

**PONTIFICIA UNIVERSIDAD
CATÓLICA DEL PERÚ**

Escuela de Posgrado



Mapeo sistemático sobre la implementación de software
usando plataformas de desarrollo Low-Code
Tesis para optar el grado académico de Maestro en Informática con
mención en Ingeniería de Software que presenta:

Luis Alberto Huasacca Quispe

Asesor:

Mag. Ing. Dennis Stephen Cohn Muroy

Lima, 2024


Informe de Similitud

Yo, *Dennis Stephen Cojn Muroy*, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada “Mapeo sistemático sobre la implementación de software usando plataformas de desarrollo Low-Code” de el autor **Luis Alberto Huasacca Quispe**, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 17%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 24/06/2024.
- He revisado con detalle dicho reporte y la tesis, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha:

San Miguel, 24 de Junio de 2024.

Apellidos y nombres del asesor / de la asesora: <u>COHN MUROY, Dennis Stephen</u>	
DNI: 43513429	Firma 
ORCID: 0000-0003-4820-0178	



Dedicado a mi abuela María Jesús.

Agradecimientos

A mi asesor, Dennis Cohn, por sus valiosos y acertados consejos que permitieron afinar la dirección del presente proyecto.



Autores señalan que existen muy pocas investigaciones que realizan mapeos y revisiones sistemáticas, y muchas de ellas no siguen un método de investigación riguroso. Prinz et al. [1] realizó una revisión de literatura, recabó información de 32 artículos primarios publicados entre 2017 y 2020, clasifica la visión actual de las Plataformas de desarrollo *Low-Code* (bajo código). Rokis et al. [2] realizó una revisión de literatura para identificar los desafíos que tienen las plataformas de bajo código y/o sin código, se organizaron en 7 fases ágiles del desarrollo de software análisis de requisitos, planeamiento, diseño, desarrollo, pruebas, despliegue y mantenimiento, analizó 23 artículos primarios publicados entre 2019 – 2021. Pinho et al. [3] presenta una revisión de literatura donde revisa plataformas *Low-Code*, plataformas *No-Code* y la usabilidad de estas, evaluó 38 artículos primarios. El objetivo de esta investigación es realizar un estudio de mapeo sistemático sobre distintas investigaciones respecto a la implementación de software usando plataformas de desarrollo *Low-Code*. Se realizó un mapeo sistemático de la literatura sobre las bases de datos: Scopus, IEEE, Web of Science y ACM Digital, se evaluaron 47 artículos primarios. Los resultados muestran a las Plataformas de desarrollo *Low-Code* identificadas, las características, las áreas de la organización que utilizan estas plataformas que son en su mayoría el Área de Compras y el Área de Ventas. Así mismo se han identificado las ventajas y desventajas del uso de las LCDP como por cada una de ellas. También se ha identificado que se desarrollan aplicaciones web y móvil como principales tipos de aplicación.

The authors point out that there are very few investigations that carried out systematic mapping and reviews, and many of them do not follow a rigorous research method. Prinz et al. [1] conducted a literature review, collected information from 32 primary articles published between 2017 and 2020, classifies the current vision of Low-Code Development Platforms, Rokis et al. [2] carried out a literature review to identify the challenges that LCDP/NCDP (Low Code Development Platform / No Code Development Platform) has, it was organized into 7 agile phases of software development, requirements analysis, planning, design, development, testing, implementation and maintenance, analyzed 23 primary articles published between 2019 – 2021. Pinho et al. [3] presents a literature review where it reviews Low-Code platforms, No-Code platforms and their usability, evaluating 38 primary articles. The objective of this research is to carry out a systematic mapping study on different research regarding software implementation using Low-Code development platforms. A systematic mapping of the literature was carried out on the databases: Scopus, IEEE, Web of Science and ACM Digital, 47 primary articles were evaluated. The results show the Low-Code development platforms identified, the characteristics, the areas of the organization that use these platforms, which are mostly the Purchasing Area and the Sales Area. Likewise, the advantages and disadvantages of using LCDPs have been identified for each one of them. It has also been identified that web and mobile applications are developed as the main types of applications.

Índice general

1. Introducción	9
1.1. Problema de la investigación	9
1.2. Justificación	10
1.3. Objetivo de la investigación	10
1.4. Marco Conceptual	11
1.5. Antecedentes	12
1.6. Organización de la Tesis	13
2. Planificación	15
2.1. Metodología	15
2.2. Preguntas bibliométricas	15
2.3. Preguntas de investigación	16
2.4. Estrategia de búsqueda	16
2.5. Cadenas de búsqueda	17
2.6. Criterios de selección de estudios	19
2.7. Proceso de extracción y clasificación	19
3. Conducción	21
3.1. Selección de estudios	21
3.2. Evaluación de la calidad	23
3.3. Extracción de datos	26
4. Análisis de resultados	29
4.1. Preguntas bibliométricas	29
4.2. Preguntas de investigación	33
5. Amenazas a la Validez	55
5.1. Validez del Constructo	55
5.2. Validez Interna	55
5.3. Validez Externa	55
5.4. Validez de las Conclusiones	55
6. Conclusiones y trabajos futuros	56
6.1. Conclusiones	56
6.2. Trabajos futuros	57
Bibliografía	58

Índice de tablas

2.1. Criterios de la técnica PICO	17
2.2. Términos de búsqueda	17
2.3. Cadenas de búsqueda para ACM	17
2.4. Cadenas de búsqueda para Web of Science	18
2.5. Cadenas de búsqueda para IEEE	18
2.6. Cadenas de búsqueda para Scopus	18
2.7. Procedimientos y Criterios de Inclusión y Exclusión	19
3.1. Resultados del proceso de extracción y clasificación	21
3.2. Resultado de publicaciones seleccionadas	23
3.3. Listado de preguntas para aseguramiento de la calidad	24
3.4. Resultados del proceso de extracción y clasificación	24
3.5. Resultado de la evaluación de los criterios de calidad	26
3.6. Estudios Primarios	28
4.1. Investigaciones clasificadas por país de publicación	29
4.2. Investigaciones clasificadas por año de publicación	30
4.3. Investigaciones clasificadas por tipo de publicación	31
4.4. Investigaciones clasificadas por medio de publicación	33
4.5. Plataformas de desarrollo Low-Code	35
4.6. Características de las plataformas de desarrollo Low-Code	37
4.7. Ventajas sobre las plataformas Low-Code	38
4.8. Desventajas sobre las plataformas Low-Code	39
4.9. Áreas de aplicación	40
4.10. Características por plataformas	47
4.11. Ventajas por plataformas	49
4.12. Desventajas por plataformas	51
4.13. Tipos de aplicaciones por plataforma	52
4.14. Investigaciones clasificadas por industrias	53
4.15. Tipos de aplicaciones por industria	54

Índice de figuras

2.1. Diagrama de flujo del proceso de mapeo sistemático Petersen et al.	15
4.1. Investigaciones clasificadas por año de publicación	30
4.2. Investigaciones clasificadas por tipo de publicación	31
4.3. Principales tipos de aplicaciones por plataforma	52
4.4. Principales industrias de aplicación	53



CAPÍTULO 1

Introducción

El objetivo de este capítulo es ofrecer una introducción a la problemática que se busca afrontar a través de la presente investigación. De igual manera, se expone un conjunto de conceptos que facilitan la comprensión de los temas expuestos.

El presente capítulo está conformado por 5 secciones. La Sección 1.1 expone la problemática que la investigación ha considerado, la Sección 1.2 presenta la justificación, la Sección 1.3 presenta los objetivos del estudio, la Sección 1.4 presenta el marco conceptual, la Sección 1.5 presenta los estudios que se realizaron con respecto a mapeos sistemáticos o revisiones sistemáticas de la literatura sobre plataformas de desarrollo *Low-Code*, la Sección 1.6 presenta la estructura del documento.

1.1. Problema de la investigación

Los equipos de desarrollo trabajan arduamente y bajo presión para poder satisfacer las crecientes expectativas y necesidades de los clientes. Lamentablemente el desarrollo de software tradicional está perdiendo presencia cuando se requiere responder a la exigencia actual de velocidad [4]. Por otro lado el *Low-Code* o bajo código es un enfoque de desarrollo de software que dispone a los desarrolladores a crear aplicaciones con poca codificación, a comparación de la programación tradicional donde los desarrolladores escriben manualmente líneas de código para producir un producto de software. En el artículo de Gartner [5] pronostica que el uso de tecnología *Low-Code* crecerá un 20 % para el 2023; además, se menciona que las organizaciones están recurriendo a la aplicación de tecnologías en bajo código para cumplir la creciente demanda de entrega de software con velocidad y flujos de trabajo de automatización altamente personalizados. Con esto, las organizaciones se permitirían alcanzar una competencia digital y la velocidad de entrega de aplicaciones en entornos ágiles y modernos [5].

El desarrollo *Low-Code* está soportado por las plataformas de aplicación *Low-Code* (LCAPs) o plataformas de desarrollo *Low-Code* (LCDPs) que permiten desarrollar software según este enfoque. Es por ello que en el ámbito académico se han incrementado las publicaciones asociadas al desarrollo en bajo código; sin embargo, la cantidad de estudios asociados a revisiones y mapeos sistemáticos de este tema son pocos [1]. También Prinz et al. [1] nos comenta que de las publicaciones evaluadas en su investigación sobre plataformas de desarrollo *Low-Code* entre 2017 y 2022, solamente dos estudios realizaron una revisión

sistemática pero no siguen un método de investigación riguroso basado en las base de datos mas relevantes, considerando así su investigación como la primera revisión de literatura científica estructurada. Por otro lado Microsoft [6] afirma que existen limitaciones; algunas plataformas tienen plantillas rígidas que pueden provocar sentimientos de frustración. Las plataformas en bajo código o sin código algunas veces no tienen tanta flexibilidad y puede ser posible que no satisfaga a un conjunto muy específico de necesidades y expectativas. En cuanto a programadores sin experiencia, aquellos pueden desarrollar en estas plataformas sin mucha complejidad, pero podrían existir inconvenientes en cuanto a desarrollo profesional y calidad de software; es, además, posible que se encuentren con obstáculos de desarrollo debido a la falta de control y flexibilidad. La seguridad también puede ser una preocupación debido a que con las plataformas sin código no se tiene tanto control sobre la gestión de seguridad de las aplicaciones, y estas podrían estar sujetas a vulnerabilidades de la plataforma [6].

1.2. Justificación

OutSystems [4] menciona que *Low-Code* es un dominio emergente y está tomando cabida con la creciente relevancia de su uso en las organizaciones, así como la creciente cantidad de publicaciones, es por ello que se requiere abordar un poco más a profundidad y completar las falencias que se encontraron en otras investigaciones documentales. A comparación del estudio de Prinz et al. [1] el presente trabajo usa una cadena de búsqueda más específica: no categoriza los artículos en cuatro cuadrantes, es un trabajo mas reciente y sigue la guía de Petersen et al. [7] actualizada al 2015. El mapeo sistemático está enfocado en preguntas asociadas al desarrollo de software con plataformas de desarrollo *Low-Code*. El término *Low-Code* se usó en el 2014 por las investigaciones Forrester, según OutSystems [4] las publicaciones entorno a este dominio comenzaron en el 2018 [8] y aún se considera que las publicaciones asociadas al desarrollo *Low-Code* están en una etapa de maduración [1]. Para contribuir a la comunidad científica se considera adecuado seguir ahondando en revisiones y mapeos sistemáticos en relación al desarrollo *Low-Code* para así proveer un mayor abanico de posibilidades documentales a los investigadores a fin de mejorar el entendimiento en este dominio. Cuanto más se investigue y publique sobre este tema en la comunidad científica, más organizado será el estudio de las LCDPs.

1.3. Objetivo de la investigación

El objetivo de la investigación es realizar un estudio de mapeo sistemático sobre distintas investigaciones respecto a la implementación de software usando plataformas de desarrollo *Low-Code*.

Los objetivos específicos para esta investigación son:

- **OE-1:** Identificar las plataformas de desarrollo *Low-Code* que se vienen utilizando para el desarrollo de software.
- **OE-2:** Describir las características de las plataformas de desarrollo *Low-Code*.

- **OE-3:** Describir las ventajas y desventajas del uso de plataformas *Low-Code*.
- **OE-4:** Identificar las áreas de la organización que usaron plataformas de desarrollo *Low-Code*.
- **OE-5:** Describir las características por cada plataforma de desarrollo *Low-Code* identificada.
- **OE-6:** Describir las ventajas y desventajas que se tienen por cada plataforma de desarrollo *Low-Code* identificada.
- **OE-7:** Identificar los tipos de aplicaciones por cada una de las plataformas de desarrollo *Low-Code*.
- **OE-8:** Identificar las industrias donde se utilizan plataformas *Low-Code*.
- **OE-9:** Identificar los tipos de aplicaciones por industrias donde se utilizan plataformas de desarrollo *Low-Code*.
- **OE-10:** Describir las tendencias de publicaciones sobre plataformas *Low-Code* en el desarrollo de software.

1.4. Marco Conceptual

De acuerdo a OutSystems [4] *Low-Code* es un enfoque de desarrollo de software que permite a los desarrolladores crear aplicaciones con mínimo esfuerzo de codificación. Los desarrolladores usan un lenguaje visual basado en modelos con una interfaz gráfica de arrastrar y soltar. *Low-Code* permite a los desarrolladores crear aplicaciones completas con interfaces de usuario modernas, integraciones con agentes externos, datos y lógica muchísimo más rápido que el desarrollo tradicional. El desarrollo en *Low-Code* está soportado por plataformas *Low-Code* o de bajo código que proporcionan un lenguaje visual, interfaz de arrastrar y soltar junto con componentes y plantillas preconstruidas; de esta manera permite al desarrollador construir aplicaciones con mayor rapidez. No existen dos plataformas que sean exactamente iguales debido a que las características y capacidades ofrecidas varían de una plataforma a otra.

Según Wazkowski [9], Las plataformas de desarrollo en bajo código (LCDP) constan de un conjunto de herramientas para programadores y no programadores, permiten una rápida generación de entrega de aplicaciones comerciales donde se requiera un esfuerzo mínimo de desarrollo, instalación, configuración del entorno, capacitación e implementación. Es considerado el uso de plataformas de desarrollo *Low-Code* como el método más rápido y probablemente también menos costoso de desarrollo de software.

Las plataformas de desarrollo *Low-Code* están basadas en la interfaz gráfica de usuario donde el desarrollador puede diseñar las aplicaciones de manera mas intuitiva y gráfica en lugar de técnicas de programación codificadas.

