

<https://doi.org/10.47460/minerva.v1iSpecial.76>

El desarrollo Low/No-code y el futuro de los desarrolladores de software

Minaya Vera Cristhian Gustavo
<https://orcid.org/0000-0003-0143-6810>
cristhian.minaya@uleam.edu.ec
Universidad Laica Eloy Alfaro de Manabí
Chone-Ecuador

Mendoza Vélez Oswaldo Vicente
<https://orcid.org/0000-0002-8823-8384>
oswaldo.mendoza@uleam.edu.ec
Universidad Laica Eloy Alfaro de Manabí
Chone-Ecuador

Arias Vera Irina Loreley
<https://orcid.org/0000-0001-6494-4471>
loreley.arias@educacion.gob.ec
Ministerio de Educación
Daule-Ecuador

Minaya Vera Andrés Alexander
<https://orcid.org/0000-0001-7081-0242>
andres.minaya@educacion.gob.ec
Secretaría de Educación Superior, Ciencia,
Tecnología e Innovación
Daule-Ecuador

Bravo Vera Henry Fabricio
<https://orcid.org/0000-0001-6213-3544>
e1310286842@live.uleam.edu.ec
Universidad Laica Eloy Alfaro de Manabí
Chone-Ecuador

Recibido(16/04/2022), Aceptado(19/05/2022)

Resumen. - Low/No-code es una tendencia de la industria del desarrollo software que procura proveer a aquellas personas que no poseen una formación específica en informática, de herramientas fáciles de usar para la creación de sus propios aplicativos o programas con el objetivo de cubrir una necesidad empresarial específica. En el mercado existen múltiples plataformas que permiten a los usuarios crear sus propias soluciones informáticas con una mínima, o en algunos casos nula necesidad de codificación. En este trabajo se exploraron los orígenes de esta tendencia y cómo ha sido tanto su evolución como su adopción dentro del mundo del desarrollo de software y empresarial. También se realizó un análisis sobre las consecuencias que esta tendencia proyecta para los profesionales del software en el futuro próximo.

Palabras clave: Low code, no code, desarrollo de software, aplicaciones informáticas.

Low/No-code development platforms and the future of software developers

Abstract. - Low/No-code is a trend in the software development industry that seeks to provide those without specific computer training with easy-to-use tools to create applications or programs to meet a particular business need. In the market, multiple platforms allow users to create software solutions with some code, or in some cases, no need for coding. This paper explores the origins of this trend and how it has evolved, and its adoption in software development. The investigation also analyzed the implications of this trend for software professionals over the near future.

Keywords: Low code, no code, software development, computer applications.

I. INTRODUCCIÓN

La migración de la información desde documentos físicos a su equivalente en formato digital inició lo que ahora se conoce como la transformación digital [1]. La digitalización de documentos, con el ahorro de recursos humanos y materiales [2], marcó el inicio de un proceso donde el desarrollo tecnológico ha transformado tanto a la sociedad como a las empresas, modelando con el tiempo tanto la manera en que las personas interactúan entre sí como también la forma en que las empresas organizan sus actividades [3]. A modo general, la transformación digital se asocia con la forma en que el desarrollo tecnológico, en su amplia acepción, está cambiando cómo se desenvuelven las actividades humanas dentro de la sociedad [4]. Esto se puede observar de las siguientes maneras: 1) en la automatización de procesos, lo que implica la incorporación de robots y de software especializado en el control de ciertos procesos industriales o sociales y 2) la transformación de servicios y productos a su equivalente digital [5].

Los seres humanos cada vez más utilizan recursos digitales para desempeñar sus actividades diarias. La tendencia de un uso cada vez mayor de herramientas web y de aplicativos móviles, para los denominados teléfonos inteligentes o smartphones, ha elevado los requerimientos de información y el flujo de datos [6]. Esta demanda cada vez mayor ha llevado a las empresas y organizaciones a la contratación de personal especializado en desarrollo de software, quienes han venido dando respuestas a estas exigencias, desarrollando aplicaciones y software según las necesidades expuestas por las compañías que los contratan. Sin embargo, el ritmo de requerimientos cada vez es mayor; las compañías, para mantenerse competitivas necesitan cada vez más soluciones que sean ágiles, pertinentes en un tiempo determinado y que cumplan con las exigencias cada vez mayores de los usuarios [1]. Es en este escenario es donde el desarrollo Low/No-code aparece como una solución a la brecha de requerimientos de programación demandado por las empresas, las organizaciones y los estados ante la falta de personal cualificado para ello. El paradigma de desarrollo de aplicaciones sin codificación (No-code) o con un grado bajo de ella (Low-code) se han convertido en una tendencia en ascenso dentro de la industria de desarrollo de software. Algunas consultoras de tecnología estiman que las aplicaciones y desarrollos de software que utilizarán esta tecnología será del 65% en el 2024 [7]. Empresas como Microsoft, Outsystem o Appian lideran el mercado de plataformas Low-code, ofertando soluciones donde las personas no formadas en áreas informáticas, pero sí conocedoras de los giros de sus negocios o de sus nichos de mercado, puedan crear aplicaciones que satisfagan sus requerimientos sin la necesidad de contratar a todo un equipo de desarrolladores que produzcan una solución a la medida de sus necesidades, pero a costes que tal vez no pueda permitirse. A medida que la inteligencia artificial y el aprendizaje automático ayudan a mejorar las plataformas en cuanto a usabilidad, seguridad, escalabilidad, complejidad e interoperabilidad, se debate cuál será el aporte de los profesionales del software dentro de las industrias en el futuro próximo.

