Universidad de Alcalá
Escuela Politécnica Superior

GRADO EN INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Trabajo Fin de Grado

Automatización de trabajos: estudio de soluciones con capacidades low-code y no-code

Autor: Álvaro Luengo Antoñanzas

Tutor: Antonio Moratilla Ocaña

2023

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

GRADO EN INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Trabajo Fin de Grado

Automatización de trabajos: estudio de soluciones con capacidades low-code y no-code

Autor: Álvaro Luengo Antoñanzas

Tutor: Antonio Moratilla Ocaña

TRIBUNAL:

Presidente: Iván González Diego

Vocal 1º: Ángel Javier Álvarez Miguel

Vocal 2º: Antonio Moratilla Ocaña

FECHA: 06/07/2023

Agradecimientos

Me gustaría sobre todo agradecer a mi familia por haberme animado durante todos estos años, su apoyo ha sido una enorme ayuda para afrontar esta etapa de mi vida. También a aquellos profesores que han demostrado la capacidad que tienen los docentes de ilusionar a sus alumnos y hacer que amen aquello que están estudiando.

Índice

1. Resumen

En un mundo donde escasean los profesionales formados en materias de informática, problema acentuado tras la última pandemia de Covid-19, se ha vuelto imprescindible el desarrollo de soluciones que permitan a las empresas suplir esta falta de profesionales, a la vez que aumentan la productividad de la compañía.

Este trabajo se enfocará en las tecnologías low-code y no-code como herramientas que mejoran el rendimiento de las empresas, eliminando tareas rutinarias mediante la automatización de procesos, así como su potencial capacidad para suplir la carencia de profesionales del sector. Estas aproximaciones al problema que suponen las tareas rutinarias, permiten a las empresas enfocarse en lo que de verdad aporta valor al negocio.

Se creará una aplicación mediante el uso de herramientas low-code y no-code, como ejemplo de su uso y capacidad para desarrollar aplicaciones complejas con un mínimo gasto de tiempo y recursos. Además, se verán ejemplos de herramientas, comparando sus características y potencia de uso, escogiendo las mejores opciones para aquellos que deseen aplicar estos conceptos al ámbito empresarial.

Palabras clave

Software, low-code, no-code, productividad, end-user, innovación, automatización.

2. Abstract

In a world where engineers are lacking, problem that has been worsened by the last pandemic of Covid-19, it has become essential to develop solutions that allow businesses to bypass the lack of qualified professionals, improving at the same time the overall productivity of the company.

This paper work will focus in low-code and no-code technologies as tools through which improve performance in companies by eliminating daily tasks, using process automatization, and the potential to eliminate the lack of qualified professionals in this sector. These approaches to the problem of daily and repetitive tasks allows companies to focus on the things that add real value to the business.

An app will be created using low-code and no-code tools as an example of their use and capability to develop complex applications with minimum time and resources. To add up, research examples of other low-code/no-code software will be presented, comparing their characteristics and use scope, choosing the best of them for those who might want to apply these concepts in the business realm.

Keywords

Software, low-code, no-code, productivity, end-user, innovation, automation.

3. Introducción

Las tecnologías low-code y no-code son cada vez más populares en el mundo empresarial por su capacidad de automatizar tareas rutinarias. Se espera que su uso no haga más que crecer en los próximos años debido a su potencia para optimizar procesos de negocio.

La principal ventaja que ofrecen estas tecnologías es que permiten a los usuarios crear aplicaciones y productos software sin necesidad de tener conocimientos de programación avanzados. Esto habilita la creación de aplicaciones de forma más rápida y eficiente. Además, permite a las empresas desarrollar soluciones tecnológicas con un modelo más accesible y económico, mejorando la productividad en el proceso de desarrollo.

De esta forma, las tecnologías low y no-code también permiten a las empresas y organizaciones adaptarse rápidamente a las necesidades del mercado, debido a que pueden crear y actualizar aplicaciones más rápidamente que si tuvieran que contratar personal especializado (en este caso programadores) para crearlas desde cero.

Teniendo en cuenta las oportunidades que ofrecen estas tecnologías al desarrollo software y a la optimización de los procesos en las empresas, se han planteados diversos campos en los cuales estas tecnologías pueden expandirse, con el propósito de aumentar sus capacidades. Así, el futuro de las tecnologías low-code y no-code es prometedor, ya que cada vez más empresas y organizaciones están adoptando estas herramientas para transformar el modelo de desarrollo de aplicaciones de negocio, empoderando al trabajador para realizar sus propias aplicaciones sin depender del apartado técnico de la misma.

En resumen, se espera que las tecnologías low-code y no-code sigan creciendo y ganando terreno en el futuro, ya que ofrecen una solución eficiente y accesible para la automatización de procesos rutinarios, los cuales consumen tiempo a las empresas, así como una mayor accesibilidad en el desarrollo de aplicaciones software.

4. Estado del arte

Desde el origen del concepto de programación, se han hecho grandes esfuerzos por facilitar el proceso de creación de código, así como su aprendizaje, haciéndolo menos técnico y más accesible a las personas. Desde la creación del lenguaje máquina, ha habido un gran proceso de abstracción que se ha desarrollado hasta llegar a los lenguajes de alto nivel utilizados hoy en día. Así, las tecnologías Low-Code y No-Code no son más que un paso más en la progresión natural de este proceso, facilitando y reduciendo, o eliminando en algunos casos, el proceso de codificación.

El término low-code fue acuñado por Forrester en 2014. Estas tecnologías se basan en el movimiento "Citizen Developer" por el cual se promueve que cualquier persona sea capaz de realizar sistemas y aplicaciones software (democratización del desarrollo).

Con el paso del tiempo, el enfoque en las empresas se ha ido alejando del cómo programar una aplicación o solución software a cómo utilizar los programas ya disponibles de forma que se consiga el mayor beneficio para el modelo de negocio de la compañía. Así, el foco se ha ido poniendo poco a poco en el "end-user" o usuario final. Estos usuarios finales carecen de la capacidad o interés de convertirse en programadores, pero pueden utilizar las herramientas que ofrecen estos sistemas para potenciar sus actividades, ej: enviar correos, crear gráficos, etc. Así, lo que pueden hacer está atado a las capacidades del software.

Ya existen herramientas que exploran a este grupo de personas, con el fin de empoderarlas en materia de desarrollo software. Un ejemplo sería la programación mediante demostración. Esta es una herramienta sencilla pero que puede llegar a ser muy efectiva para simplificar tareas rutinarias. Algunas aplicaciones ya implementan algo similar en forma de macros. De esta manera, el usuario final muestra cómo realizar una serie de acciones y el sistema codifica dichas acciones en un programa, para poder ser utilizadas infinitamente en distintos contextos.

De esta forma, se eliminan los intermediarios al permitir al usuario interactuar directamente con la aplicación mediante, por ejemplo, simples clics. Además, también se elimina la abstracción que supone un lenguaje de programación, al trabajar con instancias concretas del programa. Finalmente, se minimizan los problemas de una sintaxis compleja, haciendo que sea el sistema el que escriba el código.

Cabe aclarar que algunos productos que se ofertan como soluciones low-code o no-code ya existían antes de que surgieran estos conceptos y solo se han adaptado a este nuevo fenómeno, modificando y desarrollando funcionalidades y cambiando la forma en la que se publicita su producto. Todo esto con la finalidad de llegar a los nuevos compradores. Con todo esto, algunos de los productos que se hacen llamar Low-Code o No-Code no son nuevos, integrando incluso en algunos casos, aplicaciones ya conocidas dentro de un mismo entorno.

Características comunes de las soluciones Low-Code.

Al poner el foco en aquello que ofrecen las soluciones low-code (las más exploradas en la actualidad), se pueden encontrar características comunes que permiten entender con mayor facilidad lo que ofrecen estas tecnologías. Así, estas características representan lo mínimo que necesita una solución para ser considerada Low-Code.

La primera de estas características, es el poseer elementos para la definición de estructuras de datos, la mayoría siendo variantes del modelo de entidad-relación. Siguiente, tenemos el uso de APIs para acceder a datos externos, como por ejemplo JDBC. Esta última es existencial si se quiere que la herramienta low-code pueda conectarse a las diversas aplicaciones que se desee. Otra característica que deben poseer las soluciones Low-Code es la capacidad de diseñar interfaces gráficas de usuario (GUI) y la opción de integrarlas con los diversos elementos del programa. Los sistemas Low-Code también han de proveer un sistema básico de especificación de funciones, lo que le dota del componente "low" code en contraposición al no-code, el cual no incluye funciones ni nada que se asemeje a un proceso de codificación. La mayoría de los sistemas low-code lo hacen ofreciendo un lenguaje simple de definición de reglas de decisión, las cuales controlan el flujo.

Muchas de las herramientas también incluyen un soporte avanzado de implementación de aplicaciones, las cuales permiten desplegar la aplicación desarrollada. La forma de hacerlo varía, ya que en algunas soluciones hace falta descargar la aplicación mientras que en otras se puede hacer a través de la web. Otra funcionalidad que incluye la mayoría de las soluciones Low-Code es la posibilidad de configurar los permisos y usuarios que podrán utilizar nuestra aplicación.

Finalmente, una característica que solo incluyen las opciones más completas (ya que esta corresponde en mayor medida a las soluciones no-code) es la del diseño de workflows. Estos flujos permiten al usuario especificar, sin necesidad de programar, las acciones que ha de ejecutar la aplicación, el orden en el que ha de realizarlas y en su caso, el trigger, o detonante, que desencadena la acción.

Diferencias entre low-code y no-code

Una vez que se han explorado las características que comparten gran parte de las tecnologías low-code la pregunta más sensata sería ¿Qué diferencia entonces al no-code del low-code?

Hay tres aspectos en los que ambas tecnologías difieren. Estos son la flexibilidad, el nivel de complejidad y la velocidad en el desarrollo de aplicaciones.

Debido a su capacidad para incorporar aspectos típicos de los entornos de programación, las herramientas low-code tienen una mayor capacidad de adaptación a los cambios durante el proceso de desarrollo en comparación con las de no-code, las cuales están fuertemente basadas en plantillas que están predeterminadas. De esta forma, en los entornos low-code, el usuario es capaz de modificar los componentes y funciones del sistema en base a sus necesidades.

En relación con la anterior, el lector podrá intuir un mayor nivel de complejidad en el entorno de desarrollo low-code en comparación con el de no-code. Y es que en las

primeras el usuario ha de disponer de algún conocimiento de programación previo, si bien es cierto que este no ha de ser un conocimiento extensivo. Las herramientas no-code en cambio, no requieren ningún conocimiento previo en este campo.

Por último, se tiene la velocidad de desarrollo. Cada una de las dos tecnologías tiene un enfoque distinto, como ya se ha podido comprobar. Las tecnologías no-code, al estar enfocadas al desarrollo de aplicaciones simples, ofrecen un menor tiempo de desarrollo de los mismos. Los usuarios pueden hacer uso de plantillas predefinidas para componer una aplicación en un tiempo reducido. En cambio, las tecnologías low-code, al estar enfocadas a aplicaciones de mayor complejidad, requieren de un mayor tiempo de desarrollo.

Se podría resumir que las herramientas low-code son más complejas y flexibles, mientras que las herramientas no-code son más simples y rápidas en el desarrollo de aplicaciones de menor complejidad. Las dos opciones comparten la misma filosofía y características básicas, si bien cada una es más apropiada en un contexto o en otro por lo que es crucial realizar un estudio del sistema que se pretende desarrollar con el fin de escoger la herramienta que mejor le convenga.

LCDP vs MDE

Es importante mencionar que, en algunos casos, las aplicaciones low-code o LCDP (Low Code Development Platforms) se mencionan indistintamente con las MDE (Model-Driven Engineering) y, aunque compartan muchas similitudes, hay ciertas características que las distinguen entre sí.

MDE (Model Driven Engineering) y LCDP (Low Code Development Platforms), junto con la tecnología no-code, son tecnologías que muchas veces pueden parecer homónimas, si no la misma. Por eso es crucial discernir las diferencias y similitudes entre ambas.

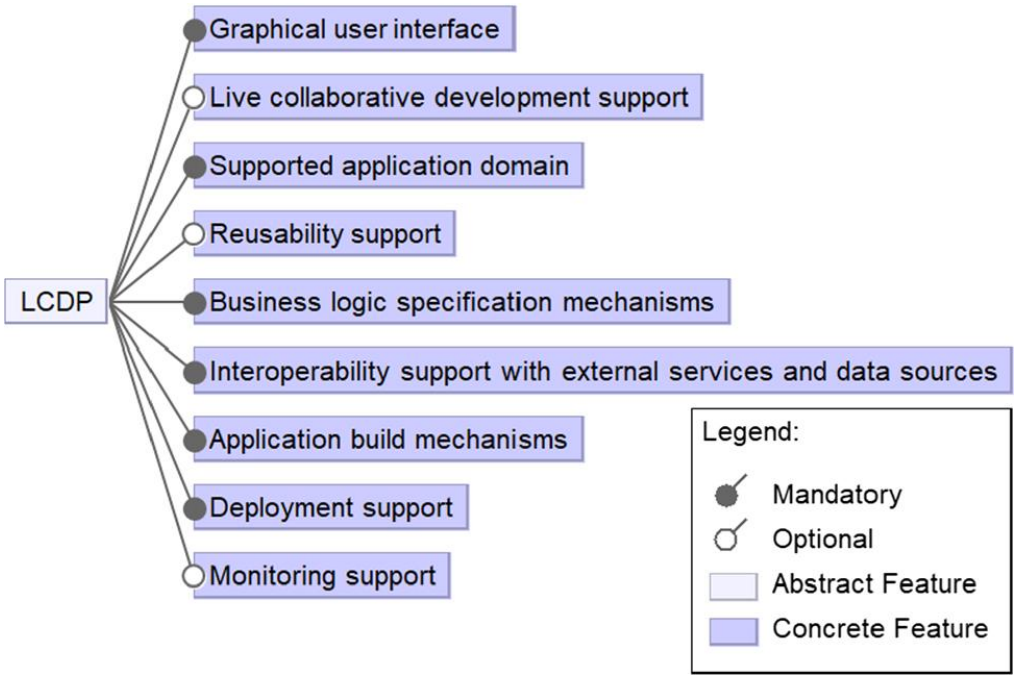
Primero, los aspectos que caracterizan a los MDE. Estos priorizan el uso de modelos como principal herramienta de desarrollo. Los modelos hacen referencia a una representación abstracta de un sistema o proceso, que se utiliza como base para la generación automática de código y la creación de aplicaciones y sistemas informáticos. De esta forma, en vez de realizar el proceso de codificación de forma manual, el desarrollador hace uso de modelos para describir el comportamiento y la funcionalidad del sistema. Una vez se ha finalizado este apartado, se utilizan herramientas de generación de código para traducir esos modelos en código funcional.

Así, se necesita un lenguaje de modelado específico para crear dichos modelos. Con este, el usuario es capaz de crear atributos, entidades, relaciones o interacciones con la base de datos, entre muchas cosas. Un buen ejemplo de esto sería el popular UML, o lenguaje de modelado universal.

De esta forma, algunos MDE, hacen uso de modelos en la automatización de procesos de verificación, simulación, optimización e ingeniería inversa, sin reducir la complejidad del código. Además, los MDE están enfocados sobre todo a profesionales, no a usuarios finales, ya que su objetivo final no es el de facilitar el entendimiento del código, sino que se centran en agilizar procesos de ingeniería. Finalmente, hay algunos MDE que reducen la complejidad o la cantidad de código, pero no gestionan el ciclo de vida de la aplicación

ni la implementación de la misma. Así, es importante resaltar que la reducción de la cantidad de código es una característica adicional que algunas soluciones ofrecen y no algo estándar a las tecnologías MDE.

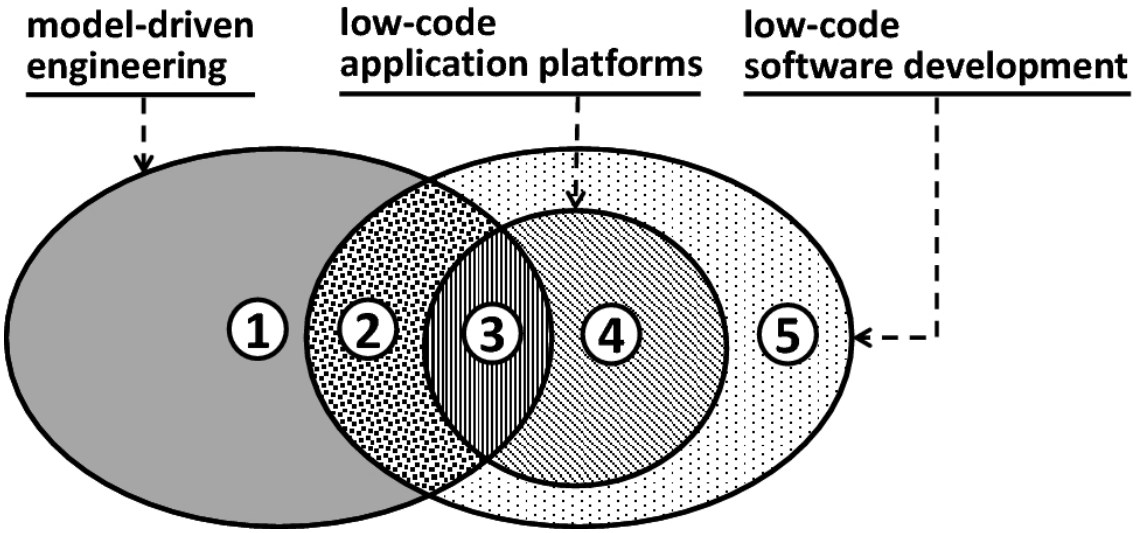
Por otra parte, las tecnologías Low-Code enfatizan el uso de interfaces como principal herramienta de desarrollo. Estas facilitan el desarrollo de aplicaciones con código reducido, con gestión del ciclo de vida y la implementación de métodos de despliegue de la aplicación. Además, la mayoría son web services, lo que libera al usuario de tener que instalar aplicaciones, o tener datos "on premise". Esto simplifica las cosas y hace más fácil el acceso de nuevos usuarios a estas aplicaciones (ya que Low-Code se enfoca a los usuarios finales). A diferencia del enfoque MDE, las tecnologías Low-Code sí aplican de base la filosofía de reducción de proceso de codificación en sus productos, ya que este es su principal valor diferencial.



Elementos de una herramienta Low Code [1]

De esta forma, mientras que los MDE se enfocan en mejorar la productividad de ciertos procesos, las tecnologías low-code y no-code en enfocan en el desarrollo y despliegue de aplicaciones mediante el uso de técnicas visuales.

