

Álex Vaughtton

Revolución

<NØ>

CODE

CONSTRUYE NEGOCIOS
SIN APRENDER A PROGRAMAR

ANAYA
MULTIMEDIA

Índice

1. Las revoluciones tecnológicas	15
Antes del papiro, el conocimiento era oral	15
Del papiro al libro.....	17
Las revoluciones exponenciales en la música	18
Democratización a través de la tecnología	20
Referencias.....	21
2. El desarrollo visual.....	23
El nacimiento de la interfaz visual	25
El nacimiento de la programación visual	27
¿Qué nivel de abstracción quieres?	29
La abstracción visual	31
La lógica también puede ser visual	33
Aprender a programar es fácil con Scratch.....	35
Bajando la barrera de entrada a la programación.....	36
Referencias.....	37
3. El movimiento No-code.....	39
El nacimiento del No-code: Bubble.io.....	41
Escribiendo código HTML, CSS y JavaScript de manera visual: el nacimiento de Webflow.....	43
¿Qué es No-code y qué no es No-code?	47
Las herramientas dan forma al ecosistema	48
Construyendo <i>chatbots</i> sin código	49
Convirtiendo datos en aplicaciones	50
La nueva era del desarrollo visual	52
Un movimiento por y para la comunidad	53
La IA y la nueva era del movimiento No-code	54
Referencias.....	57

4. De <i>side-project</i> a NoCodeHackers	59		
Lo de programar no es para mí.....	59		
Buscando herramientas para hacer más fácil el día a día.....	60		
Descubrir el No-code en pandemia	61		
Lanzando un proyecto en un fin de semana.....	63		
Un tuit para cambiarlo todo.....	65		
No validas nada hasta el primer euro facturado.....	66		
Creando una comunidad en torno al No-code.....	68		
Uniéndome a la familia de Minimum.run	69		
Escalando el proyecto hasta día de hoy.....	70		
5. Rotusoft	73		
Creando una empresa de rotulación desde Sarria, Lugo	73		
Creando un primer CRM con Notion para organizar los procesos.....	74		
Descubriendo herramientas: Notion.....	75		
Creando Rotusoft, un ERP para empresas de rotulación...	76		
Una aplicación creada sin diseñadores	77		
Involucrando al equipo en la construcción	77		
Rotusoft en números.....	78		
¿Qué puedo aprender de esta historia?.....	79		
6. SinOficina.....	81		
De arquitecto a <i>freelance</i>	81		
Descubriendo el mundo del emprendimiento	82		
Un <i>coworking</i> ¿online?.....	83		
No reinventes la rueda.....	84		
El nacimiento de SinOficina	85		
Escalando el proyecto y automatizando procesos	86		
¿Qué puedo aprender de la historia de SinOficina?.....	88		
7. Edify	89		
Encontrando una oportunidad en el mercado inmobiliario	89		
Validando la idea lo más rápido posible con No-code	90		
Validando el interés con una página web y un formulario..	92		
Zapier es el pegamento de internet	95		
Escalando el proyecto una vez validado.....	95		
		Explorando los límites del No-code	98
		Hacia el mundo Low-code	99
		Camino hacia la venta.....	100
		¿Qué puedo aprender de la historia de Edify?.....	101
		8. celiAC pLAN	103
		La pandemia y la celiaquía	103
		El nacimiento de una comunidad para celíacos: celiAC pLAN	104
		Notion como herramienta central.....	107
		Los intentos de monetizar el proyecto.....	108
		¿Qué puedo aprender de celiAC pLAN.....	108
		9. Manfred y La Bonilista	111
		Todo empieza con una <i>newsletter</i> : La Bonilista.....	111
		No todos los proyectos necesitan herramientas complejas ..	112
		Una oportunidad: Encontrar talento <i>tech</i>	113
		Sobreviviendo sin morir de éxito y el principio de Airtable	114
		Escalando la compañía y escalando Airtable.....	116
		Manfred se integra en SINGULAR para continuar escalando ..	116
		Lo que podemos aprender de Manfred.....	117
		10. Cómo Ontruck ahorró 10 000 euros al mes con No-code	119
		Ontruck, transportando mercancía de una manera más eficiente	119
		Déjame, que ya lo hago yo a mano	120
		Construyendo un MVP con No-code	121
		Recopilando la información de los transportistas con Typeform.....	122
		Automatizando el proceso con Zapier.....	124
		¿Cómo cambió el proceso?	124
		¿Qué puedo aprender de Ontruck?	125
		11. Minimalism.....	127
		Al principio, todo eran carteras	127
		Creando un <i>e-commerce</i> en Shopify.....	128
		Hacia los básicos de tu armario.....	130

