

Low Code para optimizar el tiempo de desarrollo de software

Low Code to optimize software development time

Low Code para otimizar o tempo de desenvolvimento de software

Milton Temistocles Andrade Salazar¹
Universidad de las Fuerzas Armadas ESPE Sede Santo Domingo
mtandrade@espe.edu.ec
<https://orcid.org/0000-0002-4929-3233>



Anthony Mauricio Goyes Díaz²
Universidad de las Fuerzas Armadas ESPE Sede Santo Domingo
amgoyes@espe.edu.ec
<https://orcid.org/0000-0001-6846-2488>



Joan Fernando Cevallos Sarango³
Universidad de las Fuerzas Armadas ESPE Sede Santo Domingo
jfcevallos7@espe.edu.ec
<https://orcid.org/0000-0002-0276-7307>



Selena Elizabeth Rivas Mera⁴
Universidad de las Fuerzas Armadas ESPE Sede Santo Domingo
serivas@espe.edu.ec
<https://orcid.org/0000-0001-6535-5038>



 DOI / URL: <https://doi.org/10.55813/gaea/ccri/v4/n2/261>

Como citar:

Andrade, M., Goyes, A., Cevallos, J. & Rivas, S. (2023). Low Code para optimizar el tiempo de desarrollo de software. *Código Científico Revista de Investigación*, 4(2), 927-937.

Recibido: 10/11/2023

Aceptado: 10/12/2023

Publicado: 31/12/2023

¹ Doctor en Ciencias Humanas por la Universidad del Zulia – Venezuela. Docente de la Universidad de las Fuerzas Armadas ESPE Sede Santo Domingo

² Estudiante de la Universidad de las Fuerzas Armadas ESPE Sede Santo Domingo

³ Estudiante de la Universidad de las Fuerzas Armadas ESPE Sede Santo Domingo

⁴ Estudiante de la Universidad de las Fuerzas Armadas ESPE Sede Santo Domingo

Resumen

La investigación mostró una comparativa sobre la influencia del uso de la tecnología Low Code en la optimización del tiempo de desarrollo de software mediante la implementación de un aplicativo para generar contrapuntos con relación a la programación tradicional. La investigación se basó en un paradigma cualitativo documental. Se realizó una investigación documental en la que se obtuvo información por medio de artículos científicos relacionados con la tecnología Low Code para desarrollar aplicaciones y el uso de la programación tradicional con lenguaje Java. Además, se implementó dos aplicativos usando el método tradicional de programación y el Low Code con la finalidad de dar a conocer su impacto como tecnología para desarrollo de software. Se logró determinar su eficiencia, concluyendo que el desarrollo de programas con low code es una buena alternativa para aquellas personas con pocos conocimientos de programación.

Palabras claves: Low Code, Power App, desarrollo de software, Netbeans, Java.

Abstract

This research showed a comparative study on the influence of the use of Low Code technologies in the optimization of development time, through the implementation of an application to generate counterpoints in relation to traditional programming. The research was based on a documentary qualitative paradigm. It began by conducting a documentary research in which information was obtained through scientific articles; where emphasis was placed on the use of the low code programming paradigm to develop applications and the use of traditional programming with Java language. In addition, a solution was generated through the creation of applications using the traditional programming method and the low code. It was possible to determine its efficiency, thus concluding that the development of programs with low code is a good alternative for those people with little programming knowledge.

Keywords: Low Code, Power App, software development, Netbeans, Java.

Resumo

A pesquisa mostrou um comparativo sobre a influência do uso da tecnologia Low Code na otimização do tempo de desenvolvimento de software através da implementação de uma aplicação para geração de contrapontos em relação à programação tradicional. A pesquisa baseou-se num paradigma documental qualitativo. Foi realizada uma investigação documental na qual foram obtidas informações através de artigos científicos relacionados à tecnologia Low Code para desenvolvimento de aplicações e ao uso da programação tradicional com a linguagem Java. Além disso, foram implementadas duas aplicações utilizando o método de programação tradicional e Low Code com o objetivo de divulgar seu impacto como tecnologia de desenvolvimento de software. Foi possível apurar sua eficiência, concluindo que o desenvolvimento de programas com low code é uma boa alternativa para quem tem pouco conhecimento de programação.

Palavras-chave: Low Code, Power App, desenvolvimento de software, Netbeans, Java.