Sin embargo, tanto en el MDE como en el Low-Code hay herramientas que solapan características entre estos dos tipos de tecnologías. Así, hay ejemplos de aplicaciones MDE que reducen la cantidad de código, gestionan el ciclo de vida de la aplicación o incluyen la implementación y despliegue de la misma.



Elementos comunes de Low Code y MDE [1]

De esta forma, se observa la importancia del uso de las interfaces en las aplicaciones low-code y no-code como medio para permitir a personas sin trasfondo en el mundo de la tecnología, crear y desplegar aplicaciones de negocio con facilidad, en contraposición con las MDE, las cuales se especializan en ser una solución para usuarios especializados, centrada en mejorar la eficiencia de procesos concretos.

Ventajas de las herramientas Low-Code y No-Code

Las herramientas low-code y no-code proveen de múltiples ventajas a las empresas.

Para empezar, proveen de una mayor privacidad. Al facilitar y hacer más accesible el proceso de codificación, las empresas pueden delegar dicho proceso en los propios empleados de la compañía, aunque estos no tengan perfiles con un trasfondo tecnológico, evitando tener que subcontratar servicios. De esta forma, el desarrollo se realiza de manera interna, evitando así la salida de información de la empresa.

También proveen de una mayor rapidez en el desarrollo. La parte principal del código ya está desarrollada, los usuarios solo tienen que programarla visualmente.

Con todo esto, la implementación de estos sistemas reduce el tiempo de desarrollo, lo que a su vez reduce el ciclo de vida y por lo tanto también se reducen los costes.

Las herramientas low-code reducen la complejidad de los sistemas. Al no desarrollar las aplicaciones desde cero, se parte desde una base común y como consecuencia se reduce consistentemente la complejidad.

Otra ventaja es que simplifica el mantenimiento ya que, a menor complejidad, mayor facilidad a la hora de revisar el correcto funcionamiento del sistema. También hay que resaltar el factor importante de la reusabilidad. Por ejemplo, AppCloud (Salesforce) ofrece el AppExchange, un Marketplace donde se publican aplicaciones prefabricadas, objetos y elementos que se pueden reutilizar para crear aplicaciones nuevas.

Finalmente, hay involucramiento de perfiles de negocio. Al no requerir conocimientos de programación, numerosos perfiles dentro de la empresa pueden convertirse en desarrolladores, democratizando así el proceso de desarrollo.

Desventajas de Low-Code y No-Code

Las tecnologías low-code y no-code presentan numerosas ventajas, pero también tienen desventajas que cabe destacar.

Escasez de personalización

El proceso de codificación ofrece al usuario la capacidad de personalizar en su totalidad el programa que quiere desarrollar, adaptándolo a las especiales necesidades de la empresa o persona encargada del mismo. Al desarrollar un proceso de abstracción de los distintos lenguajes de programación y al centrarse en potenciar la reutilización, se pierde capacidad de personalización en pro de una mayor rapidez en el desarrollo.

Escalabilidad

A día de hoy las tecnologías Low-Code no son escalables. Esto es comprensible teniendo en cuenta que están enfocadas al desarrollo de pequeñas aplicaciones de forma rápida. Sin embargo, no están preparadas para grandes desarrollos, donde se necesita un alto nivel de detalle y personalización. Además, algunas plataformas no están preparadas para manejar grandes volúmenes de transacciones, lo que puede afectar al rendimiento de la aplicación.

Dependencia del proveedor

A diferencia de los lenguajes de programación, las herramientas low-code y no-code son suministradas por empresas proveedoras, lo que significa que el desarrollo, mantenimiento y servicio técnico de la misma recae en la compañía que provea el servicio. De esta forma, la empresa depende de los servicios de terceros para que su aplicación funcione, lo que puede suponer problemas a futuro, por ejemplo, si el proveedor descontinúa el producto.

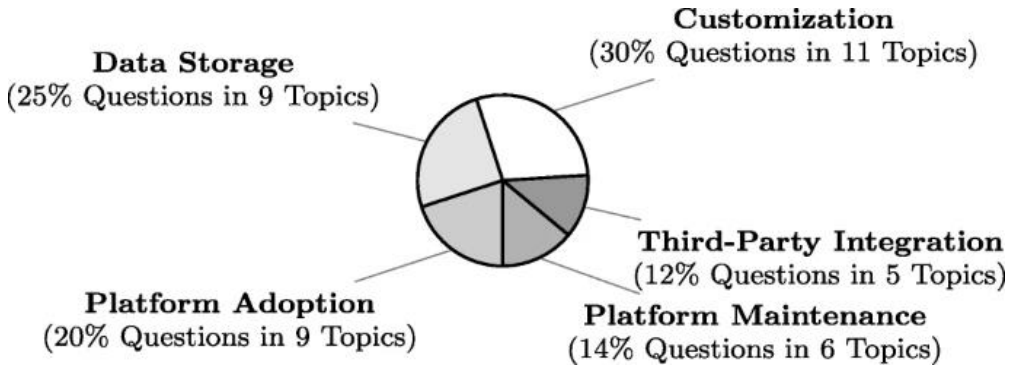
Problemas de flexibilidad

Se corresponde con el problema de personalización previamente comentado. El programador tiene poco control sobre lo que subyace a una aplicación low-code o no-code. Esto puede suponer una limitación a la hora de desarrollar aplicaciones de mayor complejidad en las cuales se necesiten funcionalidades específicas que estas aplicaciones no puedan proveer.

Obstáculos y preguntas frecuentes en la adopción de herramientas Low-Code

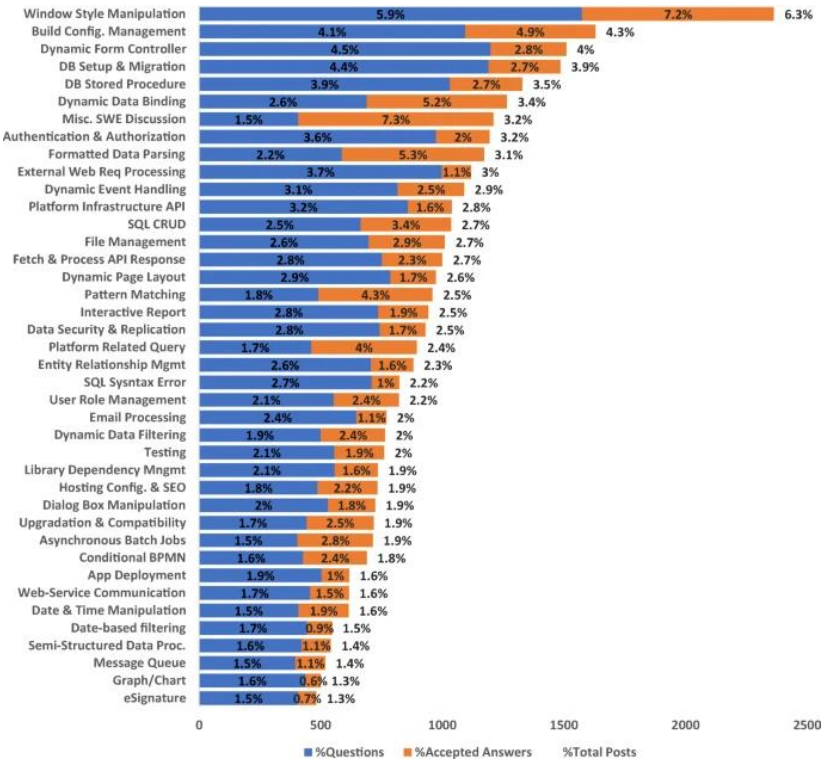
En el siguiente estudio, se utilizó la herramienta de Stack Overflow para analizar 33.700 posts relacionados con las principales dudas, preguntas y problemas encontrados a la hora de adoptar tecnologías Low-Code. Las principales conclusiones extraídas fueron las siguientes.

La categoría donde los usuarios encontraron mayores dificultades fue la de "customización" o personalización y adaptación del software a las necesidades concretas del usuario. Otra categoría recurrente es la del almacenamiento de datos en aplicaciones cloud. Finalmente, y siguiendo a la categoría anterior, la configuración cloud para gestionar todos estos sistemas, así como su mantenimiento e integración con aplicaciones de terceros.



Categorías de dudas más buscadas [2]

Estas fueron las dudas que más surgieron, condensadas en categorías que agrupan preguntas similares. Sin embargo, la imagen completa resultante de este estudio es mucho más profunda. Así, desgranando todos los resultados de estas categorías, obtenemos la imagen completa.



Problemas encontrados en herramientas Low Code [2]

Este estudio demuestra la importancia de un buen soporte y una cada vez mayor comunidad de usuario como medio para identificar, corregir o mejorar aquellos elementos que resultan de mayor complejidad para las personas que hacen uso de estas herramientas.

La tecnología no-code como herramienta de aprendizaje de DSS en estudiantes de ramas de negocio

Utilizar sistemas no-code en las universidades mejora el aprendizaje de los alumnos en ramas de negocio. Esta es la conclusión a la que ha llegado un estudio realizado por Wang Hai y Wang Shouhong en el Journal of Information Systems Education. En este estudio, se han estudiado los efectos de la implementación de herramientas no-code para el desarrollo de DSS (sistemas de soporte de decisiones) en cursos enfocados a ámbitos de negocio.

Aprender a desarrollar aplicaciones DSS es cada vez más importante para los estudiantes de las ramas de negocios. Sin embargo, los estudiantes se ven desinteresados y desincentivados, debido a la complejidad de los lenguajes de programación, a la hora de aprender las técnicas de programación necesarias para realizar dichas aplicaciones.

En este estudio han utilizado Microsoft Power Apps, ya que esta ya viene incluida con Microsoft 365 (software que ya tienen disponibles muchos estudiantes). Las conclusiones de dicho estudio arrojan una serie de datos interesantes. El primero de ellos es que antes de implementar el método no-code, solo 2 de 79 alumnos escogía el proyecto de desarrollo de DSS (mediante VBA, o Visual Basic Application), indicando que la programación intimida por su aparente complejidad a los estudiantes de campos (como el de negocio) no centrados en este ámbito. Después de implementar el método no-code, 2 de cada 10 alumnos escogían este proyecto.

Para indagar más sobre los efectos de esta decisión, los grupos que desarrollaron el DSS se dividieron en dos: los que lo desarrollaron en Visual Basic Application (VBA) y los que lo hicieron en no-code. Al primer grupo se le enseñó VBA durante 2 semestres antes de enseñarles no-code. Al segundo se le enseñó no-code directamente. Los resultados fueron que el grupo de no-code se desarrolló mejor en el proyecto. Esto es debido a que el VBA resulta menos interesante y consume más tiempo a los estudiantes de las ramas de negocio para entender y usar de forma eficiente.

Otro dato importante a resaltar es que los estudiantes no tuvieron problema en aprender lo básico del desarrollo no-code en un corto periodo de tiempo. Esto asienta las bases para que el alumno pueda avanzar (si así le interesa) a niveles más avanzados del desarrollo por su cuenta.

Estos avances no están exclusivamente restringidos a estudiantes de instituciones superiores de educación. Así, las herramientas low-code y sobre todo las no-code, debido a su ausencia total del proceso de codificación, son una gran herramienta para introducir de forma temprana a los estudiantes más pequeños en tecnologías y procesos de información con los cuales construir aplicaciones reales que puedan probar por sí mismos. Este concepto no es nuevo y ha sido probado con éxito mediante el uso de herramientas bien conocidas como lo es Scratch. Estas herramientas poseen una gran abstracción y, aun no disponiendo de una gran potencia, proveen a los usuarios más jóvenes de

mecanismos de aprendizaje con los cuales entrenar su mente desde una temprana edad para pensar de forma similar a como lo haría un programador, asentando las bases para un futuro desarrollo en ámbitos más avanzados de este campo.

Por lo tanto, las tecnologías low-code y no-code son unas buenas herramientas para empoderar a las personas con trasfondos ajenos a la programación a adentrarse en aspectos de la informática y poder así convertirse en personas capaces de optimizar procesos de sus respectivas ramas y crear aplicaciones sin necesidad de recurrir a un proceso de codificación. Además, abre las puertas a que estudiantes que, por su rama de estudios no habrían pensado en introducirse en aspectos tecnológicos, descubran sus capacidades y puedan adentrarse más en aspectos relacionados con la programación, habiendo entendido sus bases fundamentales, ahondando así cada vez más y ganando conocimientos más avanzados de programación con los cuales optimizar y/o automatizar trabajos de mayor complejidad y procesos de la propia organización.

Futuro de los profesionales de la informática

Ante estos avances, es normal que surja la pregunta de cuál será el papel de los ingenieros en un entorno laboral en el que gran parte del proceso de codificación ha sido sustituido por aplicaciones low-code y no-code.

Actualmente, como ya se ha comentado en este trabajo, las tecnologías low-code y no-code se especializan en el desarrollo de aplicaciones de un tamaño relativamente pequeño. Para grandes aplicaciones de negocio sigue siendo imprescindible desarrollar el proceso de codificación de manera tradicional, debido a que este proceso ofrece un alto nivel de detalle y personalización. Además, proporciona más control al diseñador para realizar una aplicación eficaz y eficiente. Debido a esto, actualmente no se puede sustituir el proceso tradicional de codificación completamente por estas nuevas tecnologías.

Sin embargo, el progreso de desarrollo sigue su camino y cada vez las herramientas low-code y no-code son capaces de realizar aplicaciones más complejas y con mayor nivel de eficiencia. En esto contribuye en gran manera el desarrollo de las inteligencias artificiales. Esto indica que el desarrollo de las tecnologías low-code y no-code apunta directamente a las formas tradicionales de codificación. No se puede discernir si el futuro al que se aspira es aquel en el que se pueda sustituir por completo a la codificación tradicional, ya que algunas plataformas low-code y no-code aspiran a nichos específicos y aplicaciones de pequeño tamaño altamente especializadas.

Pero, por un momento, suponga el lector el peor de los casos para el ingeniero-programador. Las herramientas low-code y no-code, y el desarrollo de la inteligencia artificial, no solo permiten automatizar todo tipo de tareas repetitivas, trabajo que ya realizan, sino que también han desechado por completo la necesidad de un proceso manual de codificación. Pues bien, las tecnologías low-code y no-code no son una caja negra de la cual solo se conocen los inputs y los outputs, son elementos complejos compuestos por numerosos subsistemas que subyacen a la aplicación principal. El funcionamiento y entendimiento de los elementos que componen las tecnologías low y no code solo pueden ser entendidos de forma eficaz por un profesional cualificado, un ingeniero. Y es que, cada vez que hay un salto en la abstracción del proceso de codificación, las tecnologías que formaban este proceso no desaparecen, podríamos decir que quedan enterradas, ya no están en la superficie a la vista de todos, pero siguen siendo

clave para el funcionamiento de la aplicación. Es imprescindible tener profesionales que entiendan y sepan manipular estas tecnologías "enterradas" ya que, si no, lo que se ha construido encima dejaría de funcionar apropiadamente. Los sistemas se están haciendo cada vez más complejos, sobre todo en estos campos que abstraen complejidad de sistemas que inherentemente son complejos, lo que hace que cada vez sea más difícil para una persona entender todo el proceso, o por decirlo de otra forma, de ver la imagen completa. Conforme pase el tiempo se necesitarán más profesionales que trabajen "en el backstage" con el fin de ser capaces de gestionar todas las tecnologías que soportan el desarrollo de las nuevas

Sin embargo, el rol del ingeniero está probablemente abocado a un cambio en el largo plazo. Aquellos que no deseen seguir el modelo del "backstage" explicado en el párrafo anterior, verán que el proceso de codificación se irá reduciendo progresivamente, lo que significa que los ingenieros necesitarán habilidades diferentes a las requeridas en la programación tradicional. En lugar de escribir código detallado, los ingenieros pueden necesitar enfocarse en la identificación de los requisitos del usuario, el modelado de datos y la integración de sistemas.

Además, es probable que estas tecnologías cambien la forma en que se organizan y administran los equipos de desarrollo de software. Se necesitará una mayor colaboración entre los desarrolladores, los diseñadores y los usuarios finales, lo que significa que los ingenieros tendrán que ahondar en habilidades de comunicación y gestión de proyectos más avanzadas.

De esta forma, se puede entender mejor el rol que ocupan las tecnologías low-code y no-code con una herramienta que ocupó en su momento un nicho similar, Excel. En su momento Excel (y herramientas similares) fueron una revolución en su campo. No solo dinamizaron el proceso de generación de tablas, sino que permitieron reducir en gran medida los procesos manuales de cálculo de datos. Estas tareas eran importantes para las personas que ocupaban puestos de administración, contabilidad, bancario, etc. Hubo gente que teorizó en su momento que las tecnologías de hojas de cálculo acabarían con los roles previamente mencionados dentro de las compañías. Sin embargo, esta tecnología se ha convertido en el mejor aliado para los trabajadores que han de manejar cálculos que requieran de tablas de almacenamiento de información. Así, Excel se ha convertido, al menos conceptualmente, en una herramienta inseparable de algunos trabajos, asociándola automáticamente a muchos de los roles que se pueden encontrar en las empresas. De esta forma, ha empoderado a los empleados, mejorando su eficiencia.

En un estudio realizado por Tech Republic [3] se obtuvo que solo el 15% de los encuestados utiliza las tecnologías low-code y no-code para crear aplicaciones nuevas. El resto, las utiliza para automatizar workflows (17%), acelerar el tiempo de desarrollo (15%), automatizar la recolección de datos (14%), entre otros. Finalmente, el 67% de los encuestados nos consideran que las tecnologías low-code y no-code resultarán en una menor demanda de ingenieros o desarrolladores.

Por todo esto, es muy pronto para dar por terminado el trabajo del ingeniero. Donde haya un sistema que sustente cierta complejidad, habrá la necesidad de disponer de ingenieros que la entiendan, manipulen, mantengan y construyan sobre ella. Así, la complejidad de estas tecnologías y su correcta implementación e integración en Low-Code y No-Code

solo puede ser entendida por empleados altamente cualificados en los campos del IT y de la ingeniería informática.