Este trabajo se enfoca en dos aspectos, uno descriptivo, donde se tocan con más profundidad, temas como las razones que permitieron el surgimiento del desarrollo Low/No-code, sus ventajas y desventajas y las plataformas que lideran estos tipos de desarrollo de software. El otro aspecto toca las consecuencias que está teniendo en el mercado de desarrolladores de software la creación y uso de este tipo de tecnología, tanto como la aparición del usuario no informático como desarrollador de aplicaciones. Para ello se realizó una exploración bibliográfica sobre el tema, escogiendo aquellos trabajos que tocaran los temas desarrollados en este trabajo y al final, mediante el análisis de los artículos y los trabajos escogidos, ofrecer resultados y conclusiones sobre la temática tratada obtenidos en la investigación.

II. DESARROLLO

Low/No-code como respuesta a la problemática del software

A. Los problemas fundamentales del desarrollo de software

Los sistemas requeridos por las compañías van ganando en complejidad con el paso de los años; al mismo tiempo, las ventanas de oportunidad que estas observan, para poder explotar ventajas de mercado, son cada vez más estrechas. Esta situación presiona a los desarrolladores a entregar productos con un mayor grado de sofisticación en períodos de tiempo cada vez más cortos. Lo que, aunado con un déficit de talento en todas las áreas relacionadas con el desarrollo del software, exacerbaban los problemas que ya arrastraba la industria desde sus inicios: sistemas que no funcionan como se requería, o que eran simplemente incorrectos; plazos de entrega de productos que no se cumplen, presupuestos rebasados o agotados, entre otros problemas. La solución más empleada para hacer frente a los plazos de entrega y la remediación de problemas era aumentar el personal en los departamentos, subcontratar personal externo para encargarse de aspectos menos técnicos de proyecto, o si no existía otra alternativa, contratar empresas para afrontar proyectos completos. La irrupción de la pandemia del COVID-19, produjo consigo un impulso exponencial a la digitalización acelerada de varios aspectos de la actividad económica, lo que provocó que el talento en desarrollo de software fuese cada vez más demandado, profundizando la crisis en el sector. Esta escasez de talento fue la razón principal en el ascenso de las aproximaciones Low/No-code dentro de la industria del desarrollo de aplicaciones, ya que este aspecto es fundamental para la creación y entrega de proyectos.

B. Low-code y No-code

Puede parecer que los términos low-code y no-code pueden ser denominaciones alternativas de un mismo concepto, sin embargo, existen diferencias importantes. Low-code es un término que acuñó en 2014 el Forrester Research [8] y que se refiere a un paradigma o enfoque de desarrollo de software en el cual no se requiere de conocimientos profundos de programación, en comparativa con el desarrollo tradicional, para la creación y puesta en funcionamiento de un proyecto de software. No obstante, la persona sí debe poseer conocimientos de programación para utilizar con propiedad y dominio las plataformas basadas en ese enfoque [3].

En cambio, las plataformas no-code están pensadas para usuarios no informáticos, pero que poseen un conocimiento importante sobre el giro del negocio de una organización o empresa. Esto le permite tener una perspectiva clara sobre las necesidades a las que debe responder un aplicativo para cubrir los requerimientos a los que debe dar solución. Por lo general el diseño empleado por los desarrolladores de estas plataformas, para permitirle al usuario el acceso a las herramientas de desarrollo, consiste en representar a las funcionalidades que ofrece la plataforma como bloques que el usuario puede arrastrar y soltar en un área donde se pueden conectar entre sí [3].

El objetivo de las plataformas Low/No-code son muy similares, proveer de entornos que agilicen el desarrollo y despliegue de aplicaciones digitales en lapsos de tiempo más cortos que los acostumbrados al utilizar un enfoque de programación tradicional. A pesar de que existen divergencias notables entre las capacidades y en el despliegue de funcionalidades entre diferentes plataformas, la tendencia general indica que, en un futuro no muy lejano, las plataformas Low/No-code tendrán un nivel de accesibilidad, adaptabilidad y potencia que serán actas para el uso tanto de usuarios con pocas nociones de informática, como de usuarios con conocimientos técnicos en desarrollo de software más avanzados.

III. METODOLOGÍA

Low/No-code como respuesta a la problemática del software

A. Los problemas fundamentales del desarrollo de software

Los sistemas requeridos por las compañías van ganando en complejidad con el paso de los años; al mismo tiempo, las ventanas de oportunidad que estas observan, para poder explotar ventajas de mercado, son cada vez más estrechas. Esta situación presiona a los desarrolladores a entregar productos con un mayor grado de sofisticación en períodos de tiempo cada vez más cortos. Lo que, aunado con un déficit de talento en todas las áreas relacionadas con el desarrollo del software, exacerbaban los problemas que ya arrastraba la industria desde sus inicios: sistemas que no funcionan como se requería, o que eran simplemente incorrectos; plazos de entrega de productos que no se cumplen, presupuestos rebasados o agotados, entre otros problemas. La solución más empleada para hacer frente a los plazos de entrega y la remediación de problemas era aumentar el personal en los departamentos, subcontratar personal externo para encargarse de aspectos menos técnicos de proyecto, o si no existía otra alternativa, contratar empresas para afrontar proyectos completos. La irrupción de la pandemia del COVID-19, produjo consigo un impulso exponencial a la digitalización acelerada de varios aspectos de la actividad económica, lo que provocó que el talento en desarrollo de software fuese cada vez más demandado, profundizando la crisis en el sector. Esta escasez de talento fue la razón principal en el ascenso de las aproximaciones Low/No-code dentro de la industria del desarrollo de aplicaciones, ya que este aspecto es fundamental para la creación y entrega de proyectos.