Las características de estas plataformas mayormente se enfocan en lo siguiente:

- Bases de datos.

- Procesos de negocio.
- Interfaz de usuario (aplicaciones basadas en Web).

Gomes [10] menciona que los términos *Low-Code* o *No-Code* se utilizan cuando se hace referencia a plataformas, aplicaciones o productos de software con una abstracción de programación de alto nivel, destinados al desarrollo a nivel de usuario final o llamados desarrolladores ciudadanos. El desarrollo de usuario final es un término que Gomes utiliza para referirse a las herramientas que permiten a los usuarios finales (también conocidos como desarrolladores ciudadanos) programar y crear aplicaciones a través de la ingeniería dirigida por modelos o *Model-Driven Engineering* (MDE). Es por ello que en su investigación descriptiva el autor se refiere a las LCDP como un lenguaje de dominio específico que recurre a los principios de MDE y tiene como objetivo elevar aún más el nivel de abstracción a un nivel donde el usuario desarrollador, aún con poco o ningún conocimiento de habilidades de programación, pueda resolver problemas nuevos, desafíos o requerimientos con esta solución tecnológica.

1.5. Antecedentes

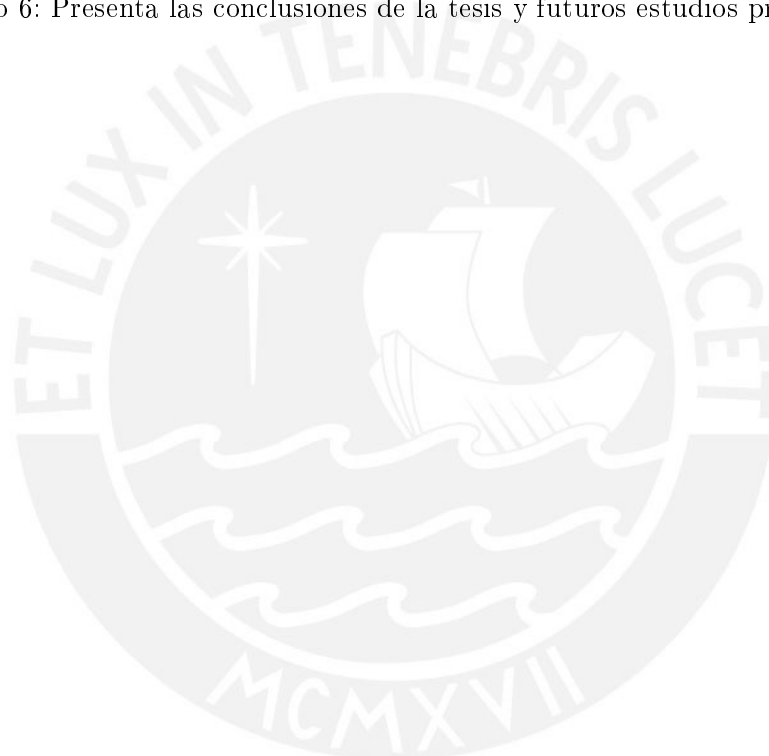
Prinz et al. [1] presentan una revisión de literatura denominada “Low-code development platforms - A literature review” la cual fue expuesta en la Vigésimo Séptima Conferencia de las Américas sobre Sistemas de Información, Montreal, 2021. Esta investigación sólo examina publicaciones en el área de Sistemas de información (IS por sus siglas en inglés) y las ciencias de la computación (CS) y se revisan 32 investigaciones publicadas entre el 2017 y 2020. Los autores consideran su investigación como la primera revisión científica y estructurada de la literatura sobre LCDP. El objetivo de su estudio fue revisar y clasificar la visión actual sobre LCDP usando el modelo STS (modelo de sistema sociotécnico), se califican las publicaciones utilizando la escala de Likert en cuatro categorías: Estructura, Personas, Tecnología y Tareas. De otro lado Rokis et al. [2] presenta una revisión de literatura denominada “Challenges of Low-Code/No-Code Software Development: A Literature Review” publicada en el Instituto de Sistemas Informáticos Aplicados de la Universidad Técnica de Riga en Letonia, el objetivo de esta publicación es revisar la literatura existente para identificar los desafíos que se tienen en el desarrollo de software en bajo código/cero código y además utilizar la literatura existente para encontrar las sugerencias, propuestas o soluciones de mitigación con respecto a estos desafíos. La búsqueda inicial se realizó el último trimestre del 2021 en diferentes bases de datos; en total se revisaron 23 artículos publicados entre 2019 y 2021 como estudios primarios relevantes para esta investigación. Los resultados obtenidos se organizaron en siete fases ágiles del desarrollo de software: análisis de requisitos, planeamiento, diseño, desarrollo, pruebas, despliegue y mantenimiento. Luo et. al [11] presenta que aunque el desarrollo de software en bajo código aportan muchos beneficios, también existen muchos desafíos y oportunidades de mejora. Algunos de los beneficios son: desarrollo rápido, fácil estudio y uso, lista para usar, amigable, etc. Los desafíos más importantes son: alta curva de aprendizaje, precios altos, poca customización, menos poderosos que la programación tradicional, etc.

Bucaioni et al. [8] presenta una revisión sistemática denominada “Modelling in low-code development: a multi-vocal systematic review”, del 2022 donde se informa sobre la planificación, ejecución y resultados de una revisión sistemática multivocal sobre el desarrollo en bajo código enfocada a la relación de la ingeniería impulsada por modelos. Inicialmente se obtuvieron 720 publicaciones y 199 fuentes de literatura gris para finalmente revisar 58 artículos primarios publicados entre 2016 y 2021. Esta investigación informa sobre la planificación, ejecución y análisis de los resultados de la revisión sistemática multivocal que proporciona una descripción general y completa del desarrollo *Low-Code* y su relación con el modelado y el MDE (*Model Driven Engineering*). La investigación concluye que el término *Low-Code* se acuñó en 2014, las primeras publicaciones en este campo aparecieron en 2018 y aún las investigaciones en desarrollo *Low-Code* se encuentran en una etapa de maduración. 27 de las 58 investigaciones tienen como tecnología central al MDE. El estudio sigue las pautas de Kitchenham y Brereton sobre revisiones sistemáticas y Garousi para revisiones de literatura multivocales en ingeniería de software. La investigación utiliza publicaciones en base de datos y literatura gris. Los autores consideran a su investigación como el primer trabajo de mapeo sistemático multivocal que proporciona un análisis de las características centrales del desarrollo *Low-Code*, en particular sobre los paradigmas funcionales, dominios de aplicación, herramientas y beneficios esperados, no necesariamente implementados en una plataforma concreta, ofrecen un catálogo de plataformas y herramientas utilizadas pero no pretenden comparar estas plataformas o herramientas. Así mismo los autores Pinho et al. [3] en su investigación “What about the usability in low-code platforms? A systematic literature Review” afirman que el desarrollo en bajo código es un concepto que está creciendo en la academia como en la industria del software y es discutido junto con otros conceptos como la ingeniería basada en modelos (MDE) y los lenguajes específicos de dominio (DSL). En las herramientas de desarrollo en bajo código es importante la usabilidad debido a que los usuarios de estas herramientas a menudo carecen de experiencia en programación. Este artículo examina la literatura actual sobre desarrollo en bajo código y sin código para investigar sobre este dominio y su relación con la usabilidad, principalmente su calidad. La investigación realizó una revisión de literatura, el protocolo de búsqueda devolvió 207 artículos de revisión de pares en cinco bases de datos y se complementaron con el proceso de bola de nieve, se filtraron y se obtuvo finalmente 38 artículos primarios relevantes que fueron analizados, sintetizados e informados. Los autores concluyen que no encontraron una definición formal de bajo código o *Low-Code* aunque se han especificado términos que podrían ayudar a su entendimiento, descubrieron que los usuarios son más conscientes de la usabilidad en herramientas *Low-Code*, los investigadores en sus artículos están considerando inconscientemente los factores de usabilidad y que el campo de desarrollo en *Low-Code* crecería si aumentara la investigación de usabilidad. Finalmente el documento sugiere una definición para el desarrollo en bajo código.

1.6. Organización de la Tesis

La organización del trabajo de tesis es la siguiente:

- Capítulo 2: Presenta la metodología del Mapeo Sistemático siguiendo el protocolo propuesto por Petersen et al. [7]. Ese capítulo refleja la metodología planeada sobre la problemática.
- Capítulo 3: Presenta la conducción del Mapeo Sistemático definido durante el planeamiento.
- Capítulo 4: Siguiendo el formulario de extracción, este capítulo presenta la recopilación de la información necesaria para responder las preguntas bibliométricas y preguntas de investigación.
- Capítulo 5: Este capítulo presenta aquellas amenazas que puedan impactar en los resultados obtenidos.
- Capítulo 6: Presenta las conclusiones de la tesis y futuros estudios propuestos.



CAPÍTULO 2

Planificación

2.1. Metodología

Petersen et al. [7] define al estudio de mapeo sistemático como un método que proporciona una estructura del tipo de reporte investigativo que usualmente ofrece un resumen visual de resultados, esto debido a que el mapa provee una visión general. Este método se define para construir un esquema de clasificación y estructurar un campo de interés.

De igual forma Petersen et al. [7] define los pasos esenciales del proceso de un estudio de mapeo sistemático, los cuales son: definición de las preguntas de investigación, realizar la búsqueda de artículos importantes, seleccionar investigaciones o estudios, revisión de estudios, clasificación de estudios, la extracción y el mapeo de datos.

En base a la guía presentada por Petersen et al. [7] se diseñó el diagrama de flujo presentado en la Figura 2.1.

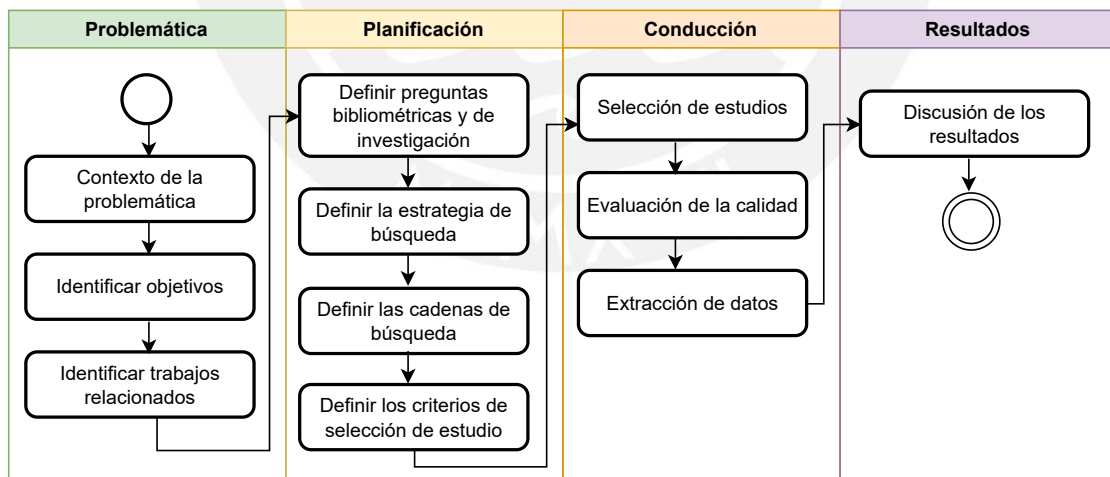


Figura 2.1: Diagrama de flujo del proceso de mapeo sistemático Petersen et al.

2.2. Preguntas bibliométricas

Esta investigación propone las siguientes preguntas de bibliometría:

- **PB-1:** ¿En qué países se publican los artículos? (OE-10).

- **PB-2:** ¿Con qué frecuencia anual se publican los artículos? (**OE-10**).
- **PB-3:** ¿Cuáles son los tipos de publicación? (**OE-10**).
- **PB-4:** ¿En dónde se publican los artículos? (**OE-10**).

2.3. Preguntas de investigación

Las preguntas que se formularon son las siguientes:

- **PI-1:** ¿Cuáles son las plataformas de desarrollo *Low-Code* que actualmente se vienen utilizando para la implementación de software? (**OE-1**).
- **PI-2:** ¿Cuáles son las características de las plataformas de desarrollo *Low-Code*? (**OE-2**).
- **PI-3:** ¿Cuáles son las ventajas y desventajas de usar plataformas de desarrollo *Low-Code*? (**OE-3**).
- **PI-4:** ¿Cuáles son las áreas de la organización donde aplicaron el uso de plataformas de desarrollo *Low-Code*? (**OE-4**).
- **PI-5:** ¿Cuáles son las características que se tienen por cada una de las plataformas *Low-Code* que se vienen utilizando? (**OE-5**).
- **PI-6:** ¿Cuáles son las ventajas y desventajas que se tienen por cada una de las plataformas *Low-Code* que se vienen utilizando? (**OE-6**).
- **PI-7:** ¿Cuáles son los tipos de aplicaciones por cada una de las plataformas de desarrollo *Low-Code*? (**OE-7**).
- **PI-8:** ¿Cuáles son las industrias donde se utilizan plataformas de desarrollo *Low-Code*? (**OE-8**).
- **PI-9:** ¿Cuáles son los tipos de aplicaciones por industrias donde se utilizan plataformas de desarrollo *Low-Code*? (**OE-9**).

2.4. Estrategia de búsqueda

Lockwood et al. [12] menciona que el marco PICo consta de la población, el fenómeno de interés y el contexto donde estos componentes debería describirse claramente, evitar ambigüedad y debe haber una congruencia entre ellos.

Las preguntas planteadas en la sección 4.2 para este mapeo sistemático se han definido con la ayuda de los criterios PICo (población, interés y contexto), el detalle se visualiza en la Tabla 2.1.

Criterio	Alcance
Población	Desarrollo de software.
Interés	Plataformas de desarrollo <i>Low-Code</i> .
Contexto	Al tratarse de un contexto amplio, se desestima este criterio.

Tabla 2.1: Criterios de la técnica PICO

Para definir la estrategia de búsqueda, se tomó como base los alcances según el criterio descrito en la Tabla 2.1; con ello se definieron términos en inglés para los conceptos respectivos, como se muestra en la Tabla 2.2.

Criterio	Término
Población	Software, Application, App, System, Development, Implementation.
Interés	Low-Code, Low Code, Platform, LCAP, LCDP.
Contexto	Al tratarse de un contexto amplio, se desestima este criterio.

Tabla 2.2: Términos de búsqueda

2.5. Cadenas de búsqueda

Con relación a las bases de datos especializadas electrónicas de artículos científicos seleccionadas para este estudio son: SCOPUS, IEEE, Web of Science, ACM Digital.

