, la respuesta insuficiente de la gerencia y la falta de acceso a datos de salud de calidad también se identifican como factores que contribuyen a la falta de calidad en varios dominios; así mismo, enfatiza la importancia de la capacitación y la adaptación continua para los desarrolladores tradicionales en este nuevo entorno de desarrollo de aplicaciones (Vera et al., 2022). Además, la falta de experiencia en desarrollo de software, las limitaciones de las plataformas Low-Code, la falta de pruebas adecuadas, la dependencia de componentes predefinidos y la falta de control sobre el código subyacente son algunos de los factores que pueden contribuir a la falta de calidad en las aplicaciones Low-Code (Arévalo, 2022).

RQ2. ¿Qué métodos y herramientas existen para evaluar la calidad de las aplicaciones Low-Code?

Santos et al. destacan las ventajas de las plataformas de bajo y sin código para la ingeniería de software, ya que aceleran el desarrollo de aplicaciones y lo vuelven más accesible a diversos perfiles de desarrolladores. Además, mencionan que para la evaluación de la calidad de estas aplicaciones existen estrategias como pruebas de funcionalidad, rendimiento y seguridad, así como revisiones de código son estrategias usadas para la evaluación de la calidad en aplicaciones (Santos, 2023).

Arévalo et al. abordan la falta de calidad de aplicaciones Low-Code con Automasoft, una solución de inteligencia artificial y aprendizaje automático que permite a usuarios sin programación crear software de calidad. La plataforma Automasoft busca mejorar la calidad al generar automáticamente código de calidad a partir de especificaciones de alto nivel proporcionadas por los usuarios (Arévalo, 2022).

Además, Vera et al., analizó el impacto de las plataformas de desarrollo de bajo código y sin código en el futuro de los desarrolladores de software, y destacó los beneficios y desafíos asociados con estas plataformas y explora cómo están cambiando la forma en que se desarrollan las aplicaciones de software mediante la combinación de métodos y herramientas que permiten evaluar y mejorar la calidad de las aplicaciones desarrolladas con plataformas Low-Code (Vera et al., 2022).

RQ3. ¿Qué estrategias pueden ser implementadas para mejorar la calidad de las aplicaciones Low-Code?

Vaello Peláez et al., destaco realizar pruebas exhaustivas, implementar buenas prácticas de desarrollo, fomentar la colaboración y el intercambio de conocimientos (Vaello Peláez, 2023). Otras estrategias que se pueden implementar: Cardona Puig et al. analizan cómo la combinación de robótica e inteligencia artificial puede transformar procesos empresariales, mejorando eficiencia y productividad. En ese contexto, destacan beneficios y desafíos de esta tecnología, proporcionando ejemplos de aplicación en distintos sectores (Cardona Puig, 2023). Pedroti et al., aborda el problema de la falta de calidad de las aplicaciones Low-Code al proponer el desarrollo de una aplicación utilizando la tecnología Manutenção 4.0 con el objetivo de reducir el tiempo de mantenimiento (Pedroti, 2023). Marceaux et al., aborda el uso del video 360 interactiva en la formación por simulación clínica en enfermería, destacando sus beneficios potenciales, limitaciones y desafíos (Marceaux et al., 2023). Beltrán Marco et al., indicaron que realizar el diseño a mano alzada sobre una pizarra o hoja de papel y que, mediante el sistema desarrollado en este TFG/TFM, automáticamente te genere el código correspondiente. Se utilizarán técnicas de Deep Learning y de visión tradicional para lograr este objetivo (Beltrán Marco, 2023).

