



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**  
**FACULTAD DE CIENCIAS DE LA EDUCACIÓN HUMANAS Y**  
**TECNOLOGÍAS**  
**CARRERA DE PEDAGOGÍA DE LAS CIENCIAS**  
**EXPERIMENTALES: INFORMÁTICA**

**Título**

“Tecnologías Low/No-Code para el desarrollo de aplicaciones educativas”

**Trabajo de Titulación para optar al título de: Licenciado en**  
**Pedagogía de la informática**

**Autor:**

Mullo Chimbolema Marco Rolando

**Tutor:**

Mgs. Fernández Acevedo Jorge Eduardo

**Riobamba, Ecuador. 2025**

## DECLARATORIA DE AUTORÍA

Yo, **Marco Rolando Mullo Chimbolema**, con cédula de ciudadanía 0605579457, autor del trabajo de investigación titulado: **Tecnologías Low/No-Code para el desarrollo de aplicaciones educativas**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 15 de junio de 2025



---

Marco Rolando Mullo Chimbolema

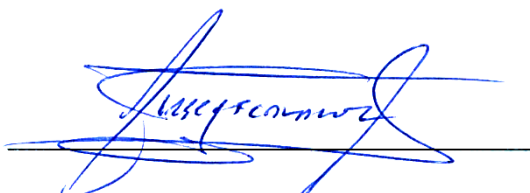
C.I: 060557945-7



## DICTAMEN FAVORABLE DEL PROFESOR TUTOR

Quien suscribe, Mg. Fernández Acevedo Jorge Eduardo catedrático adscrito a la Facultad de Ciencias de la Educación Humanas y Tecnológicas, por medio del presente documento certifico haber asesorado y revisado el desarrollo del trabajo de investigación titulado: **Tecnologías Low/No-Code para el desarrollo de aplicaciones educativas**, bajo la autoría de **Marco Rolando Mullo Chimbolema CC: 0605579457**; por lo que se autoriza ejecutar los trámites legales para su sustentación.

Es todo cuanto informar en honor a la verdad; en Riobamba, a los 26 días del mes de mayo de 2025.



Mgs. Fernández Acevedo Jorge Eduardo

C.I: 0602887630



## CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **Tecnologías Low/No-Code para el desarrollo de aplicaciones educativas**, presentado por **Marco Rolando Mullo Chimbolema**, con cédula de identidad número **0605579457**, bajo la tutoría de **Mgs. Fernández Acevedo Jorge Eduardo**; certificamos que recomendamos la **APROBACIÓN** de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 10 de junio de 2025

PHD. PATRICIO RICARDO HUMANANTE RAMOS  
PRESIDENTE DEL TRIBUNAL DE  
GRADO

MG. HERNÁN RAMIRO PAILIACHO YUCTA  
MIEMBRO DEL TRIBUNAL DE GRADO

MG. MARÍA BELÉN PIÑAS MORALES  
MIEMBRO DEL TRIBUNAL DE GRADO



Dirección  
Académica  
VICERRECTORADO ACADÉMICO



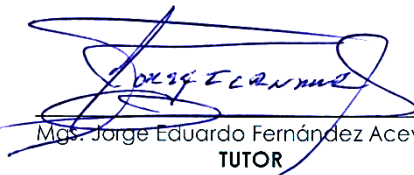
UNACH-RGF-01-04-08.15  
VERSIÓN 01: 06-09-2021

## PLAGIO

# CERTIFICACIÓN

Que, **MULLO CHIMBOLEMA MARCO ROLANDO** con CC: **060557945-7**, estudiante de la Carrera **PEDAGOGÍA DE LAS CIENCIAS EXPERIMENTALES INFORMÁTICA**, Facultad de **CIENCIAS DE LA EDUCACIÓN, HUMANAS Y TECNOLOGÍAS**; ha trabajado bajo mi tutoría el trabajo de investigación titulado **"Tecnologías Low/No-Code para el desarrollo de Aplicaciones Educativas"**, cumple con el **3%**, de acuerdo al reporte del sistema anti plagio **Compilatio Magister+**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 28 de mayo de 2025



Mg. Jorge Eduardo Fernández Acevedo  
TUTOR

## **DEDICATORIA**

Dedico con profundo amor y gratitud este logro a mis queridos padres, Manuel Mullo y Rosa Chimbolema, quienes, con su amor incondicional, su guía constante y su inquebrantable apoyo moral han sido el pilar fundamental de mi vida y formación. Les agradezco por inculcarme valores esenciales, por enseñarme la importancia de la educación y por motivarme a ser una persona íntegra y respetuosa.

Extiendo también esta dedicatoria a mis hermanos, cuya comprensión, paciencia y respaldo constante han sido de gran ayuda en los momentos más desafiantes de mi carrera universitaria.

A todos ustedes, les entrego con orgullo este trabajo de investigación, fruto de esfuerzo, dedicación y perseverancia. Este logro académico también es suyo.

## **AGRADECIMIENTO**

En primer lugar, agradezco a Dios, por brindarme sabiduría, fortaleza y perseverancia para culminar con éxito esta importante etapa académica.

A mis padres, Manuel Mullo y Rosa Chimbolema, mi más sincero agradecimiento por su amor incondicional, su apoyo constante y su incansable lucha por verme alcanzar mis sueños. Gracias por creer en mí; este logro también les pertenece.

A mis hermanos, quienes me abrieron las puertas de sus hogares y me brindaron comprensión, paciencia y apoyo durante todo este proceso. Su generosidad fue clave para mi formación académica.

Extiendo mi agradecimiento a todos mis profesores por su entrega y dedicación, y de manera especial a mi tutor de titulación, por su guía, compromiso y por compartir sus conocimientos y experiencia, lo cual fue fundamental en el desarrollo de este trabajo y en mi crecimiento profesional.

# ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

PLAGIO

DEDICATORIA

AGRADECIMIENTO

ÍNDICE GENERAL

ÍNDICE DE TABLAS

RESUMEN

ABSTRACT

|                                                 |    |
|-------------------------------------------------|----|
| CAPÍTULO I.....                                 | 16 |
| 1.1.    INTRODUCCIÓN.....                       | 16 |
| 1.2.    Antecedentes.....                       | 17 |
| 1.3.    Planteamiento del Problema .....        | 18 |
| 1.3.1.        Formulación del Problema.....     | 20 |
| 1.4.    Justificación.....                      | 20 |
| 1.5.    Objetivos.....                          | 21 |
| 1.5.1.        General.....                      | 21 |
| 1.5.2.        Específicos.....                  | 21 |
| CAPÍTULO II.....                                | 22 |
| MARCO TEÓRICO .....                             | 22 |
| 2.1.    Estado de Arte .....                    | 22 |
| 2.2.    Definición de tecnologías No-Code ..... | 24 |



|         |                                                                                                    |    |
|---------|----------------------------------------------------------------------------------------------------|----|
| 2.3.    | Definición de tecnologías Low-Code .....                                                           | 24 |
| 2.4.    | Características de las Tecnologías Low/No-Code .....                                               | 24 |
| 2.5.    | Beneficios de las Tecnologías Low/No-Code .....                                                    | 26 |
| 2.6.    | Principales Plataformas y Tecnologías Low/No-Code.....                                             | 26 |
| 2.6.1.  | PictoBlox .....                                                                                    | 26 |
| 2.6.2.  | Scratch .....                                                                                      | 28 |
| 2.6.3.  | App Inventor.....                                                                                  | 29 |
| 2.6.4.  | Microsoft Power Apps.....                                                                          | 30 |
| 2.6.5.  | Retool .....                                                                                       | 31 |
| 2.6.6.  | DronaHQ .....                                                                                      | 33 |
| 2.6.7.  | AppGyver .....                                                                                     | 34 |
| 2.6.8.  | Flutter Flow.....                                                                                  | 35 |
| 2.7.    | Comparativa general de las tecnologías Low/No code estudiadas..                                    | 36 |
| 2.8.    | Metodologías ágiles tradicionales. ....                                                            | 40 |
| 2.9.    | Metodologías ágiles en entornos No-Code.....                                                       | 40 |
| 2.10.   | Comparación entre metodologías ágiles tradicionales y metodologías ágiles en entornos No-Code..... | 41 |
| 2.11.   | Impacto social y la gestión administrativa de tecnologías de Low/No-Code .....                     | 45 |
| 2.11.1. | Impacto Social .....                                                                               | 45 |
| a)      | Acceso a la tecnología: .....                                                                      | 45 |
| b)      | Mejora en la educación:.....                                                                       | 45 |
| c)      | Desarrollando habilidades digitales:.....                                                          | 45 |
| d)      | Localización del trabajo en equipo:.....                                                           | 45 |
| e)      | Reducción de silos de información:.....                                                            | 46 |

|                    |                                                          |    |
|--------------------|----------------------------------------------------------|----|
| 2.11.2.            | Gestión Administrativa.....                              | 46 |
| a)                 | Eficiencia operativa: .....                              | 46 |
| b)                 | Optimización de recursos: .....                          | 46 |
| c)                 | Agilidad en la toma de decisiones: .....                 | 46 |
| d)                 | Adaptación a los cambios: .....                          | 46 |
| e)                 | Reducción de costos: .....                               | 47 |
| f)                 | Implementación rápida:.....                              | 47 |
| g)                 | Experiencia del cliente mejorada:.....                   | 47 |
| h)                 | Retroalimentación continua:.....                         | 47 |
| CAPÍTULO III ..... |                                                          | 48 |
| METODOLOGÍA.....   |                                                          | 48 |
| 3.1.               | Tipo de Investigación .....                              | 48 |
| 3.2.               | Diseño de la Investigación.....                          | 48 |
| 3.3.               | Enfoque de la Investigación .....                        | 49 |
| 3.4.               | Método de Análisis.....                                  | 49 |
| a)                 | Definición de los Criterios de Inclusión y Exclusión ... | 49 |
| b)                 | Búsqueda de Fuentes .....                                | 49 |
| c)                 | Análisis Comparativo .....                               | 50 |
| d)                 | Criterios de Evaluación ISO 9241 .....                   | 50 |
| 3.5.               | Fuentes de Información .....                             | 51 |
| 3.6.               | Población Documental.....                                | 51 |
| 3.7.               | Resultado de la búsqueda de fuentes bibliográficas ..... | 52 |
| 3.8.               | Metodologías de Búsqueda.....                            | 54 |
| 3.8.1.             | Definición de Criterios de Búsqueda.....                 | 54 |
| 3.8.2.             | Selección de Recursos .....                              | 54 |

|                                      |                                                                                                                               |    |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|----|
| 3.8.3.                               | Criterios de Inclusión .....                                                                                                  | 54 |
| 3.8.4.                               | Criterios de Exclusión .....                                                                                                  | 56 |
| 3.9.                                 | Criterios de Selección de la Información .....                                                                                | 57 |
| 3.10.                                | Proceso de Recolección de Datos .....                                                                                         | 57 |
| CAPÍTULO IV .....                    |                                                                                                                               | 58 |
| RESULTADOS Y DISCUSIÓN .....         |                                                                                                                               | 58 |
| 4.1.                                 | Criterios de Evaluación de las Tecnologías Low/No-code para el desarrollo de aplicaciones educativas.....                     | 59 |
| 4.1.1.                               | Métricas de selección.....                                                                                                    | 59 |
| 4.2.                                 | Análisis general de las principales tecnologías low/no-code que mejoren la experiencia de aprendizaje de la programación..... | 68 |
| CAPÍTULO V .....                     |                                                                                                                               | 73 |
| CONCLUSIONES y RECOMENDACIONES ..... |                                                                                                                               | 73 |
| 5.1.                                 | Conclusiones.....                                                                                                             | 73 |
| 5.2.                                 | Recomendaciones .....                                                                                                         | 74 |
| BIBLIOGRAFÍA .....                   |                                                                                                                               | 75 |

## ÍNDICE DE TABLAS

|                                                                                                                             |    |
|-----------------------------------------------------------------------------------------------------------------------------|----|
| <b>Tabla 1.</b> <i>Elementos clave de PictoBlox</i> .....                                                                   | 27 |
| <b>Tabla 2.</b> <i>Elementos clave de Scratch 3.0</i> .....                                                                 | 28 |
| <b>Tabla 3.</b> <i>Elementos clave de App Inventor</i> .....                                                                | 29 |
| <b>Tabla 4.</b> <i>Elementos clave de Microsoft Power Apps</i> .....                                                        | 30 |
| <b>Tabla 5.</b> <i>Elementos clave de Retool</i> .....                                                                      | 32 |
| <b>Tabla 6.</b> <i>Elementos clave de DronaHQ</i> .....                                                                     | 33 |
| <b>Tabla 7.</b> <i>Elementos clave de AppGyver</i> .....                                                                    | 34 |
| <b>Tabla 8.</b> <i>Elementos clave de Flutter Flow</i> .....                                                                | 35 |
| <b>Tabla 9.</b> <i>Comparativa de las tecnologías</i> .....                                                                 | 37 |
| <b>Tabla 10.</b> <i>Comparación entre metodologías ágiles tradicionales y metodologías ágiles en entornos No-Code</i> ..... | 43 |
| <b>Tabla 11.</b> <i>Enfoques de desarrollo de software</i> .....                                                            | 44 |
| <b>Tabla 12.</b> <i>Búsqueda de Fuentes</i> .....                                                                           | 52 |
| <b>Tabla 13.</b> <i>La herramienta es versión Free</i> .....                                                                | 59 |
| <b>Tabla 14.</b> <i>Precio mensual</i> .....                                                                                | 60 |
| <b>Tabla 15.</b> <i>Facilidad de uso</i> .....                                                                              | 60 |
| <b>Tabla 16.</b> <i>Diseño Responsivo</i> .....                                                                             | 61 |
| <b>Tabla 17.</b> <i>Soporte de Correo Electrónico</i> .....                                                                 | 61 |
| <b>Tabla 18.</b> <i>Integración</i> .....                                                                                   | 62 |
| <b>Tabla 19.</b> <i>Seguridad</i> .....                                                                                     | 62 |

|                                                                                                                               |    |
|-------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Tabla 20.</b> <i>Almacenamiento</i> .....                                                                                  | 63 |
| <b>Tabla 21.</b> <i>Capacidad de integración con otras herramientas</i> .....                                                 | 63 |
| <b>Tabla 22.</b> <i>Requerimientos del Hardware</i> .....                                                                     | 64 |
| <b>Tabla 23.</b> <i>Escalabilidad</i> .....                                                                                   | 64 |
| <b>Tabla 24.</b> <i>Personalizaciones</i> .....                                                                               | 65 |
| <b>Tabla 25.</b> <i>Calidad final de la Aplicación</i> .....                                                                  | 65 |
| <b>Tabla 26.</b> <i>Funcionabilidad</i> .....                                                                                 | 66 |
| <b>Tabla 27.</b> <i>Fiabilidad</i> .....                                                                                      | 66 |
| <b>Tabla 28.</b> <i>Mantenibilidad</i> .....                                                                                  | 67 |
| <b>Tabla 29.</b> <i>Eficiencia</i> .....                                                                                      | 67 |
| <b>Tabla 30.</b> <i>Portabilidad</i> .....                                                                                    | 68 |
| <b>Tabla 31.</b> <i>Resultado final luego de la aplicación de métricas a las tecnologías Low/No code seleccionadas.</i> ..... | 69 |

## RESUMEN

El mundo laboral y la sociedad actual exigen cada vez más el desarrollo de aplicaciones de software a medida de las necesidades, esto resulta ser una valiosa oportunidad para los graduados de la Carrera de Pedagogía de las Ciencias Experimentales: Informática. Sin embargo, para los estudiantes de los primeros semestres resulta complejo emprender en un área en la cual no tienen conocimientos básicos y experiencia. Por otra parte, es un gran reto para los docentes lograr el desarrollo de habilidades cognitivas en los estudiantes para que puedan resolver problemas con ayuda de un lenguaje de programación. Así, la presente investigación tiene como objetivo analizar el uso de las tecnologías low/no-code para el desarrollo de aplicaciones educativas, para que el proceso de enseñanza-aprendizaje sea más motivador, creativo, productivo y eficiente. Para abordar este problema, se utilizó una metodología cuali-cuantitativa, que incluyó revisiones bibliográficas. Se investigaron varias aplicaciones Low/No-Code para evaluar su eficacia y facilidad de uso en el contexto educativo. Los resultados indican que las tecnologías Low/No-Code son altamente efectivas para empoderar a educadores y estudiantes en el mundo de la programación, reduciendo significativamente el tiempo requerido para el desarrollo de aplicaciones. Además, estas tecnologías fomentan la creatividad y el aprendizaje activo, facilitando la implementación de proyectos educativos innovadores. Tras finalizar el estudio y apoyado en la literatura revisada, se recomienda tanto a estudiantes como a docentes, de la Carrera mencionada, utilizar tecnologías tales como Pictoblox, Flutter Flow o Power Apps para incursionar en el aprendizaje de la programación y el desarrollo de aplicaciones de software.

**Palabras claves:** Low-Code, No-Code, Educación, Aplicaciones, Desarrollo de Software, Innovación, Programación, Tecnología

## ABSTRACT

The current labor market and society increasingly demand the development of customized software applications, which represents a valuable opportunity for graduates of the Pedagogy of Experimental Sciences: Informatics program. However, for early-semester students, entering this field without basic knowledge or experience presents a significant challenge. Moreover, it is difficult for educators to foster the cognitive skills necessary for problem-solving through programming languages. This research aimed to analyze the use of low-code and no-code technologies in the development of educational applications to enhance the teaching-learning process, making it more motivating, creative, productive, and efficient. A qualitative-quantitative methodology was applied, primarily through bibliographic reviews. Several low-code and no-code tools were examined to evaluate their effectiveness and usability in educational contexts. The findings show that these technologies are highly effective in empowering both teachers and students in the field of programming, significantly reducing development time. Furthermore, they encourage creativity and active learning, facilitating the implementation of innovative educational projects. Based on the findings, the use of tools such as Pictoblox, Flutter Flow, and Power Automate (formerly known as Power Apps) is recommended for students and educators seeking to explore programming and software development.

**Keywords:** Low-Code, No-Code, Education, Applications, Software Development, Innovation, Programming, Technology



Firmado electrónicamente por:  
**SOFIA FERNANDA  
FREIRE CARRILLO**

Validar electrónicamente con PímaeEC

Reviewed by:

Mgs. Sofía Freire Carrillo

**ENGLISH PROFESSOR**

C.C. 0604257881

# CAPÍTULO I

## 1.1. INTRODUCCIÓN

En los últimos años ha cobrado relevancia una nueva tendencia en el desarrollo de aplicaciones conocidas como *low-code/no-code* (bajo código/sin código), la cual hace referencia a plataformas tecnológicas que permiten crear soluciones digitales con una mínima o nula necesidad de programación tradicional. Estas tecnologías están diseñadas para facilitar el desarrollo de software incluso a usuarios sin conocimientos técnicos avanzados, agilizando los procesos de diseño y automatización.