Las cadenas definidas de cada base de datos corresponden para a la Tabla 2.3, 2.4, 2.5 y 2.6 respectivamente.

Fuente	Cadena de búsqueda
ACM	(Title:(“Low-Code” OR “LowCode”) AND Title:(“Platform”) OR Title:(“LCAP” OR “LCDP”) AND Title:(“Software” OR “Application” OR “App” OR “System”))OR (Abstract:(“Low-Code” OR “LowCode”) AND Abstract:(“Platform”) OR Abstract:(“LCAP” OR “LCDP”) AND Abstract:(“Software” OR “Application” OR “App” OR “System”)) OR (Keyword:(“Low-Code” OR “LowCode”) AND Keyword:(“Platform”) OR Keyword:(“LCAP” OR “LCDP”) AND Keyword:(“Software” OR “Application” OR “App” OR “System”))

Tabla 2.3: Cadenas de búsqueda para ACM

Fuente	Cadena de búsqueda
Web of Science	(TI=("Software" OR "Application" OR "App" OR "System") AND TI=("Development" OR "Implementation") AND TI=("Low-Code" OR "LowCode") AND TI=("Platform") OR TI=("LCAP" OR "LCDP"))OR (AB=("Software" OR "Application" OR "App" OR "System") AND AB=("Development" OR "Implementation") AND AB=("Low-Code" OR "LowCode") AND AB=("Platform") OR AB=("LCAP" OR "LCDP")) OR (AK=("Software" OR "Application" OR "App" OR "System") AND AK=("Development" OR "Implementation") AND AK=("Low-Code" OR "LowCode") AND AK=("Platform") OR AK=("LCAP" OR "LCDP"))

Tabla 2.4: Cadenas de búsqueda para Web of Science

Fuente	Cadena de búsqueda
IEEE	("All Metadata":"Software" OR "All Metadata":"Application" OR "All Metadata":"App" OR "All Metadata":"System" OR "All Metadata":"Development" OR "All Metadata":"Implementation") AND ("All Metadata":"Low-Code" OR "All Metadata":"Low Code") AND ("All Metadata":"Platform" OR "All Metadata":"LCDP" OR "All Metadata":"LCAP")

Tabla 2.5: Cadenas de búsqueda para IEEE

Fuente	Cadena de búsqueda
Scopus	TITLE-ABS-KEY(("Software" OR "Application" OR "App" OR "System") AND ("Development" OR "Implementation") AND ("Low-Code" OR "Low Code")AND ("Platform") OR ("LCAP" OR "LCDP"))

Tabla 2.6: Cadenas de búsqueda para Scopus

2.6. Criterios de selección de estudios

Petersen et al. [7] menciona que los criterios de selección y exclusión se utilizan para descartar estudios que no son relevantes para responder las preguntas de investigación planteadas.

■ Criterios de inclusión

- **CI-1:** Publicaciones y artículos escritos en idioma inglés o español.
- **CI-2:** Estudios que hagan referencia al desarrollo de software.
- **CI-3:** Estudios que hagan referencia al uso de plataformas *Low-Code*.
- **CI-4:** Estudios con acceso a su contenido.

■ Criterios de exclusión

- **CE-1:** Los estudios de tipo mapeo sistemático, revisión sistemática, conference Review, Book y Proceeding como recopilatorio de investigaciones.
- **CE-2:** Publicaciones anteriores al año 2018, según Bucaioni et al. [7] las primeras investigaciones en este campo aparecieron a partir del 2018.
- **CE-3:** Las publicaciones duplicadas. En caso hubiera resultados duplicados, se considerará aquellas cuya información sea mas completa.
- **CE-4:** Estudios únicamente que hagan referencia al trabajo de otros artículos.
- **CE-5:** Estudios que en su contenido no referencien al uso de plataformas de desarrollo *Low-Code* para desarrollar software.

2.7. Proceso de extracción y clasificación

Los criterios de inclusión y exclusión se utilizan para descartar estudios que no son relevantes para responder las preguntas de investigación planteadas [7], es por ello que se propone cuatro etapas que permitirán seleccionar los estudios según los criterios de selección.

En el MSL se ha considerado el siguiente proceso, aplicando los criterios según la Tabla 2.7.

Procedimiento	Criterio de selección
Primera etapa	Búsqueda inicial
Segunda etapa	CI-1, CE-1, CE-2 y CE-3
Tercera etapa	CI-2, CI-3 y CE-4
Cuarta etapa	CI-4 y CE-5

Tabla 2.7: Procedimientos y Criterios de Inclusión y Exclusión

- **Primera etapa:** Hace referencia a la búsqueda inicial en las bases de datos especializadas.

- **Segunda etapa:** Se aplican el criterio de inclusión CI-1; se aplican los criterios de exclusión: CE-1, CE-2 y CE-3 sobre los títulos y las palabras clave de los artículos.
- **Tercera etapa:** Se aplican los criterios de inclusión CI-2, CI-3 y el criterio de exclusión CE-4 sobre los títulos, palabras clave y resúmenes de los artículos.
- **Cuarta etapa:** Se realizó una lectura rápida sobre el contenido del estudio y aplicó el criterio de inclusión CI-4 y el criterio de exclusión CE-5.



CAPÍTULO 3

Conducción

3.1. Selección de estudios

En mayo del 2023 se realizaron las búsquedas en las bases de datos especializadas electrónicas de artículos científicos, para este estudio las bases de datos son: SCOPUS, IEEE, Web of Science, ACM Digital. Las búsquedas se realizaron con las cadenas mencionadas en las tablas: Tabla 2.3, 2.4, 2.5 y 2.6 respectivamente.

En base al proceso de extracción y clasificación propuesto en la Tabla 2.7, la Tabla 3.1 muestra el detalle del refinamiento de los estudios y la Tabla 3.2 detalla los artículos clasificados hasta la etapa 4. Se utilizaron los criterios de inclusión y exclusión para descartar los estudios que no son relevantes para responder las preguntas de investigación planteadas.

Fuente	Etapas 1	Etapas 2	Etapas 3	Etapas 4
Scopus	166	132	64	42
IEEE	6	2	2	2
Web of Science	253	73	6	5
ACM	17	3	1	1
TOTAL	442	376	73	50

Tabla 3.1: Resultados del proceso de extracción y clasificación

Ref.	Título
P-01 [13]	OSTRICH: a rich template language for low-code development (extended version)
P-02 [14]	BALTICLSC: LOW-CODE SOFTWARE DEVELOPMENT PLATFORM FOR LARGE SCALE COMPUTATIONS
P-03 [15]	Analyzing business process management capabilities of low-code development platforms
P-04 [16]	EDALoCo: Enhancing the accessibility of blockchains through a low-code approach to the development of event-driven applications for smart contract management
P-05 [17]	A Low-Code Development Framework for Constructing Industrial Apps

Sigue en la página siguiente.

P-06 [18]	Improving the developer experience with a low-code process modelling language
P-07 [19]	Understanding Low-Code or No-Code Adoption in Software Startups: Preliminary Results from a Comparative Case Study
P-08 [20]	Building a Polyglot Data Access Layer for a Low-Code Application Development Platform: (Experience Report)
P-09 [21]	Student Request System Prototype Using Low-Code Development Platform
P-10 [22]	A GUI-based Low-Code Development Platform for Power Systems Analysis
P-11 [23]	Low-Code Versus Code-Based Software Development: Which Wins the Productivity Game?
P-12 [24]	Democratizing the development of recommender systems by means of low-code platforms
P-13 [25]	In Search of the Essence of Low-Code: An Exploratory Study of Seven Development Platforms
P-14 [26]	Accelerating SQL with Complex Visual Querying
P-15 [27]	Recent Trends in Software Development: Low-Code Solutions
P-16 [28]	Closing the Gap between Designers and Developers in a Low Code Ecosystem
P-17 [29]	Low-Code Development Platforms: A Descriptive Study
P-18 [30]	Low-code application platform in meeting increasing software demands Quickly: SetXRM
P-19 [31]	Using Microsoft PowerApps, Mendix and OutSystems in Two Development Scenarios: An Experience Report
P-20 [32]	An overview on how to develop a low-code application using OutSystems
P-21 [33]	Low-code platform for automating business processes in manufacturing
P-22 [34]	Solution Discovery over Feature Toggling with Built-in Abstraction in OutSystems
P-23 [35]	Towards supporting SPL engineering in low-code platforms using a DSL approach
P-24 [36]	Automated Refactoring of Unbounded Queries in Software Automation Platforms
P-25 [37]	OSTRICH - A Type-Safe Template Language for Low-Code Development
P-26 [38]	Celonis studio - A low-code development platform for citizen developers
P-27 [39]	A Dialogue Interface for Low Code Program Evolution
P-28 [40]	Factors that affect the utilization of low-code development platforms: survey study
P-29 [41]	API Management Maturity of Low-Code Development Platforms

Sigue en la página siguiente.

P-30 [42]	Low-code from frontend to backend: Connecting conversational user interfaces to backend services via a low-code IoT platform
P-31 [43]	Test mocks for low-code applications built with OutSystems
P-32 [44]	Leveraging Low Code Development of Smart Personal Assistants: An Integrated Design Approach with the SPADE Method
P-33 [45]	BalticLSC: A low-code HPC platform for small and medium research teams
P-34 [46]	Automated migration of EuGENia graphical editors to the web
P-35 [47]	Comparative Study on Various Low Code Business Process Management Platforms
P-36 [48]	Introduction of an Assistance System to Support Domain Experts in Programming Low-Code to Leverage Industry 5.0
P-37 [49]	Low-code platforms and languages: the future of software development
P-38 [50]	From Builders to Editors: Bidirectional Transformations of Low-code Models
P-39 [51]	Digital Transformation of Enterprises Using a Low-Code Platform
P-40 [52]	Aligning Software Engineering Teaching Strategies and Practices with Industrial Needs
P-41 [53]	A Use Case-based Investigation of Low-Code Development Platforms
P-42 [54]	Simple Dependent Types for OSTRICH
P-43 [55]	Improving Collaboration Efficiency Between UX/UI Designers and Developers in a Low-Code Platform
P-44 [56]	Supporting the understanding and comparison of low-code development platforms
P-45 [57]	Testing the Concept of the RUN-Time Adaptive Enterprise: Combining Organization and IT Agnostic Enterprise Models with Organization Implementation Variables and Low Code Technology
P-46 [58]	Adding Blockchain and Smart Contracts to a Low-Code Development Platform
P-47 [59]	App Development via Low-Code Programming as Part of Modern Industrial Engineering Education
P-48 [60]	Application for the Management of Sports Performance in CrossFit Supported by an Artificial Intelligence Cognitive Service
P-49 [61]	Approaches to improve shop floor management
P-50 [62]	Low Code Development Platform for Digital Transformation

Tabla 3.2: Resultado de publicaciones seleccionadas

3.2. Evaluación de la calidad

Los cincuenta estudios anteriormente seleccionados fueron sometidos a un proceso de evaluación de la calidad. En la Tabla 3.3 se detallan las preguntas de calidad. La Tabla

3.4 presenta el resumen de los resultados aplicando las preguntas de calidad basadas en la propuesta de Zarour et al. [63]. Este instrumento utiliza una escala de calificación de 3 niveles (Sí: 1 punto, No: 0 puntos, Parcialmente = 0.5 puntos), finalmente la Tabla 3.5 y la Tabla 3.6 presentan los resultados de la evaluación y los estudios primarios respectivamente.

Todas aquellas investigaciones relevantes con una calificación mayor o igual a 2.5 son consideradas como aceptadas; por el contrario, aquellas con un puntaje menor a 2.5 son descartadas.

El puntaje promedio es de 3.47. El resultado de la evaluación de la calidad demuestra que 3 publicaciones no cumplen los criterios de calidad quedando 47 estudios aceptables.

Identificador	Pregunta
QA1	¿El objetivo de la investigación ha sido suficientemente explicado?
QA2	¿La idea o el enfoque presentado ha sido claramente explicado?
QA3	¿Se han considerado amenazas a la validez?
QA4	¿Hay una adecuada descripción del contexto en donde se ha llevado a cabo la investigación?
QA5	¿Se mencionan de forma clara los hallazgos del estudio?

Tabla 3.3: Listado de preguntas para aseguramiento de la calidad

Fuente	Etapas 1	Etapas 2	Etapas 3	Etapas 4	Criterios de calidad
Scopus	166	132	69	42	39
IEEE	6	2	2	2	2
Web of Science	253	73	6	5	5
ACM	17	3	1	1	1
TOTAL	442	376	78	50	47

Tabla 3.4: Resultados del proceso de extracción y clasificación

Ref.	QA1	QA2	QA3	QA4	QA5	Calificación
P-01	0.5	1	0	1	1	3.5
P-02	1	1	0	1	1	4
P-03	1	1	1	0.5	0.5	4
P-04	1	1	1	0.5	0.5	4
P-05	0.5	0.5	1	1	0.5	3.5
P-06	1	1	1	1	0.5	4.5
P-07	0.5	0.5	1	1	0.5	3.5

Sigue en la página siguiente.

P-08	1	1	0	1	0.5	3.5
P-09	1	1	0	1	1	4
P-10	1	1	0	1	0.5	3.5
P-11	1	1	0	1	1	4
P-12	0.5	1	0	1	0.5	3
P-13	1	1	0	1	1	4
P-14	1	1	0	1	0.5	3.5
P-15	0.5	1	0	1	0.5	3
P-16	0.5	1	0	1	0.5	3
P-17	1	1	0	1	1	4
P-18	0.5	1	0	1	0.5	3
P-19	1	1	0	1	1	4
P-20	1	1	0	1	1	4
P-21	1	1	0	1	0.5	3.5
P-22	1	1	0	1	0.5	3.5
P-23	0.5	0.5	0	0.5	0.5	2
P-24	0.5	1	0	1	1	3.5
P-25	0.5	1	0	1	0.5	3
P-26	1	1	0	1	0.5	3.5
P-27	0.5	1	0	1	0.5	3
P-28	0.5	1	0	1	1	3.5
P-29	1	1	1	1	0.5	4.5
P-30	0.5	0.5	0	0.5	0.5	2
P-31	0.5	1	0	1	1	3.5
P-32	0.5	0.5	0	0.5	0.5	2
P-33	0.5	1	0	1	1	3.5
P-34	0.5	1	0	1	0.5	3
P-35	1	1	0	1	1	4
P-36	0.5	1	1	1	0.5	4
P-37	1	1	0	1	1	4
P-38	0.5	1	0	1	1	3.5
P-39	0.5	1	0	1	1	3.5
P-40	0.5	1	1	1	0.5	4
P-41	0.5	1	1	1	1	4.5
P-42	0.5	1	0	1	1	3.5
P-43	0.5	1	0	1	1	3.5
P-44	1	1	0	1	0.5	3.5
P-45	0.5	1	0	1	1	3.5
P-46	1	1	0	1	0.5	3.5
P-47	0.5	0.5	0	1	0.5	2.5

Sigue en la página siguiente.