Así mismo, Casas et al., aborda el problema de la selección de una plataforma de integración de IPaaS y propone una metodología basada en el proceso analítico jerárquico para comparar y evaluar

diferentes opciones (Casas et al., 2022). Cavallé Guzman et al., proporciona recomendaciones y pautas para implementar con éxito un plan de marketing digital basado en la inteligencia artificial en la empresa Shimoku. (Cavallé Guzman, 2022); y Díaz et al., destaca el potencial de las plataformas de videojuegos como una herramienta para la educación inicial en programación. (Díaz et al., 2022)

Además, establecer estándares de calidad, esto es, definir estándares de calidad claros y medibles para las aplicaciones desarrolladas con plataformas Low-Code puede ayudar a garantizar que se cumplan los requisitos de calidad y que se sigan las mejores prácticas de desarrollo (Arévalo, 2022).

CONCLUSIONES

Este estudio presenta un mapeo sistemático de la literatura sobre cómo los investigadores y profesionales enfrentan la falta de calidad en aplicaciones Low-Code. A través de este método se analizó el estado del arte, identificando temas abiertos mediante el análisis de evidencias en 13 estudios primarios.

A través de las respuestas encontradas en una pregunta de investigación y tres sub-preguntas de investigación, fue posible identificar evidencias que apuntan al establecimiento de normas para abordar la falta de calidad en las aplicaciones Low-Code. Los principales hallazgos de nuestro estudio son:

- La falta de experiencia en desarrollo de software, limitaciones de las plataformas Low-Code, falta de pruebas exhaustivas, y dependencia de componentes predefinidos son algunos de los factores que inciden en la calidad de las aplicaciones Low-Code.
- Algunas normas, estándares y directrices de desarrollo son para garantizar la consistencia y la calidad en todas las aplicaciones desarrolladas con plataformas Low-Code.
- Los resultados logrados por este estudio de mapeo ayudarán a desarrollar nuevos frentes de investigación en el desarrollo de aplicaciones Low-Code.

APÉNDICE A

Listado de trabajos seleccionados.

S01. Santos, P. B. dos. (2023). *Engenharia sem código: as vantagens da linguagem low-code e no-code*.

S02. Cardona Puig, E. (2023). *Automatización robótica de procesos con inteligencia artificial*. Universitat Politècnica de Catalunya.

S03. Vilca Apaza, D. Y., & Astudillo Maguiña, C. A. (2023). *Sistema web para la difusión de información en el área de Registros Civiles de la RENIEC, 2023*.

S04. Pedroti, R. P. (2023). *Manutenção 4.0: Desenvolvimento de aplicativo com a finalidade de reduzir o tempo da manutenção*.

S05. Marceaux, J., Servotte, J.-C., & Pilote, B. (2023). Utilisation de la vidéo 360 interactive pour la formation par simulation clinique en santé: Potentiels, limites et enjeux dans le cadre de la formation en soins infirmiers. *Médiations et Médiatisations*, *15*, 175–184.

S06. Vaello Peláez, J. (2023). *Análisis comparativo de costes de migraciones de sistemas de información tradicionales a entornos cloud*.

S07. Beltrán Marco, R. (2023). *Diseño rápido de páginas web mediante dibujo a mano alzada*.

S08. Vera, C. G. M., Vicente, M. V. O., Vera, I. L. A., Alexander, M. V. A., & Vera, H. F. B. (2022). Low/No-code development platforms and the future of software developers. *Minerva*, *1*(Special), 21–33.

S09. Arévalo, J. P. (2022). Automasoft: desarrollo de software para no desarrolladores. *CAOBA Express*.

S10. Herráiz Montalvo, C. I. (2022). *Análisis de pulsera inteligente para la detección de estados afectivos mediante aprendizaje no supervisado sobre series temporales*.

S11. Casas, S., Cruz, D., Vidal, G., & Constanzo, M. (2022). El portal de integración de IPaaS: propuesta de comparación aplicando el proceso analítico jerárquico. *Simposio Argentino de Ingeniería de Software (ASSE 2022)-JAIIO 51 (Modalidad Virtual y Presencial (UIA), Octubre 2022)*.

S12. Cavallé Guzman, S. (2022). *Plan de marketing digital de Shimoku: hacia la era de la inteligencia artificial*.