El presente trabajo de investigación aborda la enseñanza de la programación mediante la utilización de estas tecnologías (Low/No Code) como una alternativa de aprendizaje innovadora dirigida a estudiantes sin experiencia previa, con el ánimo de mermar las dificultades que enfrentan al aprender programación y posteriormente al crear aplicaciones educativas. Dada la creciente necesidad de habilidades digitales, sumada a la limitada inversión en infraestructura y a los sistemas educativos con escasos recursos en Ecuador, esta investigación resulta oportuna y crucial para identificar las barreras que obstaculizan el desarrollo de dichas competencias en el país.

El uso de tecnologías de Low/No Code tiene el potencial de democratizar el desarrollo de software, promoviendo una experiencia de aprendizaje más rica en el aula.

El propósito de esta investigación es analizar cómo las tecnologías low/no-code puede transformar la enseñanza de la programación, haciendo que sea más accesible y pertinente para los futuros educadores. Además, se evalúa su potencial para mejorar la preparación del sistema educativo frente a los desafíos tecnológicos actuales, contribuyendo a formar profesionales competentes en un mundo cada vez más digitalizado.

El presente informe está organizado en cinco secciones:

**Capítulo I:** en esta sección se expone los antecedentes del objeto de estudio, el planteamiento del problema que describe los desafíos que enfrentan estudiantes y educadores al enseñar programación. Se presenta la pregunta de investigación que impulsa el estudio, y se justifica la importancia del estudio en el contexto actual.



**Capítulo II:** esta sección presenta la fundamentación teórica donde se contextualiza la importancia de la programación en la educación y las limitaciones actuales en Ecuador. Esta sección incluye una revisión de la literatura sobre tecnologías Low/No-Code, sus definiciones, y su aplicabilidad en el ámbito educativo.

**Capítulo III:** esta sección comprende la metodología, la cual ofrece información sobre el tipo, diseño, enfoque de la investigación adoptado, así como las técnicas de recolección de datos. Además, describe la población de interés y los criterios de selección de la información.

**Capítulo IV:** esta sección presenta los resultados y la discusión de los hallazgos obtenidos a partir de la investigación, analizando la efectividad de las tecnologías Low/No-Code en el desarrollo de habilidades en los estudiantes. Se discuten las implicaciones de estos resultados en el contexto educativo y se comparan con estudios previos.

**Capítulo V:** finalmente, se presentan las conclusiones y recomendaciones, se resumen los principales hallazgos del estudio y se ofrecen sugerencias para la implementación de tecnologías Low/No-Code en el currículo de la carrera de Pedagogía de Ciencias Experimentales: Informática, así como para futuras investigaciones en este campo.

## **1.2. Antecedentes**

Para esta investigación, se realizó una Revisión Sistemática de Literatura (SLR) con el fin de recopilar información sobre metodologías para el desarrollo de aplicaciones móviles educativas, obtenida de diversas fuentes académicas como Tesis Doctorales, de Maestría, Trabajos de Titulación, Libros Académicos, Documentos Institucionales, Repositorios (DSpace, Scielo, etc) y Artículos Científicos, priorizando fuentes publicadas entre 2018 y 2025. Su objetivo fue contribuir al diseño, desarrollo y evaluación de estas aplicaciones, ya que actualmente no existen guías metodológicas que orienten a los desarrolladores hacia la creación de aplicaciones de calidad en este ámbito.

En el trabajo de investigación de Roig Hervás (2021) se describió las tecnologías Low-Code y No-Code como una de las propuestas más atractivas en el desarrollo de software, destacándose por la posibilidad de llevar a cabo proyectos sin requerir conocimientos técnicos y en un tiempo reducido. Pero estas tecnologías siguen siendo poco conocidas, y no está claro qué pueden hacer, ni si cumplen con todas sus promesas. El objetivo principal del TFG (Trabajo de fin de Grado) de Roig Hervás (2021) fue estudiar las tecnologías Low-

Code y No-Code para determinar sus pros y contras. Esto se gestionó realizando desafíos en los que se implementaron varios casos de uso, tratando de comparar estas tecnologías con las tradicionales y evaluando su estado frente al desarrollo regular. Las pruebas involucraron la herramienta Low-Code de Microsoft denominada: Power Apps.

El propósito de esta investigación fue analizar como las tecnologías Low/No Code ayudan en la rapidez del desarrollo de un software para Startups (Empresa creciente que se basa en la comercialización de sus productos mediante las TICs) y proyectos emergentes. La población estudiada incluyo desarrolladores experimentados en proyectos innovadores. Los resultados indicaron que las tecnologías Low/No Code permitieron crear aplicaciones con menos esfuerzo, tiempo de desarrollo, costos operativos, favoreciendo una respuesta rápida a las crecientes necesidades del mercado. El impacto mas significativo de esta investigación fue la reducción de costos y tiempo de desarrollo, permitiendo a las pequeñas empresas ser más competitivos en el mercado tecnológico, no solo a nivel social si no también en entornos educativos Carrasco (2025). Esta información es crucial para evaluar la viabilidad de implementar estas tecnologías en un propio contexto.

En el trabajo de Kirikova & Rokis (2022) menciona que las tecnologías Low/No code es un campo relativamente nuevo en constante evolución por lo que requiere comprender los desafíos que se presentan constantemente, estos desafíos pueden servir como un modelo para que también se desarrolle funcionalidades, permitiendo comparar los resultados obtenidos con tecnologías Low-Code y métodos tradicionales.

### **1.3. Planteamiento del Problema**

Actualmente, es fácil notar que todo cuanto nos rodea está vinculado con la tecnología. Uno de los dispositivos tecnológicos más empleado – el celular – da la posibilidad a los usuarios de instalar apps seleccionándolas de una base de miles de aplicaciones cuya utilidad es diversa. Cada vez son más las personas, instituciones y empresas, las que requieren contar con aplicaciones a medida de sus necesidades. En el ámbito educativo pasa algo similar, puesto que, en los docentes y estudiantes nace la necesidad de utilizar aplicaciones para apoyar procesos de aprendizaje gracias al hecho de contar con celulares o computadores portátiles.

Esta necesidad inherente de la sociedad moderna genera un espacio de oportunidades laborales para todo aquel profesional de las Ciencias de la Computación, y de manera particular para los graduados de la Carrera de Pedagogía de las Ciencias Experimentales Informática de la Universidad Nacional de Chimborazo, quienes durante su proceso educativo superior cuentan con cinco asignaturas relacionadas con el desarrollo de aplicaciones educativas. Las tres primeras se dictan en primero, segundo y tercer semestre, perteneciendo a la unidad básica de formación, mientras que las dos siguientes se dictan en cuarto y quinto semestre, perteneciendo a la unidad profesional de formación (UNACH, 2024).

Pero el desarrollo de habilidades, por parte de los estudiantes, en el campo de la programación presenta algunas dificultades en la comprensión y aplicación de conceptos básicos de programación, así como en el desarrollo del pensamiento lógico - algorítmico. Esta situación se debe en gran parte al hecho de que los estudiantes de primer ingreso no tienen una experiencia o conocimiento previo sobre programación. Por otra parte, el aprendizaje de un lenguaje de programación, como Java, se ve ralentizado por la complejidad de su sintaxis. Y los programas que se logran crear en los primeros semestres no tienen un buen impacto visual pues se desarrollan en modo texto. Estos factores confunden y a la vez desmotivan al estudiante. Así pues, con estos antecedentes, resulta ser un gran reto para los docentes de los primeros semestres lograr que sus estudiantes alcancen los resultados de aprendizaje trazados en los sílabos.

Desde la perspectiva del autor de la presente investigación, nace la iniciativa de buscar alternativas tecnológicas que permitan incentivar el aprendizaje de la programación, desarrollar habilidades cognitivas y facilitar posteriormente la creación de aplicaciones de software funcionales dotadas de un buen impacto visual, de manera que los graduados de la Carrera de Pedagogía de las Ciencias Experimentales Informática puedan responder de manera eficiente a las demandas del mercado laboral. Es así como se llega al concepto de tecnologías Low/No-Code que se presentan como una propuesta innovadora que busca democratizar el desarrollo de software.

En consideración a los problemas mencionados anteriormente y con el ánimo de generar un aporte positivo en la formación de los futuros profesionales de Pedagogía de la Informática, se ha creído conveniente desarrollar el presente estudio cuali-cuantitativa de las Tecnologías

Low/No-Code para el aprendizaje de los fundamentos de programación y su posterior aplicación en el desarrollo de aplicaciones educativas.

### **1.3.1. Formulación del Problema**

¿Cuáles son las tecnologías low-code y no-code que presentan las características más adecuadas para facilitar el aprendizaje de los fundamentos de programación y su aplicación en el desarrollo de aplicaciones educativas, según criterios de selección previamente establecidos?

## **1.4. Justificación**

Las nuevas tecnologías tales como las plataformas Low/No-Code, tienen la capacidad de fomentar la inclusividad tanto en el diseño de software como en la enseñanza. Estas tecnologías permiten que más estudiantes desarrollen habilidades críticas con menor dificultad que los lenguajes de programación tradicional (Parra, 2024).

Las tecnologías Low/No Code están proporcionando una solución temporal muy necesaria al permitir que los estudiantes desarrollen aplicaciones de una manera más sencilla. Además de permitirles alcanzar un cierto nivel de competencia técnica, también aumenta su creatividad y autoestima. De esta manera, su experiencia educativa ofrece un aprendizaje más atractivo y relevante (Albuja, Alvear, et al., 2023).

Este trabajo de investigación tiene un valor significativo para la formación de los profesionales de la carrera de Pedagogía de las Ciencias Experimentales Informática de la Universidad Nacional de Chimborazo, ya que se presenta como un referente del uso de las tecnologías Low/No-Code en el ámbito educativo. La relevancia de esta investigación radica en la integración con las competencias del perfil de egresado de la carrera, que busca formar profesionales con la capacidad de integrar tecnologías en la educación (UNACH, 2025). Además, cumple con los requisitos del Plan de Creación de Oportunidades, que busca el desarrollo de una educación de calidad innovadora e inclusiva (Gobierno del Ecuador, 2021).

La investigación proporciona información sobre las plataformas Low/No-Code más populares para el desarrollo de aplicaciones educativas, con el fin de crear una guía útil para educadores y estudiantes determinando las mejores tecnologías para sus entornos académicos.

Finalmente, las tecnologías Low/No-Code propone a los docentes enseñar los fundamentos del diseño y los conceptos educativos sin centrarse en la programación desde el inicio. Esto crea una mejor experiencia de aprendizaje para los estudiantes. Además, las habilidades tecnológicas adquiridas a través del uso de estas tecnologías representan un valor añadido en el mercado laboral actual, capacitando a los estudiantes para abordar problemas emergentes en una sociedad cada vez más digital y globalizada.

## **1.5. Objetivos**

### **1.5.1. General**

- Analizar el uso de las tecnologías low/no-code para el desarrollo de aplicaciones educativas.

### **1.5.2. Específicos**

- Examinar el estado actual de las tecnologías low/no-code y su aplicabilidad en el desarrollo de aplicaciones educativas.
- Definir criterios específicos para la selección de tecnologías low/no-code adecuadas para el desarrollo de aplicaciones educativas, considerando su aplicabilidad en el aprendizaje de la programación.
- Identificar las principales tecnologías low/no-code que mejoren la experiencia de aprendizaje de la programación en la Carrera de Pedagogía de Ciencias Experimentales: Informática, mediante una tabla comparativa de sus características clave.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1. Estado de Arte**

La presente investigación se desarrolló con base en una revisión bibliográfica de estudios recientes con relación en el uso de tecnologías Low/No Code en ámbitos educativos. Su principal objetivo fue identificar los principales conceptos, comparativas, enfoques y aplicaciones vinculadas en el desarrollo de aplicaciones sin código o con poco código. Este análisis permitió establecer teóricamente la investigación, contextualizar el problema y establecer una base sólida para desarrollar el análisis comparativo de las tecnologías Low/No Code seleccionadas.

Existen diversas investigaciones que analizan los orígenes de las tecnologías low-code y no-code, así como su adopción y evolución tanto en el desarrollo de software como en el ámbito educativo. Asimismo, se ha explorado su posible impacto en la formación de futuros profesionales, destacando cómo estas tecnologías podrían transformar las prácticas educativas y las competencias necesarias para el desarrollo de aplicaciones, anticipando cambios en el panorama de la industria y en los conjuntos de habilidades requeridos.

Las plataformas Low-Code y No-Code se encuentran actualmente entre las principales tendencias en el desarrollo de software, ya que ayudan a abordar el problema de la creación de aplicaciones para quienes no son informáticos según Minaya et al. (2022), en su trabajo de investigación titulada “El desarrollo Low/No-code y el futuro de los desarrolladores de software”. Low-code es un método de programación de bajo o muy bajo nivel de codificación, que permite a los usuarios diseñar soluciones de TI (Tecnologías de la Información) innovadoras con una codificación mínima o nula, permitiéndoles abordar eficazmente problemas comerciales específicos. En este sentido, existen una serie de plataformas que han democratizado el desarrollo de software para que organizaciones de todo tipo puedan crear sus propias soluciones.

De esta manera se investigaron mecanismos que apoyan la democratización del desarrollo de aplicaciones intensivas en datos a través de la construcción semiautomática de tecnologías no-code adaptadas al dominio y orientadas a datos, minimizando la necesidad de

configuración y conocimiento sobre computación específica gracias a la investigación de Salado (2024), en su trabajo de investigación titulado “Aplicación de flujos de trabajo científicos en dominios con procesamiento intensivo de datos”. El objetivo principal es reducir la complejidad actual en la creación de este tipo de aplicaciones, simplificando su desarrollo. Esto ayuda a crear más tecnologías intensivas en datos que pueden ser personalizadas según las necesidades de profesionales en diversos campos, todo con menor esfuerzo y costo que los enfoques anteriores que requerían conocimientos tecnológicos avanzados.

Así mismo del trabajo de investigación desarrollado por Fernández-Acevedo (2017) titulado “El Pensamiento Computacional y su relación con el desarrollo de la Creatividad en los niños y niñas del quinto grado de educación general básica de la unidad educativa San Felipe Neri de la ciudad de Riobamba” se puede resaltar algunos puntos de vistas que son importantes para la presente investigación como son:

El desarrollo de la Creatividad es esencial para formar estudiantes innovadores capaces de transformar su entorno al momento de ejercer su rol. Fernández-Acevedo (2017) menciona que el pensamiento computacional impulsa en los estudiantes la generación de ideas originales a través de actividades como la programación con tecnologías low y no-code, favoreciendo así el desarrollo creativo desde el nivel básico.

En la asignatura Desarrollo de Pensamiento Computacional es vital la formación del estudiante mediante estas tecnologías, ya que permiten trascender de usuarios pasivos a docentes capacitados en la creación de contenidos. Fernández-Acevedo (2017) recalca que enseñar programación fomenta las habilidades técnicas como la capacidad de innovar oportunamente en contextos educativos.

El pensamiento analítico, lógico y crítico se desarrolla de manera progresiva a través de las tecnologías low y no-code, permitiendo que los estudiantes estructuren procesos y tomen decisiones fundamentadas. Según Fernández-Acevedo (2017) estas competencias son fundamentales para alcanzar aprendizajes más significativos y autónomos en la asignatura.

En el perfil del egresado de los estudiantes debe incluir el uso ético, creativo y eficiente de las tecnologías low y no-code en el desempeño de sus funciones. Fernández Acevedo (2017)

argumenta que formar a los estudiantes en el desarrollo de soluciones digitales impulsan su participación crítica y productiva en el ejercicio de sus funciones.

## **2.2. Definición de tecnologías No-Code**

Las tecnologías No-Code son plataformas de desarrollo de software que permiten a los usuarios crear aplicaciones sin necesidad de escribir código manualmente. Estas tecnologías están diseñadas para ser accesibles a personas sin conocimientos en programación, lo que facilita la creación de aplicaciones de forma rápida y eficiente (Bravo Vera , 2022).

## **2.3. Definición de tecnologías Low-Code**

Las tecnologías Low-Code son plataformas de desarrollo de software que requiere cierto conocimiento en programación para poder crear aplicaciones con muy poca codificación. Estas tecnologías permiten a los usuarios obtener diferentes niveles de experiencia técnica, desde desarrolladores de nivel básico hasta el nivel avanzado (Hurlburt, 2021).

## **2.4. Características de las Tecnologías Low/No-Code**

Una de las características más importantes de las tecnologías no-code es la accesibilidad, estas plataformas están diseñadas para ser amigables para el usuario, de modo que las personas puedan centrarse en construir la apariencia y la sensación de la aplicación en lugar de preocuparse por la codificación. Como señala Ihrwe (2023), esta accesibilidad ha hecho posible que incluso más personas, incluidos emprendedores y propietarios de pequeñas empresas, puedan crear sus propias soluciones digitales.

También destaca su capacidad para democratizar el desarrollo de aplicaciones, permitiendo a cualquier usuario con o sin habilidad en programación puedan crear soluciones innovadoras a través de interfaces visuales e intuitivas. Con estas tecnologías no necesariamente requiere de escribir grandes volúmenes de código, facilitando así el agilizar el prototipado, integración de múltiples servicios, y sobre todo la personalización de las aplicaciones desarrolladas según los requerimientos de los usuarios (Kirvan & Pratt, 2024).

De acuerdo con Moreno (2022) tecnologías reducen significativamente el tiempo y los costos de desarrollo y a la vez incrementan la participación activa de la comunidad de desarrolladores en la transformación digital y el entorno educativo. Desde este punto de vista



se considera que las características de las tecnologías Low/No Code son clave para el ámbito educativo, ya que no solo promueven el aprendizaje crítico y el pensamiento lógico si no también conocimientos más avanzados en la programación.

Especialmente en el sector educativo, donde los estudiantes necesitan aprender sobre cómo diseñar software o incluso la lógica de la programación, las plataformas sin código son realmente valiosas. La tecnología sin código permite al sector educativo fomentar la creatividad y el pensamiento crítico, al permitir a los estudiantes realizar actividades de resolución de problemas e innovación en lugar de dedicar todo su tiempo a escribir código (Cantillo, 2021).

Otra característica importante de las tecnologías No-Code es la reducción de costos. Al eliminar la necesidad de desarrolladores altamente capacitados y al reducir el tiempo de desarrollo, las empresas pueden ahorrar significativamente en costos de desarrollo. Es particularmente beneficioso para startups y pequeñas empresas con recursos limitados, permitiéndoles competir en igualdad de condiciones con empresas más grandes (Mousalli Kayat, 2018).