Introducción

El desarrollo de software ayuda a resolver una gran cantidad de problemas, tanto para personas como empresas, sin embargo, para realizar dicha actividad se necesita profundos conocimientos en lenguajes de programación. Para muchas personas puede ser muy complicado aprender un lenguaje de programación es por eso que se ha pensado en tecnologías intuitivas que puedan ayudar a personas sin vastos conocimientos en programación a desarrollar productos de software. La tecnología empleada será low code, la cual supondrá una gran ayuda al momento de crear software, reemplazando el código exhaustivo por un método de programación más sencillo que use interfaz gráfica.

En la actualidad existen muchas empresas que se dedican al desarrollo de software dependiendo de sus necesidades. Sin embargo, cuando se habla de desarrollo de software existen diferentes factores que deben considerarse, estos factores pueden ser el capital, tiempo, la infraestructura, desarrolladores profesionales o personal, entre otros (Parra Arévalo, J. P., 2020). Si una empresa no cuenta con estos recursos lo más probable es que estos factores se conviertan en una barrera o un limitante para desarrollar software (Basantes Suñiga, Á. J., 2022).

La tecnología poco código (Low code) presenta una nueva forma de llevar a cabo el proceso de programación de ciertas aplicaciones, por esta razón su popularidad ha ido aumentando en conjunto con la demanda permiten que surjan diferentes soluciones con tecnología Low Code, llegando así a ser popular para el ámbito empresarial o personal. Low code tiene mucho potencial a la hora de desarrollar aplicaciones (Tapia Macías, C. A., 2021). Low-code es un software muy dinámico y fácil de usar para cualquier usuario. Para poder comprender los beneficios y rapidez que nos puede traer se pretende realizar una comparativa de rendimiento entre low-code y un lenguaje de programación. Por lado de low-code tenemos la herramienta Power Apps por Microsoft que nos ayudará a crear la aplicación desde cero sin

necesidad de programar grandes cantidades de líneas de código. Por otra parte, tenemos Eclipse - Java esta plataforma está basada en java de código abierto y se pueden incluir un gran número de plugins que nos permitirá el desarrollo de la aplicación y se debe manejar java para su elaboración (Chaudhary, H. A. A., & Margaria, T., 2021).

Como aporte se pretende introducir Low Code como alternativa para el desarrollo de software, mediante la presentación de datos comparativos obtenidos a partir de una aplicación realizada de manera tradicional, usando lenguaje de programación Java, y con la tecnología “Low Code”, además, la información recolectada a partir del análisis documental obtenido de las fichas de lecturas de los artículos científicos será utilizada para la comparativa final. Se obtuvo como resultado que para una persona particular que tenga pocos conocimientos en lenguajes de programación se ve ampliamente beneficiada con la integración de la tecnología Low Code en el desarrollo de software, reduciendo el tiempo de desarrollo y obteniendo un producto con mayor calidad (Roig Hervás, D., 2021). Se concluye que la tecnología Low Code es una alternativa poco viable para desarrolladores independientes o pequeñas y medianas empresas debido a sus limitaciones en su funcionalidad y que la mayor parte de sus soluciones proponen un sistema de paga. Sin embargo, los servicios que proponen sistemas usando tecnología Low Code cada vez son más, por lo que se espera que estas limitaciones se vean reducidas, volviéndose una excelente alternativa para el desarrollo de software.

Las plataformas que ofrecen desarrollo de software con low code disponen herramientas muy intuitivas, las cuales facilitan la tarea de desarrollar productos de software; Con esta investigación se pudo constatar que la tecnología low code es ideal para desarrollar proyectos de programación que no tengan un alto grado de complejidad.

Metodología

Se realizó una investigación profunda de documentación existente sobre la tecnología de desarrollo de software Low Code, donde se analizó sus características, ventajas y

desventajas con respecto a la programación tradicional al desarrollar una aplicación o sistema. Esta investigación fue realizada con la finalidad de evaluar todos los beneficios que conlleva adaptar este paradigma de desarrollo de software.

El trabajo realizado en (Eriksson Jirbratt, P., 2021) titulado “Análisis comparativo sobre la plataforma de desarrollo nocode glideapp y la plataforma low-code outsystems en la creación de aplicaciones web” menciona sobre la comparativa entre herramientas de software OutSystems y Low-code. Al realizar la comparativa queda claro que Low-code es superior, además de ser muy intuitiva para los diferentes profesionales ya que permite reducir el tiempo de programar código y lo único necesario para la elaboración de un aplicativo es saber lo que se va a crear y usar. Además, menciona que los profesionales al escoger entre las plataformas para desarrollo web y otras, se inclinan por Low-code ya que no presenta limitaciones y esto permite mayor flexibilidad en el desarrollo, tanto en lo funcional como en la personalización.

