

DE CERO A SAAS EN 90 DÍAS

Roadmap edition

Una guía paso a paso para lanzar tu SaaS en 90 días

Índice

Introducción – Capítulo 0

FASE 1: VALIDACIÓN DE LA IDEA (Días 1–30)

- Capítulo 1 – Por qué muchas ideas de SaaS fracasan
- Capítulo 2 – Encuentra ideas que valen la pena
- Capítulo 3 – Microvalidación rápida
- Capítulo 4 – Tu propuesta mínima de valor (MVP mental)
- Capítulo 5 – Señales de que vale la pena construir

FASE 2: CONSTRUCCIÓN DEL MVP (Días 31–60)

- Capítulo 6 – Qué aprender y qué no
- Capítulo 7 – Stack mínimo para lanzar rápido
- Capítulo 8 – Prototipo rápido
- Capítulo 9 – Testeo interno y mejoras rápidas

FASE 3: PRIMEROS CLIENTES Y VALIDACIÓN REAL (Días 61–90)

- Capítulo 10 – Canales para conseguir los primeros usuarios gratis
- Capítulo 12 – Cómo hablar con tus primeros clientes
- Capítulo 13 – Iterar según feedback real
- Capítulo 14 – Señales de éxito inicial
- Capítulo 15 – Qué haría distinto si empezara hoy

Introducción – Capítulo 0

Quién soy y por qué escribí esto

Antes de nada, déjame presentarme brevemente. Si vienes de mi canal de youtube ya me conoces un poco pero por si acaso me presento, mi nombre es Victor González soy un programador de 23 años que desde hace dos intenta ganar dinero por internet, cada fracaso me enseñó lecciones valiosas que hoy quiero compartir contigo.

Escribí esta guía porque cuando empecé, no encontré un recurso que me explicara de forma clara y práctica cómo pasar de una idea a un producto real. Todo lo que encontraba era demasiado teórico, demasiado técnico, o simplemente irrelevante para alguien que quería empezar desde cero.

Este roadmap es el que me hubiera gustado tener. No es perfecto, pero es práctico y está basado en experiencia real.

Qué NO encontrarás aquí

Quiero ser honesto desde el principio sobre las limitaciones de esta guía:

- **No es un curso de programación.** Asumo que tienes conocimientos básicos o estás dispuesto a aprenderlos por tu cuenta.
- **No es una fórmula mágica.** El éxito requiere trabajo, persistencia y un poco de suerte.
- **No promete resultados garantizados.** Cada proyecto es diferente.

Cómo usar este roadmap

Te recomiendo seguir esta guía según el punto en el que te encuentres con tu proyecto:

1. **Lee todo primero.** Lee la guía completa para tener una visión general del proceso.
2. **Respeta las fases.** No saltes a construir antes de validar. Este es el error más común y costoso.
3. **Adapta los tiempos a tu realidad.** Si estás full time en esto, quizás puedas ir más rápido. Si es un proyecto paralelo, tómatelo con más tiempo requiere paciencia.
4. **Documenta tu progreso.** Lleva un diario de decisiones y aprendizajes. Te será útil más adelante.

Ahora sí, empecemos con la primera fase: la validación de tu idea.

FASE 1

VALIDACIÓN DE LA IDEA

Días 1 – 30

Capítulo 1 – Por qué muchas ideas de SaaS fracasan

El problema más común: construir antes de validar

El cementerio de startups está lleno de productos perfectamente contruidos que nadie quería comprar. Es un patrón que se repite una y otra vez: un emprendedor tiene una idea brillante, pasa meses desarrollándose en secreto, la lanza y sorpresa no hay nadie esperando por él.

Este fenómeno tiene un nombre: "construir en el vacío". Ocurre cuando asumimos que sabemos lo que el mercado necesita sin tomarnos el tiempo de verificarlo. La emoción de crear nos ciega ante la realidad de que a lo mejor nuestra idea no es tan buena como pensamos.