A estos elementos se suman otros aspectos que fortalecen el valor de las herramientas Low/No-Code en diferentes contextos:

**Agilidad:** facilidad, simplicidad y rapidez para que la aplicación se desarrolle, modifique y adapte a los cambios que requiera el negocio.

**Experiencia de cliente mejorada:** las empresas tienen ahora mucha más capacidad para agregar valor a los clientes y generar lealtad a través de herramientas, actualizaciones de aplicaciones, colaboraciones entre usuarios, etc.

**Escalabilidad:** qué tan fácilmente la plataforma puede crecer con la empresa para permitirle evolucionar y expandirse para atender las necesidades de usuarios y clientes.

**Visualizaciones:** permite a los usuarios ver datos en tiempo real, información e interacciones con los clientes, y cómo todos los datos pueden filtrarse y editarse para cambiar las vistas de toda la información disponible.

## 2.5. Beneficios de las Tecnologías Low/No-Code

Las plataformas de este tipo y formato ofrecen una serie de beneficios para la gestión educativa y empresarial:

**Autonomía y libertad:** los usuarios no tienen que depender de un programador experto externo o del departamento de TI (Tecnologías de la Información) para obtener asistencia o solicitar una tarea. Las plataformas de Low-Code y No-Code tienen funcionalidades, informes y modelos predefinidos que pueden ser creados o editados fácilmente. Esto ahorra tiempo al asignar una tarea o una acción (Dtark Cloud, 2025).

**Inversión inicial mínima:** no se invierte en sistemas, infraestructuras o expertos en programación. Esta es una ventaja para que este tipo de plataformas sean adecuadas para las pymes o empresas que no quieren gastar una gran cantidad de dinero (Dtark Cloud, 2025).

**Colaboración:** al permitir que cualquier usuario acceda a la plataforma, la información, los datos y las aplicaciones pueden compartirse para ayudar a toda la empresa en su trabajo.

**Aumento de la automatización:** los usuarios tienen la capacidad de automatizar flujos de trabajo a través de acciones básicas para una toma de decisiones precisa, aliviando las cargas de trabajo y ahorrando horas laborables (Dtark Cloud, 2025).

## 2.6. Principales Plataformas y Tecnologías Low/No-Code

Para el desarrollo de la presente investigación, fue necesario identificar y describir las principales plataformas y tecnología Low/No code disponibles en el ámbito educativo. Este paso fue fundamental para conocer el tipo de tecnología (Low o No Code), sus funciones, características, objetivos, Soluciones, Competencias, Ámbitos de aplicación y su Misión, con el objetivo de analizar a cada una de ellas de manera comparativa. Estas tecnologías fueron elegidas tras una revisión documental en base en criterios adaptados del modelo MOSCA y los criterios de evaluación definidos por las normas ISO 9241, que brindaron una estructura sólida para definir la mejor tecnología en entornos educativo.

### 2.6.1. PictoBlox

Tipo: No code

PictoBlox, es una plataforma cuya interfaz gráfica y sencilla permite a los usuarios disfrutar y aprender de forma eficaz y dinámica. Con una sistematización gráfica, basada en el popular lenguaje de programación Scratch 3.0, este instrumento proporciona a los niños una experiencia agradable por medio de acciones divertidas y sencillas como arrastrar y soltar objetos, lo que les ayuda a perfeccionar su comprensión, imaginación y creatividad. El programa posee una serie de librerías avanzadas para que los estudiantes con mayor rango de edad y experiencia puedan incursionar en el uso y creación de aplicaciones dotadas de inteligencia artificial, la cual permite entre otras cosas el reconocimiento de imágenes, voz y procesamiento de lenguaje natural.

Gracias a los principios esenciales de la educación STEAM, esta aplicación fue creada con la meta de incorporar la ciencia, la tecnología, la ingeniería y las matemáticas como fundamentos claves para un nuevo tipo de conocimiento informático (STEAMpedia, 2024).

La Tabla 1 resume los elementos clave de PictoBlox según el modelo MOSCA, destacando su misión, objetivos, soluciones, competencias y ámbitos de aplicación:

**Tabla 1.** *Elementos clave de PictoBlox*

|                       |                                                                                                                                                                                                                                                  |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Misión                | Presentar a los usuarios el mundo de la programación y estimular su capacidad de pensar de manera lógica mediante un enfoque divertido y atractivo.                                                                                              |
| Objetivos             | <ol style="list-style-type: none"> <li>1. Desarrollar habilidades de programación.</li> <li>2. Fomentar el pensamiento computacional.</li> <li>3. Promover la creatividad en proyectos innovadores.</li> </ol>                                   |
| Soluciones            | <ul style="list-style-type: none"> <li>- Sistema de codificación por bloques que simplifica el aprendizaje.</li> <li>- Integración de Python para un avance en programación.</li> <li>- Entorno seguro y atractivo para experimentar.</li> </ul> |
| En Competencias       | <ul style="list-style-type: none"> <li>- Resolución de problemas.</li> <li>- Pensamiento crítico.</li> <li>- Colaboración en proyectos grupales.</li> </ul>                                                                                      |
| Ámbitos de Aplicación | <ul style="list-style-type: none"> <li>- Educación formal en escuelas y colegios.</li> <li>- Talleres extracurriculares y clubes de programación.</li> <li>- Formación en habilidades digitales para el futuro laboral.</li> </ul>               |

*Nota:* La tabla representa los elementos claves de PictoBlox.  
*Fuente:* Tomado de (STEAMpedia, 2024).

2.6.2. Scratch

Tipo: No-code

Scratch es un entorno de programación visual desarrollado por el MIT (Instituto de Tecnología de Massachusetts) con el objetivo de enseñar los conceptos de la informática y la lógica a estudiantes de cualquier nivel. Su nombre deriva de "Scratch", que como concepto de programación significa reutilizar fragmentos de código que son fácilmente ajustables y modificables para un nuevo uso. (Rueda López, 2024).

La Tabla 2 resume los elementos clave de Scratch según el modelo MOSCA, destacando su misión, objetivos, soluciones, competencias y ámbitos de aplicación:

**Tabla 2.** *Elementos clave de Scratch 3.0*

|                       |                                                                                                                                                                                                                |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Misión                | Simplificar la enseñanza de conceptos de computación y lógica, permitiendo a estudiantes de todas las edades aprender a programar de manera accesible y divertida.                                             |
| Objetivos             | 1. Introducir a los usuarios en la programación visual.<br>2. Fomentar la creatividad y el pensamiento lógico.<br>3. Facilitar la reutilización y adaptación de fragmentos de código.                          |
| Soluciones            | - Entorno de programación visual que elimina la necesidad de escribir código.<br>- Herramientas intuitivas para crear proyectos interactivos.<br>- Comunidad en línea para compartir y colaborar en proyectos. |
| Competencias          | - Desarrollo del pensamiento computacional.<br>- Habilidades de resolución de problemas.<br>- Trabajo en equipo y colaboración en proyectos.                                                                   |
| Ámbitos de Aplicación | - Educación formal en aulas de primaria y secundaria.<br>- Talleres de programación y actividades extracurriculares.<br>- Proyectos de aprendizaje colaborativo y creativo.                                    |

*Nota:* Esta tabla representa los elementos clave de Scratch 3.0

*Fuente:* Tomado de (Rueda López, 2024).

**2.6.3. App Inventor**

Tipo: Low-code

App Inventor es una plataforma de desarrollo de aplicaciones móviles que consiste en un diseñador con una interfaz de arrastrar y soltar para diseñar los elementos de la interfaz de usuario (UI) de la aplicación. El editor de bloques es un entorno de programación visual en el cual los inventores de aplicaciones pueden construir la lógica de sus aplicaciones encajando elementos codificados por colores como piezas de un rompecabezas para describir el programa. Para ayudar en el desarrollo y prueba de aplicaciones, App Inventor ofrece una aplicación móvil llamada App Inventor Companion (o simplemente “el Companion”) que los desarrolladores pueden ejecutar de manera independiente para verificar y modificar la funcionalidad de su aplicación en tiempo real. Esto significa que cualquiera puede crear una aplicación móvil en poco tiempo y comenzar a iterar y probar de inmediato (Salas, Pérez, & Bernal, 2024).

La Tabla 3 resume los elementos clave de App Inventor según el modelo MOSCA, destacando su misión, objetivos, soluciones, competencias y ámbitos de aplicación:

**Tabla 3.** *Elementos clave de App Inventor*

|            |                                                                                                                                                                                                                                                             |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Misión     | Facilitar el desarrollo de aplicaciones móviles para Android, permitiendo a los usuarios, especialmente estudiantes, crear aplicaciones de manera rápida y accesible.                                                                                       |
| Objetivos  | <div>1. Proporcionar una plataforma intuitiva para el desarrollo de aplicaciones.</div> <div>2. Fomentar el aprendizaje práctico a través de proyectos creativos.</div> <div>3. Permitir la publicación y monetización de aplicaciones desarrolladas.</div> |
| Soluciones | <div>- Interfaz gráfica de arrastrar y soltar que simplifica el proceso de programación.</div> <div>- Acceso a herramientas en la nube para el desarrollo de aplicaciones.</div>                                                                            |

|                       |                                                                           |
|-----------------------|---------------------------------------------------------------------------|
| Competencias          | - Posibilidad de crear juegos, animaciones y sistemas de geolocalización. |
|                       | - Desarrollo de habilidades en programación y diseño de aplicaciones.     |
|                       | - Fomento de la creatividad y la innovación.                              |
|                       | - Capacidades de trabajo colaborativo y resolución de problemas.          |
| Ámbitos de Aplicación | - Educación formal en cursos de tecnología y programación.                |
|                       | - Talleres de desarrollo de aplicaciones móviles.                         |
|                       | - Proyectos extracurriculares y emprendimientos estudiantiles.            |

*Nota:* Esta tabla representa los elementos clave de App Inventor.

*Fuente:* Tomado de (Salas, Pérez, & Bernal, 2024).

#### 2.6.4. Microsoft Power Apps

Tipo: Low-code

Se puede decir que es una plataforma para profesores y estudiantes que les proporciona características para crear aplicaciones rápidamente, a veces desde cero o a partir de plantillas. También pueden ser útiles para la gestión de clases, la recopilación de datos de los estudiantes y la automatización de tareas administrativas. Además, Power Apps proporciona capacidades de protección y gestión de datos para asegurar que las aplicaciones sean seguras y estén de acuerdo con las regulaciones de la organización. Microsoft Power Apps es una plataforma de desarrollo Low-code que permite a los usuarios crear aplicaciones personalizadas con habilidades mínimas de codificación. Funciona como una herramienta integrada con Microsoft 365 y actúa sobre las características de Excel y otras aplicaciones conocidas (Roig, 2021),

La Tabla 4 resume los elementos clave de Microsoft Power Apps según el modelo MOSCA, destacando su misión, objetivos, soluciones, competencias y ámbitos de aplicación:

**Tabla 4.** Elementos clave de Microsoft Power Apps

|                       |                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Misión                | Permitir crear aplicaciones personalizadas de manera sencilla y accesible, sin necesidad de habilidades de codificación extensas, para mejorar la eficiencia en entornos educativos.                                                                                                                                                                    |
| Objetivos             | <ol style="list-style-type: none"> <li>1. Facilitar el desarrollo de aplicaciones personalizadas para necesidades específicas.</li> <li>2. Integrar herramientas familiares de Microsoft para simplificar el proceso.</li> <li>3. Proporcionar plantillas predefinidas para acelerar el desarrollo.</li> </ol>                                          |
| Soluciones            | <ul style="list-style-type: none"> <li>- Plataforma intuitiva que permite la creación de aplicaciones a partir de plantillas o desde cero.</li> <li>- Integración con Microsoft 365, aprovechando herramientas como Excel.</li> <li>- Funciones avanzadas de seguridad y administración de datos para cumplir con políticas institucionales.</li> </ul> |
| Competencias          | <ul style="list-style-type: none"> <li>- Desarrollo de habilidades en creación de aplicaciones y gestión de datos.</li> <li>- Fomento de la innovación y la resolución de problemas específicos del entorno educativo.</li> <li>- Capacidades de trabajo colaborativo en proyectos educativos.</li> </ul>                                               |
| Ámbitos de Aplicación | <ul style="list-style-type: none"> <li>- Educación formal para la gestión de clases y recopilación de datos.</li> <li>- Automatización de procesos administrativos en instituciones educativas.</li> <li>- Proyectos de innovación y mejora en el entorno escolar.</li> </ul>                                                                           |

*Nota:* Esta tabla representa los elementos clave de Microsoft Power Apps.

*Fuente:* Tomado de (Roig, 2021).

### 2.6.5. Retool

Retool es una plataforma low-code orientada a empresas, que permite construir herramientas internas personalizadas mediante un editor visual de propiedades. Ofrece una amplia variedad de componentes, como tablas, cuadros de texto y menús desplegables, que facilitan

el diseño de interfaces interactivas. Además, Retool gestiona aspectos clave de seguridad, como autenticación, autorización y registros de auditoría. Para garantizar la privacidad, es posible implementar Retool en instalaciones locales, protegiéndolo con una VPN propia y alojándolo dentro de una VPC privada (Carmona & Monsalve, 2024).

La Tabla 5 resume los elementos clave de Retool según el modelo MOSCA, destacando su misión, objetivos, soluciones, competencias y ámbitos de aplicación:

**Tabla 5.** *Elementos clave de Retool*

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Misión                | Facilitar la creación de herramientas internas personalizadas para equipos empresariales, optimizando procesos y mejorando la eficiencia operativa.                                                                                                                                                                                                                                                                          |
| Objetivos             | <ol style="list-style-type: none"><li>1. Ofrecer una plataforma de desarrollo de bajo código que permita a los usuarios crear aplicaciones internas sin requerir una programación compleja.</li><li>2. Proporcionar elementos personalizables, tales como tablas, campos de texto y menús desplegables.</li><li>3. Garantizar la administración de la autenticación, la autorización y los registros de auditoría.</li></ol> |
| Soluciones            | Posee un editor de propiedades intuitivo para la creación rápida de herramientas. Opciones de implementación en las instalaciones para mayor seguridad de los datos. Posibilidad de bloquear conexiones de red mediante VPN y VPC.                                                                                                                                                                                           |
| Competencias          | <ul style="list-style-type: none"><li>- Desarrollo de habilidades en creación de herramientas internas y gestión de datos.</li><li>- Fomento de la colaboración y la eficiencia en equipos de trabajo.</li><li>- Capacidades de seguridad y gestión de datos en entornos empresariales.</li></ul>                                                                                                                            |
| Ámbitos de Aplicación | <ul style="list-style-type: none"><li>- Empresas que necesitan herramientas internas para optimizar procesos.</li><li>- Equipos de desarrollo y operaciones que requieren soluciones personalizadas.</li><li>- Organizaciones que priorizan la seguridad de los datos en sus aplicaciones internas.</li></ul>                                                                                                                |

*Nota:* Esta tabla representa los elementos clave de Retool.



*Fuente:* Tomado de (Carmona & Monsalve, 2024).

**2.6.6. DronaHQ**

DronaHQ es una plataforma Low-code para diseño front-end desarrollada con JavaScript, que permite diseñar interfaces mediante una interfaz de arrastrar y soltar. Ofrece acceso a numerosas plantillas predefinidas y permite utilizar consultas SQL, así como reutilizar componentes para optimizar el desarrollo (Ojeda & Casas, 2024).

La Tabla 6 resume los elementos clave de DronaHQ según el modelo MOSCA, destacando su misión, objetivos, soluciones, competencias y ámbitos de aplicación:

**Tabla 6.** *Elementos clave de DronaHQ*

|                       |                                                                                                                                                                                                                                                                                         |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Misión                | Facilitar la creación de aplicaciones y herramientas internas mediante un enfoque accesible y flexible, permitiendo a las organizaciones optimizar sus procesos.                                                                                                                        |
| Objetivos             | <div>1. Permitir la creación de aplicaciones utilizando un código JavaScript sencillo.</div> <div>2. Ofrecer características como variables, estados e importación de bibliotecas.</div> <div>3. Proporcionar una interfaz de arrastrar y soltar con plantillas predefinidas.</div>     |
| Soluciones            | <div>- Integración de tecnología de inicio de sesión único (SSO) y gestión de permisos.</div> <div>- Entornos seguros y medidas de seguridad adecuadas para grandes corporaciones.</div> <div>- Compatibilidad con SQL y reutilización de componentes para mejorar la eficiencia.</div> |
| Competencias          | <div>- Desarrollo de habilidades en programación básica y creación de aplicaciones.</div> <div>- Fomento de la colaboración y la gestión de permisos en entornos corporativos.</div> <div>- Capacidades en seguridad y gestión de datos.</div>                                          |
| Ámbitos de Aplicación | <div>- Grandes corporaciones que necesitan herramientas internas personalizadas.</div>                                                                                                                                                                                                  |

- Equipos de desarrollo que buscan optimizar procesos mediante aplicaciones.
- Organizaciones que requieren medidas de seguridad robustas y gestión de permisos.

*Nota:* Esta tabla representa los elementos clave de DronaHQ.

*Fuente:* Tomado de (Ojeda & Casas, 2024)

**2.6.7. AppGyver**

AppGyver es una plataforma de desarrollo visual No-code, que permite crear aplicaciones sin necesidad de escribir código. Ofrece integración con múltiples servicios y APIs, lo que facilita el desarrollo rápido de prototipos y aplicaciones interactivas. Su constructor visual simplifica el proceso, permitiendo a los usuarios diseñar aplicaciones de manera accesible, incluso sin conocimientos de programación. No obstante, presenta desafíos como la gestión compleja de datos y limitaciones en la personalización avanzada. (Huatatoca, 2024).

La Tabla 7 resume los elementos clave de AppGyver según el modelo MOSCA, destacando su misión, objetivos, soluciones, competencias y ámbitos de aplicación:

**Tabla 7.** *Elementos clave de AppGyver*

|                       |                                                                                                                                                                                      |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Misión                | Facilitar el desarrollo de aplicaciones y prototipos accesibles para usuarios sin conocimientos de programación.                                                                     |
| Objetivos             | 1. Proporcionar un constructor visual intuitivo.<br>2. Permitir la integración con múltiples servicios y APIs.<br>3. Soportar la creación de prototipos interactivos.                |
| Soluciones            | - Herramienta de arrastrar y soltar para el desarrollo de aplicaciones.<br>- Integración fluida con diversas APIs y servicios.<br>- Desarrollo ágil que permite iteraciones rápidas. |
| Competencias          | - Habilidades en creación de aplicaciones sin código.<br>- Innovación y experimentación en prototipos.<br>- Gestión de datos y conexiones con servicios externos.                    |
| Ámbitos de Aplicación | - Startups y pequeñas empresas que buscan rapidez en el desarrollo.                                                                                                                  |

- Equipos de diseño que necesitan prototipos interactivos.
- Organizaciones que desean optimizar procesos sin depender de desarrolladores.