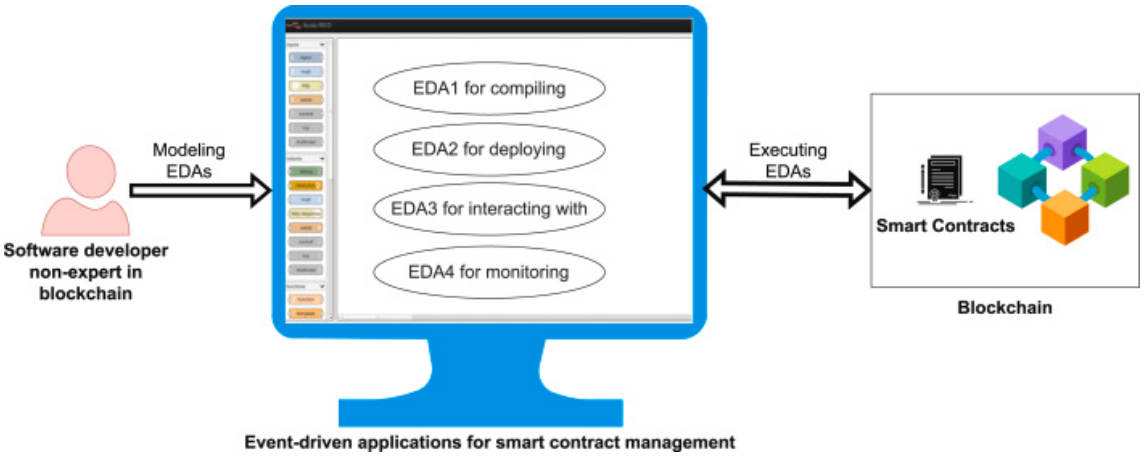
Principales herramientas Low-Code y No-Code escogidas

Node-RED Low Code Platform

La tecnología Blockchain está actualmente restringida a un reducido número de expertos software debido al tiempo que se requiere para dominar este campo. Low-Code es una herramienta eficaz para facilitar el desarrollo de estas aplicaciones mediante el uso de sistemas "event-driven" (orientados a eventos).

Node-Red facilita el acceso a blockchain y a los "smart contracts". Esto lo hace mediante el modelado de flujos, los cuales especifican las comunicaciones entre los tres actores del sistema: Los **productores de datos**, los **procesadores de datos** y los **consumidores** de datos. También permite desplegar las aplicaciones "smart contracts" sin que se requiera tener conocimientos previos de blockchain.

Como ejemplo que prueba el uso de esta herramienta, estos sistemas se han utilizado exitosamente en el estudio de las vacunas de Covid-19 de AstraZeneca y Moderna para monitorizar en tiempo real la temperatura a la que se encontraban.



Arquitectura general [4]

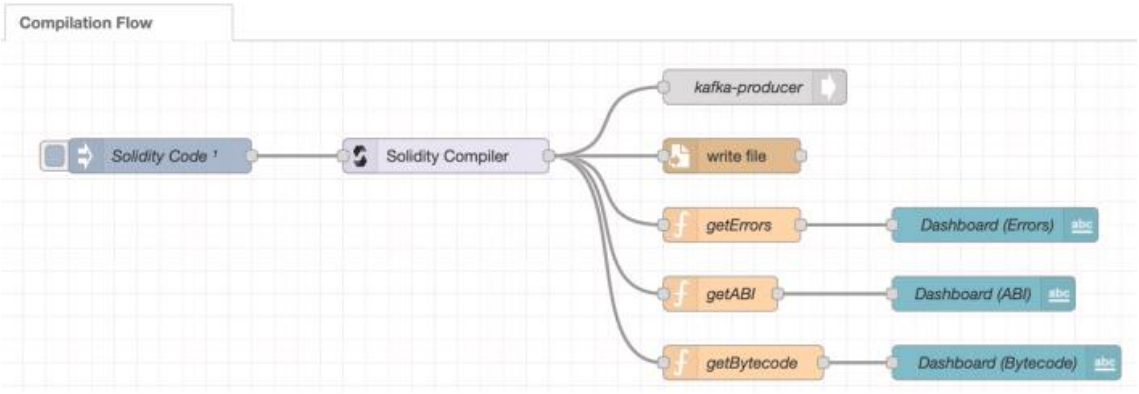
Como ya se ha comentado anteriormente, la arquitectura utilizada en esta solución está dividida entre tres actores, los productores de datos, los procesadores de datos y los consumidores de datos. A continuación, se detallará el papel que juega cada uno de estos.

Los productores de datos son los intermediarios entre el mundo real y (en este caso) el smart contract. A su vez, estos hacen uso de servicios web (los cuales realizan la función de intercomunicación e interoperabilidad entre máquinas), sensores IT y, por supuesto, los smart contracts.

Los procesadores de datos son el contenedor de la plataforma Low-Code. Realizan la función de desarrollo user-friendly mediante nodos de flujos. Estos nodos están agrupados en tres categorías: los nodos productores de datos, los nodos de procesamiento de datos y los nodos consumidores de datos. Los primeros, reciben información de los

productores de datos y los envían a los nodos de procesamiento de datos. Los segundos, compilan, despliegan o interactúan con los smart contracts, después envían los resultados a los nodos consumidores de datos. Estos últimos conectan y envían datos desde la plataforma Low-Code hasta la capa de los consumidores de datos.

Como se puede observar, esta arquitectura puede recordar al modelo-vista-controlador que se utiliza en muchas aplicaciones desarrolladas en java.



Ejemplo de flujo de trabajo en Node-RED [4]

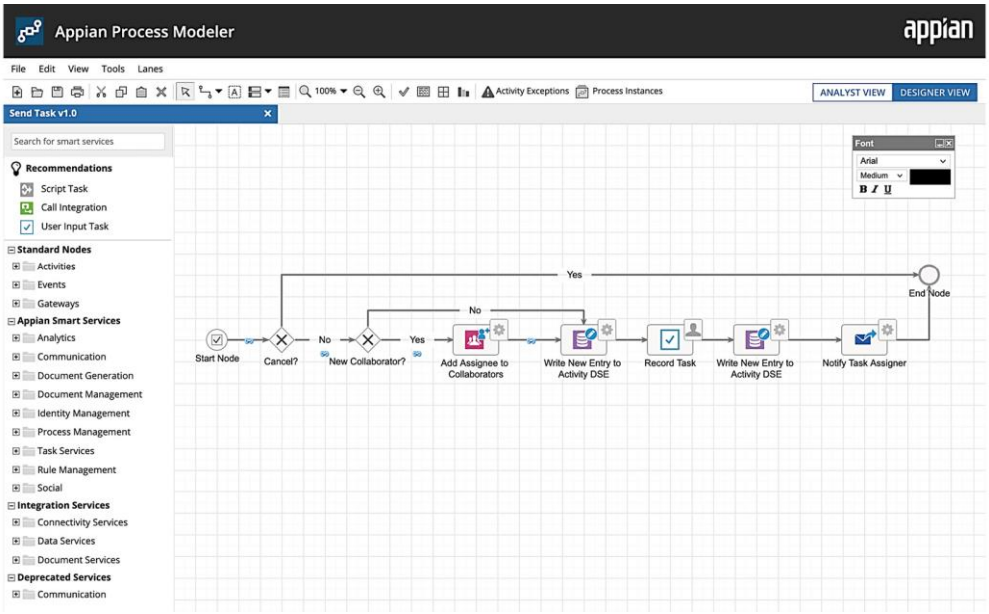
Este ejemplo representa una aplicación que permite al usuario compilar un smart-contract. El primero de los nodos es del tipo productor de datos, el cual contiene el código a ser compilado. El segundo es un nodo de procesamiento de datos el cual es el encargado de realizar de forma transparente la compilación del código. Los nodos de color amarillo son funciones que captan los posibles resultados, incluido el de error. Finalmente, los tres últimos nodos (los que se encuentran en azul) son consumidores de datos, en este caso realizando la acción de dashboard para las distintas salidas.

Appian

Appian proporciona numerosas herramientas para agilizar los procesos empresariales, en este caso el enfoque estará centrado en Appian Quick Apps. Esta es una herramienta de low-code que no requiere descarga ni instalación, lo cual mejora sustancialmente la accesibilidad y reduce el tiempo de despliegue de una potencial aplicación.

Además, Quick Apps proporciona una herramienta de “reporting” o “dashboard” para generar gráficos. Esto ofrece a las personas en puestos de responsabilidad una herramienta para la toma eficaz de decisiones.

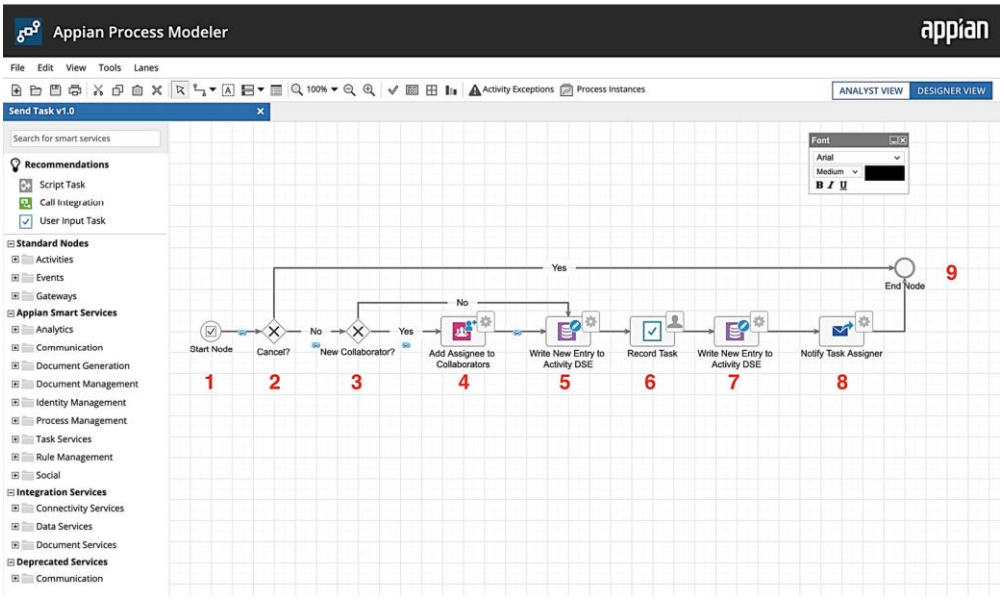
Así, al igual que en el producto anterior, el usuario tiene la libertad de modificar el flujo para conseguir el resultado deseado.



Ejemplo de flujo de trabajo en Appian [5]

Estos diagramas se asemejan en gran medida a los diagramas de flujo que un ingeniero puede realizar a la hora de diseñar una aplicación. La ventaja de este tipo de productos es que el proceso de diseño se traduce directamente a un proceso de desarrollo de la aplicación, ahorrando un tiempo significativo al programador.

Así, en el siguiendo flujo del programa se obtendría lo siguiente:

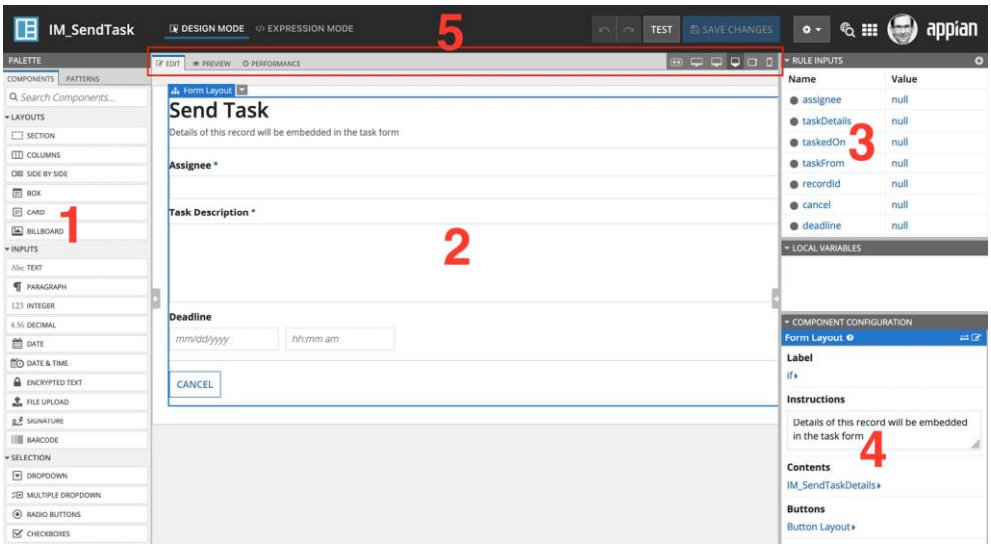


Flujo de trabajo desgranado [5]

El primer nodo es el nodo de inicio, este se encarga de mostrar la interfaz correspondiente. En el paso dos se indica una simple decisión, si el usuario da clic en cancelar se redirige al final del flujo. En el paso tres se encuentra otra decisión, esta vez se evalúa si el usuario ya tiene permiso para utilizar la aplicación. Si no es así, se le añade en el nodo número

cuatro. El nodo cinco no es más que proceso que escribe el item en la base de datos de Appian como registro para que la empresa utilice la información para realizar mejoras del sistema. El nodo seis envía una verificación en forma de correo electrónico a la persona que se le ha asignado la tarea. En el nodo siete, cuando el usuario ha terminado la tarea, se registra el evento en la base de datos. Por último, en el nodo ocho, se envía un mensaje al gerente del empleado notificándole de la finalización de la tarea. Una

Quick Apps también permite configurar las interfaces para mayor confort del usuario.



Interfaz de Appian [5]

De esta forma, ofrece una interfaz amigable y sencilla de entender para el usuario final lo que, añadido a la funcionalidad “drag and drop” (arrastrar y soltar elementos hacia la pantalla) facilita en gran medida el desarrollo.

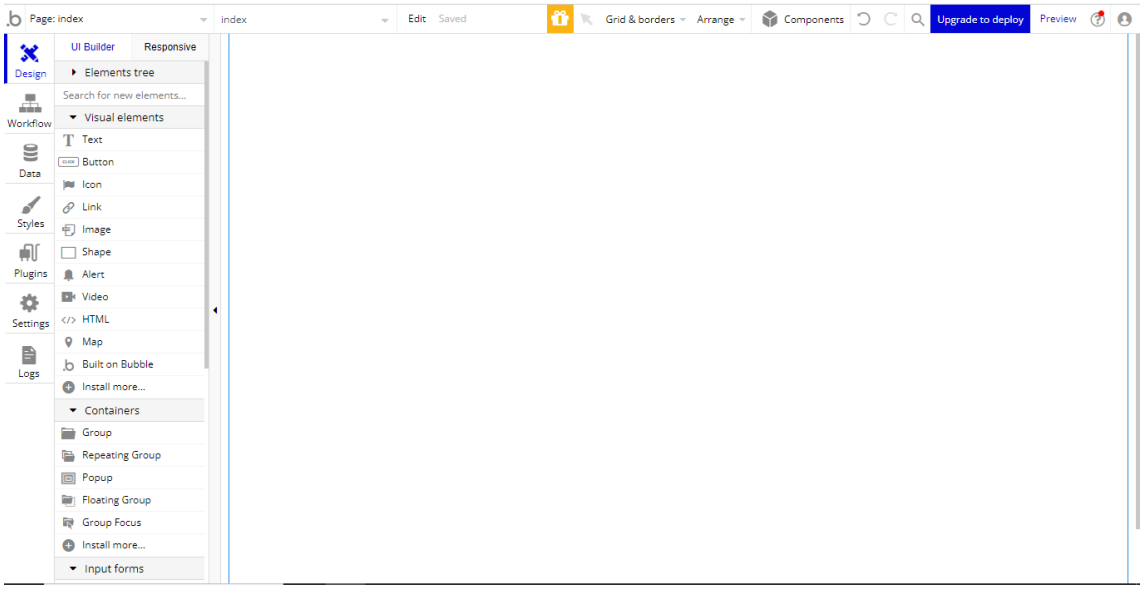
Appian Quick Apps proporciona integraciones sencillas con sistemas ya existentes, lo que significa que los usuarios pueden integrar datos de diferentes fuentes y sistemas en una sola aplicación. Esta es una característica muy extendida en los sistemas low y no-code ya que permite extender la funcionalidad de la aplicación y no limitarla a tareas de alta sencillez.

También se proporcionan mecanismos de seguridad avanzados, tales como características de autenticación, autorización y protección de datos. Esto asegura un alto nivel de fiabilidad, tanto para el desarrollador como para el destinatario de la aplicación resultante.

Finalmente, Appian asegura altas capacidades de escalabilidad en referencia a Quick Apps, asegurando que se pueden realizar aplicaciones de alta complejidad mediante esta herramienta. Esto lo sustenta mediante la arquitectura en la nube y una gran capacidad para distribuir cargas de trabajo. La nube posibilita que el sistema se pueda adaptar fácilmente a las necesidades del usuario, lo que combinado con su enfoque modular y una gran cantidad de módulos y funciones preconstruidas, permite a los desarrolladores crear aplicaciones de manera rápida y sencilla.

Bubble.io

Bubble.io es una aplicación enfocada en el diseño visual basado en el concepto de “Drag and Drop”. Siguiendo esta filosofía, el usuario tiene a su disposición una pestaña situada en el lado izquierdo de la pantalla en la cual, similar a otras soluciones web como WordPress, dispondrá de un abanico de objetos y funcionalidades las cuales podrán integrarse en la aplicación arrastrándolas a la misma.

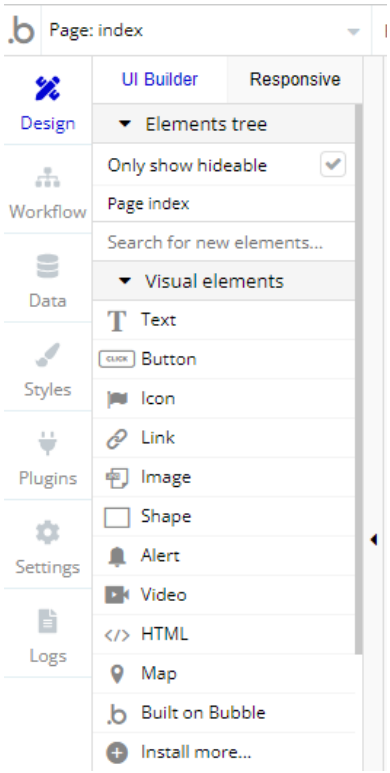


Menú de Bubble [6]

De esta forma, Bubble.io posee una gran variedad de opciones de personalización, lo que permite adaptarse no solo a las distintas situaciones que los usuarios puedan encontrarse, sino que también permite adaptar el diseño de la propia aplicación de forma rápida, sencilla e intuitiva. Así, Bubble.io ofrece distintas herramientas de personalización.

La primera de estas herramientas son los componentes personalizables ya comentados, a saber: botones, gráficos, formularios, mapas, etc. Además, ofrece al usuario la posibilidad de crear sus propios componentes personalizados, lo que añade una mayor variedad de posibilidades a la hora de crear aplicaciones.