B. Low-code y No-code

Puede parecer que los términos low-code y no-code pueden ser denominaciones alternativas de un mismo concepto, sin embargo, existen diferencias importantes. Low-code es un término que acuñó en 2014 el Forrester Research [8] y que se refiere a un paradigma o enfoque de desarrollo de software en el cual no se requiere de conocimientos profundos de programación, en comparativa con el desarrollo tradicional, para la creación y puesta en funcionamiento de un proyecto de software. No obstante, la persona sí debe poseer conocimientos de programación para utilizar con propiedad y dominio las plataformas basadas en ese enfoque [3].

En cambio, las plataformas no-code están pensadas para usuarios no informáticos, pero que poseen un conocimiento importante sobre el giro del negocio de una organización o empresa. Esto le permite tener una perspectiva clara sobre las necesidades a las que debe responder un aplicativo para cubrir los requerimientos a los que debe dar solución. Por lo general el diseño empleado por los desarrolladores de estas plataformas, para permitirle al usuario el acceso a las herramientas de desarrollo, consiste en representar a las funcionalidades que ofrece la plataforma como bloques que el usuario puede arrastrar y soltar en un área donde se pueden conectar entre sí [3].

El objetivo de las plataformas Low/No-code son muy similares, proveer de entornos que agilicen el desarrollo y despliegue de aplicaciones digitales en lapsos de tiempo más cortos que los acostumbrados al utilizar un enfoque de programación tradicional. A pesar de que existen divergencias notables entre las capacidades y en el despliegue de funcionalidades entre diferentes plataformas, la tendencia general indica que, en un futuro no muy lejano, las plataformas Low/No-code tendrán un nivel de accesibilidad, adaptabilidad y potencia que serán actas para el uso tanto de usuarios con pocas nociones de informática, como de usuarios con conocimientos técnicos en desarrollo de software más avanzados.

C. Ventajas de utilizar plataformas Low/No-code

Los objetivos de las plataformas Low/No-code son principalmente, reducir la complejidad y los tiempos de desarrollo y entrega de productos digitales [9]. Sin embargo, se pueden destacar otros beneficios que traen consigo su empleo. Entre ellas se pueden destacar:

- **Disminución de costos:** al disminuir el intervalo temporal de desarrollo de un producto y los recursos destinados a él, se contraen los costos generados en su desarrollo. Esto puede aplicar tanto si utiliza el talento humano de la empresa u organización como si se externalizar su producción.
- **Rapidez:** es uno de los aspectos de mayor importancia para las empresas. Al estar desarrollado desde el inicio gran parte o toda la programación necesaria, los desarrolladores se pueden enfocar en aspectos como la usabilidad, la configuración de la interfaz y en la experiencia del usuario. Todo esto gracias a que el tiempo de desarrollo en estas plataformas con respecto al desarrollo tradicional es entre 5 y 10 veces más veloz [8].
- **Competitividad:** esto debido a que pueden sacar productos de manera ágil, satisfaciendo los requerimientos más complejos de sus clientes, manteniendo las ventajas estratégicas frente a sus competidores [10].
- **Facilidad de uso:** al permitir a personas con escasos o nulos conocimientos de programación, puedan por sí mismo desarrollar aplicaciones que satisfagan una necesidad digital presente en sus organizaciones o empresas, provoca incentivos positivos en diferentes áreas productivas. Por lo general estas personas, que no son profesionales de la programación o el desarrollo de software, poseen otras habilidades y conocimientos operativos que les permiten identificar los que requieren tanto los clientes como sus propios departamentos de trabajo. Estos trabajadores son especialmente útiles cuando las demandas de soluciones digitales son abundantes en organizaciones con limitados recursos de TI [8].
- **Disminución de la complejidad:** como las aplicaciones se desarrolla a partir de recursos y funcionalidades ya diseñadas y que están disponibles para su implementación, los equipos de desarrollo pueden centrarse en la personalización y en la mejora continua de la experiencia del usuario al utilizar la aplicación.
- **Seguridad:** la escasez de personal de TI, con frecuencia condiciona la prioridad de los proyectos que pueden gestionar; dicha situación relega a los proyectos más pequeños al nivel de prioridades menores, por lo que los empleados de departamentos, o incluso los mismos profesionales de TI subcontratan a terceros para atender estos requerimientos. A esto se lo denomina shadow IT, o TI en las sombras [11]. Esta situación compromete la seguridad en TI de las empresas y organizaciones, ya que no estaría en su control los temas de manejo de datos y de seguridad. Con las plataformas Low/No-code el personal de TI puede coordinar a los usuarios programadores para que generen en estas plataformas aplicaciones que cumplan con sus necesidades sin para ello emplear directamente al personal de TI. Esta estrategia puede ahorrar tiempo en pruebas de seguridad, privacidad y optimización, ya que, si el software está desarrollado bajo estos parámetros, los aplicativos estarán más rápido en producción [11].
- **Bajos costos de mantenimiento:** este beneficio puede observarse cuando se requiere actualizar los servicios disponibles para mantenerse a la vanguardia de los requerimientos de los usuarios y del segmento del negocio. En el caso específico de las plataformas low-code, debido al que es poca la cantidad de programación ejecuta, es más fácil y a la vez menos costoso de actualizar y de mantener [9].