*Nota:* Esta tabla representa los elementos clave de AppGyver.

*Fuente:* Tomado de (Huatatoca, 2024).

**2.6.8. Flutter Flow**

Flutter, desarrollado por Google, es un marco de interfaz de usuario en lenguaje Dart para el desarrollo de aplicaciones móviles, web y de escritorio multiplataforma a partir de una única base de código. Esto maximiza la distribución del software. Se utiliza para crear interfaces de usuario de aplicaciones hermosas, rápidas y compiladas de forma nativa. Esto hace que el software sea más accesible para las personas. De esta manera, Flutter ayuda a crear aplicaciones móviles de manera mucho más sencilla, en una forma conocida y abreviada. A partir de una única base de código, Flutter compila aplicaciones de forma nativa para dispositivos móviles, web y de escritorio. Además, permite una rápida entrega de características con un enfoque en la experiencia nativa del usuario final. Con una arquitectura en capas clásica, Flutter ofrece una personalización completa y, como resultado, la renderización es extremadamente rápida y los diseños móviles son expresivos y flexibles. La presencia del dispositivo móvil permite que se introduzcan nuevas interacciones en el arte generativo. Flutter ofrece la clase canvas, una clase que es una interfaz para grabar operaciones gráficas. (Huatatoca, 2024).

La Tabla 8 resume los elementos clave de Flutter Flow según el modelo MOSCA, destacando su misión, objetivos, soluciones, competencias y ámbitos de aplicación:

**Tabla 8.** *Elementos clave de Flutter Flow*

|           |                                                                                                                                                                                                                                       |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Misión    | Apoyar el desarrollo de aplicaciones de forma fácil e interactiva.                                                                                                                                                                    |
| Objetivos | <div>1. Proporcionar una interfaz visual para el diseño de aplicaciones.</div> <div>2. Permitir la integración con diversas APIs y servicios.</div> <div>3. Soportar la creación de aplicaciones personalizables y funcionales.</div> |

|                       |                                                                                                                                                                                                                                                                                        |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Soluciones            | <ul style="list-style-type: none"> <li>- Plataforma de arrastrar y soltar que utiliza el framework Flutter.</li> <li>- Integración fluida con APIs y servicios externos.</li> <li>- Desarrollo ágil que permite iteraciones rápidas y pruebas.</li> </ul>                              |
| Competencias          | <ul style="list-style-type: none"> <li>- Habilidades en diseño y desarrollo de aplicaciones sin necesidad de código.</li> <li>- Innovación en el uso del framework Flutter para aplicaciones móviles.</li> <li>- Gestión de conexiones con servicios externos.</li> </ul>              |
| Ámbitos de Aplicación | <ul style="list-style-type: none"> <li>- Equipos de desarrollo que buscan acelerar el proceso de creación de aplicaciones.</li> <li>- Startups que desean lanzar productos rápidamente.</li> <li>- Diseñadores que quieren convertir sus ideas en aplicaciones funcionales.</li> </ul> |

---

*Nota:* Esta tabla representa los elementos clave de Flutter Flow.

*Fuente:* Tomado de (Huatatoca, 2024).

## 2.7. Comparativa general de las tecnologías Low/No code estudiadas

En la Tabla 9, se presenta una comparación de las diferentes plataformas de Low-code, que son tecnologías que permiten la creación de aplicaciones a través de interfaces gráficas de usuario y configuraciones en lugar de la programación tradicional con código manual. Estas plataformas se han convertido en grandes soluciones para empresas y educadores que desean reducir el proceso de desarrollo de software. Desarrollado con la ayuda de Minaya et al., (2022), esta tabla aborda esta consideración con elementos de valor agregado para ediciones gratuitas; costo; usabilidad y diseño responsivo; atención; integración; seguridad; almacenamiento; integración con otras herramientas; hardware; escalabilidad; personalización y la calidad final de las aplicaciones. A continuación, se presenta una tabla comparativa de las tecnologías seleccionadas que resumen el cumplimiento de ciertos criterios clave por cada tecnología seleccionada.

Tabla 9. Comparativa de las tecnologías

| Tecnologías          | Versión Free | Precio Mensual | Facilidad de Uso | Diseño Responsivo | Soporte de Correo Electrónico | Integración      | Seguridad | Almacenamiento | Capacidad de Integración con Herramientas | Requerimientos de Hardware | Escalabilidad | Personalizaciones | Calidad Final de la Aplicación | Funcionalidad | Fiabilidad | Mantenibilidad | Eficiencia | Portabilidad |
|----------------------|--------------|----------------|------------------|-------------------|-------------------------------|------------------|-----------|----------------|-------------------------------------------|----------------------------|---------------|-------------------|--------------------------------|---------------|------------|----------------|------------|--------------|
| PictoBlox            | Sí           | Desde \$0      | Alta             | Sí                | Sí                            | API y bloques    | Alta      | Limitado       | Alta                                      | Bajo                       | Media         | Media             | Alta                           | Media         | Alta       | Alta           | Alta       | Alta         |
| Scratch              | Sí           | \$0            | Alta             | Sí                | No                            | Limitada         | Media     | Limitado       | Media                                     | Bajo                       | Baja          | Baja              | Media                          | Baja          | Media      | Alta           | Alta       | Alta         |
| App Inventor         | Sí           | \$0            | Alta             | Sí                | No                            | API              | Media     | Limitado       | Media                                     | Bajo                       | Media         | Media             | Media                          | Media         | Media      | Alta           | Alta       | Alta         |
| Microsoft Power Apps | No           | Desde \$10     | Media            | Sí                | Sí                            | API y plantillas | Alta      | Variable       | Alta                                      | Medio                      | Alta          | Alta              | Alta                           | Alta          | Alta       | Media          | Alta       | Alta         |
| Retool               | Sí           | Desde \$10     | Media            | Sí                | Sí                            | API              | Alta      | Variable       | Alta                                      | Medio                      | Alta          | Alta              | Alta                           | Alta          | Alta       | Media          | Alta       | Alta         |
| DronaHQ              | Sí           | Desde \$15     | Media            | Sí                | Sí                            | API              | Alta      | Variable       | Alta                                      | Medio                      | Alta          | Alta              | Alta                           | Alta          | Alta       | Media          | Alta       | Alta         |
| AppGyver             | Sí           | \$0            | Alta             | Sí                | No                            | API              | Media     | Limitado       | Alta                                      | Bajo                       | Media         | Media             | Alta                           | Media         | Alta       | Alta           | Alta       | Alta         |
| BravoStudio          | Sí           | Desde \$15     | Alta             | Sí                | Sí                            | API              | Alta      | Variable       | Alta                                      | Bajo                       | Alta          | Alta              | Alta                           | Alta          | Alta       | Media          | Alta       | Alta         |
| Flutter Flow         | Sí           | Desde \$30     | Media            | Sí                | Sí                            | API              | Alta      | Variable       | Alta                                      | Medio                      | Alta          | Alta              | Alta                           | Alta          | Alta       | Media          | Alta       | Alta         |
| Thunkable            | Sí           | Desde \$0      | Alta             | Sí                | Sí                            | API              | Media     | Limitado       | Alta                                      | Bajo                       | Media         | Media             | Alta                           | Media         | Alta       | Alta           | Alta       | Alta         |
| Kodular              | Sí           | Desde \$10     | Alta             | Sí                | Sí                            | API              | Alta      | Variable       | Alta                                      | Bajo                       | Alta          | Alta              | Alta                           | Alta          | Alta       | Media          | Alta       | Alta         |

Fuente: Elaboración Propia.

A continuación, un análisis más detallado de cada variable de la tabla comparativa:

Aunque se dice que usar plataformas gratuitas es genial para estudiantes, educadores y usuarios, no hay un plan gratuito para Microsoft Power Apps. Eso puede ser un problema para aquellos que quieran probarlo antes de gastar dinero.

Los precios de las plataformas varían. Algunas, como Scratch y App Inventor, son gratuitas, lo cual es bueno para el aprendizaje. Otras, como Flutter Flow y Microsoft Power Apps, son más costosas, pero ofrecen funciones adicionales. En otras palabras, los usuarios deben decidir si lo que pagan vale la pena. Tecnologías como PictoBlox y App Inventor permiten crear aplicaciones arrastrando y soltando. Sin embargo, algunas, como Microsoft Power Apps, pueden ser difíciles de aprender. Un buen diseño responsivo hace que las aplicaciones funcionen bien en cualquier dispositivo, ya sea una computadora o un teléfono. La mayoría de las plataformas permiten esto, pero algunas tienen limitaciones.

El soporte por correo electrónico es útil para quienes necesitan ayuda. Tecnologías como PictoBlox, Microsoft Power Apps, Retool, DronaHQ, BravoStudio, Flutter Flow, Thunkable, Kodula cuentan con soporte técnico formal mediante correo electrónico. Algunas tecnologías, como Scratch, AppInventor y AppGyver no lo ofrecen, lo que puede ser un problema si hay dudas o fallos.

Integrar otras herramientas es importante. En este caso la mayoría de las Tecnologías destacan por integrarse fácilmente con APIs, pero algunas requieren saber programar lo que puede ser difícil para principiantes. Por el contrario, Scratch presenta limitaciones importantes en ese aspecto.

La seguridad es crucial, especialmente con datos sensibles. Algunas plataformas como PictoBlox, Microsoft Power Apps, Retool, DronaHQ, BravoStudio, Flutter Flow, Kodula tienen buenas medidas de seguridad, mientras que otras como Scratch, App Inventor, AppGyver y Thunkable ofrecen características mínimas de seguridad.

El almacenamiento determina cuánto espacio se tiene para datos. Algunas tecnologías como Microsoft Power Apps, Retool, DronaHQ, BravoStudio, Flutter Flow y Kodula tienen la capacidad de ofrecerte este servicio, pero se limita cuando se termina el almacenamiento básico proporcionado y se requiere adquirir una licencia adicional para extender el servicio. Otras como PictoBlox, Scratch, App Inventor, AppGyver y Thunkable requieren configurar

servicios externos para guardar información, lo que puede ser un problema si se necesita más espacio y una desventaja para los usuarios sin experiencia técnica.

En cuanto a la capacidad de integración con herramientas, conectar diferentes herramientas es importante para crear aplicaciones útiles. La mayoría de las tecnologías estudiadas permiten enlazarse con otros servicios haciendo que sea más fácil, mientras que otras como Scratch y App Inventor pueden ser más limitadas.

Algunas plataformas como Microsoft Power Apps, Retool, DronaHQ y Flutter Flow pueden requerir de computadoras de gama media, lo cual puede ser un problema para algunos usuarios que no tienen el equipo. Pero sin embargo cualquiera de las tecnologías estudiadas pueden funcionar normalmente en una computadora básica ya que consumen pocos recursos. En cuanto a PictoBlox, Scratch, App Inventor, AppGyver, BravoStudio, Thunkable y Kodula son más fáciles de acceder ya que se puede usar directamente en la web sin usar muchos recursos del computador, lo que facilita su uso en instituciones educativas con recursos limitados.

La escalabilidad es la capacidad de expandirse con el tiempo. Algunas tecnologías como Microsoft Power Apps, Retool, DronaHQ, BravoStudio, Flutter Flow y Kodula son las más sobresalientes, lo cual es importante para las empresas que buscan crecer a diferencia de Scratch que presentan restricciones significativas para ampliar los proyectos creados.

La personalización de aplicaciones es el nombre del juego. Algunas Tecnologías como Microsoft Power Apps, Retool, DronaHQ, BravoStudio, Flutter Flow y Kodula son permisivas y permiten muchas opciones, mientras que otras como Pictoblox, Scratch, App Inventor, AppGyver y Thunkable son más restrictivas.

Algunas plataformas PictoBlox, Microsoft Power Apps, Retool, DronaHQ, AppGyver, BravoStudio, Flutter Flow, Thunkable y Kodula son excelentes para construir aplicaciones de calidad, mientras que Scratch y App Inventor son menos capaces.

La funcionalidad describe cuántas cosas puede hacer una plataforma. Algunas como Microsoft Power Apps, Retool, DronaHQ, BravoStudio, Flutter Flow, Kodula son extremadamente completas, mientras que otras tienden a ser básicas y de nivel principiante.

La Fiabilidad significa que la plataforma funciona perfectamente sin errores. Algunas como PictoBlox, Microsoft Power Apps, Retool, DronaHQ, AppGyver, BravoStudio, Flutter Flow, Thunkable, Kodula son muy estables y otras como Scratch y App Inventor podrían fallar más.

La eficiencia, permite aprovechar bien los recursos del sistema, logrando aplicaciones que se desenvuelven muy rápidas, sin demoras innecesarias y también qué tan bien se ejecuta la plataforma. Todas las tecnologías estudiadas son muy eficientes, permiten el desarrollo de aplicaciones más productivos.

La portabilidad significa que la aplicación puede ser usada en otros dispositivos y sistemas operativos. Todas las tecnologías estudiadas hacen que las aplicaciones funcionen en cualquier lugar sin pérdida de funcionalidad.

## **2.8. Metodologías ágiles tradicionales.**

Las metodologías ágiles tradicionales se define como una respuesta rápida a la rigidez de los modelos clásicos, promoviendo enfoques iterativos e incrementales que priorizan el trabajo colaborativo, la entrega continua y la adaptabilidad al constante cambio (Rodelgo Álvaro, 2019). Este enfoque cambia la perspectiva de responder mejor los cambios y necesidades de los usuarios finales. Por ella se considera que estos principios puedan emplearse en las aulas, fomentando de enseñanza-aprendizaje más dinámico y personalizados.

Molina et al., (2018) sostienen que las metodologías ágiles fortalecen al momento de enfocarse en la adaptación y la independencia del equipo. Estas características fueron fundamentales para mejorar la eficiencia en el campo del desarrollo de software. Desde el ámbito educativo esta flexibilidad permite a estudiantes y docentes asumir un rol más activo y autónomo en el proceso de aprendizaje.

## **2.9. Metodologías ágiles en entornos No-Code.**

Las metodologías ágiles en entornos No-Code, adaptan principios ágiles tradicionales a plataformas visuales o por bloques donde no es un requisito primordial saber programar (Merchán et al., 2024). Esta metodología integran validación temprana y prototipado rápido, acortar barreras técnicas, esto representa una oportunidad para que los estudiantes de



distintos niveles puedan desarrollar proyectos funcionales de forma innovadora, creativa, autónoma y accesible.

Esta metodología destaca la participación activa de múltiples usuarios (técnicos y no técnicos). Esta interacción de la comunidad facilita la creación conjunta de soluciones adaptadas a necesidades reales, se amplía la participación de estudiantes en todo tipo de proyectos, impulsando un aprendizaje significativo y el desarrollo de competencias digitales (González Firmas, 2025).

## **2.10. Comparación entre metodologías ágiles tradicionales y metodologías ágiles en entornos No-Code.**

Las metodologías ágiles han transformado el desarrollo de software al priorizar la entrega rápida de valor y fomentar la colaboración entre equipos multidisciplinarios. Al usar tecnologías No-Code, estas metodologías se aplican para aprovechar al máximo sus beneficios como la flexibilidad, rapidez en el desarrollo, accesibilidad y la colaboración que apoyan a iteraciones rápidas que permiten retroalimentación continua y cambios rápidos (Parada et al., 2020). Esto hace que la construcción y modificación de aplicaciones adopte un enfoque más dinámico, como si los desarrolladores pudieran realizar actualizaciones en tiempo real y con facilidad, sin necesidad de manejar código complejo para efectuar cambios.

Además, la colaboración en entornos sin código también involucra a educadores y estudiantes, lo que permite una co-creación efectiva que responde directamente a las necesidades educativas. Como resultado, las metodologías ágiles en estos entornos fomentan un compromiso activo e inclusivo de todas las partes involucradas.

Para desarrollar la comparativa de las metodologías, se consideraron una serie de características clave para el análisis de las metodologías ágiles tradicionales entre metodologías ágiles en entornos No-Code, tales como: Iteraciones y Sprints, Colaboración Retroalimentación, Velocidad de Desarrollo, Adaptabilidad, Herramientas Utilizadas, entre otros. Según Parada et al., (2020) estas características son fundamentales al momento de evaluar tecnologías Low/No Code, ya que influyen directamente en la experiencia del usuario, desarrollo y la calidad de las soluciones generadas. A continuación, se presentan cada una de estas características:

**Iteraciones y Sprints:** Se representan como ciclos de trabajo breves y repetitivos permitiendo avanzar de forma progresiva en el desarrollo de un proyecto. En las tecnologías Low/No Code estos ciclos son imprescindibles, por que permiten validar funciones y diseños sin afectar el producto final, facilitando la mejora continua.

**Colaboración:** Es el trabajo conjunto de distintos autores para alcanzar un objetivo en común. Estas características permiten que cualquier usuario participen en la creación de un proyecto en conjunto, lo que permite ideal para entornos educativos diversos.

**Retroalimentación:** Se refiere al proceso de recibir comentarios durante o después del desarrollo de un proyecto para realizar mejoras. Estas tecnologías permiten probar diversas funcionalidades en tiempo real que facilitan ajustes en tiempo real fomentando un aprendizaje dinámico.

**Velocidad de Desarrollo:** Hace referencia al tiempo requerido para crear una solución real. Estas tecnologías destacan en gran medida por reducir significativamente este tiempo, ya que usan entornos visuales, lo que permite crear aplicaciones rápidamente sin escribir códigos por los usuarios.

**Adaptabilidad:** Es la capacidad de un sistema o herramienta para ajustarse a distintas necesidades, contextos o cambios. Esta característica permite modificar, funciones o estructura de calidad con facilidad, de manera eficaz en diversas situaciones.

**Herramientas Utilizadas:** Son herramientas que permiten crear soluciones mediante interfaces visuales, bloques de programación o diagramas de flujo. Se enfoca directamente en la eficiencia del desarrollo, calidad y la experiencia de aprendizaje.

**Participación del Usuario Final:** implica la participación activa del usuario en el diseño o evaluación de un producto ya que permiten a los usuarios contribuir con ideas, pruebas y sugerencias durante el desarrollo del proyecto.

**Costo de Desarrollo:** Son los recursos necesarios para desarrollar una aplicación, estas tecnologías reducen significativamente los costos al eliminar personal especializado y software complejo, permitiendo al sector educativo implementar soluciones tecnológicas e innovadoras con presupuestos limitados.

