Tabla de Contenidos

Millonario Sin Código

Crea aplicaciones, herramientas y negocios sin escribir una sola línea

por Joe Giler

Prefacio

No escribí este libro para venderte un sueño. Ya hay suficientes personas en internet prometiendo que puedes apretar tres botones y despertar rico. Nada de esto funciona así, y si viniste aquí buscando eso, prefiero que cierres la portada ahora y conserves tu dinero.

Lo que quiero hacer en cambio es mostrarte algo que es cierto y que, en mi experiencia, está muy subestimado: las herramientas para construir software real se han vuelto silenciosamente lo bastante buenas como para que hoy una persona sin conocimientos de programación, pero con determinación, pueda lanzar productos que antes requerían un equipo de ingeniería financiado. He visto este cambio ocurrir de primera mano. He construido herramientas sin escribir código, las he vendido, las he roto, las he reconstruido, y aprendí más de los fracasos que de cualquier éxito prolijo. Este libro es la versión honesta de lo que sé.

Una palabra rápida sobre el título. "Millonario Sin Código" es una meta, no una garantía. El millón es una forma abreviada de nombrar el resultado que muchos lectores dicen querer: un negocio que paga las cuentas, recompra su tiempo y no depende de capital de riesgo ni de un título en ciencias de la computación. Algunos de ustedes llegarán ahí. Muchos construirán algo más pequeño y aun así significativo: un ingreso extra, una habilidad para trabajar de forma independiente, una herramienta que resuelva bien un solo problema. También cuento esos casos como victorias, y no voy a fingir lo contrario para mantenerte pasando páginas.

He intentado mantener este libro concreto. Cuando menciono una plataforma, es una que he usado o que he visto usar en serio a otras personas, como Bubble, Webflow, Airtable, Zapier o Glide. Cuando menciono un número, es o bien una cifra que la propia empresa publica o un rango que he visto con mis propios ojos, y te diré cuál de los dos. He evitado deliberadamente las estadísticas inventadas y las historias de éxito sin nombres asociados, porque ese tipo de relleno es exactamente lo que vuelve inservible a la mayor parte del material de "ganar dinero por internet".

También deberías saber lo que este libro no es. No es un manual paso a paso de ninguna herramienta en particular. El software cambia demasiado rápido para eso;

una guía de Bubble captura por captura estaría desactualizada antes de imprimirse. En cambio, voy tras lo duradero: cómo pensar el mundo sin código, cómo elegir un problema que valga la pena resolver, cómo ensamblar herramientas hasta formar un negocio en funcionamiento, cómo cobrar dinero y cómo evitar las trampas que hunden a los principiantes. Los botones específicos los puedes aprender de los propios tutoriales de los proveedores, que suelen ser gratuitos y excelentes.

Una última cosa. Uso mucho la palabra "nosotros" en estas páginas, porque construir cualquier cosa real rara vez es un acto en solitario, y porque estoy en esto contigo. He cometido casi todos los errores contra los que te advierto. Donde lo haya hecho, lo diré con claridad. Manos a la obra.

Introducción

Aquí va un hecho que habría sonado absurdo hace quince años: puedes construir y lanzar un producto de software funcional este mismo mes sin contratar a un desarrollador, sin aprender un lenguaje de programación y sin reunir un solo dólar de dinero externo. No un juguete. Un producto al que la gente puede suscribirse, usar todos los días y pagarte por él. Esa capacidad es nueva, es real, y la mayoría de las personas todavía no se ha puesto al día. Este libro trata de cerrar esa brecha para ti.

Durante casi toda la historia de la informática, el software estuvo encerrado detrás del código. Si tenías una idea para una aplicación, tenías exactamente dos opciones. Podías aprender a programar, lo que en la práctica significaba un año o más antes de poder construir algo que un desconocido tolerara. O podías pagarle a alguien que ya supiera hacerlo, lo que significaba decenas de miles de dólares y una dependencia de esa persona para cada cambio futuro. Ambos caminos filtraban a la abrumadora mayoría de las personas que tenían buenas ideas pero ninguna formación en ingeniería. La barrera no era la imaginación. Era la traducción, el tedioso acto de convertir una idea en las instrucciones precisas que una máquina aceptará.