Ejemplos reales y errores típicos

Veamos algunos patrones de fracaso que he observado (y experimentado):

El síndrome del "yo lo necesito". Asumes que porque tú tienes un problema, millones de personas también lo tienen. A veces es verdad, pero muchas veces tu situación es única o el problema no es lo suficientemente doloroso para que la gente pague por solucionarlo.

La trampa de la funcionalidad. Pasas meses añadiendo características porque crees que necesitas tener "todo" para lanzar. Mientras tanto, nunca pruebas si la funcionalidad core tiene demanda real.

El miedo a compartir la idea. Trabajas en secreto por miedo a que te roben la idea. Resultado: nunca obtienes feedback hasta que es demasiado tarde. La realidad es que las ideas valen poco; la ejecución lo es todo.

Ignorar las señales negativas. Cuando alguien te dice que no pagaría por tu producto, buscas excusas en lugar de escuchar. "No entienden mi visión" es la frase favorita de quienes fracasan.

El antídoto para todo esto es simple pero difícil de ejecutar: validar antes de construir. Los próximos capítulos te mostrarán exactamente cómo hacerlo.

Capítulo 2 – Encuentra ideas que valen la pena

Cómo detectar problemas reales

Las mejores ideas de SaaS no nacen por arte de magia, sino de la observación atenta de problemas reales. Aquí tienes las fuentes más fértiles para encontrar ideas que merecen tu tiempo:

Tu propia experiencia profesional. ¿Qué tareas repetitivas haces en tu trabajo? ¿Qué herramientas usas que son frustrantes? ¿Qué procesos podrían automatizarse? Las mejores ideas vienen de problemas que conoces íntimamente.

Comunidades online. Reddit, foros especializados, grupos de Facebook y LinkedIn son minas de oro. Busca hilos donde la gente se queja de herramientas existentes o pregunta cómo resolver problemas específicos.

Competidores con mala reputación. Revisa las reseñas de 1-2 estrellas en G2, Capterra o Product Hunt. Los clientes insatisfechos te dicen exactamente qué falta en el mercado.

Tendencias emergentes. Nuevas regulaciones, cambios tecnológicos (como la IA), o transformaciones en industrias crean oportunidades. Quien llega primero con una solución adecuada tiene ventaja.

Señales de que la gente pagaría

No todos los problemas merecen una solución de pago. Busca estas señales:

1. **El problema es frecuente.** Un dolor que ocurre diariamente genera más urgencia que uno mensual.
2. **La gente ya está pagando por soluciones imperfectas.** Si existen competidores con clientes, hay mercado.
3. **El problema tiene impacto económico.** Las empresas pagan por ahorrar tiempo, reducir costos o aumentar ingresos.
4. **La gente busca activamente soluciones.** Usa Google Trends, busca volumen de keywords relacionadas. Si hay búsquedas, hay demanda.

Capítulo 3 – Validación rápida

Cómo testear sin programar

La microvalidación es el arte de confirmar interés real antes de escribir una sola línea de código. El objetivo es obtener evidencia de que la gente pagaría por tu solución invirtiendo el mínimo tiempo y dinero posible.

Hay una regla de oro: las opiniones son gratis, los compromisos cuestan. Cuando alguien dice "buena idea" no significa nada. **Cuando alguien te da su email, agenda una llamada, o paga por adelantado, eso sí es validación real.**

Herramientas de validación

Landing pages. Crea una página simple que explique tu propuesta de valor y tenga un formulario para capturar emails de interesados. Herramientas como Cardd, Webflow o incluso una página de Notion pueden servir. Mide cuántas visitas se convierten en registros.

Encuestas dirigidas. No hagas encuestas genéricas. Diseña preguntas que revelen comportamiento real: "¿Cuánto gastas actualmente en resolver este problema?" es mejor que "¿Te gustaría una herramienta que...?"

Entrevistas de problema. Habla con 10-15 personas de tu público objetivo. No les cuentes tu solución; pregúntales sobre el problema. Escucha cómo lo describen, qué han intentado, cuánto les molesta. Las entrevistas bien hechas revelan insights que ninguna encuesta puede capturar.