El proyecto se basó en documentación bibliográfica que mantenía un enfoque dirigido a los desarrolladores que comparaban la experiencia de seguir desarrollando en Power Apps y en Java. En base a la muestra bibliográfica se implementó una aplicación con tecnología Low Code y otra haciendo uso del lenguaje de programación Java. Con la ayuda de la metodología SCRUM se logró segmentar el proyecto en cinco etapas que consisten en desarrollar de manera funcional la operación suma, resta, multiplicación y división, siendo la etapa final, el diseño.

Para realizar la comparativa entre los paradigmas de programación propuestos llevaremos a cabo la creación de un programa sencillo que será una calculadora con las operaciones matemáticas básicas, dicho programa será creado primero con el lenguaje de programación Java en el IDE Eclipse y después será realizado usando la tecnología Low Code. Para el desarrollo de la aplicación tanto en Power Apps como en eclipse se aplicó la metodología de desarrollo de software Scrum, el cual se basa principalmente en la separación

de la integridad de las labores a hacer o Product Backlog que se irán desarrollando por bloques en periodos cortos de tiempo que vamos a llamar Sprints.

En la Figura 1 se muestra el diagrama de flujo propuesto para el aplicativo. Se hará uso de 2 variables para ingresar datos, además también se necesitará una variable para la opción de la operación y otra para alojar el resultado. Este flujo de trabajo se mantendrá para ambos escenarios.