Las herramientas sin código eliminan la mayor parte de esa traducción. En lugar de escribir instrucciones en un lenguaje de programación, ensamblas tu producto de forma visual: arrastrando elementos a un lienzo, conectándolos con reglas que configuras en paneles de ajustes en lenguaje sencillo y organizando datos en tablas parecidas a una hoja de cálculo. La plataforma genera el código subyacente por ti. Webflow, por ejemplo, permite a los diseñadores construir sitios web de producción manipulando un lienzo visual mientras escribe HTML y CSS limpios detrás de escena. Bubble te permite construir aplicaciones web completas, con cuentas de usuario, bases de datos y procesamiento de pagos, sin tocar un editor de texto. Zapier te permite conectar miles de aplicaciones separadas para que se pasen información entre sí de forma automática, reemplazando lo que antes era código de integración a medida.

Quiero ser cuidadoso con las definiciones, porque los términos se enturbian. "Sin código" significa que construyes sin escribir nada de código de programación; la herramienta expone todo a través de una interfaz visual. "Bajo código" significa que

construyes principalmente de forma visual, pero puedes recurrir a pequeños fragmentos de código cuando necesitas algo que la interfaz no cubre. En la práctica la línea se difumina, y los mejores constructores usan ambos con gusto. A lo largo de este libro me apoyaré en enfoques sin código porque son la puerta de entrada más accesible, pero señalaré los momentos en que un poco de bajo código, o un desarrollador pagado para una sola tarea acotada, es la jugada inteligente y no una traición a la filosofía.

Ahora, la parte honesta. El enfoque sin código no es magia, y no elimina el trabajo duro. Elimina la *programación*, que resulta ser una porción del trabajo real más pequeña de lo que los principiantes esperan. Lo que queda es todo lo que de verdad determina si ganas dinero: entender el problema de una persona concreta con la profundidad suficiente para resolverlo, diseñar algo que pueda descifrar sin un manual, conseguir a los primeros usuarios, convencerlos de que paguen, mantenerlos contentos y hacer esto una y otra vez a medida que aprendes qué funciona. Esas habilidades son el negocio. El enfoque sin código solo despeja los escombros que antes te impedían empezar.

Déjame ajustar las expectativas sobre el "millonario" del título, porque en el prefacio prometí honestidad y lo dije en serio. Llevar un producto sin código hasta el millón de dólares en ingresos es alcanzable, y hay gente que lo ha hecho, pero no es lo típico y no es rápido. Lo que es mucho más común, y aun así transforma la vida de muchas personas, es usar estas herramientas para reemplazar un sueldo, para lanzar una práctica independiente cobrándoles a clientes por los desarrollos, para sumar un flujo de ingresos rentable a un negocio existente, o para validar una idea de forma barata antes de dedicarle años. Prefiero que alcances una meta realista y estés encantado a que persigas una fantasía y renuncies en el tercer mes. A lo largo del libro mostraré la escalera completa, desde tus primeros cien dólares hasta resultados genuinamente grandes, y seré claro sobre qué peldaño alcanza cada táctica.

¿Para quién es este libro? Es para la persona con una idea y sin formación técnica que está cansada de esperar permiso. Es para el trabajador independiente o el dueño de una agencia que quiere entregar software sin una nómina de ingenieros. Es para el dueño de un pequeño negocio ahogado en trabajo manual tedioso que unas pocas automatizaciones podrían borrar. Es para el empleado que construye por su cuenta

rumbo al día en que pueda irse. No necesitas un título, una red de inversores ni ahorros para quemar. Sí necesitas curiosidad, tolerancia para ir resolviendo las cosas sobre la marcha y la disposición a lanzar algo imperfecto y mejorarlo en público.