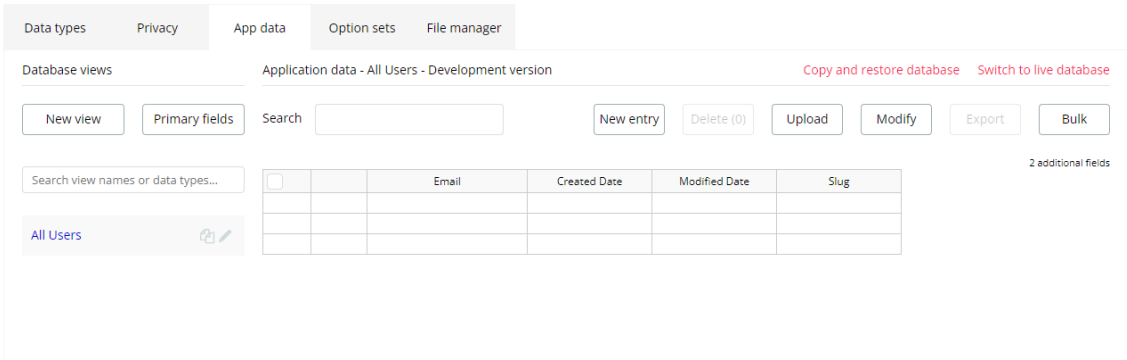
Bubble.io también ofrece mecanismos de automatización dentro de sí. Un ejemplo sería la posibilidad de crear estilos reutilizables, los cuales pueden ser aplicados a los distintos elementos de la aplicación, ahorrando tiempo al usuario.



Barra de herramientas [6]

Al igual que las herramientas que ya se han comentado previamente, Bubble.io incluye el desarrollo de flujos de trabajo como principal método de sustitución del proceso de codificación. Mediante estos, se puede especificar al programa realizar acciones concretas y manejar los distintos datos de la misma.

Además, Bubble.io ofrece un elemento del que carecen otras soluciones, una base de datos. Esta, es una base de datos sencilla pero útil para pequeñas aplicaciones que no precisan de realizar operaciones complejas con los datos allí almacenados. En dicha base de datos el usuario podrá crear tipos de datos, cada uno con las columnas que sean necesarias, al igual que se haría en otras herramientas de almacenamiento de datos.



Almacenamiento de información [6]

Bubble.io también permite incluir aplicaciones de terceros dentro de la correspondiente aplicación. Algunas de ellas pueden ser útiles para realizar transacciones dentro de la

misma, como es el ejemplo de PayPal. De esta forma, la aplicación no está limitada solo por lo que es capaz de ofrecer por sí misma, sino que es capaz de incluir dentro de sí aplicaciones que extienden su funcionalidad. Con todo esto, el usuario puede crear pequeñas aplicaciones de negocio sin tener que recurrir a un programador, abaratando costes y ahorrando tiempo de desarrollo.

Finalmente, aunque Bubble.io suele ser clasificado como una aplicación no-code, la capacidad de incluir elementos customizados de HTML, lo cual, aun no siendo un lenguaje de programación estricto, incorpora la posibilidad de realizar un cierto proceso de codificación, elemento característico de una herramienta low-code.

Microsoft PowerApps

Microsoft Power es una herramienta muy potente para realizar tareas creadas con Low-Code. Esta trae consigo muchas soluciones que permiten implementar o realizar acciones de forma rápida y efectiva para una empresa, sin tener que recurrir al largo proceso de codificarlas. Estas soluciones son: Power BI, Power Apps, Power Pages, Power Virtual Agents y por último Power Automate.

La primera solución (Power BI) está centrada en el análisis de datos. La segunda, está enfocada a la construcción de aplicaciones. Power Pages, se orienta al desarrollo de aplicaciones web enfocadas al cliente. Power Virtual Agents se centra en la interacción con los clientes. Por último, y en la que se centrará este apartado, Power Automate. Su función es la de mejorar la productividad de la empresa mediante la automatización de tareas rutinarias para las empresas.

Esta última opción, ofrece varios tipos de flujos con los que automatizar los procesos de una empresa. El primero, automated cloud flow, se ejecuta cuando se activa un determinado trigger, por ejemplo, cuando llega un email o se publica un tweet. Esto permite crear flujos de trabajo automatizados en la nube sin necesidad de escribir código. Además, permite integrar servicios en la nube, como Office 365, Dynamics 365, SharePoint, OneDrive y otros servicios de terceros, en un solo flujo de trabajo. También se pueden integrar con el correo electrónico, o aplicaciones como Teams, para enviar notificaciones. Estos flujos son capaces de gestionar grandes cantidades de datos, lo que hace sencillo su escalabilidad.

El segundo, instant cloud flow, se basa en activar el trigger manualmente, como, por ejemplo, cuando das a un botón de una aplicación. Por lo demás, comparte características con el anterior tipo de flujo. Finalmente, los instant cloud flows son accesibles en dispositivos móviles, lo que permite a los usuarios trabajar en los flujos de trabajo en cualquier lugar.

El tercero, scheduled cloud flow, es lo que el lector se puede imaginar, permite crear flujos de trabajo automatizados en la nube que se ejecutan en una fecha determinada.

Los business process flow permiten a los usuarios diseñar y visualizar el flujo de trabajo de un proceso empresarial específico. Estos flujos se diseñan de manera visual, utilizando una interfaz de usuario de tipo arrastrar y soltar, lo que hace que la creación de un flujo de trabajo sea fácil y rápida. Además, debido al marcado carácter empresarial de este flujo, los BPFs permiten a los usuarios ver de una forma visual e intuitiva el progreso del

proceso empresarial, con el fin de tener una imagen más clara a la hora de ser analizado por puestos de responsabilidad. Así, cada etapa se muestra con un porcentaje de completado y un indicador de estado.

Por último, los desktop flow realizan el papel de los RPA (Robotic Process Automation) ya que pueden grabar acciones que realizas en la aplicación para posteriormente crear un flujo que ejecute dichas acciones, similar a lo que haría una macro. Debido a esto, los desktop flows están especializados en automatizar tareas repetitivas y procesos de escritorio en Windows. Los Desktop Flows cumplen con los estándares de seguridad y cumplimiento de Microsoft, lo que garantiza que los datos y procesos empresariales estén protegidos.

N8N

A diferencia de otras herramientas se centra principalmente en conectar diferentes aplicaciones y sistemas, los cuales intercambian información entre sí, como mecanismo para expandir las capacidades de la propia herramienta. Debido a esto, N8N cuenta con una gran variedad de conectores los cuales habilitan a la aplicación conectarse con numerosos servicios, tales como Salesforce o Trello. Esto hace, que N8N funcione mejor cuando se implementa junto a otros sistemas.

Por otra parte, N8N ofrece una personalización avanzada de los elementos que provee. El usuario puede personalizar la forma en que se procesan los datos en tus flujos de trabajo, utilizar transformadores de datos para manipular la información y crear flujos de trabajo basados en eventos.

Los tipos de nodos que utiliza N8N para gestionar el flujo de información del programa son los siguientes:

Los "**trigger nodes**" como su nombre bien indica, ejercen la función trigger dentro del sistema. Estos inician los flujos de trabajo al hacer de activadores de los mismos. Para cumplir esta tarea, están especializados en detectar eventos específicos, los cuales el usuario puede configurar. Un ejemplo podría ser un botón pulsado, un dato introducido en la base de datos, etc.

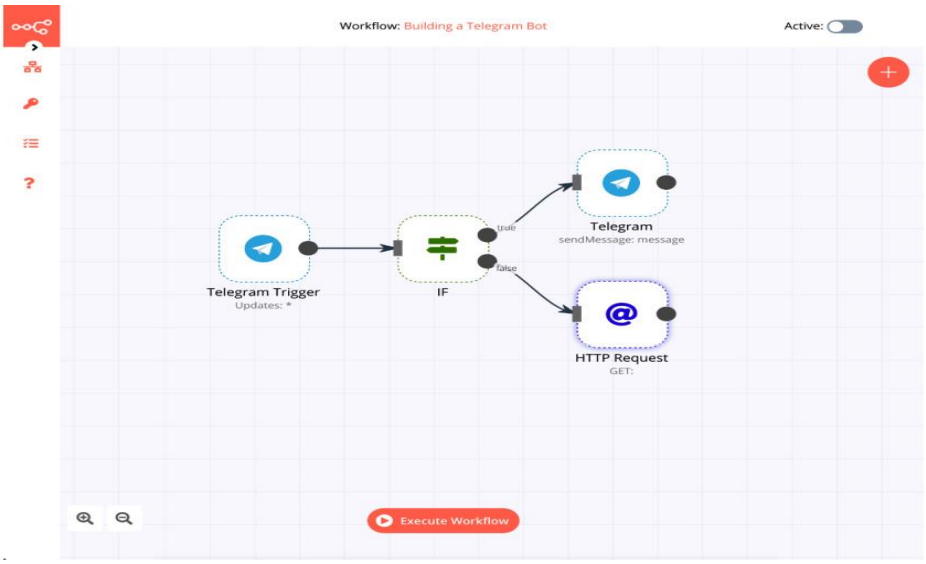
Los "**Function nodes**" permiten al usuario introducir su propia lógica personalizada al flujo de trabajo. Estos tipos de nodos, se pueden usar para agregar lógica empresarial, transformar datos o realizar operaciones personalizadas.

Los "**Action Nodes**" en cambio, realizan una tarea concreta. Esta tarea puede ser, introducir un dato en la base de datos, enviar un recordatorio de una reunión o, por ejemplo, enviar notificaciones a un grupo de personas.

Después tenemos los "**Data Nodes**" o nodos de datos. Como su nombre deja intuir, estos nodos se centran en la manipulación y transformación de datos dentro de los flujos. Así, estos son capaces de diversas tareas que van desde filtrar información hasta añadir o eliminar ciertos datos dentro del flujo de trabajo.

Finalmente, tenemos los "**Integration Nodes**". Estos son los que permiten a los flujos interactuar con otras aplicaciones y servicios. Este segmento iría en conjunción con lo comentado en el segundo párrafo de este apartado.

Ejemplo de flujo en N8N:



Ejemplo de flujo en N8N [7]

N8N también provee de herramientas de monitorización. Con ellas, el usuario puede monitorizar y registrar la actividad de los flujos de trabajo. De esta forma, se tiene una imagen más completa del funcionamiento de la aplicación, lo que permite detectar posibles problemas de rendimiento y encontrar una solución con mayor facilidad.

Apoyándose en todos estos atributos, N8N ofrece a los usuarios la posibilidad de compartir soluciones, lo que genera una comunidad en la cual poder resolver problemas y encontrar elementos que incorporar en nuestras propias aplicaciones.

Cabe mencionar que, debido a la especialización de Bubble.io en aspectos de front-end, esta aplicación es una buena herramienta a tener en cuenta a la hora de utilizar N8N. Así, estas dos soluciones funciones muy bien de forma conjunta, complementándose la una a la otra.

Por último, N8N hace más uso de objetos complejos, como podrían ser los tipos JSON por lo que se requiere cierto nivel de conocimientos en materia de programación para poder utilizar la herramienta de la forma más eficiente posible.

Salesforce

Salesforce es una herramienta bien conocida en el entorno de las compañías tecnológicas (sobre todo enfocadas a los CRM) por su capacidad de automatizar procesos de marketing que consumen tiempo innecesario, y por lo tanto dinero, a las empresas.

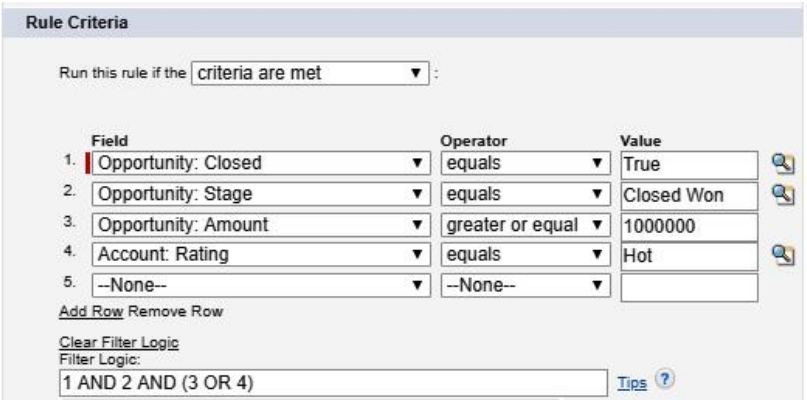
Como se acaba de comentar, Salesforce se centra en el aspecto CRM de las empresas, siendo una gran herramienta para administrar y mejorar los procesos de ventas, marketing

y atención al cliente. Así, Salesforce ofrece una amplia variedad de herramientas para gestionar los clientes potenciales, incluyendo la captura de potenciales clientes a través de formularios web, la puntuación de clientes potenciales, la segmentación y la distribución de clientes potenciales al equipo de ventas adecuado.

Salesforce cuenta con grandes herramientas para la automatización de las tareas de marketing y de ventas. Esto es de gran ayuda para los equipos de ventas y de contacto con el cliente dentro de las compañías, permitiendo crear campañas de marketing de forma automatizada y personalizar las mismas dependiendo del cliente al que vaya dirigida. En el apartado de automatización de ventas, es capaz de automatizar tareas de seguimiento, previsión de ventas y procesos de aprobación, lo que agiliza el cierre de los contratos. De esta forma, la empresa es capaz de aceptar un mayor flujo de contratos y aplicar una metodología ágil en su relación con los clientes.

Además, sus sistemas cuentan con capacidad de almacenamiento basado en cloud lo que facilita la escalabilidad de la aplicación. El sistema de Salesforce se considera Low-Code (en vez de No-Code) ya que para muchas de las opciones se necesita utilizar fórmulas para realizar funciones de automatización. Por lo tanto, se puede considerar que utiliza más "programación" que las otras opciones previamente tratadas. Por ejemplo, se utilizan ampliamente funciones propias de los lenguajes de programación tales como DATE y operadores como AND y OR.

Ejemplo de uso de operadores:



Ejemplo de uso de operadores [8]

Salesforce ofrece una plataforma para la gestión de casos y la automatización de procesos de servicio al cliente, haciendo que las empresas puedan ofrecer un servicio al cliente más rápido y personalizado, con herramientas para la gestión de casos, el enrutamiento de casos y la automatización de respuestas a clientes.

En aspectos que conciernen a la seguridad, Salesforce ofrece altos niveles de seguridad y cumplimiento con estándares de la industria, como ISO, HIPAA, GDPR, entre otros.

Como aspecto principal del funcionamiento de Salesforce, este posee, al igual que otras soluciones, una batería de flujos que pueden ser utilizados en distintas situaciones para conseguir el resultado deseado.

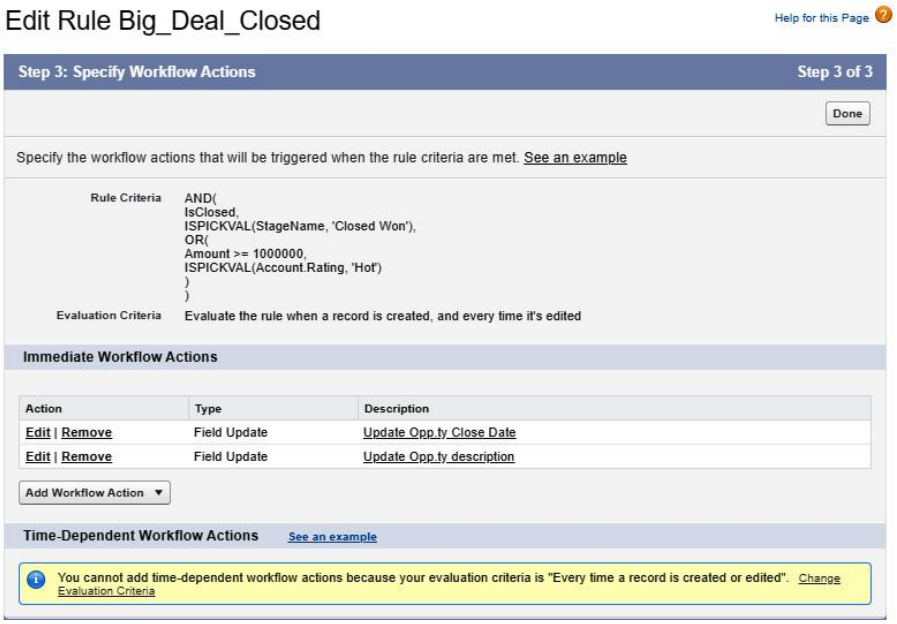
Los primeros son los workflow rules. Estos flujos permiten la automatización de tareas repetitivas mediante la activación de unas acciones automáticas previamente especificadas por el usuario. Entre estos se encuentra IFTT (If This Then That), el cual es la forma más simple de automatización. Para realizar estas tareas, los workflow rules son capaces de evaluar criterios como el valor de un campo. Por ejemplo, en la imagen previamente mostrada en este apartado, se puede observar un workflow rule, el cual solo se activará cuando se cumplan las condiciones especificadas. Estos tipos de flujos son capaces de integrarse con el resto de funcionalidades que ofrece Salesforce para crear aplicaciones de mayor profundidad.

Los segundos son los procesos de aprobación. Estos permiten automatizar las peticiones que los usuarios realizan a sus superiores para acceder o modificar ciertos archivos o elementos que requieren de permisos o condiciones de acceso como podrían ser los privilegios de usuarios. Para ello, estos flujos con capaces de evaluar los distintos criterios que definen a los usuarios de la aplicación. Para satisfacer los requisitos de seguridad que cada empresa pueda disponer, estos flujos pueden implementar un diseño de múltiples etapas, lo que ofrece un proceso jerárquico de aprobación. Al igual que otros tipos de flujos, estos proveen alertas, como por ejemplo cuando un usuario requiere acceso a un documento se le puede comunicar al responsable, además de salidas definidas en caso de que no se realicen las aprobaciones en el marco de tiempo establecido.

Tercero están a los process builders (constructores de procesos). Estos son una evolución de los workflow rules y ofrecen una mejor interfaz de usuario. A diferencia de los workflow rules, el usuario tiene la capacidad de determinar el orden de ejecución de las acciones.

Finalmente, se encuentran los lightning flows. Estos están pensados para automatizar los procesos de mayor complejidad y que requieren ciertos conocimientos de programación para poder ser utilizados de forma eficaz.