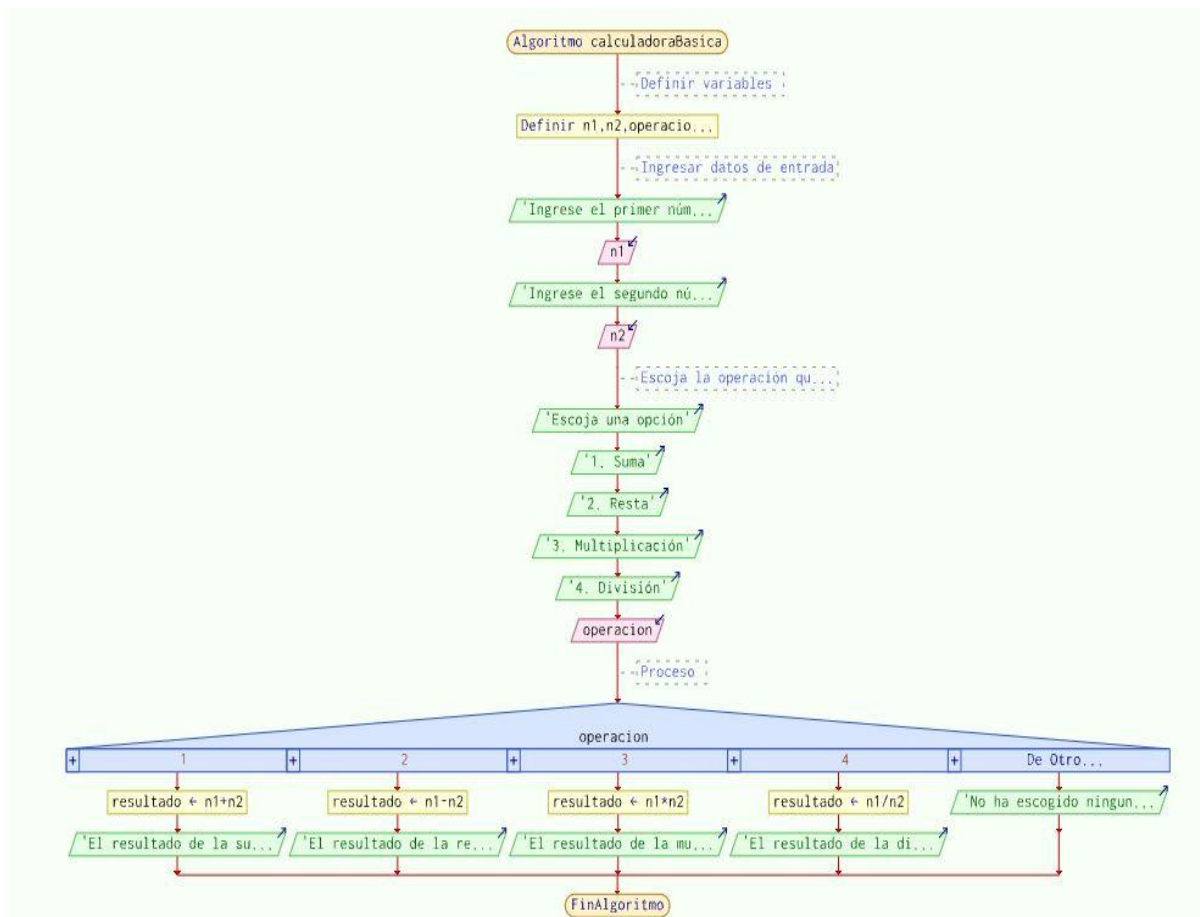


Figura 1: Diagrama de flujo del programa “calculadora”

Nota. Diagrama de flujo de la aplicación “calculadora” realizado en PSeint. Tiene la finalidad de dar a conocer la lógica de los algoritmos que se implementaron para el aplicativo, tanto con Low Code como en NetBeans con Java. En el siguiente enlace se puede visualizar la figura de mejor manera: <https://postimg.cc/Wd7bm36Z>.

Eclipse, es el entorno de desarrollo integrado (IDE) más usado para el desarrollo en Java. Es de código abierto y gratuito, por estas razones se seleccionó Eclipse como la herramienta comparativa contra la tecnología Low Code. En esta etapa se programará una calculadora con las cuatro operaciones básicas usando el lenguaje de programación Java dentro del IDE Eclipse.

Cabe destacar que para ello no usaremos ninguna herramienta de programación que le otorgue interfaz gráfica al programa, sino que será programado desde cero mediante las líneas de comando que brinda el IDE. Para obtener los datos en función al tiempo de desarrollo que permitan realizar la comparativa entre Low Code y Java se creó un programa de una calculadora básica. El resultado del programa que tiene las funciones como la suma, resta, multiplicación y división, la interfaz del programa se puede visualizar en la Figura 2.



Figura 2: Interfaz para el programa realizado en Eclipse

Nota. El aplicativo fue realizado en Eclipse con el lenguaje de programación Java. Para remarcar la diferencia del tiempo de desarrollo se realizó la interfaz solo con líneas de código.

Low Code: Considerando que Power Apps es un conjunto de plataformas, servicios y conectores que permite desarrollar aplicaciones propias de manera ágil e intuitiva. La manera más sencilla de comprobar su funcionamiento y rendimiento es mediante el uso de esta. Es así como se propone empezar por la creación del proyecto en Power Apps de Microsoft que es la tecnología de desarrollo de software Low Code que se utilizará para realizar el trabajo. Una vez creado el proyecto se debe definir el diseño que tendrá la aplicación por medio de un boceto preliminar para agilizar el proceso de desarrollo con ideas claras.

Debido a que se propuso desarrollar una aplicación básica se optó por una plantilla en blanco, de ese modo empezamos arrastrando etiquetas, combos de textos y botones que son los principales componentes de la calculadora y con lo que se le dará un diseño y buena apariencia. La parte más importante y se podría decir que la más complicada de desarrollar esta calculadora será la funcionalidad debido a que la sintaxis varía entre soluciones con tecnología Low Code.

Sin embargo, Power App tiene una gran biblioteca con guías detalladas para el desarrollo de aplicaciones con su tecnología. Para la funcionalidad únicamente se crearán variables, empezando por el primer y segundo número, que serán los datos que permitirán realizar la operación, además de los botones que accionan las operaciones básicas como suma, resta, multiplicación y división.

Para la creación de variables se utilizarán las instrucciones predefinidas por el software de Power Apps, mismas instrucciones que se colocarán en el apartado de propiedades de la componente ubicada en la parte superior, esta sección delimita el comportamiento del componente. Para las operaciones se determinó que irán variando. El resultado del software realizado en Power Apps se puede observar en la Figura 3.