P-48	1	0.5	0	0.5	1	3
P-49	0.5	0.5	0	1	0.5	2.5
P-50	1	0.5	0	1	0.5	3

Tabla 3.5: Resultado de la evaluación de los criterios de calidad

3.3. Extracción de datos

A fin de responder las preguntas planteadas se procede a la extracción de datos de los 47 artículos seleccionados, los siguientes datos fueron obtenidos de los estudios.

- **Id:** Identificador de la investigación.
- **País:** Países donde se publicaron los artículos, responde la pregunta (**PB-1**).
- **Año de publicación:** Año de publicación, responde la pregunta (**PB-2**).
- **Tipo de investigación:** Si es un artículo de revista o paper de conferencia, responde la pregunta (**PB-3**).
- **Medio de publicación:** El medio donde se publican las investigaciones, conferencia o revista, responde la pregunta (**PB-4**).
- **Plataformas *Low-Code*:** Identifica la plataforma y sus características, responde la pregunta (**PI-1, PI-2 y PI-5**).
- **Ventajas y desventajas:** Identifica las ventajas y desventajas de usar plataformas *Low-Code* y por cada una de ellas, responde la pregunta (**PI-3 y PI-6**).
- **Área organizativa:** Identifica las áreas de la organización que implementar software desarrollado en plataformas *Low-Code*, responde la pregunta (**PI-4**).
- **Tipo de aplicación:** El tipo de software que se ha implementado, responde la pregunta (**PI-7**).
- **Industrias:** las industrias que implementan software desarrollado en plataformas *Low-Code*, responde la pregunta (**PI-8**).

Las investigaciones primarias para la extracción de datos son las siguientes:

Ref.	Título
P-01 [13]	OSTRICH: a rich template language for low-code development (extended version)
P-02 [14]	BALTICLSC: LOW-CODE SOFTWARE DEVELOPMENT PLATFORM FOR LARGE SCALE COMPUTATIONS
P-03 [15]	Analyzing business process management capabilities of low-code development platforms

Sigue en la página siguiente.

P-04 [16]	EDALoCo: Enhancing the accessibility of blockchains through a low-code approach to the development of event-driven applications for smart contract management
P-05 [17]	A Low-Code Development Framework for Constructing Industrial Apps
P-06 [18]	Improving the developer experience with a low-code process modelling language
P-07 [19]	Understanding Low-Code or No-Code Adoption in Software Startups: Preliminary Results from a Comparative Case Study
P-08 [20]	Building a Polyglot Data Access Layer for a Low-Code Application Development Platform: (Experience Report)
P-09 [21]	Student Request System Prototype Using Low-Code Development Platform
P-10 [22]	A GUI-based Low-Code Development Platform for Power Systems Analysis
P-11 [23]	Low-Code Versus Code-Based Software Development: Which Wins the Productivity Game?
P-12 [24]	Democratizing the development of recommender systems by means of low-code platforms
P-13 [25]	In Search of the Essence of Low-Code: An Exploratory Study of Seven Development Platforms
P-14 [26]	Accelerating SQL with Complex Visual Querying
P-15 [27]	Recent Trends in Software Development: Low-Code Solutions
P-16 [28]	Closing the Gap between Designers and Developers in a Low Code Ecosystem
P-17 [29]	Low-Code Development Platforms: A Descriptive Study
P-18 [30]	Low-code application platform in meeting increasing software demands Quickly: SetXRM
P-19 [31]	Using Microsoft PowerApps, Mendix and OutSystems in Two Development Scenarios: An Experience Report
P-20 [32]	An overview on how to develop a low-code application using OutSystems
P-21 [33]	Low-code platform for automating business processes in manufacturing
P-22 [34]	Solution Discovery over Feature Toggling with Built-in Abstraction in OutSystems
P-24 [36]	Automated Refactoring of Unbounded Queries in Software Automation Platforms
P-25 [37]	OSTRICH - A Type-Safe Template Language for Low-Code Development
P-26 [38]	Celonis studio - A low-code development platform for citizen developers
P-27 [39]	A Dialogue Interface for Low Code Program Evolution
P-28 [40]	Factors that affect the utilization of low-code development platforms: survey study

Sigue en la página siguiente.

P-29 [41]	API Management Maturity of Low-Code Development Platforms
P-31 [43]	Test mocks for low-code applications built with OutSystems
P-33 [45]	BalticLSC: A low-code HPC platform for small and medium research teams
P-34 [46]	Automated migration of EuGENia graphical editors to the web
P-35 [47]	Comparative Study on Various Low Code Business Process Management Platforms
P-36 [48]	Introduction of an Assistance System to Support Domain Experts in Programming Low-Code to Leverage Industry 5.0
P-37 [49]	Low-code platforms and languages: the future of software development
P-38 [50]	From Builders to Editors: Bidirectional Transformations of Low-code Models
P-39 [51]	Digital Transformation of Enterprises Using a Low-Code Platform
P-40 [52]	Aligning Software Engineering Teaching Strategies and Practices with Industrial Needs
P-41 [53]	A Use Case-based Investigation of Low-Code Development Platforms
P-42 [54]	Simple Dependent Types for OSTRICH
P-43 [55]	Improving Collaboration Efficiency Between UX/UI Designers and Developers in a Low-Code Platform
P-44 [56]	Supporting the understanding and comparison of low-code development platforms
P-45 [57]	Testing the Concept of the RUN-Time Adaptive Enterprise: Combining Organization and IT Agnostic Enterprise Models with Organization Implementation Variables and Low Code Technology
P-46 [58]	Adding Blockchain and Smart Contracts to a Low-Code Development Platform
P-47 [59]	App Development via Low-Code Programming as Part of Modern Industrial Engineering Education
P-48 [60]	Application for the Management of Sports Performance in CrossFit Supported by an Artificial Intelligence Cognitive Service
P-49 [61]	Approaches to improve shop floor management
P-50 [62]	Low Code Development Platform for Digital Transformation

Tabla 3.6: *Estudios Primarios*

CAPÍTULO 4

Análisis de resultados

Se recopiló la información para responder las preguntas bibliométricas y de investigación planteadas.

4.1. Preguntas bibliométricas

- **PB-1 ¿En qué países se publican los artículos?**

La Tabla 4.1 presenta las publicaciones por países donde Estados Unidos tiene 14 publicaciones y China 5.

País	Ref.
Estados Unidos	P-01, P-10, P-15, P-19, P-20, P-21, P-25, P-26, P-28, P-31, P-37, P-38, P-39, P-41
China	P-06, P-08, P-13, P-40, P-46
Reino Unido	P-17, P-24, P-42, P-43
Japón	P-29, P-36, P-48, P-50
Grecia	P-04, P-05, P-18
India	P-03, P-09, P-16
Suiza	P-44, P-45, P-49
Turquía	P-11, P-47
Alemania	P-35
Australia	P-14
Bélgica	P-22
Eslovaquia	P-34
Eslovenia	P-33
España	P-07
Finlandia	P-12
Maldivas	P-02
Rusia	P-27

Tabla 4.1: Investigaciones clasificadas por país de publicación

■ **PB-2 ¿Con qué frecuencia anual se publican los artículos?**

Se aprecia que las primeras publicaciones son del año 2018 y 2019. La mayor cantidad de publicaciones son 17 en el 2021 y 2022 y se nota así una tendencia al crecimiento; el detalle se encuentra en la Figura 4.1 y la Tabla 4.2.

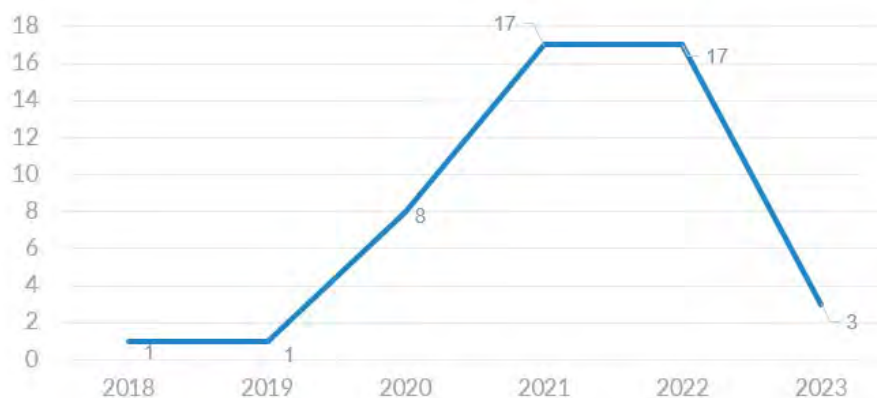


Figura 4.1: Investigaciones clasificadas por año de publicación

Año	Ref.
2018	P-06
2019	P-21
2020	P-08, P-12, P-16, P-20, P-31, P-34, P-44, P-47
2021	P-02, P-05, P-13, P-18, P-19, P-22, P-24, P-25, P-26, P-28, P-29, P-33, P-38, P-43, P-45, P-49, P-50
2022	P-01, P-07, P-09, P-10, P-11, P-14, P-15, P-17, P-27, P-35, P-36, P-37, P-39, P-40, P-41, P-42, P-48
2023	P-03, P-04, P-46

Tabla 4.2: Investigaciones clasificadas por año de publicación

■ **PB-3 ¿Cuáles son los tipos de publicación?**

En la Figura 4.2 se presentan las cantidades de artículos por tipo de publicación donde la mayor cantidad de investigaciones son de tipo Conference Paper con 38 publicaciones seguido por Journal Article con 9 investigaciones, la Tabla 4.3 tiene el detalle.

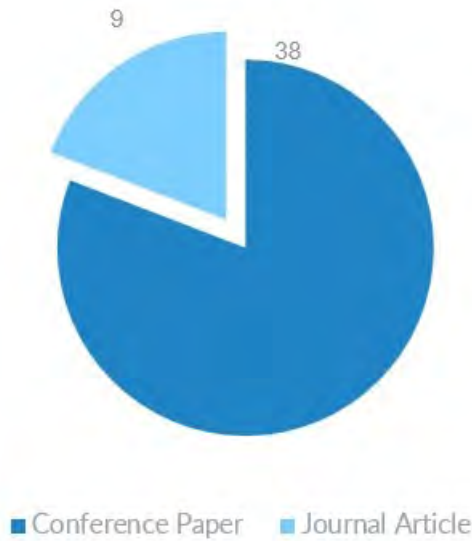


Figura 4.2: Investigaciones clasificadas por tipo de publicación

Tipo	Ref.
Conference Paper	P-06, P-07, P-08, P-09, P-10, P-12, P-13, P-14, P-15, P-16, P-17, P-18, P-19, P-20, P-21, P-22, P-24, P-25, P-26, P-27, P-29, P-31, P-33, P-34, P-35, P-37, P-38, P-41, P-42, P-43, P-44, P-45, P-46, P-47, P-48, P-49, P-50
Journal Article	P-01, P-02, P-03, P-04, P-11, P-28, P-36, P-39, P-40

Tabla 4.3: Investigaciones clasificadas por tipo de publicación

▪ **PB-4 ¿En dónde se publican los artículos?**

La gran mayoría de los investigadores prefieren publicar artículos en conferencias de ACM/IEEE como medio de publicación donde se tienen 12 artículos. El detalle se puede ubicar en la Tabla 4.4.

Medio	Ref.
ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion	P-06, P-12, P-13, P-16, P-19, P-22, P-24, P-25, P-31, P-34, P-38, P-43
International Conference, IFM, Lugano, Switzerland	P-27, P-42

Sigue en la página siguiente.

International Conference on Innovations in Bio-Inspired Computing and Applications	P-46, P-48
Software and Systems Modeling	P-01
Vol. 40 No. 4 (2021): Computing and Informatics	P-02
SOFTWARE-PRACTICE and EXPERIENCE	P-03
Computer Standards and Interfaces	P-04
CCF Conference on Computer Supported Cooperative Work and Social Computing	P-05
23rd International Conference, BPMDS 2022 and 27th International Conference, EMMSAD 2022, Held at CAiSE 2022	P-07
20th IFIP WG 6.1 International Conference, DAIS 2020, Held as Part of the 15th International Federated Conference on Distributed Computing Techniques, DisCoTec 2020	P-08
2022 International Conference on Algorithms, Data Mining, and Information Technology (ADM-IT)	P-09
2022 China Automation Congress (CAC)	P-10
IT Professional Volume 24	P-11
2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)	P-14
Proceedings of the Future Technologies Conference (FTC) 2021, Volume 3	P-15
17th Iberian Conference on Information Systems and Technologies (CISTI)	P-17
FOURTH INTERNATIONAL CONFERENCE OF MATHEMATICAL SCIENCES (ICMS 2020)	P-18
2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)	P-20
13th IFAC Workshop on Intelligent Manufacturing Systems IMS 2019	P-21
2021 Best Dissertation Award, Doctoral Consortium, and Demonstration and Resources Track at BPM, BPM-D 2021	P-26
Romanian Journal of Information Technology and Automatic Control	P-28

Sigue en la página siguiente.

International Conference on Evaluation and Modeling Methods for Systems Analysis and Development	P-29
2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)	P-33
2022 International Conference on Inventive Computation Technologies (ICICT)	P-35
IEEE Robotics and Automation Letters Volume 7	P-36
2022 20th International Conference on Emerging eLearning Technologies and Applications (ICE-TA)	P-37
Russian Engineering Research	P-39
MDPI	P-40
14th Central European Workshop on Services and their Composition, ZEUS 2022	P-41
2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)	P-44
10th Enterprise Engineering Working Conference	P-45
Proceedings of the AHFE 2020 Virtual Conference on Human Factors and Systems Interaction	P-47
Proceedings of the 3rd International Conference on Human Interaction and Emerging Technologies: Future Applications (IHIET 2020)	P-49
Information and Communication Technology for Competitive Strategies (ICTCS 2020)	P-50

Tabla 4.4: Investigaciones clasificadas por medio de publicación

4.2. Preguntas de investigación

- **PI-1 ¿Cuáles son las plataformas de desarrollo *Low-Code* que actualmente se vienen utilizando para la implementación de software?**

Se observa que las plataformas: OutSystems, Mendix y Microsoft Power Apps son las más recurrentes en los artículos. Los artículos [15], [16], [17], [25], [27], [31], [32], [33], [34], [35], [36], [40], [41], [43], [47], [48], [49], [51], [53], [56] y [62] mencionan más de una plataforma de desarrollo *Low-Code* en su publicación correspondiente. Las Plataformas de desarrollo *Low-Code* y su cantidad de menciones se detallan en la Tabla 4.5.