Ejemplo de flujo que requiere un proceso de codificación:



Ejemplo de codificación en Salesforce [8]

Un aspecto importante a considerar es que Salesforce ofrece un sistema denominado Trailhead. Este es un entorno gratuito de conocimiento, con cursos y documentación, lo que facilita el aprendizaje para sus usuarios. Por ejemplo, Trailhead ofrece cursos interactivos como herramienta para que los usuarios desarrollen conocimientos sobre el uso de la aplicación. Estos últimos, incluyen ejercicios para que los usuarios pongan en práctica sus conocimientos. Como plus, la plataforma ofrece certificaciones que el usuario podrá realizar.

Entorno a Trailhead se desarrolla una comunidad en la cual los usuarios pueden hacer preguntas y resolver dudas gracias al apoyo de otros miembros. De esta forma se crea una red de conocimientos que facilita el intercambio de ideas y soluciones.

5. Desarrollo de una aplicación mediate Bubble

El proceso estándar de desarrollo de una aplicación mediante low-code o no-code se puede describir mediante los siguientes pasos.

El primero de ellos es el del diseño del modelo de dominio, con el cual se representan conceptos y relaciones. Un ejemplo sería el UML (Universal Modelling Language). El siguiente paso es el de la definición de la interfaz de usuario. A continuación, iría la especificación de la lógica de negocio, en la cual se define el control y los flujos de datos. Siguiente está la integración con servicios externos y data sources. Seguido, se encuentra la generación de la aplicación y su posterior despliegue. Finalmente, está el mantenimiento de la aplicación, el cual, como con toda aplicación, es un proceso recurrente.

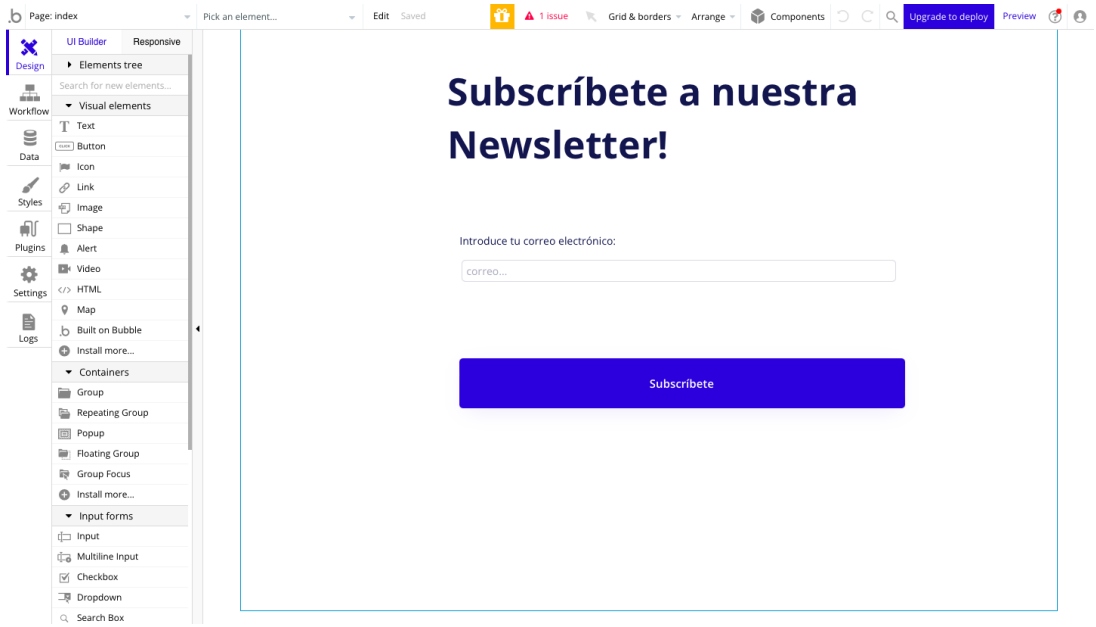
Una vez explicado el proceso común de desarrollo, detallaré el proceso concreto en una de estas aplicaciones.

Desarrollo de una Newsletter

La aplicación que se ha escogido para el desarrollo de aplicaciones no-code es Bubble. Esta aplicación combina una estética limpia y fácilmente entendible con una gran variedad de funcionalidades expresadas en forma de nodos. Además, al igual que muchas otras soluciones low-code y no-code, no hace falta instalarlo en la máquina para poder utilizarlo.

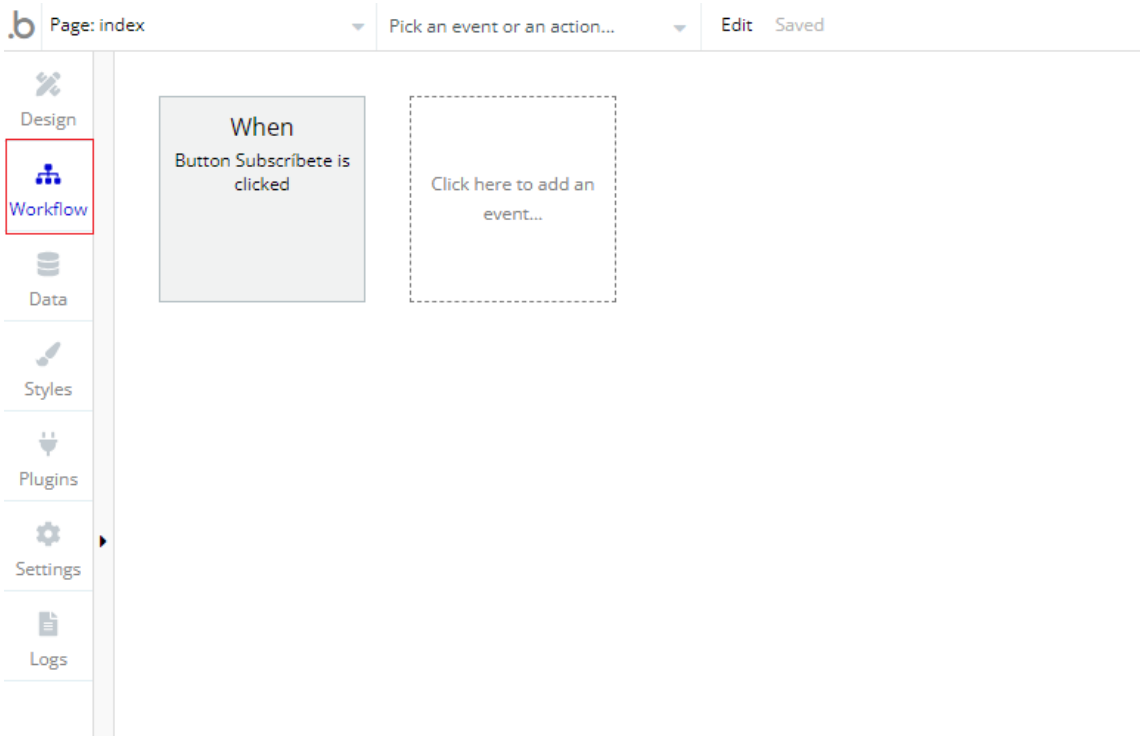
Para demostrar las capacidades de esta herramienta low-code no-code, se creará una aplicación web en la cual los usuarios podrán suscribirse a una newsletter. Una vez hecho esto, el administrador de la newsletter podrá programar publicaciones, las cuales serán enviadas por correo a los suscriptores. Todo esto se hará mediante el uso de nodos, los cuales cargarán con el flujo del programa.

Para empezar, la ventaja de Bubble es que integra dentro de sí tanto la creación de una aplicación web, como la gestión de una base de datos en la que almacenar información de la misma. Se empezará creando una pestaña en la cual se solicitará al usuario que introduzca su correo electrónico.

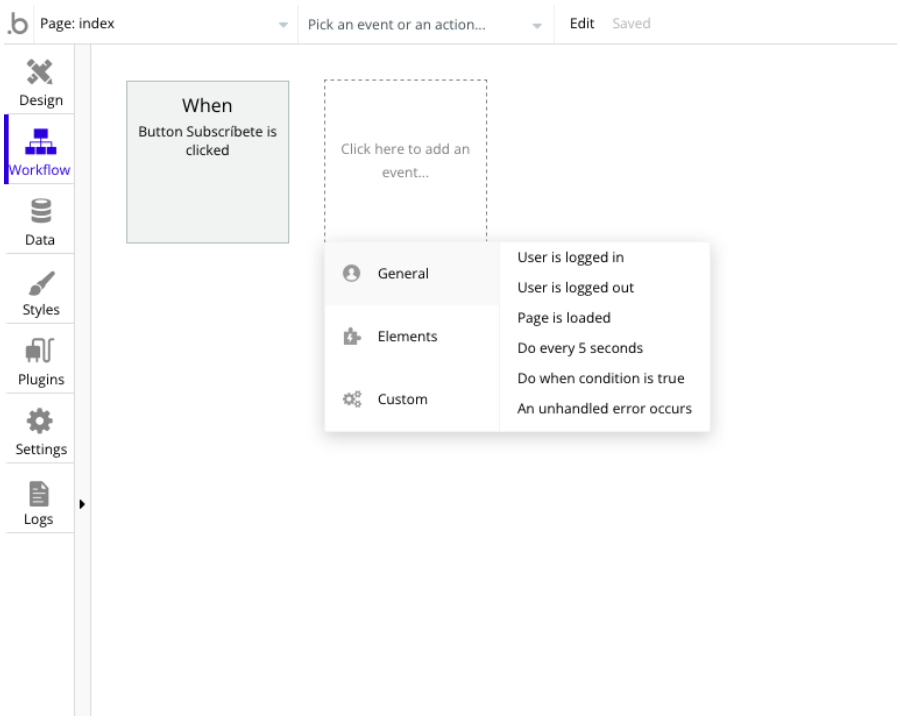


Una vez pulsado el botón de “subscríbete” el programa procederá a darlo de alta en la base de datos. En una aplicación web tradicional, habría que hacer uso de programación para conseguir implementar esta funcionalidad. Sin embargo, la tecnología no-code, mediante el uso de flujos, otorga un importante ahorro de tiempo al ofrecer la posibilidad de conseguir la misma funcionalidad que se busca en este ejemplo, sin tener que pasar por un proceso de codificación, haciendo que cualquier persona pueda hacerlo.

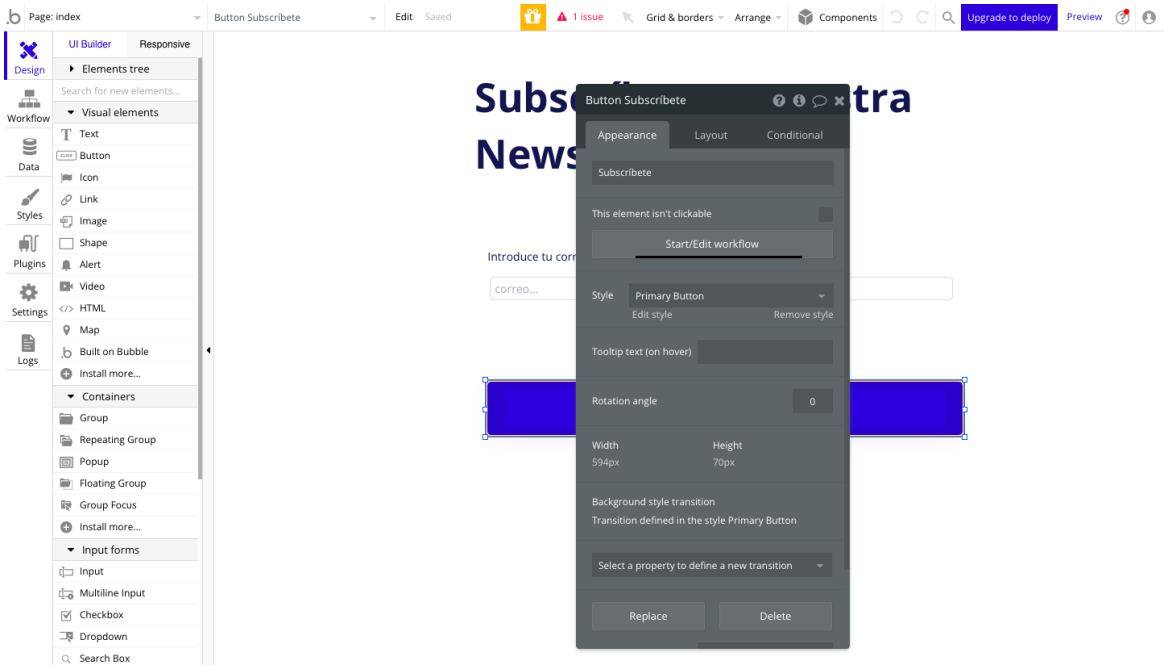
El proceso a seguir es el siguiente. Hay dos formas de empezar a generar flujos, o como los llama Bubble, workflows. La primera opción es dirigirse a la pestaña de workflows, situada en la barra de herramientas en la parte izquierda de la pantalla.



Una vez dentro se puede crear un nuevo flujo seleccionando cual ha de ser su nombre, su detonante, etc.

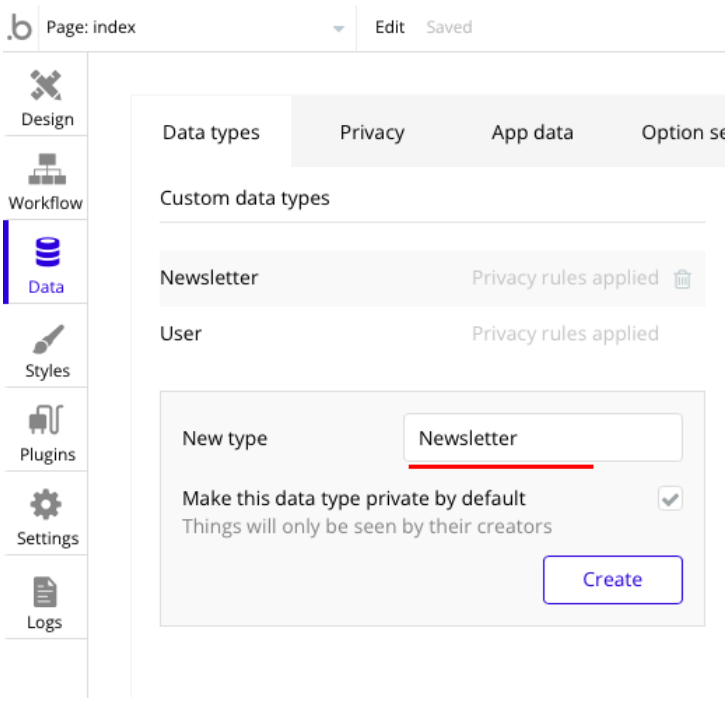


La otra opción, es la de hacer doble clic directamente sobre el elemento que ha de ser el detonante. Se abrirá un menú de opciones, desde ahí se podrá pulsar "Start/edit workflow".

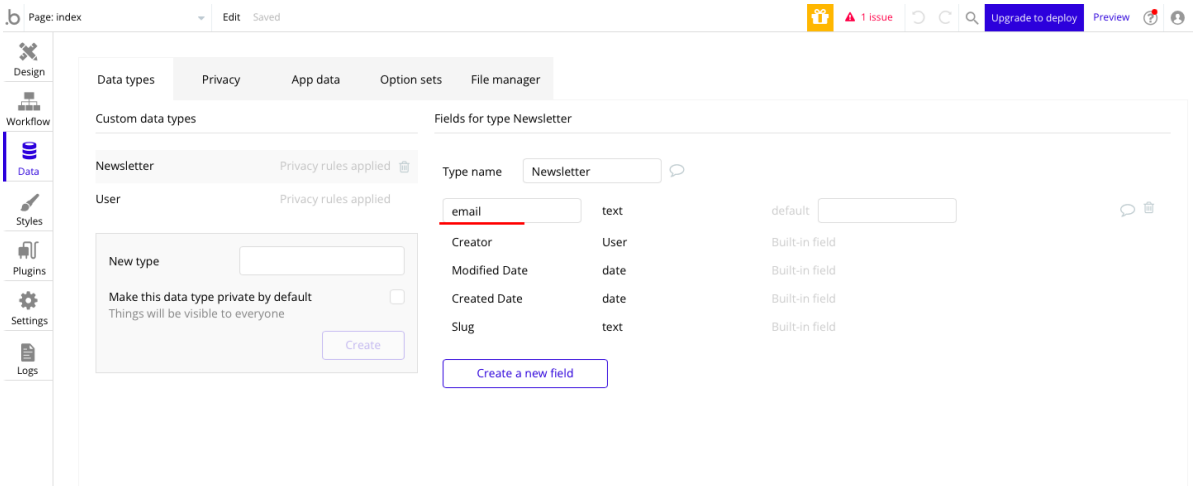


Una vez se tenga esto listo, y antes de empezar con el flujo, se creará un tipo de datos llamado "Newsletter" en el cual se almacenarán los correos de la gente que se subscriba.

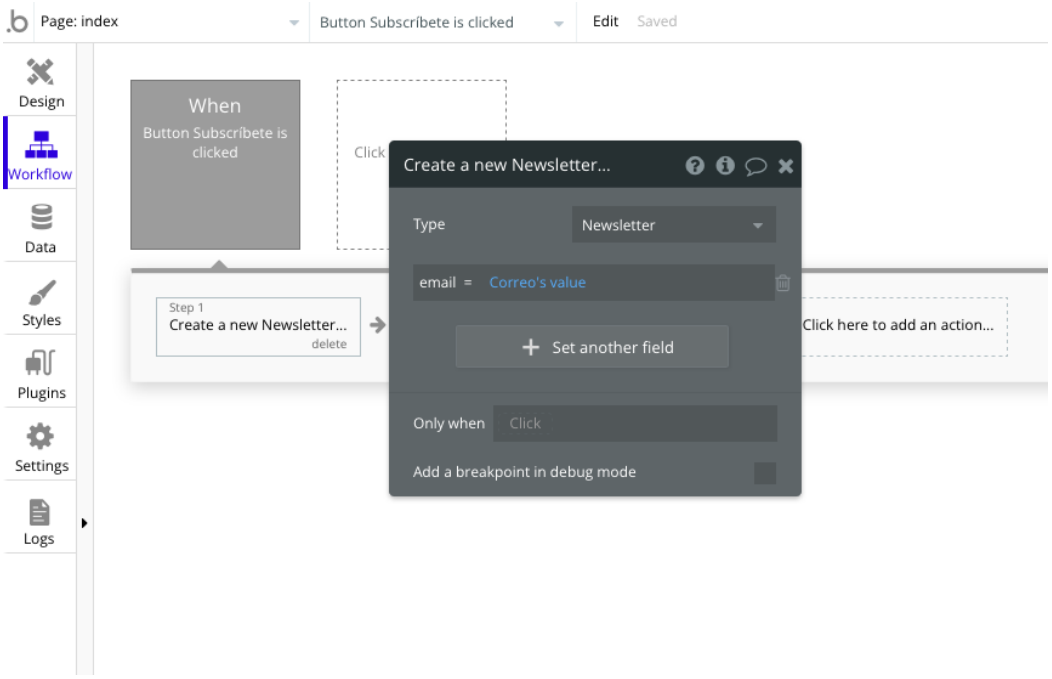
Para ello, el usuario podrá dirigirse al apartado de “data” y en la caja “new type” podrá introducir el nombre del nuevo tipo de dato.