Preventas. La validación más fuerte es que alguien te pague antes de que el producto exista. Ofrece un descuento de "early adopter" a cambio de pago anticipado. Si nadie paga, tienes tu respuesta.

Test de anuncios. Con 50-100€ en Google Ads o Facebook Ads puedes medir el interés real. Crea un anuncio que lleve a tu landing page y mide el costo por lead. Si es demasiado alto, la demanda probablemente no está ahí.

Cómo hacer entrevistas efectivas

Las entrevistas son tu herramienta más poderosa. Aquí van algunas reglas:

- Nunca preguntes "¿comprarías esto?". La gente miente para ser amable.
- Pregunta sobre el pasado, no sobre el futuro. "¿Qué hiciste la última vez que tuviste este problema?"
- Calla y escucha. En esta fase tu trabajo es robar información, no convencer.

Capítulo 4 – Tu propuesta mínima de valor (MVP mental)

Definir el núcleo de tu SaaS

Antes de construir nada, necesitas claridad absoluta sobre qué problema específico vas a resolver y para quién. Este es tu "MVP mental": la versión más simple posible de tu producto que todavía entrega valor real.

Responde estas preguntas con precisión:

1. **¿Quién es tu cliente ideal?** No "empresas" ni "profesionales". Sé específico: "Agencias de marketing de 5-20 empleados que gestionan redes sociales para múltiples clientes."
2. **¿Qué problema concreto resuelves?** No "mejorar la productividad". Algo como: "Reducir el tiempo de creación de informes de redes sociales de 3 horas a 15 minutos."
3. **¿Cuál es tu diferenciador?** ¿Por qué alguien elegiría tu solución sobre las alternativas existentes? Puede ser precio, simplicidad, un nicho específico, o una característica única.
4. **¿Cuál es la única funcionalidad esencial?** Si tu producto solo pudiera hacer UNA cosa, ¿cuál sería? Esta es tu versión MVP

Evitar características innecesarias

El síndrome de las funcionalidades infinitas es el asesino silencioso de los SaaS. Cada característica que añades multiplica la complejidad: más desarrollo, más bugs, más documentación, más soporte.

Aplica el test de la funcionalidad:

- ¿Esta funcionalidad es necesaria para que el producto entregue su valor core? Si no, elimínala.
- ¿Algún cliente real la ha pedido y ha mostrado disposición a pagar por ella? Si no, ponla en la lista de "quizás luego".
- ¿Puedo lanzar sin ella y añadirla después? Sí sí, lanza sin ella.

Recuerda: puedes añadir funcionalidades infinitas después del lanzamiento. No puedes recuperar los meses perdidos construyendo cosas que nadie quería.

Capítulo 5 – Señales de que vale la pena construir

Métricas simples de validación

Después de las semanas de validación, necesitas evaluar objetivamente si tu idea merece pasar a la fase de construcción. No te dejes llevar por el optimismo; busca evidencia concreta.

Señales positivas fuertes:

- Has conseguido al menos 100 mails de interesados en tu landing page
- Tus potenciales clientes te contactan proactivamente preguntando cuándo estará listo

Señales de alarma:

- La gente dice que es "interesante" pero nadie te da su email o tiempo
- El público objetivo es difícil de encontrar o contactar
- Ya existen soluciones muy buenas y baratas (o gratis) en el mercado

Cuándo seguir y cuándo descartar

Aquí viene la parte difícil. Si las señales son claramente negativas, es momento de pivotar o descartar la idea. Sé honesto contigo mismo: es mejor perder un mes de validación que un año de desarrollo en el producto equivocado.

Si las señales son mixtas, considera si hay un pivote que podrías hacer: ¿otro público objetivo? ¿un problema relacionado pero diferente? ¿Un modelo de negocio distinto?

Si las señales son positivas, felicidades. Es hora de pasar a la fase 2: construir tu MVP.