Añadiendo una personalización más allá de los <i>plugins</i>	131
Una marca referente en el mercado.....	132
¿Qué puedo aprender de Minimalism?.....	132
12. Velneo.....	135
Visual Basic y Velázquez.....	135
Visual Trans y Velneo.....	138
Tres de cada diez kilos de hortalizas en España son gestionadas con Velneo: Fruttec.....	139
Hasta un 60 % más rápido con Velneo.....	140
¿Qué puedo aprender de Velneo?.....	141
Referencias.....	142
13. Taxdown.....	143
Hacer la renta es algo tedioso.....	143
Creando una solución para calcular cuánto podrías ahorrarte.....	144
Desarrollando en No-code para ser más ágil.....	146
Gestionando las operaciones de Taxdown con Airtable.....	147
¿Qué puedo aprender de Taxdown?.....	149
Referencias.....	150
14. KOP Stadium.....	151
¿Se puede mejorar la experiencia de ver fútbol en un bar? ...	151
¿Cómo validas una experiencia?.....	152
El dilema para escalar: ¿Código o No-code?.....	155
Creando la primera versión con Glide y Airtable.....	156
El escalado de KOP y pasando a código.....	159
¿Qué puedo aprender de KOP?.....	160
15. Creando un simulador de controladores aéreos en Bubble.....	161
Gestionando la comunicación entre avión y torre de control.....	161
Construyendo una réplica en Bubble.....	163
Si funciona, repítelo.....	164
¿Qué puedo aprender de Julián y Eurocontrol?.....	165

16. Facturar 10M dólares con una aplicación hecha en Bubble: Betterlegal.....	167
Encontrando una oportunidad en el mundo legal.....	167
Automatizando el proceso sin código, luego con código, para volver a hacerlo sin código.....	168
Replicando Betterlegal en Bubble en noventa días.....	170
Explorando los límites de Bubble.....	171
¿Qué puedo aprender de Betterlegal?.....	173
Referencias.....	173
17. Creando una aplicación para un torneo de calistenia en Flutterflow.....	175
De un TFG a una agencia internacional de Flutterflow.....	176
Validando un proyecto lo más ágil posible.....	179
Iterar más rápido, la ventaja de Flutterflow.....	181
¿Qué puedo aprender yo de Calenge?.....	182
18. El futuro del No-code.....	185
El estado actual del No-code.....	185
La empresificación del No-code.....	187
No somos No-code, somos desarrolladores.....	189
La inteligencia artificial reemplazará a los programadores... ..	190
El futuro de la construcción de <i>software</i>	192
Construye acompañado de una IA.....	195
Nunca ha sido mejor momento para aprender a desarrollar....	196
Las herramientas actuales y el futuro.....	198
El futuro de la comunidad.....	198

2. El desarrollo visual

Hemos visto cómo, cada vez que aparece una nueva tecnología, significa cambios a nivel de sociedad. Sin embargo, cuando hablamos de No-code, la palabra correcta para definirla sería «movimiento» o «revolución»; de aquí el título de este libro.

Como tantas cosas que nos parecen nuevas e innovadoras, este movimiento lleva décadas entre nosotros. Si quisiéramos buscar su origen, tendríamos que remontarnos a principios de la década de los ochenta, momento en el que el mundo estaba dominado por la primera generación de ordenadores personales.

Ya habíamos dado un salto de gigante desde los antiguos *mainframes* de IBM, ordenadores que literalmente ocupaban lo mismo que una habitación, a ordenadores con un tamaño reducido, gracias a una pequeña compañía por aquel entonces llamada Intel.