Lo que necesitarás en la práctica es modesto: una computadora, una conexión a internet confiable y un presupuesto para herramientas que empieza cerca de cero. La mayoría de las plataformas que menciono ofrecen planes gratuitos lo bastante generosos como para construir y probar un producto real, y las que cobran suelen costar decenas de dólares al mes, no miles, hasta que tengas clientes que paguen para justificar más. Siempre señalaré a dónde va el dinero para que nunca gastes de más antes de tener pruebas de que la gente quiere lo que estás construyendo.

Así está organizado el libro. Esta primera parte explica la revolución en sí, por qué está ocurriendo ahora y qué significa para alguien como tú. De ahí pasamos a elegir qué construir, porque elegir el problema equivocado es el error más caro de todo este juego y ninguna cantidad de ejecución impecable lo rescata. Luego entramos en las herramientas mismas, agrupadas por lo que hacen y no por marca, para que el conocimiento sobreviva a la inevitable rotación del mercado de software. Después abordamos las partes de las que a nadie le gusta hablar pero que separan a los que ganan dinero de los que solo sueñan: conseguir a tus primeros usuarios, cobrar dinero, manejar las realidades operativas y escalar sin perder la cabeza. Por el camino me detendré en los fracasos, incluidos los míos, porque estudiar lo que se rompe es la forma más rápida de evitar romperse de la misma manera.

Lee esto con un proyecto en mente. El mayor predictor aislado de si alguien saca provecho de un libro como este es si construye algo mientras lo lee, aunque sea una cosita pequeña y fea. Así que, mientras avanzas, mantén una idea candidata en el bolsillo trasero y deja que estos capítulos la pongan a prueba a presión. Al final, o bien la estarás construyendo de verdad, o bien habrás aprendido lo suficiente para saber que era la idea equivocada y te habrás ahorrado meses. Ambas cosas son victorias. Empecemos por qué este momento es distinto de todos los momentos que lo precedieron.

Capítulo 1: La revolución sin código — Cómo la gente común está construyendo software que vale millones

Para entender por qué importa este momento, ayuda sentir lo absurdo que era el viejo camino. Imagina que tienes una idea en 2008 para una aplicación web sencilla, digamos una herramienta de agendamiento para tutores. Para construirla necesitarías alquilar un servidor, instalar y configurar una base de datos, aprender un lenguaje de backend para manejar la lógica, aprender un lenguaje de frontend para construir las pantallas, aprender a hacer que se comuniquen entre sí de forma segura, configurar el inicio de sesión de usuarios sin filtrar contraseñas, integrar un procesador de pagos y luego mantenerlo todo parchado y en funcionamiento. Cada una de esas cosas era una especialidad. Un solo desarrollador competente podría cobrarte entre treinta y ochenta mil dólares por ensamblarlo, y volver a cobrarte cada vez que quisieras cambiar un botón. La idea era la parte fácil. La construcción era una muralla.

Esa muralla es la que el enfoque sin código derribó, y no ocurrió de la noche a la mañana ni por accidente. Fue el fruto de una larga y poco glamorosa acumulación de infraestructura. La computación en la nube hizo que nadie tuviera que instalar sus propios servidores; podías alquilar poder de cómputo por minuto a proveedores como Amazon y Google. Los navegadores se volvieron lo bastante potentes como para ejecutar aplicaciones sofisticadas, de modo que el software podía vivir en la web en lugar de instalarse en cada dispositivo. Las interfaces de programación de aplicaciones, los conectores que permiten que una pieza de software use otra, se volvieron estándar, así que una herramienta nueva podía delegar los pagos a Stripe, la mensajería a Twilio o los mapas a Google en lugar de reconstruir todo eso desde cero. Una vez que toda esa plomería existió y fue confiable, pudo aparecer un nuevo tipo de empresa: una que envolvía la plomería en una interfaz visual amigable y vendía la capacidad de construir a personas que no sabían programar.

Qué significa realmente "sin código"

Quítale el marketing y el enfoque sin código es un intercambio simple. En el desarrollo tradicional expresas tus intenciones en texto que sigue la gramática estricta de un lenguaje de programación, y una computadora ejecuta ese texto. En el desarrollo sin código expresas las mismas intenciones manipulando una interfaz visual, y la plataforma traduce tus elecciones a ese texto por ti. La lógica sigue ahí. La base de datos sigue ahí. Las cuentas de usuario y el flujo de pago siguen ahí. Simplemente los estás especificando a través de menús, interruptores y arrastrar y soltar en lugar de sintaxis.