Plataforma	Ref.	Menciones
OutSystems	P-01, P-02, P-05, P-06, P-08, P-11, P-14, P-15, P-16, P-19, P-20, P-21, P-24, P-25, P-27, P-28, P-29, P-31, P-38, P-40, P-41, P-42, P-43, P-44, P-46, P-48, P-50	27
Mendix	P-03, P-05, P-13, P-19, P-20, P-21, P-28, P-29, P-31, P-36, P-39, P-41, P-44, P-45, P-50	15
Microsoft Power Apps	P-03, P-05, P-09, P-13, P-15, P-19, P-21, P-28, P-37, P-41, P-44, P-49	12
Appian	P-13, P-21, P-28, P-35, P-41, P-44, P-50	7
Salesforce Lightning	P-03, P-15, P-21, P-28, P-31, P-35, P-50	7
Zoho Creator	P-03, P-37, P-44	3
KissFlow	P-20, P-44, P-50	3
Google App Maker	P-21, P-44, P-50	3
Pega	P-13, P-29	2
BalticLSC	P-02, P-33	2
Quickbase	P-13, P-50	2
EDALoCo	P-04	1
JogetCloud	P-05	1
APICloud	P-05	1
DaDayun	P-05	1
H3Yun	P-05	1
JiandaoYun	P-05	1
iVx	P-05	1
Microsoft Power BI	P-07	1
ECUST	P-10	1
Google Appsheet	P-03	1
Amazon Honeycode	P-03	1
Thinkwise	P-03	1
NintexWorkflow	P-03	1
Bonita Platform	P-13	1
Xatkit	P-15	1
Ktrain	P-15	1
RESTsec	P-15	1
SetXRM	P-18	1
Wavemaker	P-13	1
Node-RED	P-04	1

Sigue en la página siguiente.

Aurea BPM	P-21	1
TrackVia	P-21	1
SAP	P-24	1
Celonis Studio	P-26	1
Kony	P-28	1
Betty Blocks	P-29	1
Unity	P-31	1
ROSE	P-34	1
Caspio	P-35	1
HAWE e Design	P-36	1
Simulink	P-36	1
LegoNXT-G	P-36	1
Netgrif	P-37	1
Airtable	P-39	1

Tabla 4.5: Plataformas de desarrollo *Low-Code*

▪ **PI-2 ¿Cuáles son las características de las plataformas de desarrollo *Low-Code*?**

Según las características recabadas en la Tabla 4.6 las más resaltantes son que utilizan elementos gráficos, lenguajes y programación visual, ofrece un desarrollo rápido y con poca codificación, además son consideradas como plataformas destinadas al desarrollo del usuario final o desarrollador ciudadano, proviene del paradigma basado en modelos; además cuenta con funciones de arrastrar y soltar y simplifica la complejidad del proceso de construcción de software. En la Tabla 4.6 se encuentran las características de las plataformas *Low-Code*.

Característica	Ref.	Menciones
Elementos gráficos, lenguajes y programación visual	P-02, P-03, P-04, P-05, P-07, P-08, P-10, P-11, P-12, P-13, P-14, P-15, P-16, P-18, P-19, P-21, P-27, P-28, P-31, P-33, P-34, P-35, P-36, P-37, P-39, P-41, P-44, P-45, P-47, P-49, P-50	31
Ofrece un desarrollo rápido y con poca codificación	P-05, P-08, P-09, P-11, P-12, P-14, P-15, P-18, P-19, P-21, P-22, P-26, P-27, P-28, P-29, P-31, P-34, P-35, P-36, P-37, P-39, P-41, P-42, P-44, P-45, P-47, P-49	27

Sigue en la página siguiente.

Destinadas al desarrollo del usuario final o desarrollador ciudadano	P-17, P-21, P-26, P-27, P-28, P-29, P-33, P-41, P-42, P-44, P-49, P-50	12
Proviene del paradigma basado en modelos	P-02, P-13, P-15, P-18, P-21, P-28, P-33, P-37, P-45, P-49, P-50	11
Cuenta con funciones de arrastrar y soltar	P-03, P-05, P-07, P-10, P-12, P-13, P-18, P-31, P-37	9
Simplificar la complejidad del proceso de construcción de software	P-01, P-02, P-03, P-15, P-25, P-26, P-36, P-37	8
Basada en la nube	P-02, P-33, P-34, P-37, P-41, P-44, P-47	7
No requiere demasiados conocimientos de tecnologías y lenguajes de programación	P-04, P-07, P-10, P-41, P-44, P-47, P-50	7
Es un paradigma emergente	P-12, P-17, P-27, P-31, P-41, P-45, P-50	7
Generación automática de código	P-21, P-47, P-49, P-50	4
Posee componentes pre diseñados	P-19, P-35, P-50	3
Permite la reutilización de código	P-04, P-25, P-35	3
Las aplicaciones no se crean desde cero y además permite su personalización	P-28, P-41, P-50	3
Integración con almacenes de datos	P-08, P-44	3
Integración con servicios externos	P-08, P-44	2
Acelera la entrega de software	P-11, P-13	2
Los costo de licencia son generalmente altos	P-11, P-13	2
En un solo entorno provee herramientas de modelado, bibliotecas, editores GUI, administración de dependencias, implementación, compilación, asistentes, etc.	P-13, P-35	2
Alto nivel de abstracción de la programación	P-17, P-41	2
Permite una comunicación armoniosa entre el negocio y TI,	P-35, P-45	2
Trabaja en línea	P-02	1
Contiene un repositorio de componentes	P-02	1
Ofrece notaciones de modelado	P-03	1

Sigue en la página siguiente.

Permite una retroalimentación rápida	P-07	1
Permite la generación y entrega de aplicaciones comerciales	P-21	1
Se centra en la velocidad y la simplicidad con ciclos de desarrollo cortos	P-26	1
Incluyen conocimiento empresarial dentro del proceso de desarrollo de software	P-26	1
Permite desarrollar software empresarial	P-29	1
Suelen gestionar APIs	P-29	1
Permite la creación de flujos de trabajo	P-35	1
Provee dashboards e informes	P-35	1
Ofrece funciones de gestión de usuarios	P-35	1
Ofrece mediciones de la complejidad en base a métricas	P-36	1
Permite la verificación automática de valores de base de datos	P-39	1
Usualmente conocidas como plataforma como servicio (PaaS)	P-44	1
Reportes preconstruidos	P-44	1
Se basa en tres conceptos: entidad, lógica e interfaz	P-45	1
Cuarta generación de lenguajes de programación, también llamada programación declarativa	P-47	1

Tabla 4.6: Características de las plataformas de desarrollo *Low-Code*

▪ **PI-3 ¿Cuáles son las ventajas y desventajas de usar plataformas de desarrollo *Low-Code*?**

Entre las ventajas del uso de las plataformas *Low-Code* tenemos que acelera el desarrollo de software [18] [22] [29] [40], acelera la entrega de software [40] [49] [53] [62], facilita la innovación y la transformación digital [24] [40] [47] [62] así mismo los autores [25] [41] [62] mencionan que permite una mayor productividad y los autores [47] [56] [62] aseguran que facilitan la escalabilidad, la Tabla 4.7 especifica las ventajas a más detalle.

Las desventajas para los autores [29] [40] los precios de licenciamiento son altos y la información sobre las plataformas es insuficiente, además los autores [29] [40] aseguran que son pocos los desarrolladores calificados y que también existe una resistencia al cambio [19];

por otro lado, no existe mucha difusión de los caso de éxito. La Tabla 4.8 especifica las desventajas a más detalle.

Ventaja	Ref.	Menciones
Acelera el desarrollo de software	P-06, P-10, P-17, P-28	4
Acelera la entrega de software	P-28, P-37, P-41, P-50	4
Facilita la innovación y la transformación digital	P-12, P-28, P-35, P-50	4
Mayor productividad	P-13, P-29, P-50	3
Facilita la escalabilidad	P-35, P-44, P-50	3
Colaboración entre desarrolladores expertos y desarrolladores ciudadanos	P-13, P-40	2
Reduce el coste de configuración, formación, despliegue y mantenimiento	P-04, P-35	2
Fácil integración	P-05, P-08	2
Menores costos de TI	P-17, P-28	2
Reduce el esfuerzo de configuración, formación, despliegue y mantenimiento	P-21, P-42	2
Mejora la experiencia de usuario	P-26, P-28	2
Alta eficiencia	P-05	1
Ciclo de vida más corto	P-10	1
Reduce la complejidad del desarrollo	P-17	1
Acelera la transformación digital	P-17	1
Rápida adopción	P-19	1
Generación de valor más rápida	P-26	1
Disminuye la carga de trabajo de los profesionales del departamento de TI	P-28	1
Reduce la dependencia de empleados con habilidades técnicas y profesionales de TI	P-28	1
Mayor capacidad de respuesta del negocio	P-28	1
Reducción del tiempo de comercialización	P-29	1
Decisiones más rápidas	P-35	1
Reduce la complejidad empresarial	P-37	1
Pueden contribuir a la optimización de procesos	P-47	1
Reducir esfuerzos de tareas rutinarias	P-13	1

Tabla 4.7: Ventajas sobre las plataformas Low-Code

Desventaja	Ref.	Menciones
Precios altos de licenciamiento	P-17, P-28	2
Información insuficiente sobre <i>Low-Code</i>	P-17, P-28	2
Pocos desarrolladores calificados	P-28, P-29	2
Baja calidad	P-28	1
Dependencia a un proveedor	P-28	1
Gestión inmadura de pruebas de software	P-31	1
Soluciones más complejas requieren un mayor conocimientos y experiencia	P-41	1
Muchas plataformas no incluyen algoritmos de aprendizaje automático	P-50	1
Baja disponibilidad de componentes y funciones	P-05	1
Resistencia al cambio tecnológico	P-07	1
No tienen mecanismos para abordar las ineficiencias en el proceso de conversión de diseño a código	P-16	1
Personalización limitada	P-17	1
Bajo rendimiento	P-17	1
Poca difusión de los casos de éxito	P-17	1
Puede generar deuda técnica con desarrolladores no expertos	P-24	1
Funciones limitadas	P-28	1
Flexibilidad limitada	P-28	1

Tabla 4.8: Desventajas sobre las plataformas *Low-Code*

- **PI-4** ¿Cuáles son las áreas de la organización donde aplicaron el uso de plataformas de desarrollo *Low-Code*?

Las áreas de aplicación son ventas, compras, finanzas y recursos humanos como áreas más mencionadas. En la Tabla 4.9 se observa el detalle.

Área	Ref.	Menciones
Ventas	P-07, P-28, P-35, P-44	4
Compras	P-07, P-28	2
Finanzas	P-07, P-35	2
Recursos Humanos	P-20, P-35	2
Salud	P-04	1
Administración	P-07	1
Atención al cliente	P-15	1
Trámite documental	P-41	1
Organización de viviendas sociales	P-45	1

Sigue en la página siguiente.

Votación electrónica	P-46	1
Educación y enseñanza	P-47	1
Línea de ensamblaje	P-49	1

Tabla 4.9: Áreas de aplicación

- **PI-5 ¿Cuáles son las características que se tienen por cada una de las plataformas *Low-Code* que se vienen utilizando?**

Se categorizan las características por plataforma *Low-Code* en la Tabla 4.10.

Plataforma	Características	Ref.	
OutSystems	Permite el desarrollo rápido de aplicaciones web y móviles de nivel empresarial.	P-01, P-07, P-11, P-14, P-15, P-20, P-22, P-38	
	Elementos gráficos, lenguajes y programación visual.	P-01, P-03, P-14, P-19, P-42, P-46, P-50	
	El IDE de la plataforma es el Service Studio que permite diseñar todos los aspectos de sus aplicaciones en un solo lugar, incluye la interfaz, la lógica empresarial, modelo de datos y la integración con sistemas externos.	P-01, P-03, P-16, P-19, P-25, P-31, P-38	
	Permite la integración de otras bases de datos externas se realiza en Integration Studio configurando la extensión para usar una conexión de base de datos y luego consultarla y usarla.	P-03, P-19, P-20, P-31, P-42, P-44	
	Cuenta con funciones de arrastrar y soltar.	P-16, P-31, P-41, P-44	
	Cubre todas las fases del desarrollo de software.	P-19, P-20, P-24, P-46	
	Provee Widgets listos para usar y se pueden personalizar con CSS y JavaScript.	P-20, P-38, P-42, P-43	
	Basado en la nube.	P-14, P-19, P-38	
	Permite el uso de plantillas de pantallas.	P-25, P-42, P-43	
	Permite desarrollar aplicaciones con alto nivel de abstracción.	P-07, P-41	
	La conexión a Email se realiza mediante un servidor SMTP.	P-19, P-41	
	Sigue en la página siguiente.		

	Permite desarrollar software a desarrolladores ciudadanos y a expertos.	P-29, P-41
	Plataforma fácil, segura y confiable.	P-05
	Permite la consulta visual de datos.	P-14
	Incluye el modelo de datos, diseño de interfaces, lógica empresarial, modelo de procesos.	P-15
	Modela todas las capas de la aplicación: procesos, interfaces, lógica y datos.	P-16
	La apariencia predefinida de una nueva aplicación se hereda de la interfaz de usuario.	P-16
	Las aplicaciones web se suelen utilizar en línea y las aplicaciones móviles se compilan como aplicaciones nativas y funciona en ambos sistemas operativos iOS y Android.	P-19
	Se puede almacenar datos e importar en base de datos.	P-19
	Tienda de aplicaciones llamada Forge.	P-19
	Es una plataforma como servicio (PaaS).	P-22
	Se implementa a través de flujos lógicos con un nodo inicio y un nodo final.	P-24
	Puede desarrollar y consumir APIs.	P-29
	Proviene de un paradigma basado en modelos.	P-38
	Simplificar la complejidad del proceso de construcción de software.	P-38
	Trabaja muy bien con Scrum.	P-40
	El IDE de desarrollo requiere ser instalado localmente.	P-41
	Tiene una capa gratuita.	P-48
	Integración a servicios externos.	P-44
Mendix	Cuenta con funciones de arrastrar y soltar.	P-13, P-19, P-44
	Integración con fuentes de datos.	P-13, P-44
	Integración a servicios externos.	P-13, P-44
	El IDE Web es Mendix Studio para desarrolladores ciudadanos.	P-19, P-28
	El IDE Desktop es Mendix Studio Pro para desarrolladores experimentados y puede extender su funcionalidad con Java y JavaScript.	P-19, P-28
	Provee objeto de datos, flujo de secuencia, divisiones, conectores.	P-03
	Sigue en la página siguiente.	