Con la invención del Intel 4004, se dio el primer paso para que los ordenadores, reservados a centros de investigación y centros educativos, pudieran reducir su tamaño hasta conseguir algo que realmente fuera práctico en un hogar. Si bien este chip no tenía potencia más que para hacer funcionar una calculadora, dio comienzo a una carrera hacia construir más y más microprocesadores, cada vez más y más potentes, permitiendo que se vislumbrara un futuro cercano en el que todo el mundo tuviera un ordenador en su hogar.

Probablemente, si ahora estuvieras viviendo esta época es muy probable que pensaras que esto era una absoluta locura, puesto que estos primeros ordenadores eran caros, difíciles de usar y de mantener. Pero el potencial estaba ahí.

Es en ese momento cuando empieza una carrera fascinante por abrir el mercado de los microprocesadores y los ordenadores más pequeños, pensados para ser usados por personas en sus casas y no por grandes empresas. Probablemente te suenen los nombres de las empresas que fueron pioneras en el movimiento, como Hewlett-Packard o Sun Microsystems, tal vez conozcas una pequeña compañía llamada Apple que se sumó a esta carrera en el año 1976 con su Apple I.

Todas ellas competían contra el gigante de aquellos tiempos, IBM, la empresa encargada de vender la mayoría de los ordenadores que utilizaban las empresas, los referentes de la innovación y el desarrollo de la computación. En 1981, deciden atender a un mercado diferente, el del gran consumo, lanzando lo que daría origen al término PC, ya que lanzaron en ese año su primer ordenador personal, el IBM Personal Computer (PC por sus siglas en inglés).



Este ordenador, aunque ahora pueda parecer rudimentario, era un paso de gigante de cara a acercar la visión de que todo el mundo tuviera un ordenador en su casa. Sin embargo, era realmente difícil de utilizar para una persona normal y corriente, sin mencionar lo

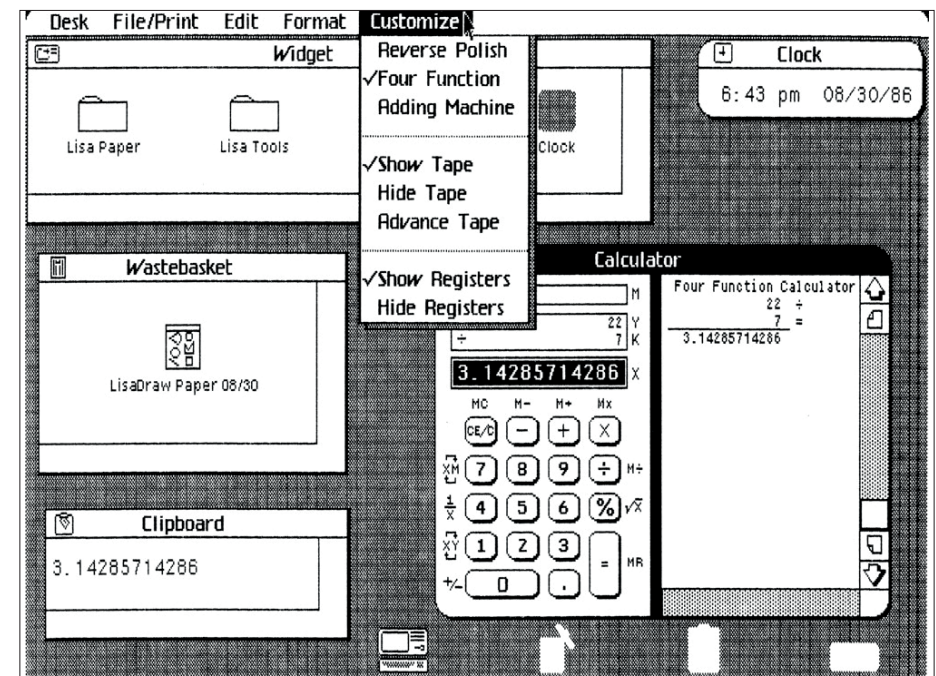
difícil que era poder permitirse un ordenador personal en aquella época. Fue el primer paso de un avance imparable, tratando de seguir las tendencias del mercado y la evolución de la velocidad de procesamiento de los microprocesadores, que aumentaba exponencialmente.