- Integración de perfiles profesionales a labores TI dentro de las organizaciones: al aprovechar los conocimientos de los profesionales en diferentes áreas de las compañías u organizaciones, para el desarrollo de productos digitales, dichas organizaciones ganan en flexibilidad y en competitividad, pues aprovechan los conocimientos integrados de todas las áreas, pues son estos quienes conocen las necesidades del negocio. Por ello el 44% de usuarios de plataformas Low/No-code son de usuarios de perfiles profesionales no asociados a TI, pero con supervisión de estos profesionales [10] [9].
- Reducción de funcionalidades innecesarias: Al poder desarrollar productos básicos para probar una cantidad mínima de ideas y funcionalidades en las plataformas Low/No-code que pueden satisfacer los requerimientos de los usuarios, evitando utilizar tiempo y recursos valiosos en funcionalidades y requerimientos que no tendrán el favor de los usuarios [9].

D.Desventajas

Entre las principales razones que se conocen por las que las empresas u organizaciones no adoptan enfoques de desarrollo Low/No-code es el desconocimiento de las mismas [10]. Luego están el depender de una plataforma particular, problemas asociados a la seguridad y la personalización de las aplicaciones y problemas con respecto a la escalabilidad. Toquemos estos temas a continuación.

Escalabilidad y personalización: por lo general, en las plataformas Low/No-code los bloques de construcción están diseñados para un conjunto básico de funcionalidades [12] que, si bien puede acortar el tiempo de desarrollo de aplicación no muy complejas, les pone por detrás de lo que puede ofrecer la programación tradicional en cuanto a personalización y escala. Por tanto, para aplicaciones que requieran un alto grado de personalización, como las utilizadas para ciencias de datos, las plataformas Low/No-code no responder a dichas exigencias. Para que las plataformas Low/No-code puedan ofrecer niveles de personalización y de complejidad que equiparen a la programación tradicional puede tardar un tiempo considerable y mucho trabajo [13].

- Fragmentación del desarrollo de aplicaciones: no todas las plataformas Low/No-code ofrecen la posibilidad de trabajar bajo diferentes enfoques de programación según los requerimientos de las empresas o compañías [9].
- Orientación al usuario: con frecuencia, las personas que utilizan las plataformas Low/No-code no provienen del área de programación o de sistemas, son profesionales de ingeniería experimentados en otras áreas, que desean aplicar sus conocimientos en los aplicativos que están desarrollando, empleando modelos y estándares conocidos por ellos [9].

E.Plataformas Low/No-code líderes

Aunque existe más de 300 plataformas que ofrecen servicios basadas en este tipo de tecnologías, solo algunas lideran en este rubro [14]. Sólo se mostrarán, para realizar un ejercicio comparativo, cuatro de las plataformas que mejor se encuentran posicionadas en el mercado.

F.Microsoft Power Apps

El gigante tecnológico en su plataforma ofrece un conjunto de herramientas integradas que le permiten a los usuarios el desarrollo de aplicaciones cuya escala se comprende entre baja a media. La plataforma de Microsoft está siendo utilizada principalmente por empresas de servicios y de manufactura, sin contar con un 43% de compañías de diversos segmentos económicos que también la usan [14].

Como se observa en la Fig. 1, Microsoft una ruta de desarrollo consistente en cuatro bloques de funcionalidad, Power BI, que es una herramienta que te ayuda a analizar y visualizar datos con el objetivo de obtener información crítica para la toma de decisiones empresariales. Power Apps, que junto con Power Automate conforman las herramientas con la cual se desarrollan las aplicaciones; tiene integradas tanto servicios como conectores y un entorno de datos para pruebas que luego se puede enlazar con las bases de datos de los clientes de la empresa. Además, se puede integrar con otros servicios de Microsoft como MS Office 365, MS Azure, MS Teams, entre otros. Por último, ofrecen un soporte para la creación, si se requiere, de asistentes virtuales con Power Virtual Agents, así pueden dar seguimiento y soluciones a problemas y solicitudes enviadas a la empresa.

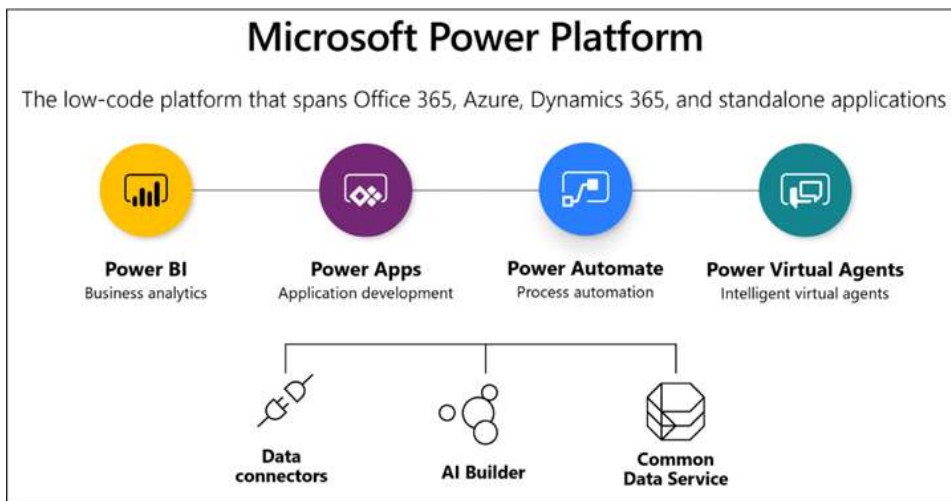


Fig. 1. Plataforma Low/No-code de Microsoft [15].