En este sentido, se encontró comparativas entre metodologías ágiles tradicionales y metodologías ágiles en entornos No-Code, donde las metodologías ágiles en entornos No-Code se adaptan fácilmente a los entornos Low/No Code haciendo que sea un trabajo más accesible, visual e inclusiva como se puede observar en la Tabla 10:

**Tabla 10.** *Comparación entre metodologías ágiles tradicionales y metodologías ágiles en entornos No-Code*

| <b>Características</b>         | <b>Metodologías ágiles tradicionales</b>                                         | <b>Metodologías ágiles en entornos No-Code</b>                                      |
|--------------------------------|----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| <b>Iteraciones y Sprints</b>   | Dividido en iteraciones cortas (sprints) para entregar incrementos del producto. | Facilita iteraciones rápidas y modificaciones instantáneas sin reescribir código.   |
| <b>Colaboración</b>            | Equipos multidisciplinares colaboran estrechamente durante el desarrollo.        | Incluye desarrolladores, educadores y estudiantes para co-creación efectiva.        |
| <b>Retroalimentación</b>       | Constante retroalimentación del usuario para ajustar funcionalidades.            | Cambios y nuevas funcionalidades pueden probarse y ajustarse casi instantáneamente. |
| <b>Velocidad de Desarrollo</b> | Velocidad mejorada, pero depende de la reescritura y ajuste del código.          | Creación rápida de prototipos e implementación de cambios en tiempo real.           |
| <b>Adaptabilidad</b>           | Alta adaptabilidad a cambios del mercado y necesidades del usuario.              | Mayor adaptabilidad debido a la facilidad de realizar ajustes y mejoras rápidas.    |
| <b>Herramientas Utilizadas</b> | Requiere herramientas de gestión y desarrollo de software complejas.             | Utiliza plataformas no-code que simplifican el proceso de desarrollo.               |

|                                        |                                                                            |                                                                                                |
|----------------------------------------|----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <b>Participación del Usuario Final</b> | Usuarios finales proporcionan retroalimentación a lo largo del desarrollo. | Usuarios finales (educadores y estudiantes) pueden involucrarse directamente en el desarrollo. |
| <b>Costo de Desarrollo</b>             | Puede ser elevado debido a la necesidad de programadores experimentados.   | Reducido, ya que no se requiere de programadores experimentados para muchas tareas.            |

*Nota:* Esta tabla presenta una comparación detallada entre las metodologías ágiles tradicionales y su aplicación en entornos no-code. Elaboración Propia.

Luego de comparar las metodologías ágiles y tradicionales es importante mencionar los principales enfoques de desarrollo de software que también ayudan y se adaptan a los entornos Low/No Code, la Tabla 11 muestra una comparativa entre tres enfoques de desarrollo de software: Métodos Tradicionales, Metodologías Ágiles y las Metodologías Ágiles en entornos No-Code. Estos enfoques se comparan directamente dos factores clave: Velocidad de Desarrollo y la Adaptabilidad lo que permite evidenciar cuál enfoque se adapta mejor a contextos educativos dinámicos en el desarrollo de aplicaciones.

**Tabla 11.** *Enfoques de desarrollo de software*

| <b>Factor</b>                  | <b>Métodos tradicionales</b> | <b>Metodologías ágiles</b> | <b>Metodologías ágiles en entornos No-Code</b> |
|--------------------------------|------------------------------|----------------------------|------------------------------------------------|
| <b>Velocidad de Desarrollo</b> | 5                            | 7                          | 9                                              |
| <b>Adaptabilidad</b>           | 4                            | 8                          | 10                                             |

*Fuente:* Elaboración Propia.

Los factores de la Tabla 11 se evaluaron como se explica a continuación:

- 1. Velocidad de Desarrollo:** Evaluada en una escala de 1 a 10, donde 1 es muy lento y 10 es muy rápido.
- 2. Adaptabilidad:** Evaluada en una escala de 1 a 10, donde 1 es muy baja adaptabilidad y 10 es muy alta adaptabilidad.

## **2.11. Impacto social y la gestión administrativa de tecnologías de Low/No-Code**

### **2.11.1. Impacto Social**

Las tecnologías Low/No-Code están cambiando el paisaje social de diferentes maneras:

#### **a) Acceso a la tecnología:**

Educadores, emprendedores y otros profesionales de diversas industrias tienen la capacidad de desarrollar aplicaciones y formular soluciones digitales sin ningún entrenamiento técnico gracias a estas plataformas (Segurola, 2024). Como se puede dar cuenta, eso lleva a una mayor inclusión digital, permitiendo a aquellos que poco o nada participaban en el desarrollo tecnológico contribuir en formas que previamente no podían. Un maestro puede crear una aplicación para seguir el progreso de los estudiantes sin ningún conocimiento de programación. La utilización responsable de estas tecnologías conduce a soluciones innovadoras en varios sectores: educación, salud, negocios, etc., mejorando la calidad de vida de las comunidades.

#### **b) Mejora en la educación:**

Lorenz (2023) menciona que los maestros crean herramientas específicamente para sus estudiantes mediante estas tecnologías, lo que lleva a un mejor aprendizaje. Por ejemplo, un maestro puede crear una plataforma que proporcione contenido educativo personalizado según el estilo de aprendizaje y nivel de comprensión de cada estudiante.

#### **c) Desarrollando habilidades digitales:**

Estas tecnologías ayudan a desarrollar competencias digitales esenciales en los estudiantes. Están aprendiendo a interactuar con herramientas tecnológicas, lo que los posiciona para el mercado laboral moderno, que depende cada vez más de habilidades digitales (Andrade-Caraguay et al., 2024).

#### **d) Localización del trabajo en equipo:**

Esto permite la colaboración interdisciplinaria para un proyecto al permitir que equipos multidisciplinarios colaboren en proyectos a través de plataformas de Low-code y No-code según (Andrade-Caraguay et al., 2024). Un equipo de marketing puede crear una aplicación de gestión de campañas independientemente del equipo de TI.

#### **e) Reducción de silos de información:**

Varias organizaciones están aprovechando Azure DevOp puesto que esta crea una cultura de transparencia y colaboración al permitir que diferentes áreas de la organización desarrollen y gestionen sus propias soluciones. Esto permite que la información fluya más libremente entre departamentos, mejorando tanto la comunicación como la eficiencia (Pazmiño Barahona, 2024).

#### **2.11.2. Gestión Administrativa**

Cuando se trata del campo de la gestión administrativa, tanto las tecnologías Low-code y No-code ofrecen inmensos beneficios según Hernández (2025):

##### **a) Eficiencia operativa:**

Estas tecnologías automatizan tareas repetitivas, procesos administrativos y ayudarán a los trabajadores humanos a realizar tareas rutinarias consumiendo menos recursos y tiempo. Cosas simples como enviar correos electrónicos de seguimiento o generar informes se pueden hacer de forma automática sin intervención humana (Lorenz, 2023).

##### **b) Optimización de recursos:**

Los recursos humanos y tecnológicos se optimizan al habilitar a los empleados para desarrollar sus propias soluciones. Esto permite que los equipos de TI se concentren en iniciativas más estratégicas en lugar de enterrarse bajo una montaña de solicitudes de desarrollo (Gobierno de España, 2023).

##### **c) Agilidad en la toma de decisiones:**

Andrade-Caraguay et al., (2024) menciona que mejoran los métodos de paneles y análisis para que los ejecutivos identifiquen rápidamente la información que necesitan. Esto es clave para la toma de decisiones inteligentes durante los procesos en tiempo real.

##### **d) Adaptación a los cambios:**

Las organizaciones pueden adaptarse a los cambios en el entorno empresarial mucho más rápido. Un departamento puede satisfacer las necesidades del mercado rápidamente sin largos ciclos de desarrollo de productos (Andrade-Caraguay et al., 2024).

**e) Reducción de costos:**

Reducción de la Dependencia del Personal Técnico para Desarrollo de Soluciones: Debido a la reducción de la dependencia del personal técnico para el desarrollo de soluciones, también se minimiza la necesidad de desarrollo de software. Un estudio desarrollado por el Gobierno de España (2023) “Esto se debe tanto a su rápida implementación como por no necesitar tanto personal especializado, con el gran ahorro del coste en programadores - que muchas pymes no pueden permitirse-. Este último punto sucede sobre todo con los softwares de No-Code” (p. 7). Es particularmente útil para las pequeñas y medianas empresas que tal vez no tienen el presupuesto para un equipo de desarrollo completo.

**f) Implementación rápida:**

Reduce el tiempo de inactividad y mejora la eficiencia organizacional al crear y desplegar rápidamente aplicaciones (Andrade-Caraguay et al., 2024). Esto asegura que las empresas puedan mantenerse a la vanguardia en un mercado en constante cambio.

**g) Experiencia del cliente mejorada:**

Un estudio desarrollado por el Gobierno de España (2023) las organizaciones pueden diseñar aplicaciones adaptadas específicamente a los requisitos de sus clientes, lo que mejora aún más la experiencia del usuario. Una empresa, para facilitar la explicación, puede crear un portal para clientes que también está personalizado y proporciona soporte inmediato.

**h) Retroalimentación continua:**

Las tecnologías Low/No-Code permiten la implementación rápida de mejoras basadas en la retroalimentación del cliente. Esto lleva a un servicio más ágil y receptivo que construye lealtad del cliente (Lorenz, 2023).

## CAPÍTULO III

### METODOLOGÍA

#### 3.1. Tipo de Investigación

El presente estudio es de tipo documental, conforme a la clasificación propuesta por Hernández-Sampieri, Fernández Baptista (2014) quienes señalan que una investigación documental se caracteriza por la recolección, análisis e interpretación de información proveniente de fuentes secundarias, como artículos científicos, libros y recursos digitales. En este caso, el estudio se fundamenta en la recopilación de información sobre tecnologías Low/No-Code, complementada con una observación directa de dichas tecnologías por parte del autor, lo cual permite un análisis más profundo.

#### 3.2. Diseño de la Investigación

El diseño de la investigación es de tipo **no experimental y descriptivo-comparativo**, ya que se centra en la recolección, análisis y comparación de información existente sobre tecnologías Low/No-Code, sin que el investigador manipule las variables o realice intervenciones directas en el entorno educativo (Mousalli Kayat, 2018).

Se opta por un diseño **descriptivo** porque permite documentar y caracterizar el uso de tecnologías Low/No-Code en el ámbito educativo, brindando una visión clara de sus características, aplicaciones y beneficios potenciales. Este enfoque es adecuado para explorar un fenómeno emergente como las tecnologías Low/No-Code, sin necesidad de alterar las condiciones naturales en las que son utilizadas.

Además, el componente **comparativo** se justifica debido a que el estudio tiene como objetivo analizar y comparar diversas tecnologías Low/No-Code, determinando cuál de ellas presenta las características más adecuadas para apoyar el aprendizaje de la programación en estudiantes de la carrera de Pedagogía de las Ciencias Experimentales Informática. Esta comparación permite establecer similitudes y diferencias entre las herramientas seleccionadas, proporcionando una base sólida para sugerir la más adecuada.



### 3.3. Enfoque de la Investigación

El enfoque de la investigación es **mixto**, ya que combina aspectos cualitativos y cuantitativos. Desde el **enfoque cualitativo**, se centra en el análisis profundo de las tipologías, características, significados y contextos relacionados con las tecnologías Low-Code y No-Code en el ámbito educativo. A través de este enfoque, se analiza cómo estas tecnologías son percibidas y utilizadas, partiendo de un estudio sistemático de la literatura y de la observación directa de las plataformas seleccionadas.

Por otro lado, el **enfoque cuantitativo** se refleja en el uso de métricas específicas para comparar las tecnologías Low/No-Code, permitiendo establecer una evaluación objetiva de sus características clave. Esta combinación permite obtener una visión integral, fundamentada tanto en la interpretación cualitativa como en el análisis comparativo cuantitativo, garantizando una mayor precisión en la selección de la herramienta más adecuada para apoyar el aprendizaje de la programación.

### 3.4. Método de Análisis

#### a) Definición de los Criterios de Inclusión y Exclusión

Criterios de inclusión y exclusión: establecer los límites de la revisión de la literatura, determinando qué estudios son relevantes para la investigación. Los criterios de inclusión fueron definidos con claridad para garantizar la coherencia y relevancia de las fuentes seleccionadas en el proceso de revisión y abarcar las siguientes categorías: el tipo de estudios (por ejemplo, investigación empírica, revisiones sistemáticas, estudios de caso), el contexto educativo (por ejemplo, escuelas, universidades, educación en línea) y el enfoque en tecnologías Low/No-Code. Entre los criterios de exclusión pueden estar los estudios que no están relacionados con el ámbito educativo, estudios que no utilizan metodologías rigurosas o estudios que no reportan resultados significativos. Fue necesario depurar los datos, lo cual ayudó a filtrar la información de modo que la revisión solo incluya fuentes correctas y útiles para que los hallazgos de esta investigación sean válidos y útiles.

#### b) Búsqueda de Fuentes

Las fuentes utilizadas para la búsqueda de datos contribuyeron una parte muy importante de la metodología de análisis, ya que esta dimensión hace que el análisis sea más general y profundo. Esta búsqueda se realizó en bases de datos académicas reconocidas como Google

Scholar, JSTOR, ERIC, SciELO, DSpace, Icalia, Tdea, FloKzu y Dialnet donde sería posible encontrar artículos, tesis y libros relevantes que discutan las tecnologías Low/No-Code en educación. Se utilizó un conjunto de palabras clave y combinaciones de términos que capturaron el alcance de la investigación, por ejemplo, "Low-Code", "No-Code", "tecnologías educativas" y "en el contexto de la educación". Además, se revisaron las referencias de los estudios seleccionados para examinar fuentes adicionales que podrían haberse pasado por alto. Tal enfoque sistemático es crucial para recopilar una literatura diversa, sentando las bases para la síntesis y análisis de la información.

### **c) Análisis Comparativo**

Este apartado ofrece una metodología de análisis comparativo que puede ser utilizada con criterios específicos dados para la comparación de cada trabajo y cada herramienta. En el contexto de esta investigación, se realizó una comparación entre las tecnologías Low/No-Code disponibles en el mercado, así como de los estudios revisados respecto a su uso en la educación. Una parte adicional de este análisis comparativo fue la inclusión de tablas que resumieron los aspectos más importantes de cada herramienta a comparar: facilidad de uso, costos, funcionalidades, soporte técnico, etc. Esto permitió detectar tendencias comunes, ventajas, desventajas, resultados generales de los estudios individuales, así como lagunas en la investigación que necesitan ser abordadas. Estas tecnologías ayudaron no solo a elegir las tecnologías mejor ajustadas para el contexto educativo; también ofrecieron ideas sobre cómo estas tecnologías pueden transformar el aprendizaje y la enseñanza.

### **d) Criterios de Evaluación ISO 9241**

Según Pérez (2023) establecer criterios de evaluación en el proceso de selección de tecnologías Low/No-Code es un paso clave, ya que, a partir de estos criterios, las tecnologías serán evaluadas/examinadas en cuanto a su relevancia para el contexto elegido. Estos criterios (solo por nombrar algunos) deben incluir:

**Facilidad de uso:** cuán intuitiva es la interfaz y experiencia de usuario (si hay alguna);

**Costo:** gastos iniciales y recurrentes;

**Funcionalidad:** qué características específicas proporciona cada una de las tecnologías para construir aplicaciones educativas. Por ejemplo, la disponibilidad de soporte técnico y una comunidad de usuarios activa también puede afectar tu elección, ya que puede impactar la implementación y experiencia de uso de la herramienta.

Estos criterios de evaluación, utilizados y aplicados, aseguran que las tecnologías seleccionadas formen parte de una evaluación objetiva centrada en las necesidades educativas y, por lo tanto, maximizan el éxito de su implementación y uso subsiguiente en el aula.

### **3.5. Fuentes de Información**

Para recabar información y llevar a cabo esta investigación, se emplearon diversas fuentes bibliográficas que ofrecieron un horizonte sobre las tecnologías Low-code y No-code en el ámbito educativo. Las fuentes seleccionadas incluyeron:

- Libros
- Artículos de revistas científicas
- Tesis y trabajos de investigación
- Informes y documentos técnicos

### **3.6. Población Documental**

La población documental es la unidad de estudio de varios textos o un conjunto de textos/documentos/artículos/libros/ relevantes para el estudio. Esta investigación estudia la población documental de las tecnologías Low/No-Code aplicadas al campo educativo. Para asegurar la relevancia y actualidad de la información, solo se consideraron fuentes fechadas en los últimos cinco años.

Se presenta la Tabla 12, que detalla cómo y dónde se llevará a cabo la búsqueda.

### 3.7. Resultado de la búsqueda de fuentes bibliográficas

Este análisis es fundamental para entender el contexto actual de la educación en Ecuador y cómo estas tecnologías pueden contribuir a reducir brechas educativas y mejorar la calidad del aprendizaje de la programación como se muestra en la tabla 12.