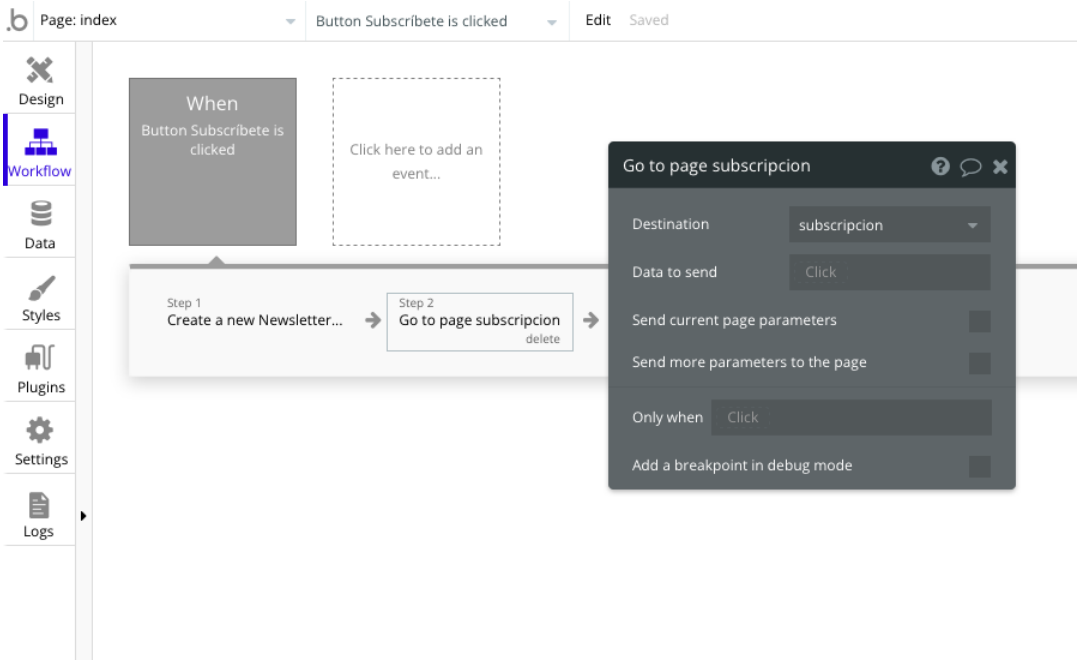
Una vez creado, aparecerá en el lado derecho, junto con unos campos predeterminados que vienen incluidos de facto (Creador, fecha de creación, fecha de modificación y el “slug” de la página). Ahora, se procederá a incluir el campo que interesa en este caso, el de email. Dentro de este campo, se almacenarán los correos de los usuarios que se suscriban a la newsletter.



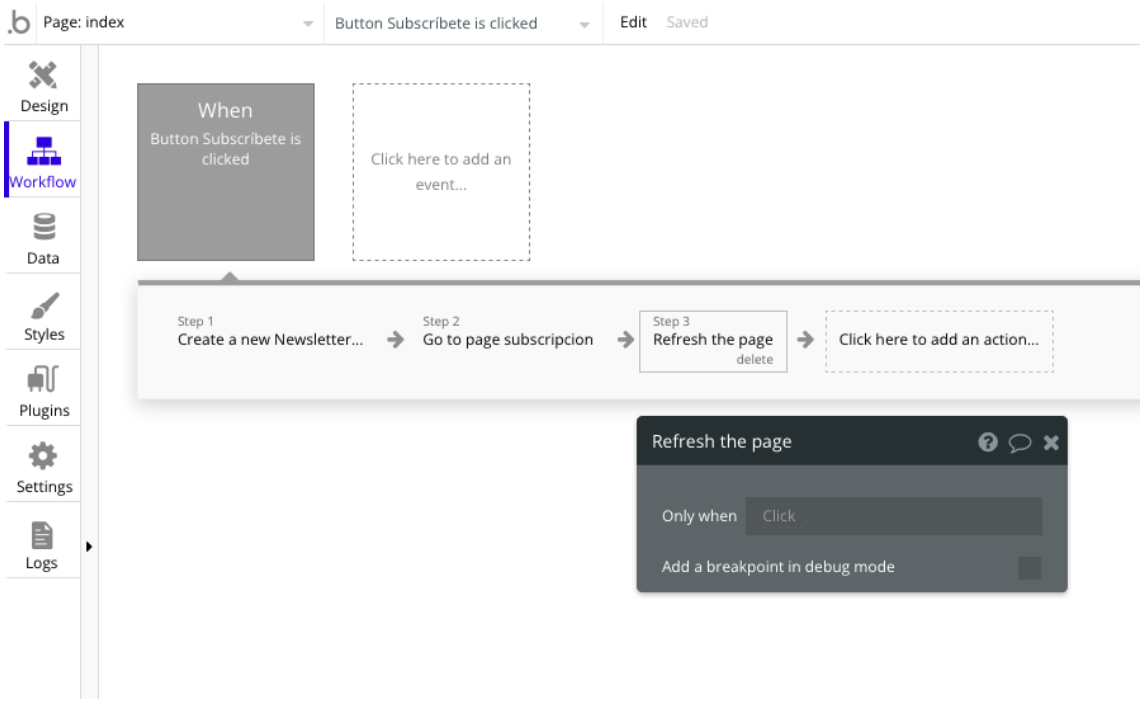
Una vez terminado este paso, se puede comenzar a trabajar con el flujo que especificará a la aplicación que acción realizar una vez que se pulse el botón de suscribirse. El primer nodo de este flujo, corresponderá a la creación de un registro en la base de datos (especificando el tipo de dato deseado, en nuestro caso Newsletter). Este registro cogerá el texto que se encuentre en el campo email.



El segundo nodo indicará a la aplicación que redirija al usuario a la ventana de confirmación de su suscripción previamente creada.

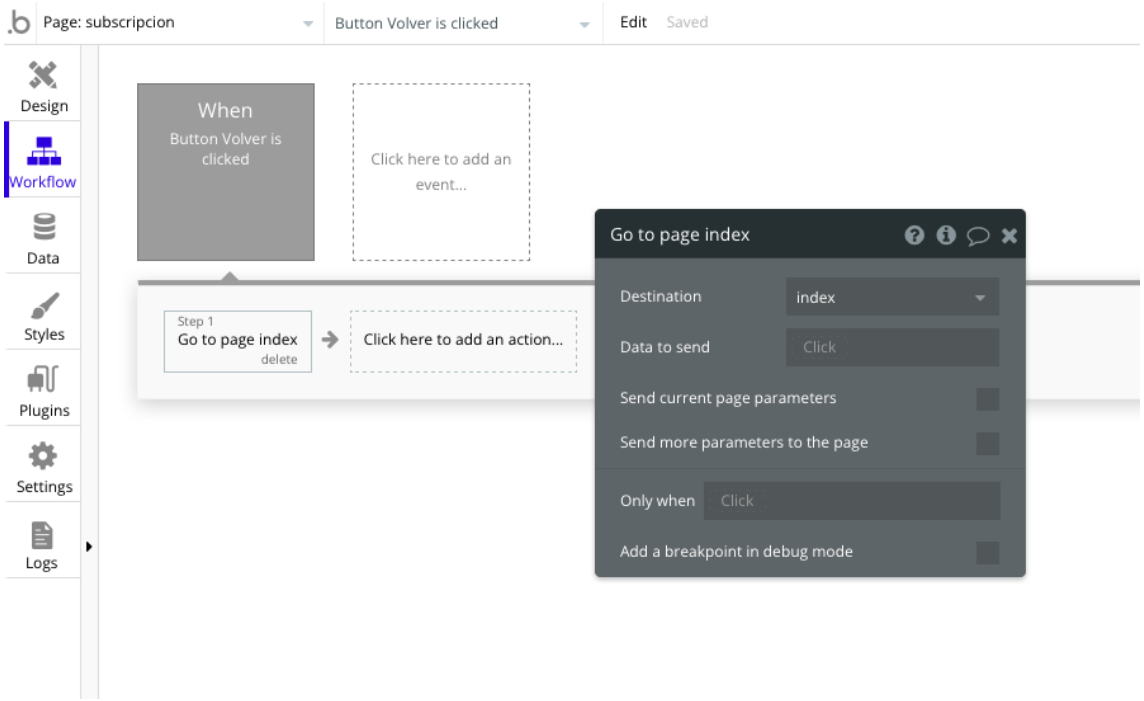


Finalmente, el último nodo especificará al programa que refresque la página actual para que, como se observará a continuación, cuando el usuario vuelva atrás, se encuentre la página con el campo email restablecido.



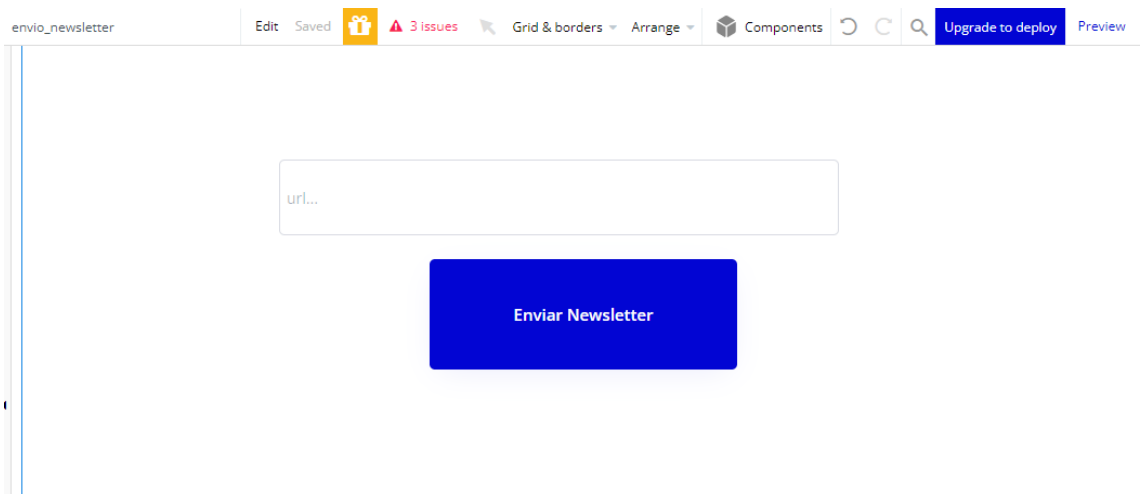
Una vez el usuario se haya registrado a la newsletter y se le haya mostrado la pantalla de confirmación, podrá pulsar el botón de volver a la ventana anterior.

Este botón, también tiene un flujo asociado. En este caso, consta de solo un nodo. En este, se le indica que ha de volver a la ventana de “index”, en la cual se encuentra el cuadro de suscripción.



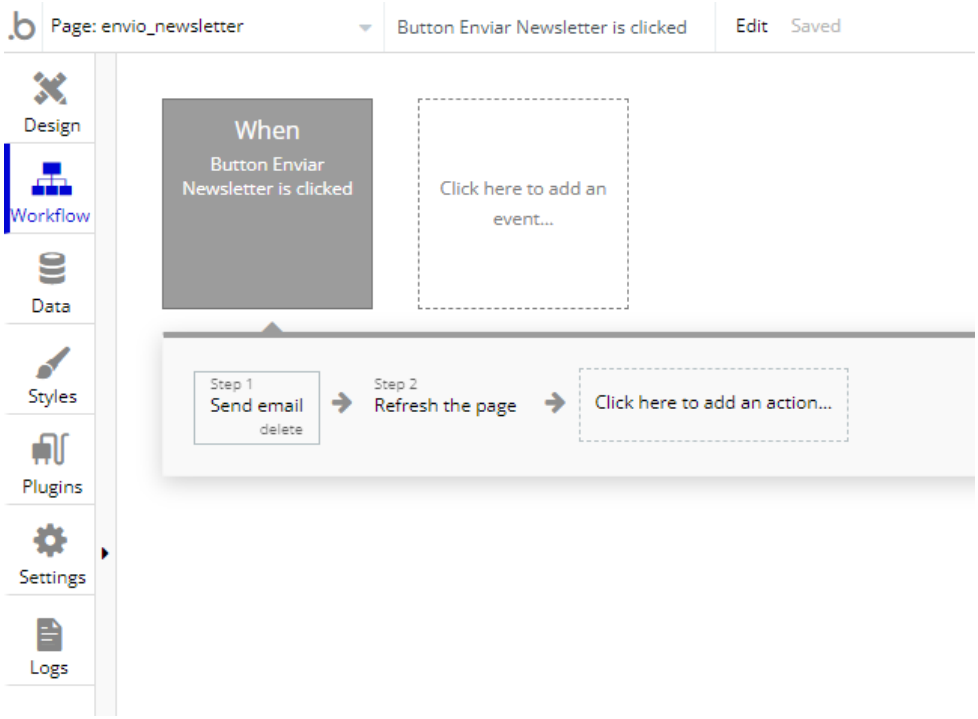
A continuación, para realizar las newsletters se utilizará WordPress como plataforma para escribir las diferentes entradas y para crear el diseño de las mismas.

Para empezar, se creará una forma para que el gerente de la newsletter pueda enviar una entrada en la newsletter con un simple click a todos los usuarios subscritos.



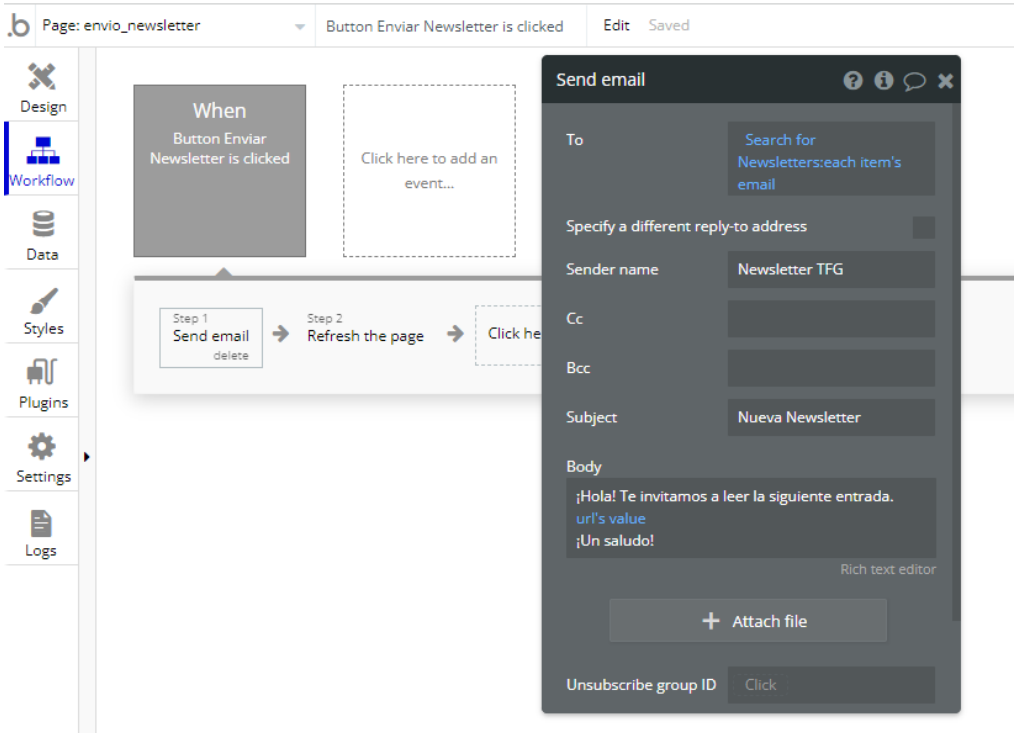
Como se puede observar, hay un campo donde el administrador introduce la url de la entrada que quiera enviar. Pulsando el botón de “Enviar Newsletter” se distribuirá un correo con formato establecido, el cual se comentará a continuación, junto con la url especificada a todos los usuarios de la newsletter.

Así, el flujo que se ejecuta al pulsar el botón es el siguiente.

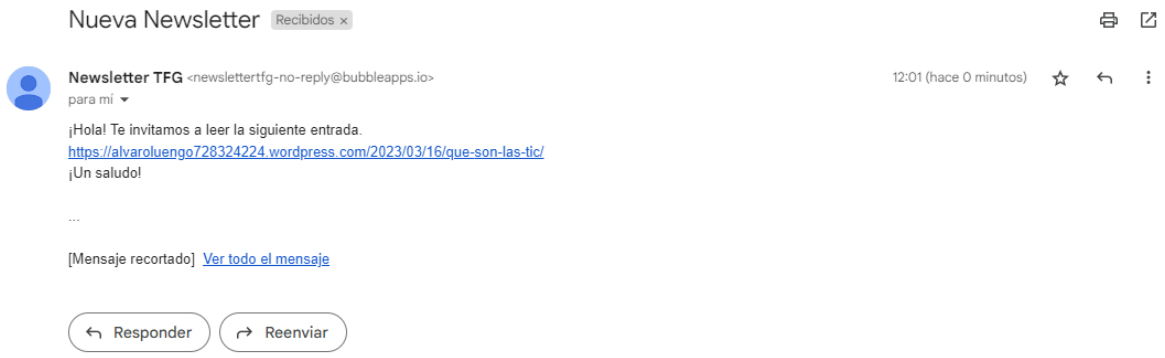


Consta de dos simples pasos. Primero, se distribuye el correo a todos los usuarios y una vez este proceso ha finalizado, se refresca la ventana.

Entrando más en detalle en el apartado del correo. El nodo de envío del correo a los usuarios utilizará el link previamente especificado, adquirido en formato dinámico, y lo situará en un correo con un formato previamente especificado. Este formato puede ser reutilizado infinitamente y puede ser modificado al gusto del administrador. Finalmente, se recorrerá la tabla de usuarios de la newsletter, enviando el correo a cada uno de ellos.



El resultado de enviar uno de estos correos sería el siguiente.



Y el resultado de pinchar el link redirigirá al usuario a la entrada de WordPress que se había creado con anterioridad.