G.Appian

Appian es otra plataforma de desarrollo Low/No-code (Fig.2). Tiene una gran presencia en la industria de los servicios financieros de donde provienen el 40% de sus clientes [14]. Se destaca principalmente por ofrecer una interfaz de desarrollo que posibilita la construcción de aplicaciones con un mayor grado de complejidad, en donde se requiera un nivel de automatización y de análisis de datos avanzados. En el apartado de análisis, Appian ha incorporado Process Mining, una herramienta de análisis con Inteligencia Artificial, a su plataforma; esta permite, a través de un análisis histórico, la determinación de todas aquellas variables que pueden introducir ineficiencias en el sistema de tal forma de predecirlos y evitarlos. Otra funcionalidad que ha introducido Appian es la posibilidad de seguir trabajar offline desde un móvil [16].

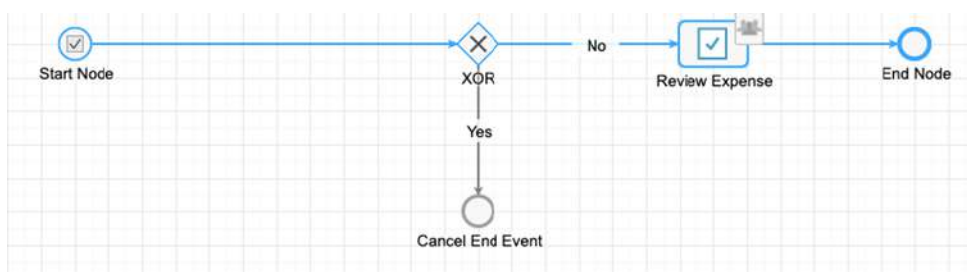


Fig. 2. Modelo simplificado de Procesos de Appian [17].

B.Outsystems

OutSystems fue pionera en el establecimiento de las plataformas Low/No-code como alternativa a la programación tradicional [1], enfocándose en ofrecer la construcción rápida de soluciones de software (Fig. 3), tanto para ecosistemas móviles como para web [18]. La plataforma ofrece un entorno integrado que abarca de principio a fin el ciclo de vida de desarrollo: desarrollo, pruebas de calidad, implementación, monitoreo y soporte. En la fig. 3 se puede observar de manera esquemática la integración de la arquitectura de trabajo, donde se muestran los dispositivos de acceso a la plataforma, los servicios, las interfaces de usuario, los módulos de desarrollo, de control calidad e implementación, donde además se puede monitorear mediante reportes de desempeño. Outsystem, al igual que otras plataformas como Appian y Microsoft Power Platform, se apoyan en el uso de Inteligencia Artificial y Aprendizaje automático para brindar apoyo a los usuarios brindándoles recomendaciones, automatización de procesos y ciclos de pruebas y validaciones de desempeño de las aplicaciones desarrolladas en la plataforma [11].

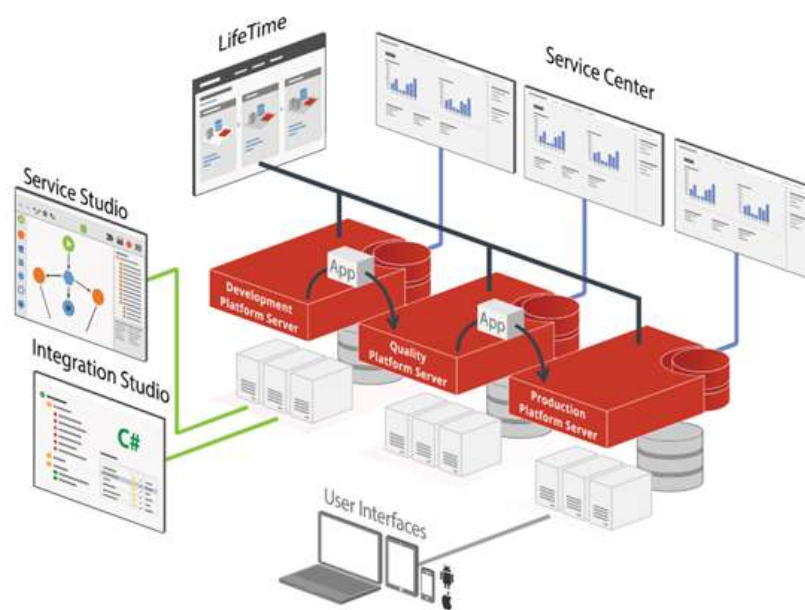


Fig. 3. Arquitectura de trabajo en Outsystems [18].