**Tabla 12.** *Búsqueda de Fuentes*

| Número | Autor(es)                                                                               | Título                                                                                                                                         | Año  | Fuente                                                                                          | Tipo de Documento | Notas                                                                               |
|--------|-----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------------------|-------------------|-------------------------------------------------------------------------------------|
| 1      | Ibuja Loachamin, Alvear Llor, & Sarango Romero                                          | Technological Inequalities in Education in Ecuador: Addressing the Educational Gap                                                             | 2023 | (Albuja, Alvear, & Sarango, 2023)                                                               | Tesis             | Analiza las desigualdades tecnológicas en la educación ecuatoriana.                 |
| 2      | Aristizabal Alzate, Moreno González, Patiño Santisteban, Ruano Cadena, & Cárdenas Ramos | Integración del proceso de manufactura del CP Factory con procesos tácticos y operativos de la red de valor a través de un BPMS                | 2023 | (Aristizabal Alzate, Moreno González, Patiño Santisteban, Ruano Cadena, & Cárdenas Ramos, 2023) | Tesis             | Proporciona un contexto sobre la integración de procesos en la educación.           |
| 3      | BASANTES SUÑIGA                                                                         | Análisis comparativo sobre la plataforma de desarrollo nocode glideapp y la plataforma low-code outsystems en la creación de aplicaciones web. | 2023 | (Basantes Suñiga, 2022)                                                                         | Tesis             | Útil para comparar plataformas low-code en el desarrollo educativo.                 |
| 4      | Bravo Vera                                                                              | El desarrollo Low/No-code y el futuro de los desarrolladores de software                                                                       | 2022 | (Bravo Vera , 2022)                                                                             | Tesis             | Examina el futuro del desarrollo low/no-code en la educación.                       |
| 5      | Brosnan, Andrew                                                                         | Exploring the nature of risk in digital transformation: a problematization perspective of low-code/ no-code platform risk                      | 2024 | (Brosnan, 2024)                                                                                 | Tesis             | Aborda riesgos asociados con plataformas low-code, relevante para la investigación. |
| 6      | (Carmona Vásquez & Monsalve Giraldo                                                     | Plataforma de ciberseguridad para el aprendizaje y entrenamiento de hacking ético para estudiantes universitarios                              | 2024 | (Carmona & Monsalve, 2024)                                                                      | Tesis             | Proporciona un enfoque en la ciberseguridad en la educación.                        |
| 7      | Deza Villacorta                                                                         | Plataforma web Low-code para realizar el seguimiento y la gestión financiera de TI                                                             | 2024 | (Deza Villacorta, 2024)                                                                         | Tesis             | Relevante para entender aplicaciones de gestión en educación.                       |
| 8      | Casas Sandra                                                                            | El Portal de Integración del PaaS Propuesta de comparación aplicando el proceso analítico jerárquico                                           | 2021 | (Casas, 2021)                                                                                   | Tesis             | Ayuda a comprender la integración de plataformas en educación.                      |
| 9      | Diego Roig Hervás.                                                                      | Tecnologías Low-Code y No-Code: Un caso práctico para estudiar su potencial y limitaciones                                                     | 2021 | (Roig, 2021)                                                                                    | Tesis             | Revisión sobre tecnologías low-code en educación                                    |

|    |                                  |                                                                                                                                                                                                                   |      |                           |                    |                                                                                         |
|----|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------------------|--------------------|-----------------------------------------------------------------------------------------|
| 10 | Fernández Cantillo               | Diseño y uso pedagógico de una página web en wix para el fortalecimiento de las habilidades de lectura crítica a los estudiantes del grado séptimo i de la institución educativa el Carmen.                       | 2021 | (Cantillo, 2021)          | Tesis              | Ejemplo práctico de uso pedagógico de herramientas digitales.                           |
| 11 | Lucas Garcias                    | Programación no-code no ensino superior: possibilidades de avaliação e aprendizagem ativas                                                                                                                        | 2020 | (Lucas Garcia, 2021)      | Tesis Seminario    | Estudio de caso sobre implementación en escuelas                                        |
| 12 | Milton Andrade                   | Low Code para optimizar el tiempo de desarrollo de software                                                                                                                                                       | 2023 | (Salazar, 2023)           | Revista Científica | Relevante para la optimización en el desarrollo de software educativo.                  |
| 13 | Ana Torres                       | Buscando la excelencia educativa: Gestión de procesos académicos y administrativos en Instituciones Públicas de Educación mediante BPM                                                                            | 2015 | (Flores, 2015)            | Tesis              | Evaluación de resultados en estudiantes                                                 |
| 14 | Carlos Hervá                     | Transformando la Educación: tecnología, innovación y sociedad en la Era Digital.                                                                                                                                  | 2024 | (Hervas, 2024)            | Artículo           | Perspectivas sobre el futuro educativo                                                  |
| 15 | Parra Arévalo                    | Análisis del desarrollo de software en no desarrollador                                                                                                                                                           | 2024 | (Parra Arévalo, 2024)     | Tesis              | Analiza el desarrollo de software en contextos no técnicos.                             |
| 16 | Valeria Eliana Ojeda Muñoz       | Caso de Estudio Comparativo de Plataformas Low-Code                                                                                                                                                               | 2024 | (Ojeda & Casas, 2024)     | Tesis              | Comparativos de LOW-CODE                                                                |
| 17 | Deza Villacorta, Andrés          | Plataforma web Low-code para realizar el seguimiento y la gestión financiera de TI                                                                                                                                | 2024 | (Deza Villacorta, 2024)   | Tesis              | Comparativos de LOW-CODE                                                                |
| 18 |                                  |                                                                                                                                                                                                                   |      |                           |                    |                                                                                         |
| 19 | Mendoza Armijos                  | Libro de Memorias II A Congreso internacional de investigación los Andes Innova 2023                                                                                                                              | 2023 | (Mendoza Armijos, 2023)   | Revista            | Perspectivas sobre el futuro educativo                                                  |
| 20 | Tamayo Ibáñez,                   | Low-code como alternativa a la digitalización de empresas. Desarrollo de un caso práctico con Appian.                                                                                                             | 2023 | (Tamayo Ibáñez, 2023)     |                    | Proporciona un caso práctico sobre la digitalización mediante low-code.                 |
| 21 | Fernández Acevedo, Jorge Eduardo | El pensamiento computacional y su relación con el desarrollo de la creatividad en los niños y niñas del Quinto Grado de Educación General Básica de la Unidad Educativa San Felipe Neri de la ciudad de Riobamba. | 2017 | (Fernández-Acevedo, 2017) |                    | Perspectivas sobre los principales enfoques para iniciar en la programación.            |
| 22 | Yumisaca Ortiz, Andrea Elizabeth | Propuesta didáctica para el desarrollo del pensamiento computacional                                                                                                                                              | 2024 | (Ortiz & Fernández, 2024) |                    | Proporciona un caso práctico sobre la aplicación de la metodología No code en la UNACH. |

*Fuente:* Elaboración Propia.

A continuación, se exploró el impacto social y administrativa de las tecnologías Low/No Code, destacando su relevancia y beneficios en contextos ágiles e innovadores y su capacidad para transformarla interacción social y organizativa hacia un entorno más inclusivo y eficiente.

### 3.8. Metodologías de Búsqueda

El proceso utilizado para realizar la búsqueda bibliográfica en el estudio sobre "Tecnologías Low/No-Code para el Desarrollo de Aplicaciones Educativas" se dividió en varias etapas, las cuales se explicación a continuación:

#### 3.8.1. Definición de Criterios de Búsqueda

**Tema:** Tecnologías de Low/No Código y su uso en el campo educativo

**Referencia:** Solo se incluyeron fuentes de los últimos cinco años (2018-2024) para asegurar la información más actualizada.

**Tipo de fuente:** Se incluyeron revistas académicas, libros, tesis, informes y documentos técnicos.

#### 3.8.2. Selección de Recursos

Para llevar a cabo esta investigación, se utilizarán diversos recursos (UR) que facilitarán la recopilación de información relevante:

**Bases de Datos Académicas:** Google Académico, para acceder a artículos académicos, tesis y libros relacionados con tecnologías Low/No-Code.

**Revistas Científicas:** IEEE Xplore, para obtener artículos sobre desarrollo de software y tecnologías emergentes. SpringerLink, para acceder a publicaciones sobre metodologías de enseñanza y aprendizaje en contextos digitales.

**Tesis y Trabajos de Investigación:** DSpace de universidades en las cuales se han realizado estudios que abordan el uso de tecnologías Low/No-Code en educación.

**Informes y Documentos Técnicos:** Organizaciones Educativas, informes de organizaciones como la UNESCO sobre el uso de tecnologías en la educación.

#### 3.8.3. Criterios de Inclusión

- Apropiación y especificidad temática para el dominio de tecnologías Low/No-Code.
- Publicaciones dentro de los últimos cinco años.

- Datos hasta octubre de 2023 (todo en el libro de texto, revistas revisadas por pares, editores reconocidos). Criterios de Inclusión en la búsqueda de información.

En cuanto a la metodología de búsqueda de literatura utilizada, se organizará en fases de la siguiente manera:

### **Definición de Palabras Clave:**

Se definieron palabras clave que delimitaron el alcance de la investigación, como "Bajo Código" (Low Code), "Sin Código (No Code)", "tecnologías educativas", "implementación en la educación" y "efectividad del aprendizaje". El uso de estas palabras clave servirá como semillas para buscar en otros recursos.

### **Selección de Recursos:**

Se seleccionó una variedad de los más relevantes, dándose prioridad a aquellos que mejoren el acceso a literatura revisada por pares y publicaciones reconocidas en el campo de la educación y la tecnología.

### **Búsqueda Sistemática:**

Se realizó una búsqueda sistemática utilizando las palabras clave definidas, mientras se filtran por relevancia y fecha para asegurar que la información registrada esté actualizada y relacionada. Cualquier referencia encontrada se almacenará para futura consideración.

### **Revisión de Resultados:**

Una vez compiladas las fuentes, los resultados fueron revisados críticamente por su calidad y relevancia en relación con cada artículo o publicación. Se centrará particularmente en las metodologías utilizadas en los estudios revisados y los resultados más destacados.

### **Compilación de Datos:**

Los datos se compilaron en categorías basadas en temas que ayuden a identificar patrones de tendencias. Esto permitió que sea más claro y menos congestionado.

## Síntesis de Resultados:

Se sintetizaron los hallazgos considerados más relevantes para generar un informe que resuma las principales conclusiones sobre el uso de tecnologías de Low/No-code en el campo de la educación, con el fin de consolidar la base sobre la cual se realizarán las primeras etapas de la investigación.

Esta son las particularidades o característica que debían cumplir las fuentes para ser consideradas en la investigación. Algunos criterios se deben incluir:

- **Relevancia Temática:** Las fuentes deben estar orientadas a abordar directamente el tema de tecnologías Low/No-Code en la educación.
- **Fecha de Publicación:** Se establecieron por rango de años (publicaciones de los últimos 5 años) para asegurar que la información sea actual.
- **Tipo de Fuente:** Se incluyeron artículos académicos, libros, informes de conferencias, y documentos como tesis.
- **Calidad de la Fuente:** se debe priorizar fuentes de editoriales reconocidas, revistas científicas con impacto, o autores con experiencia en el campo.

### 3.8.4. Criterios de Exclusión

Estas son las peculiaridades que rechazan a las fuentes de ser consideradas en la investigación.

- **Falta de Relevancia:** Fuentes que no abordan el tema de tecnologías Low/No-Code o que lo hacen de manera superficial.
- **Fecha de Publicación:** Excluir documentos muy antiguos (por ejemplo, anteriores a 2018) que puedan no reflejar el estado actual del tema.
- **Tipo de Fuente:** Excluir blogs, artículos de opinión, o fuentes no revisadas por pares que no cumplan con estándares académicos.
- **Sesgo o Falta de Objetividad:** Fuentes que presenten información sesgada o que carezcan de rigor científico.
- **Accesibilidad:** Excluir fuentes que no estén disponibles en formato completo o que requieran suscripción para acceder.



### 3.9. Criterios de Selección de la Información

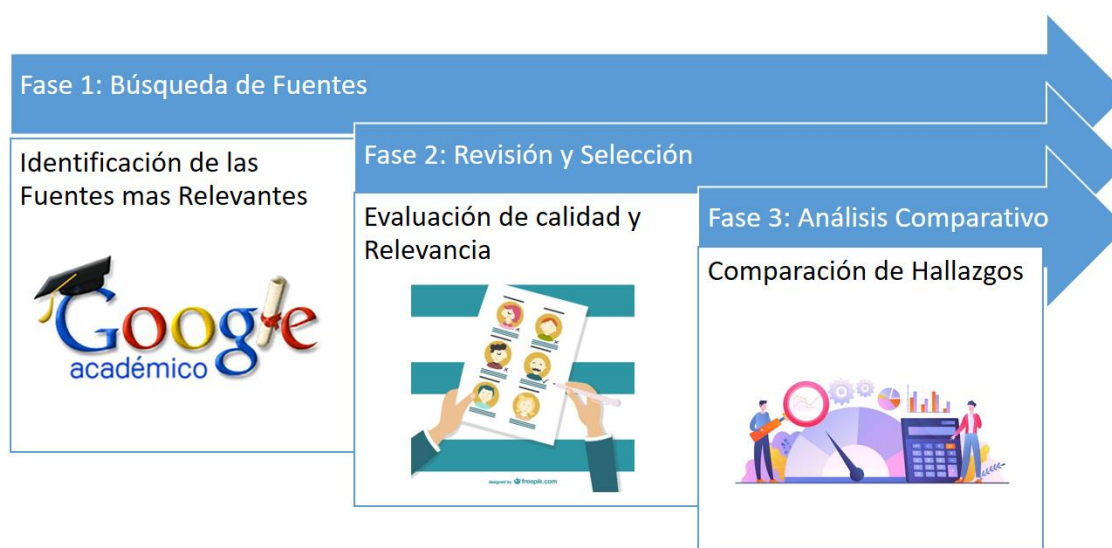
Para lograr obtener la información esencial, de calidad y objetiva en base a las tecnologías Low/No-Code fue necesario establecer criterios específicos para la selección de las fuentes. Estos criterios incluyeron:

- Calidad
- Diversidad de Perspectivas

### 3.10. Proceso de Recolección de Datos

En el desarrollo del trabajo investigativo, la recolección de datos siguió una secuencia estructurada que permitió garantizar la rigurosidad del análisis documental. La figura 1 detalla las fases que se llevaron a cabo, desde la identificación de las fuentes más relevantes hasta el análisis comparativo.

**Figura 1.** Fases del proceso de recolección de datos.



*Nota:* Esta figura representa las fases que se llevara a cabo el proceso de recolección de datos. Elaboración Propia.

Estas fases facilitaron una recolección sistemática y organizada de la información, asegurando que los datos obtenidos fueran adecuados, recientes y relevantes para los objetivos del estudio.

## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

En este capítulo se presenta una exploración detallada de los resultados obtenidos a partir de una exhaustiva revisión bibliográfica sobre las tecnologías Low/No-Code y su aplicabilidad en el desarrollo de aplicaciones educativas. El propósito de este análisis no es solo exponer los hallazgos más relevantes, sino también contextualizar la importancia de estas tecnologías en el ámbito educativo actual.

La revisión de la literatura realizada fue crucial para identificar tendencias, beneficios y desafíos en el uso de tecnologías Low/No-Code en la educación. Utilizando este marco, el objetivo es ofrecer un punto de partida sólido sobre el cual los docentes, investigadores y autoridades educativas puedan basar sus decisiones con respecto al uso de estos recursos en sus prácticas educativas.

El análisis comparativo permitió realizar hallazgos significativos en la investigación sobre tecnologías Low-Code y No-Code. En primer lugar, gracias a la recopilación de información (ver tabla 12), se observa un creciente interés en la integración de estas plataformas en el entorno educativo, lo que sugiere una búsqueda activa de tecnologías que faciliten la enseñanza y el aprendizaje. La diversidad de autores y enfoques refleja un panorama multidimensional, donde se exploran tanto las aplicaciones prácticas como los desafíos y riesgos asociados con estas tecnologías.

Además, varios estudios (ver tabla 12) se centran en la comparación de plataformas; ambos necesitan explorar sus propiedades y contienen ventajas. Tal comparación es crucial para que los institutos y desarrolladores encuentren la herramienta adecuada para sus requisitos.

Por el contrario, la amplitud de la investigación que cubre el impacto social y la gestión administrativa ilustra la importancia de que las tecnologías de Low-Code y No-Code se tengan en cuenta fuera del desarrollo de software, indicando que su utilidad general incluye cambiar la forma en que trabajan las instituciones y se desarrollan las habilidades blandas y duras de los estudiantes.

La recopilación de estos estudios proporciona una base sólida para futuras investigaciones y prácticas en el ámbito educativo. Las tecnologías Low-Code y No-Code no solo tienen el

potencial de transformar la forma en que se desarrollan aplicaciones, sino que también pueden jugar un papel crucial en la modernización y mejora de la educación y en otros contextos similares (Mohammedaziz, 2025). La continua exploración y análisis de estas tecnologías son vitales para maximizar sus beneficios y mitigar sus riesgos, asegurando así un avance significativo en la calidad educativa.

#### 4.1. Criterios de Evaluación de las Tecnologías Low/No-code para el desarrollo de aplicaciones educativas

Para determinar qué aplicación tiene las mejores características, es esencial establecer un conjunto de métricas que permita una comparación objetiva y sistemática de las diferentes plataformas disponibles en el mercado. Estas métricas no solo proporcionaron información sobre los aspectos en los que destaca cada herramienta y en qué áreas falla, sino que también nos permitieron compararlas entre sí, lo cual desempeña un papel crucial en la influencia del proceso de toma de decisiones, gracias a las normas ISO 9241. (International Organization for Standardization, 2025) es una norma que establece los estándares de usabilidad y ergonomía de los sistemas interactivos, se establecieron parte de los criterios y métricas de selección de la Tecnología:

##### 4.1.1. Métricas de selección.

La tabla 13 presenta la métrica “La herramienta es versión Free”, para las tecnologías educativas analizadas en la que se indica si cuenta con una modalidad gratuita o de paga para los usuarios:

**Tabla 13.** *La herramienta es versión Free*

| Respuesta | Ponderación |
|-----------|-------------|
| Si        | 1           |
| No        | 0           |

*Fuente:* Elaboración Propia

La tabla presenta una métrica sencilla significativa, la disponibilidad de una versión gratuita para las tecnologías educativas analizadas. Esta métrica se clasifica en dos categorías: "Sí" y "No", asignando un valor de 1 para las tecnologías que ofrecen una versión gratuita y un valor de 0 para aquellas que no lo hacen.

La tabla 14 presenta la métrica “Precio mensual”, para las tecnologías educativas analizadas en la que se indica si cuenta con precios accesibles o versión gratuita para los usuarios:

**Tabla 14.** *Precio mensual*

| <b>Precio</b> | <b>Ponderación</b> |
|---------------|--------------------|
| \$0           | 2                  |
| \$10          | 1                  |
| \$15          | 0                  |

*Fuente:* Elaboración Propia.

Como se muestra en la tabla de precios mensual, los diferentes niveles de precios importan cuando se trata de la percepción de valor en tecnologías educativas.

La opción de \$0 (cero dólares) recibe la mayor ponderación (2), lo que sugiere que las tecnologías gratuitas son muy deseables porque las escuelas y los sistemas educativos a menudo tienen presupuestos limitados.

Por ejemplo, un valor de \$10 (diez dólares) obtiene una ponderación positiva (1), indicando que se considera razonable y asequible para la mayoría de los usuarios.

El precio de \$15 no recibe ponderación, por lo que se considera excesivo en términos de lo que se ofrece en valor, lo cual podría disuadir a los usuarios.

Pero, en general, esta métrica es importante para los educadores y los administradores escolares para garantizar que las opciones que consideran no rebasen el presupuesto mientras extraen el mayor valor de su inversión.

La tabla 15 presenta la métrica “Facilidad de uso”, para las tecnologías educativas analizadas en la que se indica si su entorno es amigable con el usuario:

**Tabla 15.** *Facilidad de uso*

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Baja             | 0                  |
| Media            | 1                  |
| Alta             | 2                  |

*Fuente:* Elaboración Propia.