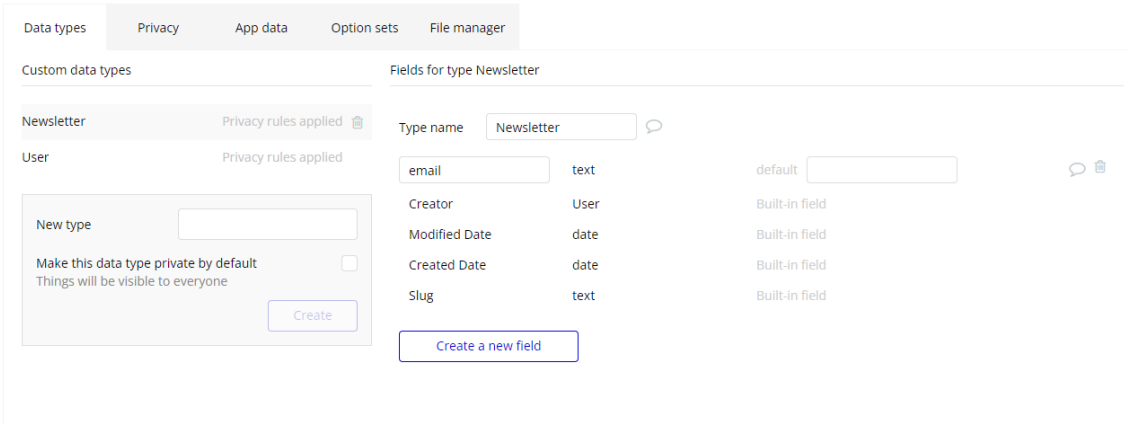
Este sistema es muy flexible ya que lo único necesario es disponer de un link de acceso a la entrada. De esta forma no se limita al uso de una sola aplicación, dando libertad al usuario para usar las herramientas que más le convengan.

Elección de la aplicación

Como ya se ha comentado en el apartado en el cual se realizaba la comparación de las distintas características y capacidades de las herramientas escogidas, Bubble.io es una gran herramienta para personas que no dispongan de la formación en programación necesaria para crear una aplicación y no dispongan del tiempo o las ganas de aprender las complejas tareas de configuración, y en algunos casos de codificación, que poseen otras soluciones de low-code. Así, esta herramienta tiende más al no-code, buscando ofrecer el mayor nivel de abstracción posible para que el usuario final pueda utilizar la aplicación desde el primer instante, sin necesidad de revisar un manual o buscar tutoriales en internet. Esto se puede observar en cómo han situado los elementos con los que interactúa el usuario. Si uno se fija en la interfaz de diseño, aparenta ser más una herramienta de diseño de páginas webs, como WordPress, que una herramienta con la cual desarrollar una aplicación de negocio. Esto hace que un usuario que ya haya estado en contacto con estas herramientas se familiarice más rápido con la interfaz de Bubble.io.

Para facilitar las tareas al usuario, Bubble.io ofrece dentro de sí una herramienta de diseño de interfaces, así como una de almacenamiento de información, las cuales pueden ser accedidas de forma rápida y sencilla a través del propio menú de diseño. La forma simple y accesible con la que se muestra al usuario facilita el entendimiento a este. Si bien es cierto que esta simpleza es su mayor debilidad, impidiendo crear aplicaciones de una gran complejidad, es una buena herramienta para aquellas numerosas soluciones que las empresas pueden necesitar y que pueden ser creadas de forma sencilla y rápida con esta aplicación, en comparación de si se tuviera que hacer con un proceso manual de codificación.

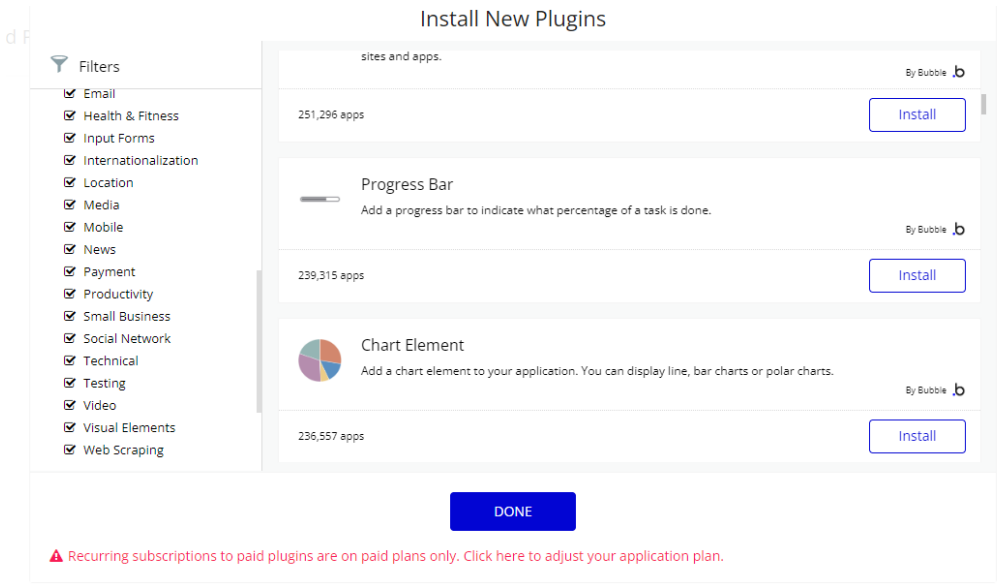
La base de datos, si bien no es del mismo nivel de complejidad como otras soluciones dedicadas en exclusiva a esta tarea, tales como PostgreSQL y su IDE pgadmin, por ejemplo, si es capaz de suplir la mayoría de necesidades que una aplicación al uso podría necesitar. Permite crear tipos de datos personalizados por el usuario y los campos del mismo son personalizados según las necesidades concretas.



Tipos de datos [6]

Bubble.io permite la implementación sencilla de plugins que amplían las funcionalidades de la aplicación sin necesidad de recurrir a herramientas de terceros. Así, Bubble.io es capaz de conectarse con servicios de alojamiento en la nube, plataformas para realizar transacciones monetarias, servicios de marketing, etc. También se pueden implementar diseños propios mediante la inclusión de HTML personalizado allá donde lo desee el usuario. Cabe tener en cuenta que para poder instalar nuevos plugins dentro de la aplicación, el usuario deberá estar subscrito a alguno de los planes de pago.

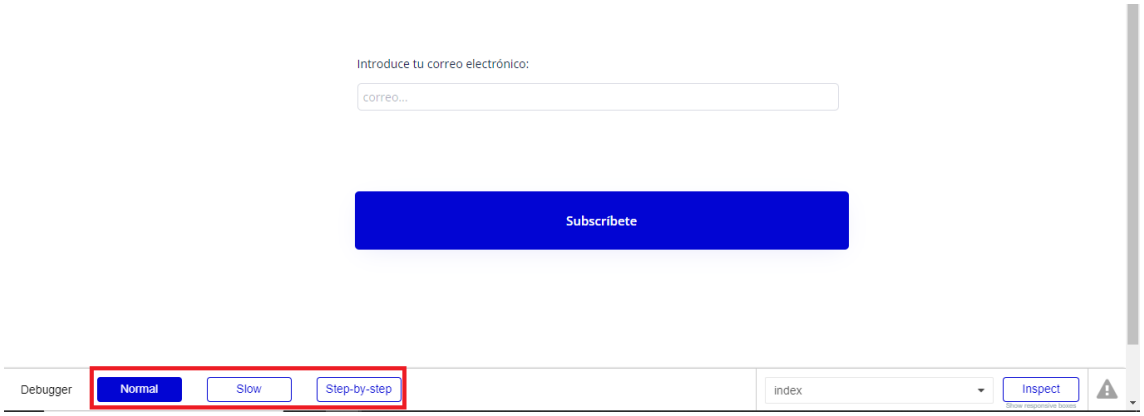
Plugins en Bubble.io:



Plugins [6]

Además, una característica que hizo que la solución Bubble.io fuera más atractiva que el resto, es que esta, permite probar la funcionalidad de la aplicación sin tener que desplegarla y por tanto sin tener que pagar. De esta forma, y gracias a la libertad que el usuario tiene a la hora de diseñar interfaces (alto y ancho de la pantalla, por ejemplo) el usuario puede diseñar su aplicación teniendo en cuenta distintos contextos, ya sea en un ordenador, tablet o móvil y comprobar el correcto funcionamiento de las características implementadas.

Bubble.io también permite, a la hora de probar la aplicación, aplicar lo que se podría denominar un debug, por el cual los pasos de la ejecución se realizan de forma pausada. Esto resulta una característica verdaderamente útil a la hora de entender la aplicación para los nuevos usuarios, así como descubrir los potenciales errores, esto gracias a que Bubble.io, al estar en modo debug, indica con total claridad las acciones que se están realizando en la instancia actual, mostrando los datos de entrada o los de salida que se generen en dicho instante. Bubble.io también ofrece, a la hora de probar la aplicación, un modo de ejecución lento, en el cual se simula como sería el uso de la misma en momentos de estrés, como podría ocurrir en el caso de que muchos usuarios estuvieran utilizándola simultáneamente.



Estas características no están presentes en el resto de soluciones que se han evaluado en este documento, lo que dota a Bubble.io de una particularidad de gran utilidad que lo distingue del resto.

A todo lo ya comentado, se suma una característica a notar y es que posee un plan gratuito. Si bien para acceder a algunas características, como instalar plugins nuevos o utilizar APIs y backend workflows, o tener un máximo de 200 visitas por mes, el plan gratuito es lo suficientemente potente como para construir aplicaciones simples. Esto es de gran utilidad para automatizar características internas a una compañía las cuales, al no estar de cara al público, no necesitan tener un gran flujo de usuarios. Aun así, Bubble.io ofrece planes de pago con precios no muy altos que van desde 25€ al mes hasta 115€ al mes. También se ofrece un plan empresarial, su precio dependería de una evaluación de la empresa ya que esta solución se enfoca a grandes corporaciones. Sin embargo, la versión gratuita ofrece de por sí soluciones para automatizar tareas rutinarias a negocios que no disponen de una gran facturación pero que, como muchos negocios, ven la necesidad de realizar un proceso de modernización mediante herramientas destinadas a optimizar el proceso empresarial y, con las cuales, añadir valor a la organización en su conjunto.

6. Tendencias de la tecnología

Las tecnologías low-code y no-code han ganada gran popularidad en los últimos años gracias a su capacidad de crear aplicaciones y soluciones personalizadas sin la necesidad de habilidades avanzadas de programación. Gracias a esto, más personas tienen acceso al desarrollo software, haciendo que empresas de menor volumen de facturación puedan desarrollar aplicaciones que ayuden a automatizar aspectos del modelo de negocio, sin depender de empresas externas o sin tener que contratar personal adicional especializado en aspectos de programación.

Con la atención que han recibido estas tecnologías, sobre todo tras la última pandemia de SARS Covid-19, se puede esperar tendencias que apunten al crecimiento de este tipo de tecnologías en aspectos que permitan una mayor capacidad de desarrollo, por la cual aplicaciones de una mayor complejidad puedan ser creadas usando la filosofía que estas dos tecnologías proponen, potenciando herramientas que carezcan de procesos de codificación como medio para crear soluciones de mayor tamaño y complejidad inherente.

De esta forma, a continuación, se tratará qué caminos pueden seguir estas tecnologías para poder entender cuáles de ellas son las más prometedoras, siguiendo estudios que apuntan a una clara evolución de las mismas.

Un campo prometedor para ambas tecnologías, low-code y no-code, es el de las inteligencias artificiales. Las inteligencias artificiales tienen aspectos que encajan con el objetivo de las tecnologías que se tratan en este estudio. En primer lugar, las IA son capaces de automatizar procesos de forma sumamente eficiente, eliminando si cabe un mayor componente de interacción por parte del usuario hacia el desarrollo de dicha automatización. Así, sumando la gran capacidad que las inteligencias artificiales han alcanzado en el campo de la comprensión en los últimos años, un usuario solo habría de especificar lo que desea a la inteligencia artificial y está sería capaz de automatizar dicha tarea. El usuario no tendría que disponer de conocimientos de programación, ni siquiera utilizar un lenguaje de programación, ya que las IA son capaces, a día de hoy, de interpretar con un alto grado de exactitud el lenguaje que los seres humanos utilizamos en nuestro día a día.

Esta forma de utilizar las IA posee un enfoque más cercano al no-code que al low-code, debido a la ausencia del aprendizaje y utilización de un lenguaje de programación. Sin embargo, la versatilidad que en los últimos años han alcanzado las tecnologías de inteligencia artificial, han conseguido que se pueda disponer de enfoques variados en cuando a la complementariedad de esta con otros sistemas, haciendo que el nivel de abstracción conseguido con su implementación afecte en mayor o menor medida al proceso de codificación.

Debido a esto, también han surgido tecnologías que utilizan las inteligencias artificiales como forma de reducir la carga de trabajo a través de un enfoque que se asemeja en mayor medida al de las tecnologías low-code. Estas adoptan en ocasiones la forma de un asistente que ayuda al programador en el proceso de codificación, autocompletando código, sugiriendo mejoras o resaltando errores en la aplicación. De esta forma, no solo

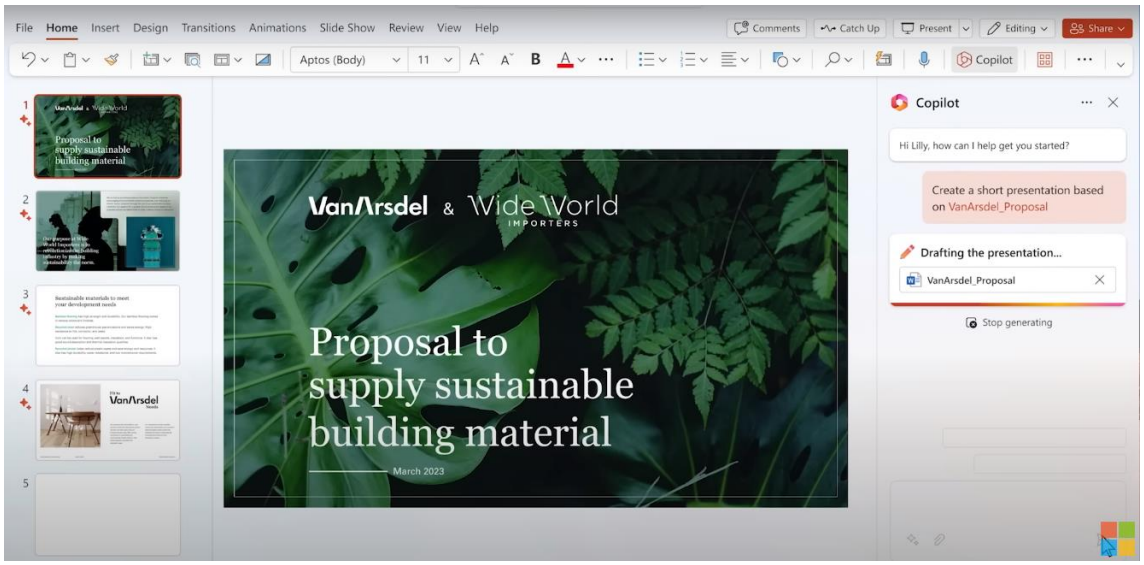
se consigue una reducción del trabajo por parte del programador, sino que también aumenta el rendimiento del trabajo que realiza.

Un ejemplo que ha ganado cierta popularidad ha sido el GitHub Copilot el cual, aun no siendo de por sí una herramienta low-code, es capaz de realizar funciones de revisión, sugerir y corregir posibles defectos en el proceso de codificación del usuario. Este concepto es similar al proceso de "pair programming" ya usado en algunos entornos, pero sustituyendo a la persona con una inteligencia artificial. De esta forma, este no es un enfoque low-code, por mucho que la IA sea capaz de desarrollar funciones por si sola, se entrará en más detalle en el próximo párrafo. Sin embargo, posee algunos elementos de accesibilidad que caracterizan a las low-code, a saber: necesidad de una cierta habilidad para utilizar esta solución de forma eficiente y la necesidad de entender el lenguaje de programación que se está utilizando.

Ejemplo de uso de GitHub Copilot [9]

Como ya se ha mencionado, GitHub Copilot no es una herramienta low-code ni no-code ya que se enfoca en la ayuda al programador en el proceso de codificación y, aunque es capaz de reducir la cantidad de codificación realizada por el usuario, por ejemplo, redactando funciones por sí misma en base a un análisis del programa, no busca el objetivo que caracterizan a las tecnologías que se tratan en este documento, el cual es la construcción de aplicaciones en su totalidad mediante un acercamiento que reduzca el proceso de codificación y ofrezca técnicas visuales para su realización. Así, GitHub Copilot estaría enfocado a aquellos programadores que quisieran mejorar su productividad, mientras que las herramientas low-code/no-code se enfocarían en el desarrollo de aplicaciones para personas con poco o nulo trasfondo tecnológico.

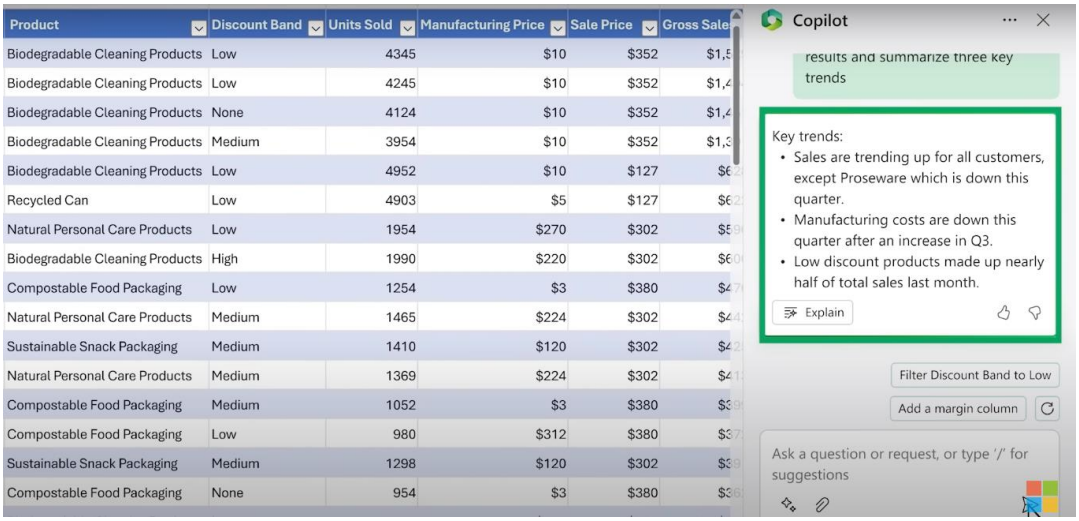
Otro ejemplo del uso de la IA en estas tecnologías es el Microsoft 365 Copilot. Esta tecnología utiliza ampliamente las inteligencias artificiales como medio de automatización de tareas. Permite realizar acciones de cierta complejidad, como podría ser el diseño y/o creación de un PowerPoint de forma automática. El usuario solo ha de especificar la información que dispondrá la herramienta para realizar dicha presentación.