B.Mendix

Es otra plataforma que lidera el segmento de desarrollo Low/no-code (Fig. 4). De la misma forma que las otras plataformas, brinda apoyo al desarrollador mediante técnicas de inteligencia artificial y aprendizaje automático; sin embargo, existen dos modalidades diferenciadas según el usuario. Para el usuario con el perfil de desarrollador ciudadano Mendix le ofrece el entorno Mendix Studio, donde la IA le puede dar al usuario principiante, guías, alternativas o sugerencias de desarrollo, una ruta de pruebas de desempeño y calidad, además de detección de errores [11]. Para desarrolladores más experimentados, Mendix tiene Medix Pro, con un entorno de desarrollo similar a los utilizados por los profesionales de software. Consiste en un entorno web en el cual se puede acceder y ver los proyectos en curso. Se puede conectar el proyecto a una base de datos propia o a una facilitada por Mendix. También se puede crear una hoja de cálculo y Mendix puede crear un proyecto. Por lo expuesto, el enfoque es similar al de un proyecto de ingeniería de software, se puede hacer seguimientos del proyecto, guardar versiones, y hacer validaciones de funcionalidad del proyecto en entornos de prueba. En este sentido, la plataforma de Midex se comporta con lo haría un IDE [19]. En la fig. 4 se ve un ejemplo de validación de un formulario en el tablero de diseño de Medix Studio.

Mendix visual designer

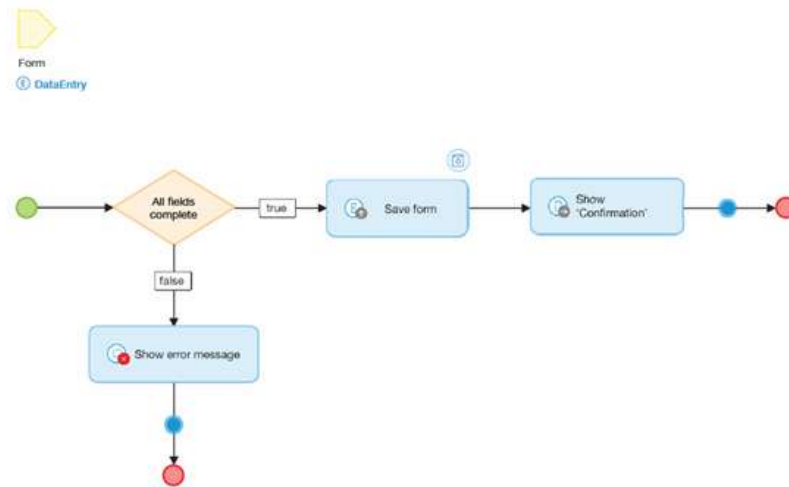


Fig. 4. El tablero de diseño de Medix [19].

II. METODOLOGÍA

En esta investigación se utilizaron fuentes bibliográficas referente a temas de desarrollo low-code y no-code y que tuviesen relevancia en torno a la temática. Se consultaron fuentes de acceso abierto de bases de datos como Scopus, IEEE, ArxivX, y portales especializados en consultoría y tecnología. Para ellos se utilizaron palabras clave para la inclusión de documentos, las cuales fueron las siguientes:

- Low code.
- No code.
- Desarrollo de software.
- Desarrollo de aplicaciones.

Además de los criterios antes mencionados, se consideraron aspectos como:

- Especificaciones de funcionamiento de plataformas Low/No-code.
- Estadísticas importantes sobre el impacto de las plataformas de desarrollo Low/No-code.
- -Trabajos que desarrollaran aplicaciones basadas en Low/No-code.
- -Análisis sobre la importancia del desarrollo Low/No-code de cara al futuro por consultores y expertos.

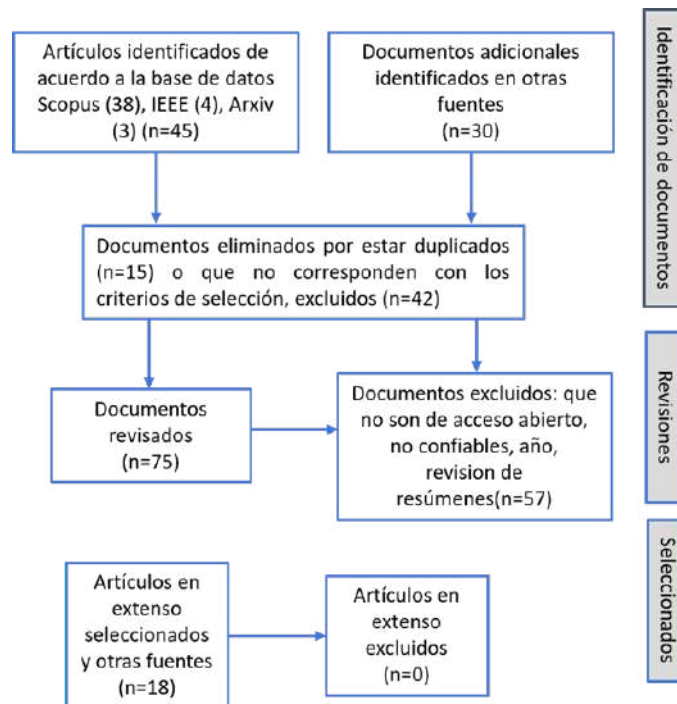


Fig. 5. Criterio de selección bibliográfica.

De la selección de archivos se realizó según se muestra en la Fig. 5, donde primero se hizo una búsqueda exploratoria por las bases de datos identificadas, realizando una revisión de 99 documentos que relacionados con la temática del desarrollo con aproximación Low/No-code, de ellos se escogieron 10 artículos que se comprendían dentro de los parámetros de selección. Además, se agregaron 8 archivos de páginas de consultoría y de expertos entorno al desarrollo Low/No-code, para un total de 18 referencias.