Pero todo cambia en el año 1983.

El nacimiento de la interfaz visual

Aunque ahora el término «escritorio» o abrir una ventana de una aplicación pueda sonar a algo común y obvio, los ordenadores antes de 1983 eran algo totalmente diferente. Líneas de comandos, pantallas monocromáticas...; la interacción con un ordenador era mayormente basada en texto.

Es entonces cuando Steve Jobs anuncia en 1983 el Apple Lisa, el primer ordenador del mundo que tiene una interfaz visual.



3. El movimiento No-code

En el año 2014, Forrester lanza un informe en el que, por primera vez, aúna todas estas iniciativas que permiten acelerar el desarrollo, ya sea mediante una interfaz visual o mediante componentes pre-construidos, en lo que denominan Low-code.

En su informe, Forrester, una de las plataformas que marcan la tendencia de cómo evoluciona el mercado, especialmente en el mundo de las grandes empresas, dice:

Hand-coding is too slow to develop and deliver many of the applications that companies use to win, serve, and retain customers. Some firms are turning to new, «Low-code» application platforms that accelerate app delivery by dramatically reducing the amount of hand-coding required. Faster delivery is the primary benefit of these application platforms; they also help firms respond more quickly to customer feedback after initial software releases and provision mobile and multichannel apps. Usage of Low-code platforms is gaining momentum for customer-facing applications.

Y es que uno de los grandes problemas a los que se enfrentan todas las compañías, pero especialmente las grandes corporaciones, es la dificultad que tienen para desarrollar la tecnología que les permita innovar y continuar ofreciendo valor a sus usuarios. Hay muchísimos factores que contribuyen a esto, pero sin duda uno de ellos

es que muchas veces internamente optan por desarrollar «desde cero», en los que deciden hacer todas y cada una de las partes del producto de manera interna. Sin embargo, aunque esto pueda tener sentido para algunas, muchas veces es reinventar la rueda.

Imagínate que tu aplicación necesita un formulario para introducir la información del usuario, un sistema de autenticación para poder iniciar sesión y una manera de poder enseñar gráficas basadas en la base de datos de los usuarios. Podrías hacer cada una de estas partes desde cero, diseñando un sistema para gestionar la autenticación de usuarios, otro sistema para que puedas construirte tus propios formularios incluyendo toda la lógica y la validación necesarias y otro que te permita pintar gráficas a partir de los datos que tienes. Sin embargo, lo realmente importante para el usuario no es esa parte, sino que estos son pasos accesorios a que el usuario pueda utilizar el software. Realmente, estas partes no le aportan valor al usuario, pero son necesarias para que funcione la aplicación.

Si decides acometer todas estas partes del producto desde cero, estás invirtiendo tiempo de tus desarrolladores (algo realmente limitado) en cosas que realmente no aportan valor a los usuarios. Es ahí donde puede tener todo el sentido acudir a una herramienta que ya te proporcione todos estos sistemas preconfigurados, para que tú no tengas que desarrollarlo todo desde cero, apoyándote en desarrollos de terceros que son especialistas en cada una de esas áreas, como puede ser Chart.js para las gráficas, Typeform para los formularios o Auth0 para la autenticación. De esta manera, estás aprovechando el trabajo de equipos que probablemente hagan un mejor producto en esa área en concreto y de algún modo aceleras el desarrollo de este producto, centrándote en lo que realmente aporta valor al usuario.

Las herramientas Low-code que proporcionan es precisamente un entorno en el que tengas todas estas pequeñas partes ya desarrolladas, como son bases de datos, autenticación, lógica o formularios y que te permiten desarrollar cualquier aplicación a base de ir conectando cada una de estas partes según los requisitos del producto, de manera muy similar a como los nodos permitían a los desarrolladores de videojuegos crear animaciones o texturas en sus juegos. Precisa-

mente el término «Low-code» viene de tratar de reducir la necesidad de escribir ese código a mano, dándote la opción de hacerlo cuando sea necesario, pero aprovechando partes preconstruidas para ahorrar tiempo.