Generación de presentaciones PowerPoint con Copilot [10]

Es de notar que en las empresas actuales se realizan una gran cantidad de presentaciones, con el fin de representar de forma clara las ideas. En cuanto a este aspecto, siendo Microsoft PowerPoint una de las herramientas más utilizadas, es de gran utilidad disponer de soluciones que permitan agilizar tareas en ellas. Así, esta herramienta es capaz de integrarse con todas las soluciones de Microsoft 365, las cuales son ampliamente utilizadas en las compañías.

Con esto, se puede observar la potencia de esta solución y la capacidad de reducir cargas de trabajo que esta tecnología trae consigo. Microsoft 365 copilot también es capaz de realizar tareas de interpretación y de análisis. El usuario puede pedirle, por ejemplo, que analice los datos de una tabla de Excel y que devuelva un resultado en base a ellos, como los datos más repetidos, por ejemplo. Estas capacidades ya no solo automatizan la tarea por parte de una persona de pensar y realizar el proceso necesario para crear dicha operación, o aprender lenguajes especializados en cálculos estadísticos, como R, sino que, mediante una simple consulta a la IA, esta es capaz de realizarlo de forma automática.

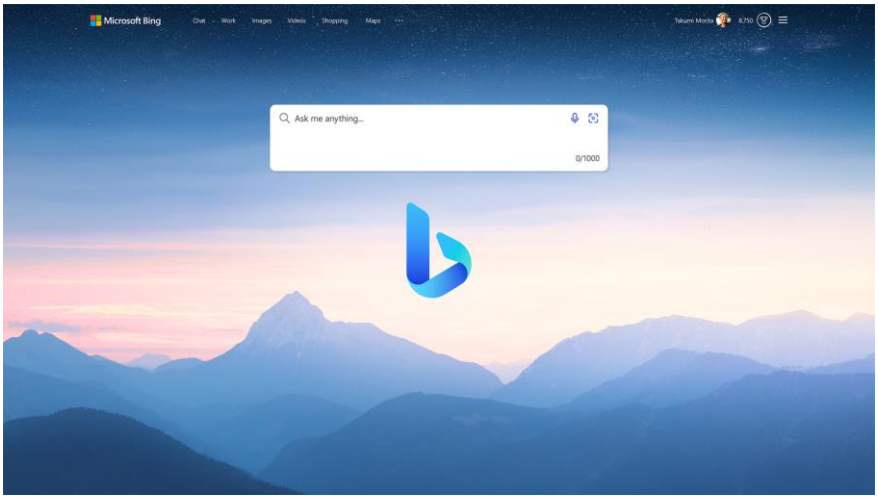


Análisis de datos Excel con Copilot [10]

Esto empodera al usuario, permitiéndole dedicar mayor tiempo a encontrar soluciones más que a pensar cómo implementar dichas soluciones, lo que también aporta beneficios a la propia empresa. De esta forma, el usuario se comunica con la herramienta de copilot con lenguaje corriente, sin necesidad de aprender complejos lenguajes de programación.

Como el propio nombre "copilot" indica, esta herramienta inició su desarrollo integrada en una extensión de Visual Studio, como apoyo a los programadores en el proceso de codificación. Sin embargo, esta ha crecido hasta abarcar cada vez más campos y convertirse, como ya se ha comentado, en una útil herramienta para tareas de automatización y reducción de carga de trabajo en el conjunto de soluciones de Microsoft 365.

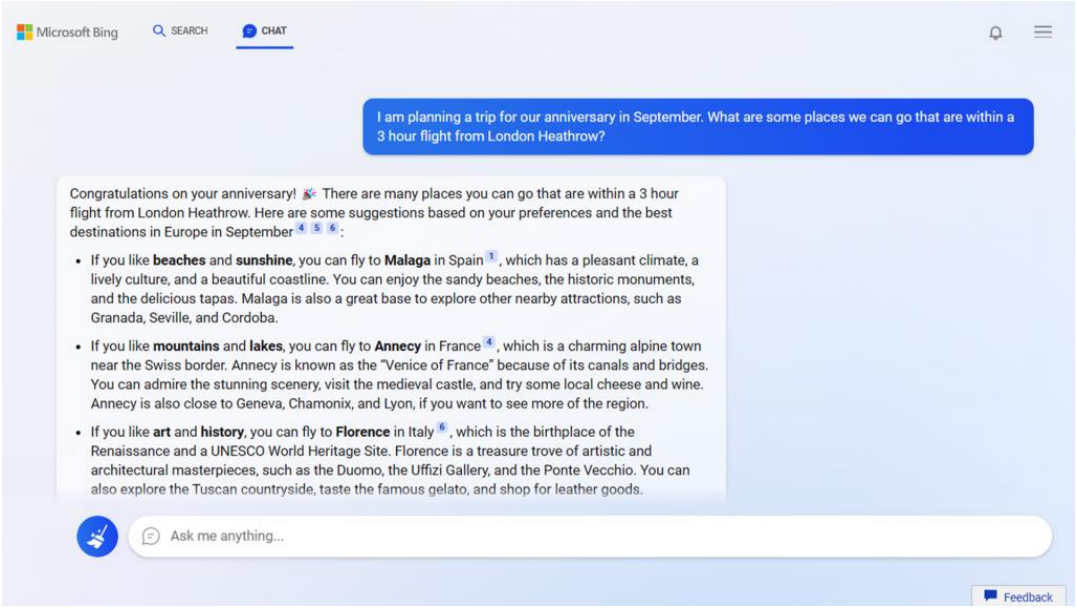
Estas tecnologías de copilot también se están empezando a utilizar en los motores de búsqueda, tanto de Edge como de Bing. Esto, con el objetivo de optimizar las búsquedas realizadas por los usuarios de estas plataformas.



Buscador Bing [11]

Esta implementación trae varias novedades a la hora de realizar consultas complejas. Por ejemplo, un usuario puede consultar a los motores de búsqueda las mejores opciones para realizar un viaje y el nuevo motor alimentado por la IA le contestará directamente los resultados, sin necesidad de que el usuario haya de realizar la búsqueda de entre los resultados que le mostrase. De esta forma, se implementa un chat interactivo en el que usuario e IA interactúan. Además, esta nueva herramienta no solo es capaz de responder a preguntas realizadas por los usuarios, también es capaz de realizar acciones de cierta complejidad, como escribir correos electrónicos, preparar entrevistas de trabajo, etc. De esta forma, se ofrece una experiencia personalizada al usuario, con respuestas dinámicas y adaptadas a las búsquedas. Aparte, es capaz de realizar resúmenes de la información contenida en un artículo o una página web o sacar los puntos claves que se tratan en el mismo.

Con esto, se puede comprobar lo íntimamente ligados que van las inteligencias artificiales al concepto de reducción y optimización del trabajo que sostienen las tecnologías low-code y no-code. Bien aplicadas, pueden ser incluso utilizadas para construir aplicaciones completas.



Chat de Microsoft Copilot a través del buscador Bing [11]

Para implementar estas características, Microsoft ha realizado cuatro avances técnicos. El primero es una nueva generación de modelos de OpenAI. Este es de mayor potencia que el ya más que conocido ChatGPT y está especializado en la realización de búsquedas. De esta forma, es más rápido, fiable y certero. En segundo lugar, han enfocado un nuevo modelo de trabajo, llamado Prometheus, para aprovechar al máximo las capacidades del modelo de OpenAI. Por último, han aplicado la tecnología de IA en el algoritmo de búsqueda para realizar búsquedas de mayor precisión.

Desde Microsoft aseguran que este desarrollo es el mayor avance en la tecnología de los motores de búsqueda de los últimos veinte años y apuntan a la implementación de la inteligencia artificial como el futuro de este sector.

Otro probable camino que estas tendencias recorrerán será el de la seguridad. Como ya se ha comentado previamente en este documento, la modularidad y reutilización que proveen estas tecnologías son un gran punto fuerte a la hora de construir aplicaciones de una forma rápida y sencilla. Sin embargo, esos mismos aspectos traen consigo ciertas características que no son muy favorables a la hora de evaluar la seguridad de las mismas. La modularidad y reutilización de ciertos elementos software pueden permitir, a un hacker que descifre su funcionamiento, tirar fácilmente las aplicaciones que hayan hecho uso de esas características.

Debido a esto, es previsible que la tecnología avance en técnicas de seguridad informática a la vez que trabaja en crear aplicaciones de mayor tamaño. Esto debido a que, si estas tecnologías han de poder ser capaces, con el tiempo, de crear soluciones para grandes empresas, estas deberán proveer de medidas de seguridad que puedan proporcionar cierto grado de confianza en el proceso de desarrollo, sobre todo si se trabaja con información sensible. Sin en esta confianza, el proceso de adopción de estas tecnologías podría restringirse a las pequeñas y medianas empresas.

Cabe considerar que, en algunos entornos, se considera que el verdadero potencial de estas tecnologías se encuentra en el low-code. Opinan que el no-code es demasiado básico y que incurre en numerosas limitaciones, sobre todo en el ámbito de la escalabilidad y la personalización, siendo por lo tanto herramientas destinadas exclusivamente para soluciones simples o para propósitos educacionales. Desde estos entornos, se enfatiza la capacidad de los lenguajes de programación como herramienta destinada a empoderar al usuario y permitirle crear soluciones adaptadas a sus necesidades. Proponen centrarse en el desarrollo de nuevos lenguajes de programación, sin dejar de buscar la reducción del proceso de codificación propia de las herramientas low-code. Con estos desarrollos de nuevos lenguajes de programación, se busca que estos sean capaces de representar con una mayor precisión las necesidades actuales de los usuarios, adaptándose así a las nuevas necesidades de mayor funcionalidad con menor cantidad de código.

Es importante tener en cuenta que no todas las tendencias han de ser positivas. Todas las tecnologías que están en proceso temprano de adopción han de superar ciertos aspectos clave antes de ser adoptados de forma generalizada.

El primer obstáculo es uno que ya he comentado en varias secciones de este documento y es el de escalabilidad. Si las tecnologías low-code/no-code aspiran a ser adoptadas por las grandes empresas estas han de ser capaces de desarrollar software de gran tamaño y complejidad de forma segura, para poder suplir las necesidades de las mismas.

Con todo esto, solo cabe destacar que ambas tecnologías prometen un futuro con amplias posibilidades de expansión y adaptación a las nuevas necesidades de las empresas. Solo el tiempo dirá cuáles de ellas serán las que predominen en este emergente mercado.

7. Conclusiones

Las tecnologías low-code y no-code han supuesto un nuevo paradigma a la hora de enfocar el desarrollo de software. Han surgido con la intención de cubrir un nicho que, aunque existente, se ha visto potenciado en los años acaecidos después de la pandemia de Covid-19.

Se han estudiado las características que ofrecen estas tecnologías a las empresas, así como el tipo de compañías a las que más beneficia su uso. A su vez, se ha podido comprobar como estas son capaces de reducir sustancialmente el proceso de codificación que el usuario ha de realizar, o en el caso de las no-code, eliminarlo por completo, permitiendo, mediante el uso de estructuras modulares, empoderar al usuario final para que este sea capaz de realizar soluciones software sin necesidad de recurrir a personal cualificado. Así, son una gran herramienta para paliar la falta de profesionales en un sector en el que existe una falta de ingenieros.

También se han estudiado cinco herramientas low-code y no-code, sus principales características y las ventajas y desventajas.

Se ha estudiado el proceso de desarrollo estándar de una aplicación no-code y su posterior implementación en la aplicación Bubble para aplicar dichos conceptos.

Finalmente, se ha indagado en el posible impacto que estas tecnologías podrán tener en los profesionales de la ingeniería, así como las tendencias que puede haber a futuro.

Para concluir, estas herramientas han cobrado protagonismo en las empresas gracias a su capacidad para automatizar tareas manuales, mejorar la eficiencia operativa y reducir el proceso de codificación, ahorrando tiempo y dinero a las compañías, permitiéndolas así enfocarse en las tareas de mayor relevancia para el modelo de negocio de las mismas.

8. Presupuesto

A continuación, se expondrá el presupuesto estimado resultante de ejecutar el proyecto expuesto en el presente trabajo.

Materiales y Licencia

Descripción	Importe
Ordenador portátil, Windows 10, 8 Gb RAM, i5 2.00 GHz	549 €
Licencia Aplicación	58 € (29 €/mes x 2 meses)

Mano de obra

Descripción	Horas
Análisis del proyecto	260
Estudio de soluciones	240
Desarrollo	40
Total	**328**

El precio de la mano de obra se valorará a 35 €/hora.

Horas totales	Precio hora	Coste Mano de Obra
328	35 €/hora	11.480 €

Coste Material + Mano de Obra

Precio desglosado	Importe total
Ordenador → 549 €	**12.087 €**
Licencia Aplicación → 58 €	
Mano de Obra → 11.480 €	

Gastos fijos variables

Categoría	Importe Total
Luz	304,29 €
Agua	150 €
Gas	758,46 €
Telefonía	477,36 €
Total	**1.690,11 €**

Gastos fijos no variables

Categoría	Coste mensual	Meses	Total
Comunidad	95,13 €	7	665,91 €

Es este apartado se especifican los gastos de la comunidad desde la cual se ha desarrollado este proyecto.

No se tiene en cuenta la cuota de autónomo debido a que, a 2023, el gobierno de la Comunidad de Madrid, lugar desde el que se realiza el presente trabajo, bonifica el 100% de las cotizaciones durante el primer año de actividad (pudiendo ampliarse a dos si los ingresos anuales son iguales o inferiores al salario mínimo interprofesional). Puesto que el trabajo se ha realizado dentro de este marco temporal, no se aplica dicho importe.

En el apartado de impuestos, tampoco se tendrá en cuenta el IAE (Impuesto sobre Actividades Económicas) ya que las empresas y autónomos están exentos de pagar este impuesto durante los primeros dos años de actividad.

Coste de Ejecución

Concepto	Importe
Coste Material + Mano de Obra	12.087 €
Gastos Fijos Variables	1.690,11 €
Gastos Fijos No Variables	665,91 €
Total	**14.443,02 €**

IBI

Impuesto sobre bienes inmuebles de la vivienda en la que se ha realizado el trabajo

Concepto	Coste Mensual	Meses	Total
IBI	48,29 €	7	338,03 €

Seguridad Social e IRPF

Concepto	Ingresos Brutos	Importe
Seguridad Social	14.443,03 €	1.241,53 €
IRPF		38,91 €

Presupuesto de ejecución por contrata

Coste	Total
Coste de ejecución	14.443,03 €
Seguridad Social	1.241,53 €
Coste antes de Impuestos	**15.684,56 €**
IBI	338,03 €
IRPF	38,91 €
Coste después de Impuestos	**16.061,5 €**
Beneficio industrial (13%)	2.087,99 €
Total	**18.149,5 €**

IVA

Finalmente, se aplica el IVA correspondiente a este tipo de servicios.

Concepto	Porcentaje	Precio	Importe
IVA	21%	18.149,5 €	3.811,40 €

Presupuesto Final

Concepto	Precio
Costes antes de IVA	18.149,5 €
IVA	3.811,40 €
Total	**21.960,89 €**

9. Bibliografía

[1] D. Di Ruscio, D. Kolovos, J. de Lara, A. Pierantonio, M. Tisi y M. Wimmer, «Low-code development and model-driven engineering: Two sides of the same coin?,» vol. 21, pp. 437-446, 2022.

[2] M. A. Al Alamin, G. Uddin, S. Malakar, S. Afroz, T. Haider y A. Iqbal, «Developer discussion topics on the adoption and barriers of low code software development platforms,» vol. 28, 2022.

[3] «Tech Republic,» [En línea]. Available: https://www.techrepublic.com/resource-library/research/research-increased-use-of-low-code-no-code-platforms-poses-no-threat-to-developers/. [Último acceso: 20 04 2023].

[4] J. Rosa-Bilbao, J. Boubeta-Puig y A. Rutle, «EDALoCo: Enhancing the accessibility of blockchains through a low-code approach to the development of event-driven applications for smart contract management,» vol. 84, 2023.

[5] S. Helzle, Low-Code Application Development with Appian : The Practitioner's Guide to High-Speed Business Automation at Enterprise Scale Using Appian., Birmingham: Packt Publishing, Limited, 2022.

[6] «Bubble.io,» [En línea]. Available: https://bubble.io/. [Último acceso: 02 2023].

[7] J. McFeetors y T. Pant, Rapid Product Development with n8n, Packt Publishing, 2022.

[8] E. Murru, Hands-On Low-Code Application Development with Salesforce, Packt Publishing, 2020.

[9] «Copilot x,» Github, [En línea]. Available: https://github.com/features/preview/copilot-x. [Último acceso: 17 04 2023].

[10] Microsoft, «The Microsoft 365 Copilot AI Event in Less than 3 Minutes,» https://www.youtube.com/watch?v=hGb9UZ8DyDc&list=WL&index=17.

[11] «Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web,» Microsoft, 07 Febrero 2023. [En línea]. Available: https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/.

[12] G. Hurlburt, «Low-Code, No-Code, What's Under the Hood?,» vol. 23, pp. 4-7, 2021.

[13] A. C. Bock y U. Frank, «Low-Code Platform,» vol. 63, pp. 733-740, 2021.

[14] A. Cypher, M. Dontcheva, T. Lau y J. Nichols, No Code Required, Morgan Kaufmann, 2010.

[15] R. Sanchis, Ó. García-Perales, F. Fraile y R. Poler, «Low-Code as Enabler of Digital Transformation in Manufacturing Industry,» MDPI AG, 2020.

[16] H. Wang y S. Wang, «Improving Student Performance by Introducing a No-Code Approach: A Course Unit of Decision Support Systems,» Journal of Information Systems Education, West Lafayette, 2022.

[17] A. Bucaioni, A. Cicchetti y F. Ciccozzi, «Modelling in low-code development: a multi-vocal systematic review,» vol. 21, p. 1959–1981, 2022.

[18] E. Mendoza, Microsoft Power Apps Cookbook - Second Edition, Packt Publishing, 2022.

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITECNICA
SUPERIOR