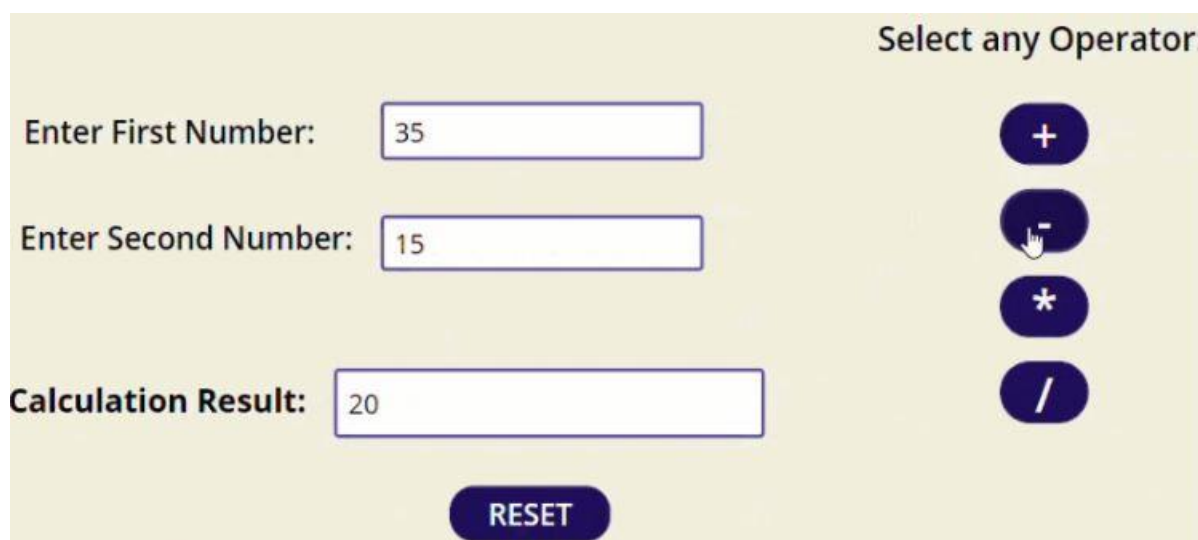


Figura 3: Interfaz para el programa realizado con tecnología Low Code

Nota. El aplicativo fue realizado con la tecnología Low Code usando la solución de Microsoft “Power Apps” usando su manual de desarrollo para concluir correctamente con el desarrollo del aplicativo.

Resultados

La presente investigación ha adoptado un enfoque comparativo entre la plataforma Low Code Power Apps y el desarrollo de software tradicional con lenguaje de programación Java. Con las etapas claramente identificadas se logró capturar el rendimiento de ambos paradigmas. La información obtenida se contabilizó en minutos de desarrollo, siendo esta la complejidad que establecimos, entre más tiempo de desarrollo equivale a más dificultad en su

implementación. Los resultados obtenidos fueron tabulados y presentados en la Tabla 1, donde se puede observar que el tiempo de desarrollo se mantiene similar en las primeras cuatro etapas, sin embargo, en la etapa de diseño se observa una clara diferencia del tiempo estimado. La razón de este comportamiento se debe a que existen componentes encapsulados que no necesitan programarse lo que ahorra de manera significativa el tiempo estimado en su implementación, logrando dedicar más tiempo al problema mencionado en lugar de la arquitectura del programa y las asignaciones porque están automatizadas.

Tabla 1
Comparativa entre Low Code y lenguaje Java en el tiempo de desarrollo

Paradigma	Etapas de desarrollo (en minutos)			
	Suma	Resta	Multiplicación	Diseño
Low Code	4.12	5.10	5.30	12.51
Eclipse - Java	5.26	5.44	7.51	29.55

Nota. El desarrollo de ambos aplicativos fue segmentado en cuatro etapas: suma, resta, multiplicación y el diseño; donde las tres primeras etapas hacen referencia a la funcionalidad y la última al diseño. Los tiempos presentados están dados en minutos.

La Figura 4 muestra de manera gráfica la tendencia que se obtuvo en relación con la etapa con el tiempo estimado para su implementación. Se puede observar claramente que en todas las etapas de la tecnología Low Code se tiene un menor tiempo por lo que su implementación es mucho más eficiente. La tecnología Low Code también tiene sus desventajas como la limitación en sus funcionalidades, si la acción requerida no se encuentra ya creado como un componente arrastrable (elemento que encapsula su comportamiento y estructura y que solo requiere usarse), no se podrá implementar. Este problema no existe con el desarrollo de software con lenguaje Java. Sin embargo, cada vez son más las aplicaciones que permiten trabajar con la tecnología Low Code, por lo que las posibilidades en sus funcionalidades se van a ir expandiendo conforme su popularidad.