	Ofrece dos modos de desarrollo: desarrollo de sitios web, que es simple y rápido, desarrollo de clientes, que es completo y complejo.	P-05
	Puede interactuar con Siemens IoT.	P-05
	Ofrece programación como bucles de rama.	P-05
	Presenta un componente de modelado de procesos patentado para definir las transiciones entre las páginas de la interfaz de usuario.	P-13
	Presenta funcionalidades para definir roles y derechos.	P-13
	Ofrece 2 entornos de desarrollo sincronizados de forma bidimensional.	P-19
	Cuando la aplicación esté lista para publicarse, se puede iniciar en el entorno de la nube o reproducirla localmente.	P-19
	Cubre todas las fases del desarrollo de software.	P-19
	La conexión a Email se realiza mediante un servidor SMTP.	P-19
	Tiene un asistente con inteligencia artificial.	P-28
	Arquitectura flexible y abierta que combina colaboración, velocidad y control.	P-28
	Funciones de aprendizaje automático para el desarrollo.	P-28
	Ofrece retroalimentación de errores.	P-28
	Permite automatizar procesos.	P-45
	Contiene Widgets prediseñados.	P-44
	Plataforma como servicio.	P-50
	Elementos gráficos, lenguajes y programación visual.	P-03
Microsoft Power Apps	Se integra con otros productos de Microsoft como Microsoft Azure, MS Teams, MS Excel y KissFlow.	P-28, P-41, P-44, P-49
	Cuenta con funciones de arrastrar y soltar.	P-13, P-15, P-19
	Integración a servicios externos.	P-13, P-44
	Basada en la nube que ofrece un editor de aplicaciones, una plataforma de datos y conectores para extensión.	P-19, P-49
	Integración con fuentes de datos.	P-13, P-44
	Permite la creación de flujos de trabajo.	P-09
	Permite la creación de tareas sobre formulario de solicitud.	P-09
	Sigue en la página siguiente.	

	Presenta funcionalidades para definir roles y derechos.	P-13
	Tiene un diseñador GUID con una paleta de Widgets predefinidos.	P-13
	Tiene similitudes con Microsoft Excel.	P-15
	Ofrece un desarrollo rápido y con poca codificación.	P-19
	Crear aplicaciones desde cero.	P-19
	Crear aplicaciones a partir de modelos.	P-19
	Categorías de los componentes: 1. Datos, 2. UI, 3. Lógica, 4. Visualización.	P-19
	Centrado en pequeñas aplicaciones que apuntan a flujos de trabajo.	P-28
	Su lenguaje <i>Low-Code</i> es PowerFx.	P-37
	Pantallas y plantillas pre diseñadas.	P-44
Appian	Cuenta con funciones de arrastrar y soltar.	P-13, P-44
	Elementos gráficos, lenguajes y programación visual.	P-28, P-41
	Integración con servicios externos.	P-41, P-44
	Presenta funcionalidades para definir roles y derechos.	P-13
	Permiten el acceso a base de datos.	P-13
	Tiene un diseñador GUID con una paleta de Widgets predefinidos.	P-13
	Permite integraciones por API.	P-13
	Se centra en los procesos complejos de negocios y aplicaciones que necesitan un alto nivel de automatización y análisis.	P-28
	Usa aprendizaje automático para brindar al usuario algunas recomendaciones sobre el siguiente paso a seguir en el proceso de desarrollo.	P-28
	Lenguaje de programación personalizado, componentes y objetos del Appian Designer.	P-41
	Envío de correos mediante la conexión a un servidor SMTP.	P-41
	Integración con fuentes de datos.	P-44
	Plataforma como servicio.	P-50
BalticLSC	Plataforma <i>Low-Code</i> para computación de alto rendimiento y cálculos avanzados.	P-02, P-33
	Facilita el procesamiento por lote.	P-02
	El término es Baltic Large Scale Computing.	P-33
	Sigue en la página siguiente.	

	Basa en un lenguaje visual llamado Computation Application Language.	P-33
ECUST	Plataforma para el diseño de simulaciones y análisis de sistemas de energía.	P-10
	Es una herramienta de modelado y cálculo para sistemas de energía eléctrica modernos.	P-10
	Elementos gráficos, lenguajes y programación visual.	P-10
	Equipada con una interfaz front-end para la visualización de procesos de simulación y análisis.	P-10
	Equipada con una interfaz back-end para facilitar el desarrollo de aplicaciones en términos de un entorno visual fácil de usar.	P-10
	Cuenta con funciones de arrastrar y soltar.	P-10
	Bibliotecas de componentes integradas.	P-10
	Puede acceder a fuentes de datos externas.	P-10
Wavemaker	Utilizan menús básicos y Scripts de captura de eventos.	P-13
	Permiten el acceso a base de datos.	P-13
	Tiene un diseñador GUID con una paleta de Widgets predefinidos.	P-13
	Cuenta con funciones de arrastrar y soltar.	P-13
	Permite integraciones por API.	P-13
Pega	Presenta funcionalidades para definir roles y derechos.	P-13
	Permiten el acceso a base de datos.	P-13
	Tiene un diseñador GUID con una paleta de Widgets predefinidos.	P-13
	Cuenta con funciones de arrastrar y soltar.	P-13
	Permite integraciones por API.	P-13
	Fácil integración con almacenes de datos y servicios.	P-29
Quickbase	Plataforma básica enfocada en la gestión de datos.	P-13
	Utilizan menús básicos y Scripts de captura de eventos.	P-13
	Presenta funcionalidades para definir roles y derechos.	P-13
	Permiten el acceso a base de datos.	P-13
	Tiene un diseñador GUID con una paleta de Widgets predefinidos.	P-13
	Cuenta con funciones de arrastrar y soltar.	P-13
	Sigue en la página siguiente.	

	Permite integraciones por API.	P-13
Bonita Platform	Es un sistema de gestión de flujo de trabajo.	P-13
	Presenta funcionalidades para definir roles y derechos.	P-13
	Permiten el acceso a base de datos.	P-13
	Tiene un diseñador GUID con una paleta de Widgets predefinidos.	P-13
	Cuenta con funciones de arrastrar y soltar.	P-13
	Permite integraciones por API.	P-13
Xatkit	Proporciona un conjunto de DSL (lenguajes específicos de dominio).	P-15
Ktrain	Permite crear modelos de aprendizaje automático.	P-15
	Elementos gráficos, lenguajes y programación visual.	P-15
	Ofrece un desarrollo rápido y con poca codificación.	P-15
RESTsec	Ofrece un desarrollo rápido y con poca codificación.	P-15
	Respalda los rápidos requisitos de seguridad que las empresas necesitan para proteger sus datos cuando utilicen un modelo de negocio en línea.	P-15
DreamWeaver	Elementos gráficos, lenguajes y programación visual.	P-15
	No requiere escribir ningún código.	P-15
SETXRM	Viene siendo utilizada por empresas públicas y corporativas.	P-16
	Permiten el acceso a base de datos.	P-16
	Puede integrar código nativo.	P-16
	Múltiples proyectos en un solo servidor sin costes.	P-16
Aurea BPM	Permite preparar una aplicación en diferentes niveles de complejidad.	P-21
	Provee una arquitectura combinada de tiempo de ejecución y modelado.	P-21
	Convierte diagramas de procesos en aplicaciones web funcionales.	P-21
	Es una herramienta que brinda un amplio soporte para el modelado, automatización, gestión y optimización de procesos de negocio.	P-21
	Sigue en la página siguiente.	

	Funciona en entornos Microsoft Windows, Linux, Unix con cualquier servidor de aplicaciones y servidor Web y utiliza el sistema de gestión de bases de datos Oracle.	P-21
SAP	Elementos gráficos, lenguajes y programación visual.	P-24
	Ofrece un desarrollo rápido y con poca codificación.	P-24
Celonis Studio	Permite crear, implementar y mantener aplicaciones de minería de procesos operativos y analíticos en un espacio centralizado.	P-26
	Provee plantillas de interfaces.	P-26
	Provee mecanismos de extensión.	P-26
	Provee control de versiones.	P-26
	Permite desencadenar acciones mediante lógica condicional y más de 100 automatizadores pre-diseñados.	P-26
Betty Blocks	Permite desarrollar software a desarrolladores ciudadanos y a expertos.	P-29
	Puede consumir pero no desarrollar APIs.	P-29
Salesforce Lightning	Conección con componentes de inteligencia artificial.	P-35
	Conección con tecnología IoT.	P-35
	Ofrece un desarrollo rápido y con poca codificación.	P-35
	Plataforma como servicio.	P-50
Caspio	Basada en la nube.	P-35
	Elementos gráficos, lenguajes y programación visual.	P-35
	Fácil integración con almacenes de datos y servicios.	P-35
HAWE e Design	Programar componentes hidráulicos.	P-36
Simulink	Elementos gráficos, lenguajes y programación visual.	P-36
Zoho	Su lenguaje <i>Low-Code</i> es Deluge.	P-37
	Cuenta con funciones de arrastrar y soltar.	P-44
	Integración a servicios externos.	P-44
	Integración con fuentes de datos.	P-44
Netgrif	Su lenguaje <i>Low-Code</i> es Petriflow.	P-37
Google App Maker	Permite crear y publicar aplicaciones impulsada por la plataforma G Suite. Sigue en la página siguiente.	P-44

		Basado en la nube.	P-44
		Cuenta con funciones de arrastrar y soltar.	P-44
		Utiliza y extiende sus funcionalidades con HTML, Javascript y CSS.	P-44
		Integración con fuentes de datos.	P-44
Kissflow		Basado en la nube.	P-44
		Plataforma de automatización de flujo de trabajo.	P-44
		Consumo de APIs.	P-44
		Integración a servicios externos.	P-44
		Integración con fuentes de datos.	P-44
Salesforce App Cloud		Basado en la nube.	P-44
		funciones de arrastrar y soltar.	P-44
		Plataforma para la automatización de procesos.	P-44
		Integración a servicios externos.	P-44
		Integración con fuentes de datos.	P-44

Tabla 4.10: Características por plataformas

- **PI-6** ¿Cuáles son las ventajas y desventajas que se tienen por cada una de las plataformas *Low-Code* que se vienen utilizando?

la Tabla 4.11 especifica las ventajas por plataforma y la Tabla 4.12 precisa las desventajas por plataforma.

Plataforma	Ventaja	Ref.
OUTSYSTEMS	Mantenimiento más sencillo.	P-11, P-20
	Facilitar, acelerar y optimizar procesos a través de la interacción visual.	P-14, P-42
	Puede mapear modelos de datos relacionales y la gran mayoría de modelo de datos NoSQL con una conversión mínima o una sobrecarga conceptual.	P-05
	Privacidad.	P-11
	Reducción de complejidad.	P-11
	Facilita el proceso de consulta de datos.	P-14
	Provee Widgets listos para usar y se pueden personalizar con CSS y JavaScript.	P-16
	Tiene bibliotecas Live Style Guides.	P-16
	Es posible crear automáticamente las páginas del listado y detalle arrastrando y soltando la entidad de base de datos.	P-19

Sigue en la página siguiente.

	Reducción de la necesidad de escribir código.	P-20
	Reutilizar fragmentos de código desarrollado por expertos.	P-25
	Permitela escalabilidad y seguridad.	P-28
	Utiliza IA para proporcionar recomendaciones, automatización y validación de la aplicación desarrollada.	P-28
	No requiere amplias habilidades de programación.	P-38
	Alta disponibilidad.	P-38
	Escalabilidad de número de usuarios.	P-44
	Escalabilidad de tráfico de datos.	P-44
	Escalabilidad de almacen de datos.	P-44
Mendix	Rápido desarrollo de aplicaciones.	P-28, P-36
	Reducción de complejidad.	P-36
	Escalabilidad de número de usuarios.	P-44
	Escalabilidad de tráfico de datos.	P-44
	Escalabilidad de almacen de datos.	P-44
	Tiene una capa gratuita.	P-44
Microsoft Power Apps	Para departamentos u organizaciones pequeñas y medianas que tienen presupuestos y desarrolladores de software limitados.	P-09
	Tasas de satisfacción superiores a la media.	P-15
	Escalabilidad de número de usuarios.	P-44
	Escalabilidad de tráfico de datos.	P-44
	Escalabilidad de almacen de datos.	P-44
Appian	Mejora la productividad de los profesionales de las TIC.	P-28
	Escalabilidad de número de usuarios.	P-44
	Escalabilidad de almacen de datos.	P-44
Node-RED	Se puede ejecutar en software contenedores, dando como resultado una solución portátil que se puede utilizar en cualquier máquina independientemente de aspectos como configuración o sistema operativo.	P-04
	Se puede ejecutar en la mayoría de los dispositivos livianos por lo que cualquier persona o entidad sin necesidad de muchos recursos puede utilizar nuestra propuesta.	P-04

Sigue en la página siguiente.