FASE 2

CONSTRUCCIÓN DEL MVP

Días 31 – 60

Capítulo 6 – Qué aprender y qué no

Lenguajes y herramientas mínimos

Una de las mayores trampas para los principiantes es el **"tutorial hell"**: pasar meses aprendiendo tecnologías que nunca usarás. El objetivo no es convertirte en experto; es construir algo funcional lo más rápido posible.

Lo que necesitas saber:

- **HTML/CSS básico:** Suficiente para entender cómo funcionan las páginas web y hacer ajustes simples.
- **Un lenguaje de programación:** JavaScript (con Node.js) o Python son las opciones más versátiles y con mejor documentación.
- **SQL básico:** Saber hacer consultas simples a bases de datos.
- **Git básico:** Para gestionar versiones de tu código (commit, push, pull). Vas a trabajar solo

Lo que NO necesitas saber ahora:

- Kubernetes, microservicios
- Machine Learning o IA (a menos que sea core de tu producto)
- Arquitecturas complejas y patrones de diseño avanzados

Qué conocimientos se pueden aprender sobre la marcha

La mayoría de las cosas las aprenderás cuando las necesites. Google, Stack Overflow, ChatGPT y la documentación oficial serán tus mejores amigos. No intentes memorizar todo; aprende a buscar eficientemente.

Problemas específicos como configurar pagos, autenticación, o envío de emails tienen soluciones ya hechas. No reinventes la rueda; usa servicios existentes.

Capítulo 7 – Stack mínimo para lanzar rápido

Componentes esenciales

Frontend (lo que ve el usuario)

Prioridad: velocidad, UX decente y cero sobre-ingeniería.

Opción recomendada (default 2025):

- **Next.js + Tailwind CSS**
 - En 2025, con Next.js puedes hacer **todo**:
 - Landing pages
 - Dashboard
 - Backend (API Routes / Server Actions)
 - Autenticación
 - SSR, SEO y performance
 - Un solo repo para todo el producto
 - Ecosistema enorme y probado

No-code / low-code para MVPs express:

- **Webflow** → landings de alta conversión
 - **Bubble** → apps completas sin código
 - **Framer** → páginas simples y rápidas
-

Backend (lógica del negocio)

Prioridad: escribir la menor cantidad de backend posible.

Con **Next.js** puedes evitar un backend separado usando:

- API Routes
- Server Actions
- Edge Functions

Opciones comunes en 2025:

- **Supabase** → base de datos, auth, storage y funciones
- **Firebase** → alternativa sólida

Si necesitas backend tradicional:

- **Node.js + Express / Hono**
- **Python + FastAPI**

💡 Si tu SaaS es CRUD + auth + pagos, no necesitas un backend complejo.

Base de datos

Elección por defecto:

- PostgreSQL

Opción indie hacker ideal:

- Supabase (PostgreSQL + UI + APIs)

Ventajas:

- SQL estándar
 - Escala sin migraciones dolorosas
 - Excelente DX para founders solos
-

Pagos (monetiza desde el día 1)

Estándar de la industria:

- Stripe
 - Suscripciones
 - Webhooks
 - Billing portal
 - Documentación excelente

Alternativas que simplifican impuestos:

- Lemon Squeezy
- Paddle

🧠 Si no quieres lidiar con VAT, facturación y compliance, paga la comisión y usa un merchant of record.

Autenticación (no la construyas tú)

Nunca implementes auth desde cero.

Opción recomendada en 2025:

- Better Auth
 - Simple
 - Flexible
 - Diseñada para stacks modernos con Next.js
 - Ideal para indie hackers

Otras opciones:

- Clerk
- Supabase Aut

Incluye:

- Email / password
- OAuth (Google, GitHub)
- Magic links

Stack recomendado

Si hoy tuvieras que lanzar un SaaS solo:

- **Framework:** Next.js + Tailwind
- **Backend:** Next.js (Server Actions) + Supabase
- **DB:** PostgreSQL (Supabase)
- **Auth:** Better Auth
- **Pagos:** Stripe o Lemon Squeezy
- **Hosting:** Vercel + Supabase

👉 Con este stack puedes pasar de idea a producción en **un fin de semana**.