S13. Díaz, J., Bustamante, A., Ramírez V, G. M., & Hochstetter, J. (2022). Video Games Platforms: A Gateway to New Trends for Initial Programming Education. *Proceedings - International*

Conference of the Chilean Computer Science Society, SCCC, *2022-Novem*, 1–6. https://doi.org/10.1109/SCCC57464.2022.10000326

REFERENCIAS

Al Alamin, M. A., Malakar, S., Uddin, G., Afroz, S., Haider, T. Bin, & Iqbal, A. (2021). An Empirical Study of Developer Discussions on Low-Code Software Development Challenges. *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, 46–57. https://doi.org/10.1109/MSR52588.2021.00018

ALSAADI, H. A., RADAIN, D. T., ALZAHRANI, M. M., ALSHAMMARI, W. F., ALAHMADI, D., & FAKIEH, B. (2021). Factors that affect the utilization of low-code development platforms: survey study. *Romanian Journal of Information Technology \& Automatic Control/Revista Român{\u{a}} de Informatic{\u{a}} Şi Automatic{\u{a}}*, *31*(3).

Arévalo, J. P. (2022). Automasoft: desarrollo de software para no desarrolladores. *CAOBA Express*.

Beltrán Marco, R. (2023). *Diseño rápido de páginas web mediante dibujo a mano alzada*.

Botto-Tobar, M., van den Brand, M. G. J., & Serebrenik, A. (2022). Cross-Language Plagiarism Detection: Methods, Tools, and Challenges: A Systematic Review. *International Journal on Advanced Science, Engineering and Information Technology*, *12*(2), 589–599.

Budgen, D.; Turner, M.; Brereton, P.; Kitchenham, B. (2008). Using Mapping Studies in Software Engineering. In *Proceedings of PPIG 2008* (pp. 195–204). Lancaster University.

Cardona Puig, E. (2023). *Automatización robótica de procesos con inteligencia artificial*. Universitat Politècnica de Catalunya.

Casas, S., Cruz, D., Vidal, G., & Constanzo, M. (2022). El portal de integración de IPaaS: propuesta de comparación aplicando el proceso analítico jerárquico. *Simposio Argentino de Ingeniería de Software (ASSE 2022)-JAIIO 51 (Modalidad Virtual y Presencial (UIA), Octubre 2022)*.

Cavallé Guzman, S. (2022). *Plan de marketing digital de Shimoku: hacia la era de la inteligencia artificial*.

Chaudhary, H. A. A., Guevara, I., John, J., Singh, A., Margaria, T., & Pesch, D. (2022). Low-code internet of things application development for edge analytics. *IFIP International Internet of Things Conference*, 293–312.

Díaz, J., Bustamante, A., Ramírez V, G. M., & Hochstetter, J. (2022). Video Games Platforms: A Gateway to New Trends for Initial Programming Education. *Proceedings - International Conference of the Chilean Computer Science Society, SCCC*, *2022-Novem*, 1–6. https://doi.org/10.1109/SCCC57464.2022.10000326

Galhardo, P., & Silva, A. R. da. (2022). Combining Rigorous Requirements Specifications with Low-Code Platforms to Rapid Development Software Business Applications. *Applied Sciences*, *12*(19). https://doi.org/10.3390/app12199556

Herráiz Montalvo, C. I. (2022). *Análisis de pulsera inteligente para la detección de estados afectivos mediante aprendizaje no supervisado sobre series temporales*.

Heuer, M., Kurtz, C., & Böhmann, T. (2022). *Towards a governance of low-code development platforms using the example of microsoft powerplatform in a multinational company*.

Ihirwe, F., Di Ruscio, D., Gianfranceschi, S., & Pierantonio, A. (2022). Assessing the Quality of Low-Code and Model-Driven Engineering Platforms for Engineering IoT Systems. *2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS)*, 583–594. https://doi.org/10.1109/QRS57517.2022.00065

Keele University. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, Ver. 2.3 EBSE Technical Report. EBSE* (Issue EBSE 2007-001).