Para la ponderación de facilidad de uso se puede apreciar que la respuesta Baja obtiene una ponderación de cero, mientras que la media llevaría una ponderación de 1 punto, las respuestas que son altas tienen una ponderación de 2 la cual será la ponderación con más puntaje.

La tabla 16 presenta la métrica “Diseño Responsivo”, para las tecnologías educativas analizadas que indican si se adaptan a diferentes dispositivos ofreciendo flexibilidad y accesibilidad en diversos entornos de aprendizaje:

**Tabla 16.** *Diseño Responsivo*

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Si               | 1                  |
| No               | 0                  |

*Fuente:* Elaboración Propia.

Para la tabla 16, en referencia a las métricas de diseño Responsive, se obtiene una ponderación si es responsivo un punto siendo la única con ponderación, indicando que las plataformas que se adaptan a diferentes dispositivos son valoradas positivamente, ya que permiten a los usuarios acceder al contenido de manera eficiente y sin problemas. Por otro lado, la opción No, obtiene una ponderación de 0, lo que sugiere que las tecnologías que no son responsivas son consideradas menos deseables, ya que pueden limitar el acceso y la usabilidad, afectando negativamente la experiencia de aprendizaje. En resumen, esta métrica es clave para seleccionar tecnologías que ofrezcan flexibilidad y accesibilidad en diversos entornos de aprendizaje.

La tabla 17 presenta la métrica “Soporte de Correo Electrónico”, para las tecnologías educativas analizadas que indican si ofrecen soporte a través de correo electrónico:

**Tabla 17.** *Soporte de Correo Electrónico*

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Si               | 1                  |
| No               | 0                  |

*Fuente:* Elaboración Propia.

Mientras que la tabla 17 que contiene las métricas sobre la disponibilidad de soporte de correo electrónico para las tecnologías Now-Code. La opción Sí recibe una ponderación de 1, (uno), indicando que las plataformas que ofrecen este tipo de soporte son valoradas positivamente, coordinando asistencia a través de correo electrónico mientras que, en contraste, la opción No obtiene una ponderación de 0, lo que sugiere que las tecnologías sin soporte de correo electrónico.

La tabla 18 presenta la métrica “Integración”, para las tecnologías educativas analizadas que indican si ofrecen API y bloques o API y plantillas para la facilidad de uso de los usuarios:

**Tabla 18. Integración**

| <b>Respuesta</b>                | <b>Ponderación</b> |
|---------------------------------|--------------------|
| Limitada                        | 0                  |
| API                             | 1                  |
| API y bloques o API y Plantilla | 2                  |

*Fuente:* Elaboración Propia.

La tabla 18, la cual se refiere a las métricas de integración de tecnologías educativas. La opción Limitada recibe una ponderación de 0 (cero), lo que indica que las tecnologías con esta categorización carecen de integración y son menos deseables. La opción API obtiene una ponderación de 1 (uno), lo que sugiere que las tecnologías que permiten integración a través de API, aunque de manera moderada. Finalizando con las opciones que ofrecen API y bloques o API y plantillas reciben la mayor ponderación de 2, indicando que estas son altamente valoradas por su capacidad de integración avanzada.

La tabla 19 presenta la métrica “Seguridad”, para las tecnologías educativas analizadas que indican el nivel de seguridad que ofrecen a los usuarios:

**Tabla 19. Seguridad**

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Baja             | 0                  |
| Media            | 1                  |

|      |   |
|------|---|
| Alta | 2 |
|------|---|

*Fuente:* Elaboración Propia.

En contraste con la tabla 19 es la opción Alta recibe la mayor ponderación de 2 (dos), lo que refleja que las tecnologías que brindan un alto nivel de seguridad son altamente valoradas y preferidas, ya que garantizan la protección de datos y la privacidad de los usuarios. Esta métrica es esencial para que los educadores y administradores seleccionen tecnologías que aseguren un entorno de aprendizaje seguro, mientras que la media obtiene una ponderación de 1 (uno). Finalizando con cero ponderaciones para las respuestas a baja.

La tabla 20 presenta la métrica “Almacenamiento”, para las tecnologías educativas analizadas que indican la capacidad de almacenamiento que ofrece para almacenar los recurso:

**Tabla 20.** *Almacenamiento*

| Respuesta | Ponderación |
|-----------|-------------|
| Limitado  | 0           |
| Variable  | 1           |

*Fuente:* Elaboración Propia.

La tabla 20 presenta una evaluación de las métricas de capacidad de almacenamiento de las tecnologías Now-Code. La opción limitada recibe una ponderación de 0, mientras que la ponderación de variable es de uno (1).

La tabla 21 presenta la métrica “Capacidad de integración con otras herramientas”, para las tecnologías educativas analizadas la capacidad de integrarse con otras herramientas:

**Tabla 21.** *Capacidad de integración con otras herramientas*

| Respuesta | Ponderación |
|-----------|-------------|
| Baja      | 0           |
| Media     | 1           |
| Alta      | 2           |

*Fuente:* Elaboración Propia.

La tabla 21 la cual mide las métricas de capacidad de integración de las tecnologías Now-Code. La opción Baja recibe una ponderación de 0 (cero), indicando que las plataformas con esta categorización tienen limitaciones significativas en su capacidad para integrarse con otras herramientas, La opción Media obtiene una ponderación de 1, mientras que la opción Alta recibe la mayor ponderación de 2.

La tabla 22 presenta la métrica “Requerimientos del Hardware”, para las tecnologías educativas analizadas que indican si son compatibles para una amplia gama de dispositivos:

**Tabla 22.** *Requerimientos del Hardware*

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Baja             | 2                  |
| Media            | 1                  |
| Alta             | 0                  |

*Fuente:* Elaboración Propia.

La tabla 22 presenta una evaluación sobre los requerimientos del hardware de las tecnologías educativas. La opción baja recibe la mayor ponderación de 2, indicando que las plataformas que requieren poco hardware son altamente valoradas, ya que son accesibles para una amplia gama de dispositivos, lo que las hace ideales para entornos educativos diversos. La opción, mientras que la media obtiene una ponderación de 1 (uno), mientras que por último la opción alta no obtiene ninguna ponderación.

La tabla 23 presenta la métrica “Escalabilidad”, para las tecnologías educativas analizadas que indican si pueden adaptarse a un crecimiento progresivo en la funcionalidad o complejidad del proyecto:

**Tabla 23.** *Escalabilidad*

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Baja             | 0                  |
| Media            | 1                  |
| Alta             | 2                  |

*Fuente:* Elaboración Propia.



La opción Alta recibe una ponderación de 2, lo que refleja que las tecnologías con alta escalabilidad son altamente valoradas, ya que permiten un crecimiento fluido y continuo, adaptándose a las necesidades cambiantes de los usuarios. Esta métrica es esencial para que los educadores seleccionen tecnologías que sean sostenibles a largo plazo y que puedan crecer junto con sus instituciones, mientras que la opción media obtiene una ponderación de uno (1) y por último la opción baja no obtiene ninguna ponderación.

La tabla 24 presenta la métrica “Personalización”, para las tecnologías educativas analizadas que indican la posibilidad de poder modificar o ajustar elementos según los requerimientos del usuario:

**Tabla 24. Personalizaciones**

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Baja             | 0                  |
| Media            | 1                  |
| Alta             | 2                  |

*Fuente:* Elaboración Propia.

La tabla 24 presenta una evaluación de las métricas personalizaciones de las tecnologías Now-Code. La opción Baja recibe una ponderación de 0 (cero), indicando que las plataformas con esta clasificación ofrecen pocas o ninguna opción de personalización, La opción Media obtiene una ponderación de 1, En contraste, la opción Alta recibe la mayor ponderación de 2, lo que refleja que las tecnologías que ofrecen amplias opciones de personalización son altamente valoradas.

La tabla 25 presenta la métrica “Calidad final de la Aplicación”, para las tecnologías educativas analizadas si permiten generar proyectos funcionales, estables y visualmente atractivos para los usuarios:

**Tabla 25. Calidad final de la Aplicación**

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Baja             | 0                  |
| Media            | 1                  |

|      |   |
|------|---|
| Alta | 2 |
|------|---|

*Fuente:* Elaboración Propia.

La opción Alta de las métricas de Calidad final de la Aplicación recibe la mayor ponderación de 2, reflejando que las aplicaciones de alta calidad son altamente valoradas por su usabilidad y rendimiento, lo que fomenta un aprendizaje efectivo y atractivo. Esta métrica es fundamental para que los educadores seleccionen tecnologías que maximicen el compromiso y los resultados académicos de los estudiantes.

La tabla 26 presenta la métrica “Funcionabilidad”, para las tecnologías educativas analizadas que indican si la tecnología permite ejecutar correctamente las acciones y procesos esperados del producto final:

**Tabla 26. Funcionabilidad**

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Baja             | 0                  |
| Media            | 1                  |
| Alta             | 2                  |

*Fuente:* Elaboración Propia.

La tabla 26 de las métricas de Funcionabilidad, la opción Alta recibe la mayor ponderación de 2, reflejando que las aplicaciones con alta funcionabilidad son altamente valoradas, ya que proporcionan tecnologías efectivas y completas que facilitan un aprendizaje más dinámico y productivo. Esta métrica es esencial para que los educadores elijan aplicaciones que realmente apoyen el proceso de enseñanza-aprendizaje.

La tabla 27 presenta la métrica “Fiabilidad”, para las tecnologías educativas analizadas que indican si la tecnología mantiene un funcionamiento estable a lo largo del tiempo:

**Tabla 27. Fiabilidad**

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Baja             | 0                  |
| Media            | 1                  |
| Alta             | 2                  |

*Fuente:* Elaboración Propia.

La opción Alta recibe la mayor ponderación de 2, reflejando que las aplicaciones con alta fiabilidad son altamente valoradas, ya que ofrecen un rendimiento consistente y estable, lo que es crucial para un aprendizaje efectivo. Esta métrica es fundamental para que los educadores seleccionen tecnologías que aseguren una experiencia de aprendizaje sin interrupciones.

La tabla 28 presenta la métrica “Mantenibilidad”, para las tecnologías educativas analizadas que indican si la tecnología permite modificar, corregir o actualizar fácilmente las aplicaciones desarrolladas a lo largo del tiempo:

**Tabla 28.** *Mantenibilidad*

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Baja             | 0                  |
| Media            | 1                  |
| Alta             | 2                  |

*Fuente:* Elaboración Propia.

La opción Alta recibe la mayor ponderación de 2, reflejando que las aplicaciones con alta mantenibilidad son altamente valoradas, ya que permiten una gestión eficiente y efectiva de los recursos, asegurando que las tecnologías educativas se mantengan actualizadas y funcionales. Esta métrica es esencial para que los educadores seleccionen aplicaciones que faciliten un uso prolongado y sostenible en el entorno educativo.

La tabla 29 presenta la métrica “Eficiencia”, para las tecnologías educativas analizadas que indican si la tecnología optimiza el uso de recurso y un rendimiento optimo:

**Tabla 29.** *Eficiencia*

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Baja             | 0                  |
| Media            | 1                  |
| Alta             | 2                  |

*Fuente:* Elaboración Propia.

La tabla 29, en la opción Alta recibe la mayor ponderación de 2, reflejando que las aplicaciones con alta eficiencia son altamente valoradas, ya que ofrecen un rendimiento óptimo y rápido, lo que contribuye a un aprendizaje más fluido y efectivo. Esta métrica es crucial para que los educadores seleccionen tecnologías que maximicen la productividad y la satisfacción del usuario en el entorno educativo.

La tabla 30 presenta la métrica “Portabilidad”, para las tecnologías educativas analizadas que indican si la tecnología permite ejecutar las aplicaciones en distintos sistemas operativos o dispositivos sin perder la funcionalidad:

**Tabla 30. Portabilidad**

| <b>Respuesta</b> | <b>Ponderación</b> |
|------------------|--------------------|
| Baja             | 0                  |
| Media            | 1                  |
| Alta             | 2                  |

*Fuente:* Elaboración Propia.

La tabla 30 en contraste, la opción Alta recibe la mayor ponderación de 2, reflejando que las aplicaciones con alta portabilidad son altamente valoradas, ya que permiten a los usuarios acceder a las tecnologías educativas desde múltiples dispositivos y plataformas sin inconvenientes. Esta métrica es esencial para que los educadores seleccionen aplicaciones que faciliten el aprendizaje en un entorno flexible y accesible.

#### **4.2. Análisis general de las principales tecnologías low/no-code que mejoren la experiencia de aprendizaje de la programación**

A Continuación, se presenta los resultados obtenidos al aplicar las métricas de evaluación a las tecnologías Low/No analizadas, mostrando las fortalezas y debilidades de cada tecnología basados en el número de criterios cumplidos (Ver la tabla 31).

**Tabla 31.** Resultado final luego de la aplicación de métricas a las tecnologías Low/No code seleccionadas.

| Plataforma           | Versión Free | Precio Mensual | Facilidad de Uso | Diseño Responsivo | Soporte de Correo Electrónico | Integración | Seguridad | Almacenamiento | Capacidad de Integración con Herramientas | Requerimientos de Hardware | Escalabilidad | Personalizaciones | Calidad Final de la Aplicación | Funcionalidad | Fiabilidad | Mantenibilidad | Eficiencia | Portabilidad | Total |
|----------------------|--------------|----------------|------------------|-------------------|-------------------------------|-------------|-----------|----------------|-------------------------------------------|----------------------------|---------------|-------------------|--------------------------------|---------------|------------|----------------|------------|--------------|-------|
| PictoBlox            | 1            | 2              | 2                | 1                 | 1                             | 2           | 2         | 0              | 2                                         | 2                          | 1             | 1                 | 2                              | 1             | 2          | 2              | 2          | 2            | 28    |
| Microsoft Power Apps | 0            | 1              | 1                | 1                 | 1                             | 2           | 2         | 1              | 2                                         | 1                          | 2             | 2                 | 2                              | 2             | 2          | 1              | 2          | 2            | 27    |
| Retool               | 1            | 1              | 1                | 1                 | 1                             | 1           | 2         | 1              | 2                                         | 1                          | 2             | 2                 | 2                              | 2             | 2          | 1              | 2          | 2            | 27    |
| BravoStudio          | 1            | 0              | 1                | 1                 | 1                             | 1           | 2         | 1              | 2                                         | 2                          | 2             | 2                 | 2                              | 2             | 2          | 1              | 2          | 2            | 27    |
| Kodula               | 1            | 1              | 1                | 1                 | 1                             | 1           | 1         | 1              | 2                                         | 2                          | 2             | 2                 | 2                              | 2             | 2          | 1              | 2          | 2            | 27    |
| DronaHQ              | 1            | 0              | 1                | 1                 | 1                             | 1           | 2         | 1              | 2                                         | 1                          | 2             | 2                 | 2                              | 2             | 2          | 1              | 2          | 2            | 26    |
| Flutter Flow         | 1            | 0              | 1                | 1                 | 1                             | 1           | 2         | 1              | 2                                         | 1                          | 2             | 2                 | 2                              | 2             | 2          | 1              | 2          | 2            | 26    |
| AppGyver             | 1            | 2              | 2                | 1                 | 0                             | 1           | 1         | 0              | 2                                         | 2                          | 1             | 1                 | 2                              | 1             | 2          | 2              | 2          | 2            | 25    |
| Thunkable            | 1            | 2              | 1                | 1                 | 1                             | 1           | 1         | 0              | 2                                         | 2                          | 1             | 1                 | 2                              | 1             | 2          | 2              | 2          | 2            | 25    |
| App Inventor         | 1            | 2              | 2                | 1                 | 0                             | 1           | 1         | 0              | 1                                         | 2                          | 1             | 1                 | 1                              | 1             | 1          | 2              | 2          | 2            | 22    |
| Scratch              | 1            | 2              | 2                | 1                 | 0                             | 0           | 1         | 0              | 1                                         | 2                          | 0             | 0                 | 1                              | 0             | 1          | 2              | 2          | 2            | 18    |

Fuente: Elaboración Propia.

Gracias al análisis de la investigación se pudo determinar a PictoBlox como la plataforma con mejores opciones para usarla como tecnologías Low/No-Code, obteniendo una puntuación total de 28 puntos. Entre sus fortalezas está la capacidad de integración, permitiendo a los estudiantes conectar fácilmente diversas herramientas y servicios. Como segunda opción quedaron tres tecnologías: Microsoft Power Apps, Retool y BravoStudio, las cuales con una puntuación de las métricas obtenidas estarían como alternativas secundarias de aprendizaje de programación.

Después de haber realizado la consolidación de la información de las tecnologías Low/No-code, y tomando como punto referencial el contexto educativo de la Carrera de Pedagogía de las Ciencias Experimentales Informática de la Universidad Nacional de Chimborazo, se han seleccionado tres tecnologías específicas para ser sugeridas en los distintos niveles de formación, así:

### **1. Para Desarrollo del Pensamiento Computacional: PictoBlox.**

Se ha considerado esta herramienta porque es una tecnología educativa diseñada específicamente para introducir a los estudiantes en el Desarrollo del Pensamiento Computacional y la programación mediante una interfaz gráfica intuitiva (Programación Visual por Bloques). Un estudio realizado en la Universidad Nacional de Chimborazo implementó una guía didáctica utilizando PictoBlox para estudiantes del primer semestre de la carrera de Pedagogía de Ciencias Experimentales; Informática, sus autores Ortiz & Fernández (2024), concluyeron lo siguiente:

La ejecución de una prueba piloto del uso de la guía didáctica en el grupo de estudiantes del primer semestre de la Carrera de Pedagogía de las Ciencias Experimentales: Informática, permitió evidenciar que los estudiantes estaban motivados al poder crear proyectos basados en juegos con la herramienta PictoBlox, a pesar de que dicho software era nuevo para todos. (p.66)

En términos de la escalabilidad, PictoBlox permite la creación de nuevos proyectos que se puede adaptar a diferentes niveles educativos, desde la educación Inicial hasta la educación Superior. Aunque es una herramienta orientada a la programación visual por bloques, permite ver y editar el código generado, lo que a futuro facilitará una adaptación gradual hacia lenguajes de programación más avanzados. El soporte técnico es considerablemente

estable ya que cuenta con una comunidad de educadores y desarrolladores que comparten sus experiencias y recursos mediante múltiples plataformas.

## **2. Para Lenguajes de Programación: Flutter Flow**

Flutter Flow, es una de las tecnologías más destacadas desarrollado por Google en cuanto a las tecnologías Low-Code. Esta herramienta posee una estructura de código abierto enfocado a la creación de aplicaciones nativas en diversas plataformas utilizando el lenguaje de programación Dart. A diferencia de las tecnologías No-Code, este tipo de herramientas requiere medianamente conocimientos de programación, pero brinda una mayor flexibilidad y control durante el desarrollo de aplicaciones para Android e iOS. Según una comparación realizada por TrustRadius (2025) en tiempo real, en base a las experiencias de múltiples usuarios al momento de crear aplicaciones frente otras herramientas similares, Flutter Flow destaca entre las diferentes tecnologías por su capacidad para crear interfaces de usuario altamente personalizados y por su rendimiento nativo.