El nacimiento del No-code: Bubble.io

Siguiendo con esta tendencia, en 2012 nace una de las primeras herramientas que podríamos catalogar como No-code: Bubble.

Y nace con la visión de construir un lenguaje de programación visual —de manera muy similar a cómo lo planteaba Visual Basic— que permitiera a cualquier persona sin conocimientos de programación construir aplicaciones realmente funcionales. En aquella época existían herramientas de programación visual muy centradas en permitirte construir la parte visual o interfaz de la herramienta, pero no existía una manera de construir toda la lógica de una manera visual.

Además estábamos en el origen del movimiento de «aprender a programar» que desembocaría en los *bootcamps* de programación actuales. Es entonces cuando Josh Hash, un programador que había aprendido con Visual Basic, descubre el mundo de techcofounder.com, una página pensada para ayudar a los fundadores no técnicos a encontrar perfiles con experiencia en programación que les ayude a construir su visión. Se vio bombardeado con mensajes de proyectos que buscaban cofundador técnico y decidió apostar por una pequeña herramienta de etiquetado de imágenes de stock con su socio, Jody.

No salió tan bien como se esperaban, por lo que, después de un año de intentar montar el negocio sin inversión, Josh tuvo que tomar la decisión de abandonar la compañía, no sin sentirse fatal por dejar en la estacada a su compañero y socio Jody. De ahí nace la idea que acabaría desembocando en lo que es Bubble.io hoy en día, una plataforma que buscaba permitir que todos los Jody del mundo (perfiles no técnicos) pudieran construir tecnología sin tener que depender de un cofundador técnico.

4. De *side-project* a NocodeHackers

Muchas veces, una simple decisión puede cambiar el curso de una vida. Al menos, eso es lo que me ha pasado a mí con NocodeHackers, un proyecto que empezó como una tontería de fin de semana y que tres años después cuenta con una comunidad de más de 15 000 alumnos y seis personas que trabajamos para que siga adelante. Y todo, por supuesto, sin tener que escribir una sola línea de código.

Lo de programar no es para mí

Y es que lo de programar nunca fue santo de mi devoción. En la carrera (estudí Ingeniería Mecánica) me introdujeron al mundo de Visual Basic y, aunque me pareció sencillo al principio, pronto vi que realmente no era algo que me apasionara. Construir la interfaz de la aplicación me flipaba, pero escribir el código que lo hiciera funcionar por detrás me llevaba por la calle de la amargura. Simplemente me parecía muy frustrante el que no fuera capaz de traducir lo que quería hacer a las líneas de código exacto que hacían lo que yo buscaba. Me pasaba horas buscando los errores y, aun así, no era algo que se me diera bien.

Pero siempre me ha fascinado aprender, por lo que en mis tiempos libres me dedicaba a aprender cosas *random*: desde edición de vídeo a producción de música pasando por pintar Warhammer o crear una página web. Y es ahí donde descubrí el mundo de

WordPress y empecé a hacer mis primeros proyectos de páginas web para un canal de YouTube con mi padre (*filosofíasencilla*) o para diversos proyectos que pasaron sin pena ni gloria. Con el paso del tiempo descubrí que había *plugins* como Divi o Elementor que me hacían la vida mucho más fácil para construir exactamente lo que yo buscaba. Ya no tenía que depender de la plantilla que encontrara y las posibilidades que me diera para modificarla, sino que tenía cierta libertad para modificar el aspecto visual de la página web. Me sentía con libertad.

Allá por 2018, lancé lo que sería mi primer proyecto real, de la mano de mi gran amigo Xoán Hermelo: una comunidad de fotógrafos gallegos en Instagram. Básicamente buscábamos fotos que subieran fotografías de Galicia a Instagram utilizando el hashtag #clickgalicia, y cada día nos ocupábamos de subir una imagen. Con ello también construimos su página web, una comunidad en Discord y un montón de accesorios tecnológicos que permitían que el proyecto funcionara. Sin embargo, lo que realmente hacía que fuera posible que el proyecto funcionara era Airtable.