Khorram, F., Mottu, J.-M., & Sunyé, G. (2020). Challenges & Opportunities in Low-Code Testing. *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. https://doi.org/10.1145/3417990.3420204

Landman, D., Serebrenik, A., & Vinju, J. J. (2017). Challenges for static analysis of java reflection-literature review and empirical study. *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, 507–518.

Marceaux, J., Servotte, J.-C., & Pilote, B. (2023). Utilisation de la vidéo 360 interactive pour la formation par simulation clinique en santé: Potentiels, limites et enjeux dans le cadre de la formation en soins infirmiers. *Médiations et Médiatisations*, *15*, 175–184.

Martín-Martín, A., Thelwall, M., Orduna-Malea, E., & López-Cózar, E. D. (2021). Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: a multidisciplinary comparison of coverage via citations. In *Scientometrics* (Vol. 126, Issue 1,

pp. 907–908). Springer. https://doi.org/10.1007/s11192-020-03792-z

Martínez, A., Onofre, H., Estrada, H., Torres, D., & Maquinay, O. (2018). Diseño y desarrollo de una arquitectura IoT en contexto con la plataforma FIWARE. *Research in Computing Science*, *147*(8), 95–106. https://doi.org/10.13053/rcs-147-8-7

Martins, R., Caldeira, F., Sa, F., Abbasi, M., & Martins, P. (2020). An overview on how to develop a low-code application using OutSystems. *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, 395–401.

Pedroti, R. P. (2023). *Manutenção 4.0: Desenvolvimento de aplicativo com a finalidade de reduzir o tempo da manutenção*.

Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M. (2008). Systematic mapping studies in software engineering. *12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 68–77.

Pin García, L. J., Toala Zambrano, M. M., & Álava Cruzatty, J. E. (2020). LA INTELIGENCIA ARTIFICIAL EN LA CALIDAD DEL SOFTWARE: UNA REVISIÓN SISTEMÁTICA DE LA LITERATURA. *UNESUM-Ciencias. Revista Científica Multidisciplinaria. ISSN 2602-8166*, *4*(1), 75–86. https://doi.org/10.47230/unesum-ciencias.v4.n1.2020.208

Sahay, A., Indamutsa, A., Di Ruscio, D., & Pierantonio, A. (2020). Supporting the understanding and comparison of low-code development platforms. *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 171–178. https://doi.org/10.1109/SEAA51224.2020.00036

Sanchis, R., García-Perales, Ó., Fraile, F., & Poler, R. (2020). Low-Code as Enabler of Digital Transformation in Manufacturing Industry. *Applied Sciences*, *10*(1). https://doi.org/10.3390/app10010012

Santos, P. B. dos. (2023). *Engenharia sem código: as vantagens da linguagem low-code e no-code*.

Sundberg, L., & Holmström, J. (2023). Democratizing artificial intelligence: How no-code AI can leverage machine learning operations. *Business Horizons*.

Talesra, K., & Nagaraja, G. S. (2021). Low-code platform for application development. *International Journal of Applied Engineering Research*, *16*(5), 346–351.

Vaello Peláez, J. (2023). *Análisis comparativo de costes de migraciones de sistemas de información tradicionales a entornos cloud*.

Vera, C. G. M., Vicente, M. V. O., Vera, I. L. A., Alexander, M. V. A., & Vera, H. F. B. (2022). Low/No-code development platforms and the future of software developers. *Minerva*, *1*(Special), 21–33.

Vilca Apaza, D. Y., & Astudillo Maguiña, C. A. (2023). *Sistema web para la difusión de información en el área de Registros Civiles de la RENIEC, 2023*.

Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*, *52*(10), 376–381.

Yan, Z. (2021). The impacts of low/no-code development on digital transformation and software development. *ArXiv Preprint ArXiv:2112.14073*.