	Permite desarrollar colaborativamente aplicaciones basadas en eventos.	P-04
	Permite una fácil importación y exportación de los flujos creados.	P-04
Xatkit	Fácil acceso.	P-15
	Mayor eficiencia.	P-15
	Mejor gestión del tiempo de los empleados.	P-15
Ktrain	Contiene una biblioteca Phyton.	P-15
Salesforce Lightning Platform	Cuenta con el respaldo de cinco mil aplicaciones de terceros.	P-15
	Puede integrarse con una gran cantidad de fuentes para mejorar la escalabilidad de aplicaciones del cliente.	P-15
	Desarrollo rápido de aplicaciones a costos bajos.	P-28
	Faciles personalizaciones de colores y branding.	P-28
	Reutilización de componentes.	P-28
Aurea BPM	Puede reducir significativamente el costo y el tiempo de implementación, desarrollo y mantenimiento de procesos.	P-21
LegoNXT-G	Tiempo de desarrollo más corto.	P-36
	Reducción de complejidad.	P-36
Salesforce App Cloud	Escalabilidad de número de usuarios.	P-44
	Escalabilidad de tráfico de datos.	P-44
	Escalabilidad de almacen de datos.	P-44
KissFlow	Escalabilidad de número de usuarios.	P-44
	Escalabilidad de tráfico de datos.	P-44
	Escalabilidad de almacen de datos.	P-44
Google App Maker	Escalabilidad de número de usuarios.	P-44
	Escalabilidad de tráfico de datos.	P-44
	Escalabilidad de almacen de datos.	P-44
Zoho Creator	Escalabilidad de número de usuarios.	P-44
	Escalabilidad de tráfico de datos.	P-44
	Escalabilidad de almacen de datos.	P-44

Tabla 4.11: *Ventajas por plataformas*

Plataforma	Desventaja	Ref.
OUTSYSTEMS	Licencias muy costosas para pequeñas y medianas empresas.	P-05
	Las interfaces para consultas de datos tienen problemas de usabilidad para usuarios menos experimentados.	P-14
	No existe un mecanismo de trazabilidad de código.	P-22
	No existe una solución para la alternancia de funciones.	P-22
	No existe un soporte nativo para pruebas unitarias.	P-31
	No es un lenguaje orientado a objetos.	P-31
	Crear o personalizar Widgets puede ser una tarea complicada a partir de entradas de diseño.	P-42
Mendix	Para conectarse a diferentes fuentes de datos requiere módulos de integración por separados y para desarrolladores no expertos no es fácil.	P-03
	Limitación de personalización de componentes prediseñados.	P-19
Microsoft Power Apps	No recomendada para aplicaciones complejas que requieren mayor personalización.	P-09
	Microsoft requiere suscripciones de pago para utilizar sus servicios y herramientas.	P-15
	Los componentes predefinidos no permiten tanta libertad de diseño y personalización.	P-19
	La documentación no provee información de buenas prácticas sobre integración de Power Apps.	P-41
	Plataforma no es tan <i>Low-Code</i> si no como menos código.	P-41
	No trabaja offline.	P-44
BalticLSC	BalticLSC no tiene un entorno de pruebas para desarrolladores.	P-02
ECUST	Solo admite operaciones fuera de línea.	P-10
	Solo funciona como software desktop.	P-10
Xatkit	Mayores costos de desarrollo y mantenimiento.	P-15

Sigue en la página siguiente.

Salesforce Lightning Platform	La empresa esta diversificando e integrando su cartera lo que complica el nivel de innovación y evolución.	P-15
	Está en proceso de crecimiento y le faltan productos eficientes.	P-15
Aurea BPM	Pocos desarrolladores.	P-21
Zoho Creator	No trabaja offline.	P-44
Google App Maker	No tiene interoperabilidad con servicios externos.	P-44

Tabla 4.12: Desventajas por plataformas

- **PI-7 ¿Cuáles son los tipos de aplicaciones por cada una de las plataformas de desarrollo *Low-Code*?**

La Tabla 4.13 expone que la gran mayoría de plataformas permiten el desarrollo de aplicaciones web y móvil, mientras que algunas pocas permiten el desarrollo de aplicaciones de escritorio, programas de procesamiento de datos, programas de aprendizaje automático, chatbot y robot de voz. La Figura 4.3 muestra gráficamente los principales tipos de aplicaciones por plataforma.

Plataforma	Tipo de aplicación	Menciones	Ref.
OutSystems	Web	18	P-03, P-06, P-11, P-15, P-16, P-19, P-20, P-25, P-27, P-28, P-31, P-39, P-40, P-41, P-43, P-44, P-46, P-48
	Móvil	18	P-03, P-05, P-06, P-15, P-16, P-19, P-20, P-25, P-27, P-28, P-31, P-39, P-40, P-41, P-43, P-44, P-46, P-48
Mendix	Web	3	P-19, P-44, P-45
	Móvil	2	P-19, P-45
Microsoft Power Apps	Web	2	P-10, P-19
	Móvil	2	P-19, P-50
Appian	Web	2	P-28, P-44
	Móvil	2	P-28, P-44
ThinkWise	Escritorio	1	P-03
	Web	1	P-03

Sigue en la página siguiente.

	Móvil	1	P-03
BalticLSC	Web	1	P-02
	Programas para el procesamiento de datos	1	P-33
Amazon Honeycode	Web	1	P-03
	Móvil	1	P-03
Node-RED	Web	1	P-04
ECUST	Escritorio	1	P-10
Ktrain	Programas de aprendizaje automático	1	P-15
Xatkit	Chatbot	1	P-15
	Robots de voz	1	P-15
SETXRM	Web	1	P-18
	Móvil	1	P-18
Aurea BPM	Web	1	P-21
Salesforce Lightning	Móvil	1	P-35
Capio	Web	1	P-35
Zoho Creator	Web	1	P-44
	Móvil	1	P-44
	Escritorio	1	P-44

Tabla 4.13: Tipos de aplicaciones por plataforma

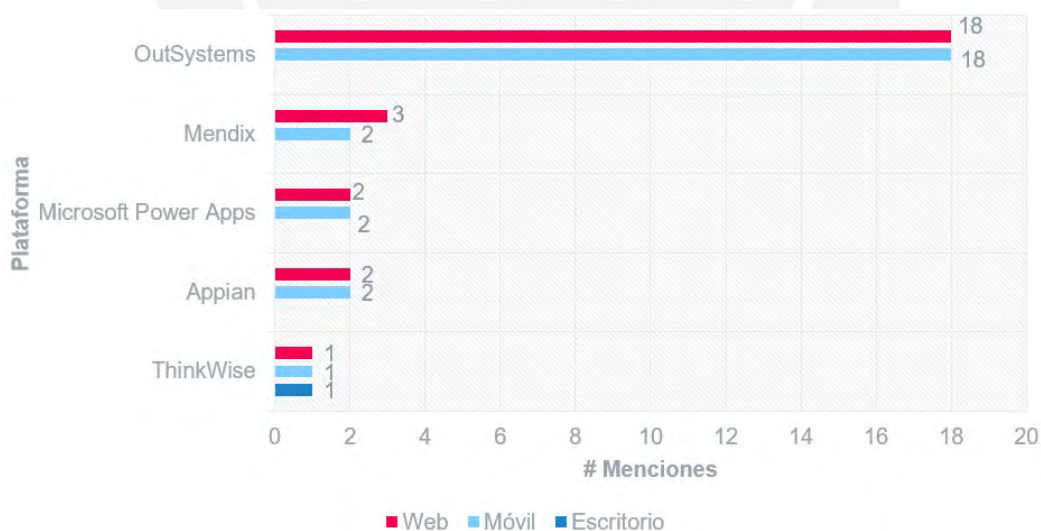


Figura 4.3: Principales tipos de aplicaciones por plataforma

- **PI-8 ¿Cuáles son las industrias donde se utilizan plataformas de desarrollo *Low-Code*?**

En la Figura 4.4 presenta las principales industrias de aplicación de las Plataformas de desarrollo *Low-Code*. Para los autores [30] [40] [47] [49] las Plataformas *Low-Code* se

utilizan en la industria de la Salud con cuatro menciones dentro de nuestra población de investigaciones primarias; por otro lado, para los autores [27] [40] [56] son aplicadas en la industria del comercio electrónico con tres menciones, para el autor [59] son utilizadas en el Gobierno e Ingeniería Industrial. La Tabla 4.14 aclara este panorama brindando una mejor referencia a detalle.

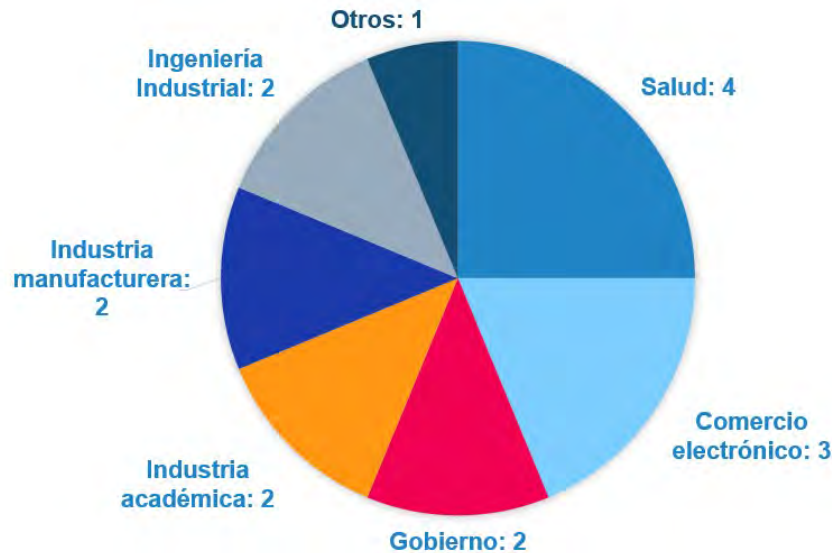


Figura 4.4: Principales industrias de aplicación

Industrias	Menciones	Ref.
Salud	4	P-18, P-28, P-35, P-37
Comercio electrónico	3	P-15, P-28, P-44
Gobierno	2	P-35, P-47
Industria académica	2	P-40, P-41
Industria manufacturera	2	P-17, P-18
Ingeniería industrial	2	P-47, P-49
Gestión organizacional	1	P-28
Administraciones financieras	1	P-35
Automatización de procesos	1	P-37
Seguros	1	P-37
Arrendamiento	1	P-37
Servicios públicos	1	P-37
CRM	1	P-44
ERP	1	P-44
Inteligencia Empresarial	1	P-44
Deporte	1	P-48

Tabla 4.14: Investigaciones clasificadas por industrias

- **PB-9** ¿Cuáles son los tipos de aplicaciones por industrias donde se utilizan

plataformas de desarrollo *Low-Code*?

La tabla 4.15 muestra los tipos de aplicación por industria que explícitamente se detallaron en los artículos primarios. De las industrias identificadas, 20 casos no mencionan los tipos de aplicaciones.

Tipos de aplicación	Industrias	Menciones	Ref.
Web	Salud	1	P-04
Chatbots	Comercio electrónico	1	P-15
Robots de voz	Comercio electrónico	1	P-15
Web	Gobierno	1	P-46
Móvil	Gobierno	1	P-46

Tabla 4.15: *Tipos de aplicaciones por industria*



CAPÍTULO 5

Amenazas a la Validez

Para el autor [64] aquellas amenazas que puedan afectar a los resultados de tesis deben ser discutidas. Se han identificado 4 posibles amenazas a la validez.

5.1. Validez del Constructo

Las cadenas de búsqueda elaboradas en este estudio, que permiten recabar información a fin de responder las preguntas de investigación, son los principales constructos a considerar. Estas cadenas de búsqueda se han definido con la ayuda de los criterios PICO, han sido ejecutadas sobre las bases de datos más relevantes seleccionadas previamente y existe una posibilidad de que algunos estudios relevantes no fueron considerados debido a que podrían haber estado indexados a fuentes de datos no elegidas o en casos donde no incluyeran los términos elegidos.

5.2. Validez Interna

Es posible la introducción de sesgo al extraer y analizar los datos del estudio. Para aliviar este riesgo se han incluido actividades de validación de la calidad.

5.3. Validez Externa

Está asociada a la capacidad de poder generalizar los resultados obtenidos en el presente estudio; para mitigar este riesgo, el proceso de búsqueda es efectuado varias veces.

5.4. Validez de las Conclusiones

Es posible que se hayan excluido investigaciones erróneamente para este mapeo sistemático; a fin de disminuir el riesgo se han detallado cuidadosamente criterios de selección, exclusión y validación de la calidad con el fin de evitar la exclusión de estudios relevantes.

CAPÍTULO 6

Conclusiones y trabajos futuros

En esta sección se analizarán los resultados del mapeo y se detallan opciones de trabajos futuros para la continuidad del proyecto.

6.1. Conclusiones

El presente estudio realizó un mapeo sistemático que tenía como objetivo evaluar distintas investigaciones respecto a la implementación de software usando plataformas de desarrollo *Low-Code*. Se elaboraron 4 preguntas bibliométricas y 9 preguntas de investigación que fueron respondidas sobre la base de los datos recabados en 47 artículos relevantes identificados.

De los 47 estudios seleccionados, 14 fueron publicados en Estados Unidos; se aprecia que las primeras publicaciones son del año 2018 y 2019, la mayor cantidad de publicaciones son 17 en el 2021 y 2022 y se nota así una tendencia al crecimiento, 38 investigaciones son Conference Paper; 12 artículos fueron publicados en “ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion” siendo el principal medio de publicación.

El estudio identificó Plataformas de desarrollo *Low-Code* como OutSystems, Mendix, Microsoft Power Apps y Appian como las más comunes. El estudio permitió identificar las características de las Plataformas como la utilización de una programación visual; además ofrecen un desarrollo rápido y con poca codificación, esto porque cuentan con funciones de arrastrar y soltar y simplifica la complejidad del proceso de construcción de software.

En relación a las ventajas y desventajas identificadas sobre el uso de las Plataformas *Low-Code* se concluye que acelera el desarrollo y la entrega de software, facilita la innovación y la transformación digital, permiten una mayor productividad y aseguran una adecuada escalabilidad. Sin embargo, los precios de licenciamiento son altos, la información sobre LCDP es insuficiente, baja demanda de desarrolladores calificados y falta de difusión de casos de éxito. Así mismo se han identificado las ventajas y desventajas del uso de las LCDP como por cada una de ellas.

Las áreas de la organización identificadas donde se utilizan plataformas de desarrollo *Low-Code* son en su mayoría el Área de Compras y el Área de Ventas. Así mismo los tipos de aplicaciones de software son aplicaciones web y aplicaciones móvil inevitablemente por el uso y la tendencia actual.

El estudio demuestra que la industria de la salud y el comercio electrónico son las más mencionadas en las investigaciones evaluadas. Se concluye también que las aplicaciones web y móviles son las más utilizadas en las industrias de la salud, comercio electrónico, recursos humanos y gobierno.

6.2. Trabajos futuros

Como parte de trabajos futuros, se pueden realizar comparaciones entre las Plataformas de desarrollo *Low-Code* encontradas en el mapeo para ilustrar las diferencias y semejanzas. Como por ejemplo: Outsystems, Mendix, Microsoft Power Apps y Appian cuentan con funciones de arrastrar y soltar e implementan software web y móvil.

Otra sugerencia de trabajo futuro en base a comparaciones entre plataformas es la recomendación de plataformas para entornos específicos, tal como en el comercio electrónico, los autores [27, 40, 56] mencionan que Outsystems, Mendix y Appian permiten la implementación de software web y móvil en esta industria.