Y es que Xoán me descubrió que gracias a esta herramienta podíamos organizar todas las publicaciones, generando el texto de manera prácticamente automática y añadiendo los hashtags ya predefinidos. Esto nos ahorra una cantidad brutal de tiempo y nos permitía organizar todas las publicaciones de la semana en minutos, en vez de tener que estar pendientes todos los días de subir la publicación.

Aún no lo sabía, pero descubrir esta herramienta me cambiaría la vida.

Buscando herramientas para hacer más fácil el día a día

En ese año estaba trabajando en ABANCA Innova, la rama de innovación abierta de ABANCA. Al ser un equipo pequeño, todos teníamos que hacer un poco de todo, pero mi principal papel era el de buscar y configurar las herramientas que utilizábamos en nuestro día

a día, ya fuera encontrar una herramienta para gestionar reservas de salas en nuestro *coworking* o construir la web en WordPress y Divi.

Sin embargo, estaba la pega de que yo no sabía programar. Mi compañero Pope sí (y me daban cierta envidia las cosas que podía hacer), por lo que mi principal recurso cuando me asignaban una tarea era buscar una herramienta que me permitiera hacerlo sin tener que programar. Fue ahí donde mi experiencia con mis otros proyectos y el tiempo libre que dedicaba a aprender cosas *random* empezó a dar frutos.

Como había descubierto Airtable y estaba enamorado de ella, empecé a utilizarla —de manera un poco clandestina— para distintas partes de mi trabajo como gestor de innovación, desde rellenar un formulario que me permitiera recoger *feedback* de los programas que hacíamos a coordinar un evento con decenas de reuniones que estaban sucediendo simultáneamente. Poco a poco fui consiguiendo que el equipo aceptara esta herramienta como parte de nuestro trabajo y hasta acabamos convirtiéndolo en algo habitual dentro del departamento. Gracias a Airtable, podíamos hacer herramientas nuestras que nos ahorraran mucho tiempo, y eso siendo un equipo de cuatro personas era una auténtica maravilla.

Todo iba sobre ruedas, pero llegando a 2019 todo cambia con el COVID.

Descubrir el No-code en pandemia

Cuando llegó el COVID, se paró prácticamente todo el mundo, y nuestro trabajo no era una excepción. Al trabajar con *startups* haciendo pilotos, realmente era difícil que pudiéramos avanzar cuando había cosas realmente mucho más importantes que lo nuestro. Por lo que, en cuanto nos mandaron a casa a teletrabajar, de repente tuvimos mucho más tiempo libre o, más bien, no había otra cosa que pudiéramos hacer.

Por ese entonces había descubierto una herramienta que prometía ser mejor que WordPress para hacer páginas web completamente

personalizadas: Webflow. Era básicamente todo lo que había soñado y buscado durante tanto tiempo, ya que me ofrecía la libertad de construir cualquier diseño que tuviera en mente, por lo que dediqué mucho de este nuevo tiempo libre a aprender a utilizar esta herramienta, siguiendo tutoriales en YouTube y los vídeos que tenían en la academia. Me enamoré de esta herramienta.

Tirando del hilo, vi que se mencionaba mucho una palabra en el entorno de Webflow, que era No-code. Fue una auténtica revelación, ya que descubrí que había miles de personas de todo el mundo que tenían el mismo problema que yo: querían hacer cosas pero no sabían programar, así que buscaban herramientas que se lo permitieran.

Entré en un *rabbit hole*, que dirían los ingleses, de buscar toda la información existente sobre este mundo del No-code y, a cada paso que daba y descubría una herramienta nueva, aparecían otras diez herramientas que merecía la pena explorar y dedicarles un tiempo. Fue ahí cuando conocí herramientas como Zapier, Make, Landbot, Bubble, Adalo, con las que pasé un buen rato de mi pandemia, jugando con ellas, intentando construir aquello de lo que nunca había sido capaz. Gracias a estas herramientas, de repente podía hacer cosas que hasta ese momento habían sido imposibles, desde páginas web que se veían profesionales a pequeñas aplicaciones que me permitieran darle un poco más de poder a Airtable. Descubrí también que gracias a Zapier podía conectarlas todas entre sí para saltarme un poco sus limitaciones y hacer cosas un poco más complejas.