Esto importa porque la parte más difícil de aprender a programar, para la mayoría de las personas, no es el pensamiento. Es la precisión implacable de la sintaxis, donde un solo carácter faltante lo rompe todo y los mensajes de error dan por sentado que ya sabes lo que estás haciendo. El enfoque sin código elimina esa crueldad en particular. No elimina la necesidad de pensar con claridad sobre cómo debe comportarse tu producto. Si acaso, expone esa necesidad, porque una vez que desaparece la barrera de la sintaxis, la calidad de tu pensamiento se vuelve todo el juego.

Vale la pena nombrar las principales familias de herramientas sin código, porque las combinarás constantemente. Están los constructores visuales de web y aplicaciones como Bubble, Webflow, Glide y Softr, que producen lo que tus usuarios realmente ven y tocan. Están las herramientas de base de datos y hoja de cálculo como Airtable y Google Sheets, que guardan tu información de forma estructurada. Están las herramientas de automatización como Zapier y Make, que mueven datos entre aplicaciones y disparan acciones de manera automática. Y hay incontables herramientas especializadas que hacen bien una sola tarea, desde formularios hasta correo electrónico o pagos. Un producto real suele coser varias de estas entre sí, y aprender a verlas como bloques de construcción y no como competidoras es uno de los cambios mentales más valiosos que puedes hacer.

Por qué ahora, y no cinco años antes

A veces la gente pregunta por qué, si las piezas existían desde hace tiempo, la ola sin código alcanzó su punto máximo solo hace poco. Parte de la respuesta es la madurez. Los primeros constructores visuales eran torpes, limitados y propensos a romperse, así que los constructores serios los evitaban y las herramientas se mantenían en

nichos. A lo largo de años de iteración, las plataformas líderes se volvieron genuinamente capaces, capaces de manejar cargas reales de usuarios, pagos reales y complejidad real. Bubble, por ejemplo, pasó de ser una curiosidad a una plataforma que las startups usan para operar negocios de producción. Webflow pasó de ser un juguete de diseñadores a una herramienta sobre la que grandes empresas montan sus sitios de marketing.

La otra parte de la respuesta es que el ecosistema que lo rodea se completó. Una herramienta es mucho más útil cuando miles de otras herramientas se conectan a ella, y la difusión de integraciones estandarizadas hizo que una aplicación sin código pudiera enchufarse a todo el universo del software. Una vez que pudiste aceptar pagos a través de unas pocas pantallas de configuración en lugar de un ingeniero de pagos, y una vez que pudiste enviar correos transaccionales o mensajes de texto de la misma manera, las últimas excusas para necesitar un equipo completo de ingeniería se evaporaron para una enorme clase de productos. La revolución es menos una única invención que un umbral finalmente cruzado, donde las herramientas se volvieron lo bastante buenas y lo bastante conectadas al mismo tiempo.

Qué está construyendo realmente la gente común

Déjame aterrizar esto en los tipos de negocios que estas herramientas genuinamente sostienen, porque el abanico sorprende a la gente. En el extremo más simple, hay quienes construyen productos informativos y comunidades: un sitio de membresía de pago, un directorio que cobra por listar, una bolsa de trabajo de nicho. Herramientas como Webflow combinado con una capa de membresía, o una solución todo en uno como Softr montada sobre una base de datos de Airtable, hacen que esto sea muy accesible, y pueden generar ingresos recurrentes reales con un esfuerzo modesto.

Un peldaño más arriba en complejidad, la gente construye productos genuinos de software como servicio, donde los usuarios inician sesión y realizan trabajo dentro de la aplicación. Este es el terreno propio de Bubble. Piensa en una pequeña herramienta de relación con clientes para un nicho específico, un sistema de reservas para un tipo particular de proveedor de servicios, o una herramienta interna por la que una empresa paga para usar. Estas son más difíciles de construir y más difíciles de vender,