Bibliografía

- [1] N. Prinz, C. Rentrop, and M. Huber, “Low-code development platforms-a literature review.,” in *AMCIS*, 2021.
- [2] K. Rokis and M. Kirikova, “Challenges of low-code/no-code software development: A literature review,” in *Perspectives in Business Informatics Research: 21st International Conference on Business Informatics Research, BIR 2022, Rostock, Germany, September 21–23, 2022, Proceedings*, pp. 3–17, Springer, 2022.
- [3] D. Pinho, A. Aguiar, and V. Amaral, “What about the usability in low-code platforms? a systematic literature review,” *Journal of Computer Languages*, p. 101185, 2022.
- [4] OutSystems, “Low-code platforms.” <https://www.outsystems.com/guide/low-code/>, Abril 2023.
- [5] Gartner, “Gartner forecasts worldwide low-code development technologies market to grow 20 % in 2023.” <https://www.gartner.com/en/newsroom/press-releases/2022-12-13-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-20-percent-in-2023>, Abril 2023.
- [6] Microsoft, “Desarrollo de aplicaciones sin código o con poco código.” <https://powerapps.microsoft.com/es-es/low-code-no-code-development-platforms/>, Abril 2023.
- [7] K. Petersen, S. Vakkalanka, and L. Kuzniarz, “Guidelines for conducting systematic mapping studies in software engineering: An update,” *Information and software technology*, vol. 64, pp. 1–18, 2015.
- [8] A. Bucaioni, A. Cicchetti, and F. Ciccozzi, “Modelling in low-code development: a multi-vocal systematic review,” *Software and Systems Modeling*, vol. 21, no. 5, pp. 1959–1981, 2022.
- [9] R. Waszkowski, “Low-code platform for automating business processes in manufacturing,” *IFAC-PapersOnLine*, vol. 52, no. 10, pp. 376–381, 2019.
- [10] P. M. Gomes and M. A. Brito, “Low-code development platforms: A descriptive study,” in *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–4, IEEE, 2022.
- [11] Y. Luo, P. Liang, C. Wang, M. Shahin, and J. Zhan, “Characteristics and challenges of low-code development: the practitioners’ perspective,” in *Proceedings of the 15th*

-
- ACM/IEEE international symposium on empirical software engineering and measurement (ESEM)*, pp. 1–11, 2021.
- [12] C. Lockwood, Z. Munn, and K. Porritt, “Qualitative research synthesis: methodological guidance for systematic reviewers utilizing meta-aggregation,” *JBIEvidence Implementation*, vol. 13, no. 3, pp. 179–187, 2015.
- [13] H. Lourenço, C. Ferreira, J. Costa Seco, and J. Parreira, “Ostrich: a rich template language for low-code development (extended version),” *Software and Systems Modeling*, pp. 1–19, 2022.
- [14] K. Marek, M. Śmiałek, K. Rybiński, R. Roszczyk, and M. Wdowiak, “BalticLSC: low-code software development platform for large scale computations,” *Computing and Informatics*, vol. 40, no. 4, pp. 734–753, 2021.
- [15] A. Sahay, D. Di Ruscio, L. Iovino, and A. Pierantonio, “Analyzing business process management capabilities of low-code development platforms,” *Software: Practice and Experience*, vol. 53, no. 4, pp. 1036–1060, 2023.
- [16] J. Rosa-Bilbao, J. Boubeta-Puig, and A. Rutle, “Edaloco: Enhancing the accessibility of blockchains through a low-code approach to the development of event-driven applications for smart contract management,” *Computer Standards & Interfaces*, vol. 84, p. 103676, 2023.
- [17] J. Wang, B. Qi, W. Zhang, and H. Sun, “A low-code development framework for constructing industrial apps,” in *Computer Supported Cooperative Work and Social Computing: 15th CCF Conference, Chinese CSCW 2020, Shenzhen, China, November 7–9, 2020, Revised Selected Papers 15*, pp. 237–250, Springer, 2021.
- [18] H. Henriques, H. Lourenço, V. Amaral, and M. Goulão, “Improving the developer experience with a low-code process modelling language,” in *Proceedings of the 21th acm/ieee international conference on model driven engineering languages and systems*, pp. 200–210, 2018.
- [19] U. Rafiq, C. Filippo, and X. Wang, “Understanding low-code or no-code adoption in software startups: Preliminary results from a comparative case study,” in *International Conference on Product-Focused Software Process Improvement*, pp. 390–398, Springer, 2022.
- [20] A. Nunes Alonso, J. Abreu, D. Nunes, A. Vieira, L. Santos, T. Soares, and J. Pereira, “Building a polyglot data access layer for a low-code application development platform: (experience report),” in *IFIP International Conference on Distributed Applications and Interoperable Systems*, pp. 95–103, Springer, 2020.
- [21] I. Rassameeroj, P. Jomkhamsri, and N. Thaitawee Wattana, “Student request system prototype using low-code development platform,” in *2022 International Conference on Algorithms, Data Mining, and Information Technology (ADMIT)*, pp. 190–194, IEEE, 2022.

- [22] X. Zhong, X. Zhou, Y. Xia, G. Bai, J. Dong, and D. Xue, “A gui-based low-code development platform for power systems analysis,” in *2022 China Automation Congress (CAC)*, pp. 6080–6085, IEEE, 2022.
- [23] A. Trigo, J. Varajão, and M. Almeida, “Low-code versus code-based software development: Which wins the productivity game?,” *IT Professional*, vol. 24, no. 5, pp. 61–68, 2022.
- [24] C. Di Sipio, D. Di Ruscio, and P. T. Nguyen, “Democratizing the development of recommender systems by means of low-code platforms,” in *Proceedings of the 23rd ACM/IEEE international conference on model driven engineering languages and systems: companion proceedings*, pp. 1–9, 2020.
- [25] A. C. Bock and U. Frank, “In search of the essence of low-code: an exploratory study of seven development platforms,” in *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 57–66, IEEE, 2021.
- [26] P. S. Rodrigues, *Accelerating SQL with Complex Visual Querying*. PhD thesis, 2021.
- [27] R. Benac and T. K. Mohd, “Recent trends in software development: Low-code solutions,” in *Proceedings of the Future Technologies Conference (FTC) 2021, Volume 3*, pp. 525–533, Springer, 2022.
- [28] M. Bexiga, S. Garbatov, and J. C. Seco, “Closing the gap between designers and developers in a low code ecosystem,” in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 1–10, 2020.
- [29] P. M. Gomes and M. A. Brito, “Low-code development platforms: a descriptive study,” in *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–4, IEEE, 2022.
- [30] E. Sahinaslan, O. Sahinaslan, and M. Sabancioglu, “Low-code application platform in meeting increasing software demands quickly: Setxrm,” in *AIP Conference Proceedings*, vol. 2334, AIP Publishing, 2021.
- [31] F. Gürcan and G. Taentzer, “Using microsoft powerapps, mendix and outsystems in two development scenarios: an experience report,” in *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 67–72, IEEE, 2021.
- [32] R. Martins, F. Caldeira, F. Sa, M. Abbasi, and P. Martins, “An overview on how to develop a low-code application using outsystems,” in *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, pp. 395–401, IEEE, 2020.
- [33] R. Waszkowski, “Low-code platform for automating business processes in manufacturing,” *IFAC-PapersOnLine*, vol. 52, no. 10, pp. 376–381, 2019.

-
- [34] B. Lopes, S. Amorim, and C. Ferreira, “Solution discovery over feature toggling with built-in abstraction in outsystems,” in *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 47–56, IEEE, 2021.
- [35] A. Bragança, I. Azevedo, N. Bettencourt, C. Morais, D. Teixeira, and D. Caetano, “Towards supporting spl engineering in low-code platforms using a dsl approach,” in *Proceedings of the 20th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences*, pp. 16–28, 2021.
- [36] I. P. Fernandes, M. Terra-Neves, and J. C. Seco, “Automated refactoring of unbounded queries in software automation platforms,” in *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 417–426, IEEE, 2021.
- [37] H. Lourenço, C. Ferreira, and J. C. Seco, “Ostrich-a type-safe template language for low-code development,” in *2021 ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pp. 216–226, IEEE, 2021.
- [38] C. Ullrich, T. Lata, and J. Geyer-Klingenberg, “Celonis studio-a low-code development platform for citizen developers,” in *BPM (PhD/Demos)*, pp. 102–105, 2021.
- [39] L. Carvalho, “A dialogue interface for low code program evolution,” in *International Conference on Integrated Formal Methods*, pp. 357–360, Springer, 2022.
- [40] H. A. ALSAADI, D. T. RADAIN, M. M. ALZHRANI, W. F. ALSHAMMARI, D. ALAHMADI, and B. FAKIEH, “Factors that affect the utilization of low-code development platforms: survey study,” *Romanian Journal of Information Technology & Automatic Control/Revista Română de Informatică și Automatică*, vol. 31, no. 3, 2021.
- [41] M. Overeem, S. Jansen, and M. Mathijssen, “Api management maturity of low-code development platforms,” in *International Conference on Business Process Modeling, Development and Support*, pp. 380–394, Springer, 2021.
- [42] I. Weber, “Low-code from frontend to backend: Connecting conversational user interfaces to backend services via a low-code iot platform,” in *Proceedings of the 3rd Conference on Conversational User Interfaces*, pp. 1–5, 2021.
- [43] A. Jacinto, M. Lourenço, and C. Ferreira, “Test mocks for low-code applications built with outsystems,” in *Proceedings of the 23rd ACM/IEEE international conference on model driven engineering languages and systems: companion proceedings*, pp. 1–5, 2020.
- [44] E. Elshan, P. Ebel, M. Söllner, and J. M. Leimeister, “Leveraging low code development of smart personal assistants: an integrated design approach with the spade method,” *Journal of Management Information Systems*, vol. 40, no. 1, pp. 96–129, 2023.

- [45] R. Roszczyk, M. Wdowiak, M. Śmiałek, K. Rybiński, and K. Marek, “Balticisc: A low-code hpc platform for small and medium research teams,” in *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 1–4, IEEE, 2021.
- [46] F. Rani, P. Diez, E. Chavarriaga, E. Guerra, and J. de Lara, “Automated migration of eugenia graphical editors to the web,” in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pp. 1–7, 2020.
- [47] N. Krishnaraj, R. Vidhya, R. Shankar, and N. Shruthi, “Comparative study on various low code business process management platforms,” in *2022 International Conference on Inventive Computation Technologies (ICICT)*, pp. 591–596, IEEE, 2022.
- [48] E.-M. Neumann, B. Vogel-Heuser, F. Haben, M. Krüger, and T. Wieringa, “Introduction of an assistance system to support domain experts in programming low-code to leverage industry 5.0,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10422–10429, 2022.
- [49] G. Juhás, L. Molnár, A. Juhásová, M. Ondrišová, M. Mladoniczky, and T. Kováčik, “Low-code platforms and languages: the future of software development,” in *2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pp. 286–293, IEEE, 2022.
- [50] J. Ramalho, H. Lourenço, and J. C. Seco, “From builders to editors: bidirectional transformations of low-code models,” in *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 108–117, IEEE, 2021.
- [51] A. Ostroukh, N. Kuftinova, V. Gaevskii, N. Filippova, and E. Subachev, “Digital transformation of enterprises using a low-code platform,” *Russian Engineering Research*, vol. 42, no. 11, pp. 1203–1206, 2022.
- [52] J. Metrôlho, F. Ribeiro, P. Graça, A. Mourato, D. Figueiredo, and H. Vilarinho, “Aligning software engineering teaching strategies and practices with industrial needs,” *Computation*, vol. 10, no. 8, p. 129, 2022.
- [53] R. Lichtenthäler, S. Böhm, J. Manner, and S. Winzinger, “A use case-based investigation of low-code development platforms,” in *ZEUS*, pp. 76–83, 2022.
- [54] J. Parreira, “Simple dependent types for ostrich,” in *International Conference on Integrated Formal Methods*, pp. 361–364, Springer, 2022.
- [55] J. Pacheco, S. Garbatov, and M. Goulão, “Improving collaboration efficiency between ux/ui designers and developers in a low-code platform,” in *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 138–147, IEEE, 2021.

-
- [56] A. Sahay, A. Indamutsa, D. Di Ruscio, and A. Pierantonio, “Supporting the understanding and comparison of low-code development platforms,” in *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 171–178, IEEE, 2020.
- [57] M. Op’t Land, M. R. Krouwel, and S. Gort, “Testing the concept of the run-time adaptive enterprise: Combining organization and it agnostic enterprise models with organization implementation variables and low code technology,” in *Advances in Enterprise Engineering XIV: 10th Enterprise Engineering Working Conference, EEWC 2020, Bozen-Bolzano, Italy, September 28, October 19, and November 9–10, 2020, Revised Selected Papers 10*, pp. 228–242, Springer, 2021.
- [58] A. Barbosa, N. Dâmaso, D. Pacheco, S. Nicola, and N. Bettencourt, “Adding blockchain and smart contracts to a low-code development platform,” in *International Conference on Innovations in Bio-Inspired Computing and Applications*, pp. 868–877, Springer, 2022.
- [59] B. Adrian, S. Hinrichsen, and A. Nikolenko, “App development via low-code programming as part of modern industrial engineering education,” in *Advances in Human Factors and Systems Interaction: Proceedings of the AHFE 2020 Virtual Conference on Human Factors and Systems Interaction, July 16-20, 2020, USA*, pp. 45–51, Springer, 2020.
- [60] J. Oliveira, S. Nicola, P. Graça, S. Martins, and T. Gafeira, “Application for the management of sports performance in crossfit supported by an artificial intelligence cognitive service,” in *International Conference on Innovations in Bio-Inspired Computing and Applications*, pp. 590–598, Springer, 2021.
- [61] S. Hinrichsen, B. Adrian, and A. Schulz, “Approaches to improve shop floor management,” in *Human Interaction, Emerging Technologies and Future Applications III: Proceedings of the 3rd International Conference on Human Interaction and Emerging Technologies: Future Applications (IHET 2020), August 27-29, 2020, Paris, France*, pp. 415–421, Springer, 2021.
- [62] V. S. Phalake and S. D. Joshi, “Low code development platform for digital transformation,” in *Information and Communication Technology for Competitive Strategies (ICTCS 2020) Intelligent Strategies for ICT*, pp. 689–697, Springer, 2021.
- [63] M. Zarour, A. Abran, J.-M. Desharnais, and A. Alarifi, “An investigation into the best practices for the successful design and implementation of lightweight software process assessment methods: A systematic literature review,” *Journal of Systems and Software*, vol. 101, pp. 180–192, 2015.
- [64] A. Jedlitschka, M. Ciolkowski, and D. Pfahl, “Reporting experiments in software engineering,” *Guide to advanced empirical software engineering*, pp. 201–228, 2008.