IV. RESULTADOS

Una vez escogido el material de trabajo, segmentada entre trabajos académicos, páginas de consultoría en tecnología, conferencias y material de las propias desarrolladoras de plataformas Low/No-code se identificaron los siguientes resultados:

- Las tecnologías Low/No-code (de baja o nula programación) se caracterizan por hacer mejorar tres aspectos inherentes a la programación y al desarrollo del software, que son: en primera instancia, la curva de aprendizaje de la plataforma. Todos los lenguajes de programación y todas las plataformas de desarrollo tradicional poseen curvas de aprendizaje mayores que las que se observan en el aprendizaje y manejo de las plataformas con un enfoque de poca o nula programación. La asistencia virtual resulta ser una ventaja potente como guía al principiante; a esto se le añade la disposición de los elementos dentro de la interfaz de usuario, que facilitan su identificación y uso. Todo esto les permite a las personas que inician en el desarrollo de soluciones de software en estas plataformas, alcanzar un nivel de experticia adecuado para emprender proyectos más rápido que en las plataformas tradicionales. Otro aspecto importante es la programación en sí misma. Al no requerir programación o requerir de muy poca, los profesionales de otras áreas que no están relacionadas con informática y computación no necesitan conocer los detalles más técnicos del desarrollo de software y pueden centrarse en la lógica y la abstracción del funcionamiento que necesitan en las aplicaciones que requieren crear. Por último, las plataformas de desarrollo Low/No-code, al involucrar en el desarrollo de soluciones de software a personal de las organizaciones sin tener que contratar más personal o subcontratar a terceros, les permite ahorrar recursos valiosos, apoyando la competitividad y la eficiencia organizativa.

- Las investigaciones muestran que existen segmentos económicos en los cuales los desarrollos en plataformas basadas en poca o ninguna codificación ganan espacio. Los sectores manufactureros, de servicios financieros y de servicios en general, muestran una clara tendencia en la adopción de estas plataformas [9] y una distribución de uso que abarca importantes zonas geográficas de del mundo [11].
- Las plataformas Low/No-code se han posicionado como la solución de desarrollo de software con mayor acogida frente a la problemática de la falta de personal cualificado para cubrir todas las necesidades que requieren las empresas y organizaciones actualmente. Sin embargo, aún no se percibe como una solución completamente viable para sustituir el desarrollo tradicional, debido a que la curva de demanda de personal en estas áreas es todavía elevada. Si bien muchos analistas ven una evolución positiva en las capacidades tecnológicas de las plataformas low/No-code, todavía la industria está en una fase temprana tanto de adopción como de desarrollo.
- La evidencia más clara de la fase temprana en que se encuentran las plataformas de desarrollo Low/No-code es las falencias que más destacan de estas plataformas. Problemas de escalabilidad, dependencia y obsolescencia en el soporte, profundidad y personalización de las aplicaciones indican que existe todavía mucho trabajo por hacer en estas áreas a fin de que pueda ser un competidor de pesos en los años venideros. Las consultoras como Gartner o Forrester pronostican el establecimiento de estas tecnologías con una tasa de penetración de hasta el 75% para 2024, sin embargo, está por confirmarse si las compañías desarrolladoras pueden conseguir el crecimiento tecnológico que requieren y mantenerse en la vanguardia del sector.

IV. CONCLUSIONES

La transformación digital empuja a las compañías a cada vez más digitalizar su presencia en los mercados en los cuales funcionan. Si además de añaden factores inmanentes como la pandemia del COVID-19, el resultado es la aceleración de la demanda de personal tecnológicamente cualificado en áreas de software y desarrollo de aplicaciones móviles y web. El desarrollo de las plataformas Low/No-code nace como respuesta a la carencia sistemática de dicho personal. Esta aproximación, que nace como una propuesta de solución a las necesidades de crecimiento y adaptación de las compañías y organizaciones a sociedades que utilizan cada vez más soluciones digitales en su día a día, ha traído a discusión temas entorno a esta realidad, como los roles del profesional de la programación en un futuro cercano y la manera en que se está configurando incluso el futuro de las carreras que se asocian a este sector, las comunidades que se formaran entorno a los desarrollos Low/No-code e incluso la desaparición de la programación tradicional.

En primera instancia, en cuanto a la pertinencia del desarrollo Low/No-code, es indudable que su adopción positiva indica que realmente está dando respuestas a necesidades reales del mercado de desarrollo de aplicaciones. La cantidad de plataformas que existen, que son 300 aproximadamente, indican también que, en poco tiempo, aquellas que mejor se adapten a las necesidades de las empresas y los usuarios serán las que sobrevivan y seguirán creciendo, mientras que las demás desaparecerán o se convertirán en plataformas de nichos específicos. En esta línea, se observa que aquellas compañías que desarrollan sus herramientas apoyándose en inteligencia artificial y aprendizaje automático están progresando con mayor velocidad. La evolución del mercado y de los requerimientos de los usuarios hace prever que la nueva transición será hacia un desarrollo preponderantemente No-code o sin programación.