Su escalabilidad permite potencialmente el desarrollo de aplicaciones educativas que pueden crecer progresivamente tanto en complejidad y además en funcionalidad sin restricciones significativas durante su desarrollo. Su arquitectura modular facilita ampliamente el trabajo y colaboración en equipos durante el desarrollo de aplicaciones de diferentes tamaños según las necesidades o requerimientos que se va a construir. El soporte técnico está bien respaldado por la misma empresa de Google y su comunidad global activa de múltiples usuarios quienes están empezando en el mundo del desarrollo de software, asimismo usuarios experimentados quienes dan más soporte al resto de la comunidad y sobre todo recursos oficiales disponibles.

## **3. Para Desarrollo de Aplicaciones Educativas: Microsoft Power Apps**

Otra de las tecnologías que destaca en el estudio realizado es Microsoft Power Apps que permite a los usuarios crear aplicaciones personalizadas con “resultados rápidos” y mínima o ninguna codificación, enfocándose en la automatización de procesos y la solución de necesidades específicas permitiendo a socios comerciales “ver” la solución junto al cliente y agregar, eliminar o corregir aspectos de ella por sí mismo. Microsoft (2025) destaca las características claves de Power Apps para el uso en entornos educativos, enfocándose no

solo en el desarrollo de aplicaciones móviles sino también en la creación de aplicaciones web y la automatización de tareas para mejorar la comunidad.

Está diseñada para ser escalable y encajar en el ecosistema de Microsoft, lo que facilita su adopción en diferentes niveles de entornos educativos. Se autodefine como una herramienta sin código, pero, si se desea utilizar JavaScript, TypeScript o C#, también lo soporta. Su soporte técnico cuenta con un fuerte respaldo de la infraestructura de Microsoft y una comunidad de usuarios activa.

La integración de tecnologías Low/No Code al proceso de aprendizaje de la programación y el desarrollo de aplicaciones, ofrece ventajas significativas en términos de accesibilidad (democratización de la programación) y rapidez en el desarrollo de aplicaciones.

Tecnologías como PictoBlox hoy en día ha facilitado el acercamiento de los ciudadanos al mundo de la programación, al desarrollo del pensamiento computacional y la creatividad. Por otra parte, tecnologías como Microsoft Power Apps ha permitido tanto a los docentes como a estudiantes, provistos de conocimientos básicos en programación, crear soluciones personalizadas e innovadoras sin una gran inversión de tiempo y recursos, más allá de los facilitados por la misma herramienta. Sin embargo, este estudio ha determinado que los proyectos más complejos, que requieren recursos significativos, están dirigidos específicamente a desarrolladores experimentados. No obstante, tecnologías como Flutter han permitido un mejor desempeño al combinar tecnologías Low/No-Code, facilitando el trabajo y mejorando la eficiencia en la creación de aplicaciones. Frameworks como Flutter proporcionan las capacidades necesarias para desarrollar aplicaciones, aunque implican una curva de aprendizaje más pronunciada, tanto en contextos educativos como en aplicaciones generales.

Finalmente, además de la facilidad de uso, es fundamental considerar otros factores al seleccionar un lenguaje y entorno de programación para el desarrollo de aplicaciones educativas en los estudiantes de la carrera de Pedagogía de las Ciencias Experimentales: Informática. Entre estos factores se incluyen la escalabilidad, el soporte técnico y, especialmente, las oportunidades laborales. Puesto que estos estudiantes, que se preparan para educar a las nuevas generaciones, requieren herramientas actualizadas y efectivas que faciliten la enseñanza y motiven el aprendizaje. La elección debe alinearse con los objetivos educativos específicos, el nivel de experiencia y las necesidades a largo plazo.



## CAPÍTULO V

### CONCLUSIONES y RECOMENDACIONES

#### 5.1. Conclusiones

- La revisión del estado actual de la literatura ha determinado que las tecnologías Low/No-code están en una fase de creciente adopción en el ámbito educativo. Estas tecnologías no solo facilitan el desarrollo de aplicaciones, sino que también democratizan el acceso al aprendizaje de la programación, al permitir que educadores y estudiantes sin experiencia técnica puedan crear soluciones personalizadas. Así pues, se observa un cambio significativo en la forma en que se aborda la enseñanza y el aprendizaje en entornos digitales.
- Gracias al análisis de los criterios de evaluación obtenidos de las Normas ISO 9241 y al estudio de las tecnologías No-Code, se identificaron los criterios fundamentales para seleccionar la plataforma con mayores beneficios educativos, tanto para los estudiantes como para el docente. Estos criterios incluyen la facilidad de uso, la capacidad de integración con otras herramientas, la seguridad de los datos y la escalabilidad de las aplicaciones desarrolladas.
- Las tecnologías sugeridas permitirán a futuros investigadores proponer opciones basadas en el desarrollo de aplicaciones sin necesidad de pasar codificando por extensos períodos de tiempo. Estas tecnologías dan paso a soluciones reales tanto en el ámbito educativo como en el mercado laboral ya que fortalecen el interés por aprender mediante entornos interactivos y accesible pedagógicamente.

## 5.2. Recomendaciones

- Se sugiere fomentar un ambiente de innovación en el aula mediante el uso de proyectos colaborativos que utilicen tecnologías low/no-code. Los estudiantes deben ser animados a trabajar en equipos para desarrollar aplicaciones que resuelvan problemas reales o que mejoren procesos educativos.
- Se recomienda integrar el uso de tecnologías Low/No-Code en el currículo de la carrera de Pedagogía de Ciencias Experimentales: Informática. Esto permitirá a los estudiantes familiarizarse con tecnologías actuales que facilitan el desarrollo de aplicaciones, fortaleciendo sus competencias tecnológicas desde el inicio de su formación.
- Se sugiere implementar proyectos prácticos en el currículo educativo, donde los estudiantes apliquen conceptos teóricos en la creación de soluciones reales. Esto promoverá el desarrollo de habilidades críticas y creativas, preparándolos mejor para los desafíos del mercado laboral y fomentando su interés por el aprendizaje mediante experiencias interactivas.

## BIBLIOGRAFÍA

- Aristizabal Alzate, S., Moreno González, N., Patiño Santisteban, N., Ruano Cadena, M., & Cárdenas Ramos, A. (2023). *Integración del proceso de manufactura del CP Factory con procesos tácticos y operativos de la red de valor a través de un BPMS*. Bogota: Pontificia Universidad Javeriana.
- Albuja, L. L., Alvear, L. J., & Sarango, R. V. (2023). *Technological Inequalities in Education in Ecuador: Addressing the Educational Gap*. ESPE Universidad de las Fuerzas Armadas. Obtenido de <https://doi.org/10.55813/gaea/ccri/v4/n2/239>
- Basantes Suñiga, A. J. (2022). *Análisis comparativo sobre la plataforma de desarrollo No code glideapp y la plataforma Low-code Outsystems en la creación de aplicaciones Web*. Babahoyo: Universidad Técnica de Babahoyo.
- Bravo Vera , H. (2022). *El desarrollo Low/No-code y el futuro de losdesarrolladores de software*. Guayaquil: Universidad Laica Eloy Alfaro de Manabí.
- Brosnan, A. (2024). *Exploring the nature of risk in digital transformation: A Problematisation Perspective of Low-code/ No-code Platform Risk*. Cork: University College Cork.
- Cantillo, F. (2021). *Diseño y uso pedagógico de una página web en Wix para el fortalecimiento de las habilidades de lectura crítica a los estudiantes del grado séptimo I de la Institución Educativa El Carmen*. Cartagena: Universidad de Cartagena.
- Carmona, V. J., & Monsalve, G. E. (2024). *Plataforma de ciberseguridad para el aprendizaje y entrenamiento de hacking ético para estudiantes universitarios*. Medellin: Escuela de Ingeniería de Antioquia.
- Carrasco, J. L. (2025). *No-Code versus Low-Code en 2025: ¡Analizando los pros y los contras!* Equipo editorial de Quixy. Obtenido de <https://quixy.com/blog/no-code-versus-low-code/>

- Casas, S. (2021). *El Portal de Integración del PaaS Propuesta de comparación aplicando el proceso analítico jerárquico*. Patagonia Austral: Universidad Nacional de la Patagonia Austral.
- Deza Villacorta, A. (2024). *Plataforma web Low-code para realizar el seguimiento y la gestión financiera de TI*. Lima: Universidad Peruana de Ciencias Aplicadas .
- Dtark Cloud. (15 de 02 de 2025). Obtenido de 02: <https://starkcloud.com/starkcloud-blog/que-es-no-code-y-low-code-ejemplos-y-beneficios/#:~:text=El%20low%20code%20no%20code,infraestructura%20y%20expertos%20en%20programaci%C3%B3n>.
- Fernández-Acevedo, J. E. (Junio de 2017). *El Pensamiento Computacional y su relación con el desarrollo de la Creatividad en los niños y niñas del quinto grado de educación general básica de la Unidad Educativa San Felipe Neri de la ciudad de Riobamba*. Trabajo de titulación, Escuela Superior Politécnica de Chimborazo, Riobamba. Obtenido de <http://dspace.esPOCH.edu.ec/handle/123456789/6649>
- Flores, A. (2015). *Buscando la excelencia educativa: Gestión de procesos académicos y administrativos en Instituciones Públicas de Educación mediante BPM*. Valencia: Universidad Internacional de Valencia.
- Gobierno de España. (2023). *Low-Code (y No-Code), herramientas para democratizar la digitalización de empresas*. Fondo Europeo de Desarrollo Regional. España: red.es. Obtenido de [https://comercio.acelerapyme.gob.es/sites/acelerapyme/files/2023-04/Low-Code%20\(Y%20No-Code\)%2C%20herramientas%20para%20democratizar%20la%20digitalizaci%C3%B3n%20de%20empresas\\_0.pdf](https://comercio.acelerapyme.gob.es/sites/acelerapyme/files/2023-04/Low-Code%20(Y%20No-Code)%2C%20herramientas%20para%20democratizar%20la%20digitalizaci%C3%B3n%20de%20empresas_0.pdf)
- Gobierno del Ecuador. (2021). *Plan de Creación de Oportunidades 2021-2025*. Obtenido de <https://www.planificacion.gob.ec/wp-content/uploads/2021/09/Plan-de-Creacio%CC%81n-de-Oportunidades-2021-2025-Aprobado.pdf>
- González Firmas, X. (2025). *Metodologías ágiles vs. tradicionales: ¿cuál es mejor para la gestión de proyectos?* OBS Business School, Barcelona. Obtenido de

<https://www.obsbusiness.school/blog/metodologias-agiles-vs-tradicionales-cual-es-mejor-para-la-gestion-de-proyectos>

Hernández, E. (2025). *Preinscripción de Optativas GII Basado en Plataforma Office 365*. Madrid: Universidad Politécnica de Madrid.

Hernández-Sampieri, R., Fernández, C., & Baptista, P. (2014). *Metodología de la Investigación* (Sexta ed.). México: McGraw-Hill.

Hervas, C. (2024). *Transformando la Educación: tecnología, innovación y sociedad en la Era Digital*. Sevilla: Universidad de Sevilla.

Huatatoca, G. (2024). *Investigación de las plataformas No-Code, existentes en la actualidad para los desarrolladores de software orientadas al sitio web*. Tena: Instituto Superior Tecnológico Tena. Obtenido de <http://localhost:8080/jspui/handle/123456789/513>

Hurlburt, G. F. (2021). *Low-Code, No-Code, What's Under the Hood?* STEMCorp, USA. Obtenido de <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9655393>

Ihirwe, J. F. (2023). *Ingeniería de bajo código para el Internet de las cosas*. L'Aquila: University of L'Aquila. doi:<http://dx.doi.org/10.2139/ssrn.4539001>

International Organization for Standardization. (17 de 03 de 2025). *Normas ISO*. Obtenido de <https://normasiso.org/norma-iso-9241/>

Kirikova, M., & Rokis, K. (2022). *Desafíos del desarrollo de software de bajo código/sin código: una revisión de la literatura*. Universidad Técnica de Riga, Letonia. doi:[http://dx.doi.org/10.1007/978-3-031-16947-2\\_1](http://dx.doi.org/10.1007/978-3-031-16947-2_1)

Kirvan, P., & Pratt, M. K. (2024). *Platforms, low-code and no-code development*. Obtenido de <https://www.techtarget.com/searchsoftwarequality/definition/low-code-no-code-development-platform>

Lorenz, K. (2023). *Cómo las herramientas no-code están transformando la innovación*. Konrad Lorenz Fundación Universitaria. Obtenido de <https://www.konradlorenz.edu.co/blog/como-las-herramientas-no-code-estan-transformando-la-innovacion/>

- Lucas Garcia, J. (2021). *Programação No-code no ensino superior: Possibilidades de Avaliação e Aprendizagem Ativas*. Lisboa: Centro Universitário UNA.
- Mendoza Armijos, H. (2023). *Libro de Memorias II ACongreso internacional de investigación los Andes Innova 2023*. Quito: Congreso Internacional los Andes Innova.
- Minaya Vera , C., Arias Vera, I., Mendoza Vélez , O., & Minaya Vera , A. (2022). *El desarrollo Low/No-code y el futuro de losdesarrolladores de software*. Chone: Universidad Laica Eloy Alfaro de Manabí.
- Minaya, C., Mendoza, O., Arias, I., Minaya, A., & Bravo, H. (2022). *Low/No-code development platforms and the future of software developers*. (Vol. 1). Ecuador: MINERVA. doi:<https://doi.org/10.47460/minerva.v1iSpecial.76>
- Mohammedaziz, T. (24 de 03 de 2025). *UST*. Obtenido de <https://www.ust.com/es/insights/how-low-code-slash-no-code-supercharges-application-modernization>
- Molina, M. B., Cevallos, V. H., & Dávila, C. J. (2018). *Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software*. Espirales Revista Multidisciplinaria De investigación. Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=8466424>
- Moreno, J. (2022). *Low-Code vs. No-Code – La guía definitiva*. Flokzu. Obtenido de <https://flokzu.com/es/bpm-es/low-code-y-no-code-plataformas-de-workflow-y-bpm-dan-aire-a-it/>
- Mousalli Kayat, G. (2018). *Métodos y Diseños de Investigación Cuantitativa*. Merida: Universidad de los Andes.
- Ojeda, M. V., & Casas, S. (2024). *Caso de Estudio Comparativo de Plataformas Low-Code*. Santa Cruz: Unidad Académica Río Gallegos.
- Ortiz, Y. A., & Fernández, A. J. (2024). *Propuesta didáctica para el desarrollo del pensamiento computacional mediante la programación visual por bloques*. Riobamba: Universidad Nacional de Chimborazo.

- Parada, C. J., Puentes, P. R., & Rodríguez, J. d. (2020). *Análisis de las Metodologías Ágiles para el Desarrollo de Software*. Universidad Francisco de Paula Santander, Cúcuta. Obtenido de [https://www.tdea.edu.co/images/tdea/galeria/sello\\_editorial/ebooks/investigacion\\_innovacion\\_ingenieria\\_software\\_2.pdf](https://www.tdea.edu.co/images/tdea/galeria/sello_editorial/ebooks/investigacion_innovacion_ingenieria_software_2.pdf)
- Parra Arévalo, J. (01 de 11 de 2024). *Análisis del desarrollo de software en no desarrolladores*. Bogota: FundaciónUniversitariaSanMateo. Recuperado el 01 de 11 de 2024
- Pazmiño Barahona, C. P. (2024). *Herramientas para la gestión del desarrollo ágil de una aplicación web con el marco scrum*. Quito: Escuela Politécnica Nacional.
- Pérez, F. (2023). *Usabilidad (Norma ISO 9241)*. Icalia. Obtenido de <https://www.icalia.es/w/usabilidad>
- Rodelgo Álvaro. (2019). *Gestión ágil vs Gestión tradicional de proyectos ¿Cómo elegir?* Escuela de Negocios FEDA. Obtenido de <https://www.escueladenegociosfeda.com/blog/50-la-huella-de-nuestros-docentes/471-gestion-agil-vs-gestion-tradicional-de-proyectos-como-elegir>
- Roig, H. D. (2021). *Tecnologías Low-Code y No-Code: Un caso práctico para estudiar su potencial y limitaciones*. Valencia: Universitat Politècnica de València.
- Rueda López, N. (2024). *Scratch para la práctica Lógico Matemática con estudiantes de Sexto de básica de la Unidad educativa “Comandante General Atahualpa”*. Sangolquí: Instituto Superior Tecnológico Universitario RUMIÑAHUI.
- Salado, R. (2024). *Aplicación de flujos de trabajo científicos en dominios con procesamiento intensivo de datos*. Cordoba: Universidad de Córdoba.
- Salas, C. R., Pérez, C. C., & Bernal, B. J. (2024). *Gamificación mediante la App Inventor como didáctica educativa*. Quito: Red de Investigadores Latino Americanos.
- Salazar, M. T. (2023). *Low Code para optimizar el tiempo de desarrollo de software*. Santo Domingo: Universidad de las Fuerzas Armadas ESPE.

- Seguro, E. (2024). *No-Code y Triple Impacto: Una alianza innovadora*. Design & innovation company. Obtenido de <https://www.dmo.company/lab/www-dmo-company-lab-no-code-y-triple-impacto-una-alianza-innovadora>
- STEAMpedia. (03 de 12 de 2024). *Agilo Research Pvt. Ltd.* Obtenido de [https://thetempedia.com/product/pictoblox/?srsltid=AfmBOopoeD0CNC\\_bbxkdn2P4biFK1z9uKSSW697xh89pEwUu7cRjnhBg](https://thetempedia.com/product/pictoblox/?srsltid=AfmBOopoeD0CNC_bbxkdn2P4biFK1z9uKSSW697xh89pEwUu7cRjnhBg)
- Tamayo Ibáñez, J. (2023). *Low-code como alternativa a la digitalización de empresas. Desarrollo de un caso práctico con Appian*. Barcelona: d'Enginyeria Industrial de Barcelona.
- UNACH. (2024). Ajuste Curricular Sustantivo y No Sustantivo. 46. Riobamba.
- UNACH. (2025). *Perfil de Egresado de la carrera de Pedagogía de las Ciencias Experimentales: Informática*. Universidad Nacional de Chimborazo, Riobamba. Obtenido de <https://www.unach.edu.ec/licenciatura-en-pedagogia-de-la-informatica-ele/>