Sin embargo, había algo que me chocaba: todo el contenido formativo estaba en inglés. Toda la gente que hablaba de No-code era de habla inglesa, todos los tutoriales y cursos de YouTube eran en inglés y, por mucho que buscaba, no encontraba a nadie que estuviera explicando cosas en español. Por suerte, para mí esto no era un problema, pero me imaginé que a otras muchas personas esto sí que podría suponerles una gran barrera para poder acercarse a este mundo.

Sucedió otra cosa curiosa también, que fue que cuando hablaba de estas herramientas a mi entorno cercano, todo el mundo se quedaba con una cara extraña del tipo: ¿de qué me está hablando Álex?

Yo no entendía cómo podían no estar tan asombrados como lo estaba yo. Así que me decidí a cambiar eso.

Lanzando un proyecto en un fin de semana

Siempre he sido bastante inquieto y de meterme en fregaos en los que nadie me pedía que me metiera. No me bastaba con mi trabajo a secas, sino que buscaba siempre estar haciendo cosas en mi tiempo libre, ya sea aprender a sacar fotos —cosa que acabó convirtiéndome en fotógrafo ocasional— o grabar vídeos —cosa que también me llevó a grabar algunos vídeos para empresas y grupos de música o a organizar el carnaval en el instituto—. Ese gen siempre ha estado ahí de una forma u otra, aunque es fácil echar la vista atrás y conectar los puntos, no me resultaba obvio que esto de emprender fuera una salida para mí.

Como parte de mi trabajo en ABANCA, me formé bastante en *design thinking* y *lean startups*, dos metodologías de innovación que usábamos en nuestro programa de intraemprendimiento. La idea principal de estas dos metodologías es detectar un problema que existe en el mercado, diseñar una solución y validar lo más rápido posible que esto es un problema por el que la gente está dispuesta a pagar. Incluso íbamos a dar clase sobre esto. Pero en el fondo me sentía un poco impostor por no haber creado nada propio sobre lo que explicar estas metodologías con cierto rigor.

Un poco por aburrimiento y otro poco por demostrarme a mí mismo que era capaz de seguir los principios del *design thinking*, me embarqué en la misión de lanzar y validar un proyecto lo más rápido posible. Fue el momento en el que recordé que en mi camino de aprendizaje del No-code no había encontrado contenido en español y lo bien que estaban funcionando comunidades como No-code France o Makerpad, cada una en su idioma. Tomé la decisión de intentar cambiar esto.

Aquí es donde surge el maravilloso síndrome del impostor: ¿qué voy a enseñar yo que realmente a alguien le pueda ser útil?

**¿Alguna vez has tenido una idea brillante pero te has sentido frenado porque no sabes programar?
¿Te ha pasado que piensas que necesitas gastar una fortuna en una agencia de desarrollo para hacerla realidad?**

Revolución NO-CODE es el primer libro en español que aborda en profundidad el movimiento No-code. En sus páginas, se presentarán casos reales de emprendedores y empresas que han utilizado el No-code para validar ideas, mejorar operaciones y construir negocios que han escalado sin una sola línea de código. Ejemplos inspiradores como el de Manfred, que escaló su empresa hasta tener más de 20 empleados y la vendió; o la gente de Ontruck, que ahorró decenas de miles de euros con Zapier, demuestran las infinitas posibilidades del No-code.

Con la publicación de este libro, buscamos empoderar a las personas que no son técnicas, el 97 % del mundo. A los diseñadores, a la gente de marketing, de negocio. Podrán ver que es posible construir negocios importantes sin tener que desarrollar tecnología compleja. Que ellos pueden.

Este libro es una receta paso a paso que te introducirá al maravilloso mundo del No-code, en el que además te daremos una guía detallada para empezar tu idea de negocio sin necesitar inversión y sin tener que aprender a programar.

¡Únete a la revolución No-code!

ANAYA
MULTIMEDIA

2315284
ISBN: 978-84-415-5276-0