Con respecto al rol del profesional del software en medio de esta nueva transición tecnológica, mucho se comenta de la democratización del desarrollo de aplicaciones y de la figura del desarrollador ciudadano como nuevo actor en la industria. Sin embargo, tanto las consultoras de tecnología como las investigaciones académicas no ahondan en lo que permite a las nuevas plataformas Low/No-code seguir progresando en su presencia de mercado. Dentro de cada plataforma estarán profesionales con amplios

Como no existe una investigación académica extensa del tema, y muy poca como tal se centra en las necesidades de formación de los profesionales que quieran estar capacitados en esta área, una investigación sistemática sobre metodologías de enseñanza que puedan dar respuesta a esta necesidad se revela como muy necesaria. sin embargo, se puede ya esperar que, cualquier enfoque o aproximación al tema debe tomar en cuenta la necesidad de incluir un tratamiento multidisciplinario de la enseñanza, tomando en cuenta que en este nuevo escenario, no solo los profesionales del desarrollo de software estarán interesados en formarse en el conocimiento de las temas relacionados con informática, y que, según las estadística sobre la cantidad de personas formándose en tecnologías asociadas al desarrollo, no llegan á más allá del 2% de la población mundial. Por último, y teniendo en consideración aquellas actividades en las cuales el desarrollo Low/No-code está creciendo, como la manufactura, los servicios financieros, y los servicios en general, se requiere mayor investigación en relación entre las tecnologías asociadas a la industria 4.0 con las plataformas de desarrollo de este tipo. Cuál sería el grado de interoperabilidad o funcionalidad requerida y cómo pueden transformar los procesos industriales y de qué manera pueden afectar a la sociedad.

Por tanto, es necesario realizar más investigaciones sobre las plataformas Low/No-code, se requiere hacer pruebas de operación, desarrollo, funcionalidad e implementación, haciendo las pruebas de flexibilidad, accesibilidad, seguridad, diseño, tiempo de entrega de productos entre otros aspectos y compararlos con resultados similares utilizando una aproximación de desarrollo tradicional.

REFERENCIAS

- [1] H. A. Alsaadi, D. T. Radain, A. M. M., W. F. Alshammary, D. Alahmadi y B. Fakieh, «Factors that affect the utilization of low-code development platforms: survey study,» *Romanian Journal of Information Technology and Automatic Control*, vol. 31, n° 3, pp. 123-124, 2021.
- [2] J. C. Metrolho, F. Ribeiro y A. Ricardo, «A strategy for facing new employability trends using a low-code development platform.,» de 14th International Technology, Education and Development Conference., Valencia, 2020.
- [3] M. Moskal, «No-code application development on example of Logotec App Platform,» *Informatyka, Automatyka, Pomiar y Gospodarcze i Ochronie Środowiska*, vol. 11, n° 1, pp. 54-57, 2021.
- [4] E. Stolterman y A. C. Fors, «Information and technology for the good life,» *International Federation for Information Processing*, vol. 143, n° 1, pp. 687-692, 2004.
- [5] E. Henriette, M. Feki y I. Boughzala, «The Shape of Digital Transformation: A Systematic,» de MCIS 2015 Proceedings, Samos, 2015.
- [6] Y.-H. Chang y C.-B. Ko, «A Study on the Design of Low-Code and No Code Platform for Mobile Application Devdelopment,» *International Journal of Advanced Smart Convergence*, vol. 6, n° 4, pp. 50-55, 2017.
- [7] G. Hurlburt, «Low-Code, No-Code, What's Under the Hood?,» Meryland, 2021.
- [8] C. Richardson y J. R. Rymer, «The Forrester Wave™: Low-Code Development Platfirms Q2 2016,» Forrester, Washington D. C., 2016.
- [9] R. Sanchis, O. García-Perales, F. Fraile y R. Poler, «Low-Code as Enabler of Digital Transformation in Manufacturing Industry,» *Applied Sciences*, vol. 10, n° 1, pp. 12-30, 2021.
- [10] OutSystems, «The State of Application Development Is IT Ready for Disruption?,» Outsystems, Boston, 2019.
- [11] Z. Yang, «Arxiv.com,» 21 december 2021. [En línea]. Available: <https://arxiv.org/abs/2112.14073>. [Último acceso: 30 May 2022].
- [12] M. Woo, «The Rise of No/Low Code Software Development—No Experience Needed?,» *Engineering*, vol. 6, n° 2020, pp. 960-961, 2020.
- [13] J. Stangarone, «MRC,» michael, Ross & Cole, ltd. , 28 March 2019. [En línea]. Available: <https://www.mrc-productivity.com/blog/2019/03/pros-and-cons-of-low-code-development-platforms/>. [Último acceso: 29 May 2022].
- [14] . Kulkarni, «Deciphering Low-Code/No-Code Hype – Study of Trends, Overview of Platforms, and Rapid Application Development Suitability,» *International Journal of Scientific and Research Publications*, vol. 11, n° 7, pp. 536-540, 2021.

-
- [15] Microsoft, «Microsoft News Center Latinoamérica,» Microsoft, 30 November 2021. [En línea]. Available: <https://news.microsoft.com/es-xl/aplicaciones-low-code-de-microsoft-para-potenciar-la-transformacion-digital/>. [Último acceso: 1 junio 2022].
- [16] B. Fernández, «Silicon Technology Powering Business,» 27 Abril 2022. [En línea]. Available: <https://www.silicon.es/la-plataforma-de-appian-mucho-mas-que-low-code-2456612>. [Último acceso: 20 Mayo 2022].
- [17] Appian, «Process Modeling Tutorial,» Appian, 2022. [En línea]. Available: https://docs.appian.com/suite/help/22.1/Process_Modeling_Tutorial.html. [Último acceso: 1 June 2022].
- [18] R. Martins, F. Caldeira, F. Sá, M. Abbasi y P. Martins, «An overview on how to develop a low-code application using OutSystems,» de 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, 2020.
- [19] M. Heusser, «Tech Target,» 23 March 2021. [En línea]. Available: <https://www.techtarget.com/searchsoftwarequality/tip/A-closer-look-at-6-popular-low-code-development-platforms>. [Último acceso: June 5 2022].