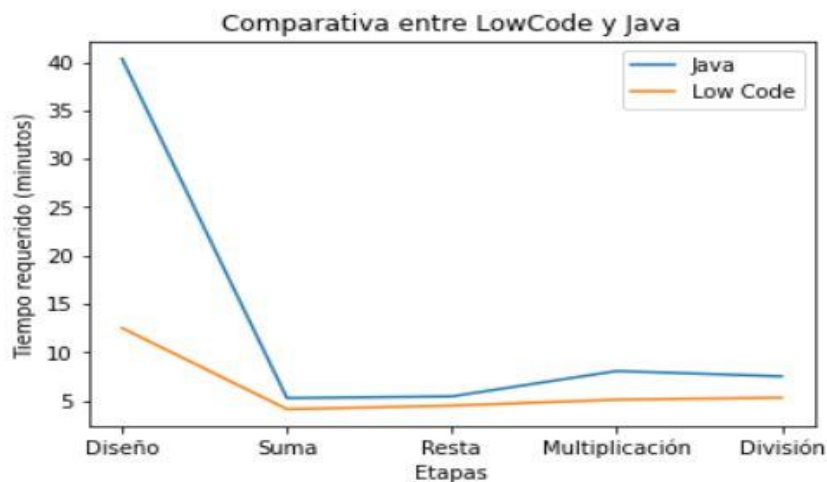


Figura 4: Tabulación del tiempo estimado de desarrollo entre Low Code y Java

Nota. La gráfica estadística muestra las cuatro etapas de desarrollo en relación con el tiempo en minutos que requirieron. Los datos propuestos están basados según la Tabla 1.

Conclusiones

La diferencia existente al momento de implementar aplicaciones propias entre la tecnología Low Code en Power Apps y la programación tradicional en Eclipse utilizando el lenguaje de programación Java, es la fácil implementación y buenos resultados que se obtienen siguiendo las guías detalladas que cada propuesta brinda. Por lo tanto, Low Code permite desarrollar aplicaciones en un menor tiempo y con mejores resultados, en un enfoque hacia la optimización del tiempo de desarrollo.

Las existentes limitaciones de Low code en la plataforma Power Apps respecto a su funcionalidad como las pocas funciones al momento de implementar, incluso la limitación de creatividad la vuelve poco atractiva de utilizar en grandes empresas. Sin embargo, es una buena alternativa para empresas pequeñas o para desarrolladores independientes que requieran una tecnología poco robusta y fácil de manipular, obteniendo como resultado un producto final atractivo. Además, el uso de la tecnología permite, a los desarrolladores, dedicar más tiempo en generar una solución al problema en lugar de enfocarse en la estructura que tendrá, haciendo uso de las tareas automatizadas.

A pesar de que programar en Java supone una mayor dificultad para desarrollar software, esta opción sigue siendo más eficiente a comparación de Low Code en escenarios

donde una organización o empresa quisiese desarrollar algún software complejo para satisfacer alguna necesidad específica por sus limitaciones en su funcionalidad y que la mayoría de las plataformas que trabajan con la tecnología Low Code son de paga. Sin embargo, los servicios que proponen sistemas usando tecnología Low Code están aumentando, por lo que se espera que estas limitaciones se vean reducidas significativamente en un futuro cercano, volviéndose una excelente alternativa para el desarrollo de software.

Referencias bibliográficas

- Basantes Suñiga, Á. J. (2022). Análisis comparativo sobre la plataforma de desarrollo No-Code Glideapp y la Plataforma Low-Code Outsystems en la creación de aplicaciones web (Bachelor's thesis, Babahoyo: UTB-FAFI. 2022).
- Chaudhary, H. A. A., & Margaria, T. (2021). Integration of micro-services as components in modeling environments for low code development. Труды Института системного программирования РАН, 33(4), 19-30.
- Eriksson Jirbratt, P. (2021). The difference between software development on Low Code platform Camunda and software development in Java.
- Parra Arévalo, J. P. (2020). Análisis del desarrollo de software en no desarrolladores (Doctoral dissertation, Editorial Universitaria San Mateo).
- Roig Hervás, D. (2021). Tecnologías Low-Code y No-Code: Un caso práctico para estudiar su potencial y limitaciones (Doctoral dissertation, Universitat Politècnica de València).
- Tapia Macías, C. A. (2021). Desarrollo de aplicaciones multiplataforma usando Frameworks y plataformas Low-code: Un caso de estudio.