No complicar con frameworks innecesarios

Resiste la tentación de usar la última tecnología de moda. Un MVP feo pero funcional que genera ingresos es infinitamente mejor que una arquitectura perfecta que nunca se lanza.

Capítulo 8 – Prototipo rápido

Cómo construir una versión funcional sin perfección

El objetivo de esta fase es crear algo que funcione, no algo perfecto. Tu MVP debería ser casi vergonzoso en su simplicidad. Si no te da un poco de vergüenza lanzarlo, probablemente has tardado demasiado.

Principios del prototipo rápido:

1. **Funcionalidad sobre estética.** Usa templates, frameworks CSS prediseñados. No pierdas tiempo en diseño personalizado.
2. **Hardcodear primero, generaliza después.** Si solo tienes un tipo de usuario, no construyas un sistema de roles complejo.
3. **Usa servicios externos.** Envío de emails (Resend, SendGrid), almacenamiento (S3, Cloudinary), etc.
4. **Automatiza lo mínimo.** Algunas tareas pueden ser manuales al principio. ¿Necesitas un dashboard de administración? Quizás una hoja de cálculo sirve por ahora.

Capítulo 9 – Testeo interno y mejoras rápidas

Feedback temprano

Antes de mostrar tu producto al mundo, necesitas una ronda de testeo interno. Esto no significa esperar a que esté perfecto; significa encontrar los bugs y problemas más obvios que arruinarían la primera impresión.

Quién debería testear:

- **Tú mismo:** Usa el producto como si fueras un usuario real. Haz el flujo completo varias veces.
- **Amigos:** Que prueben sin que les expliques nada. Observa dónde se confunden.
- **1-2 personas de tu lista de espera:** Los early adopters más entusiastas a menudo están dispuestos a probar versiones beta.

Cómo iterar sin frustrarse

El feedback va a doler. Tu "bebé" tiene defectos que no habías visto. Esto es normal y necesario.

Estrategia de iteración:

1. **Clasifica el feedback:** Crítico (bugs que impiden usar el producto), Importante (confusiones frecuentes), Nice-to-have (mejoras estéticas).
2. **Arregla solo lo crítico:** Antes del lanzamiento, enfócate exclusivamente en lo que impide el uso básico.
3. **No diseñes:** Si alguien no entiende algo, añade una explicación o tooltip. No reconstruyas la interfaz.

FASE 3

PRIMEROS CLIENTES Y VALIDACIÓN REAL

Días 61 – 90

Capítulo 10 – Canales para conseguir los primeros usuarios gratis

Estrategias orgánicas

Tus primeros 10-50 clientes no vendrán de anuncios pagados. Vendrán de tu esfuerzo directo y de participar en las comunidades correctas.

Canales que funcionan:

Tu waitlist: Empezaste a construirla en la fase de validación. Ahora es momento de enviar el email de lanzamiento. Ofrece un descuento especial para los primeros en probar.

Twitter/X: Comparte tu proceso, la gente ama seguir historias de indie hackers. Muestra tu progreso, tus aprendizajes, tus errores. Construye en público.

Comunidades de nicho: Reddit (subreddits específicos de tu industria), grupos de Facebook, comunidades de Slack. No hagas spam; aporta valor y menciona tu producto cuando sea relevante.

Indie Hackers y Product Hunt: Plataformas diseñadas para lanzar productos nuevos. Prepara bien tu lanzamiento, sólo tienes una oportunidad.

Contacto directo: Identifica potenciales clientes y escríbeles personalmente. Un email bien escrito y personalizado puede abrir muchas puertas.

Qué NO hacer

- **No gastes en publicidad todavía.** Hasta que no tengas claro tu mensaje y tu cliente ideal, estarás quemando dinero.
- **No hagas spam.** Arruina tu reputación y raramente funciona.
- **No esperes viralidad.** Los primeros clientes se consiguen uno a uno, con trabajo manual.

Capítulo 11 – Cómo hablar con tus primeros clientes

Preguntas clave para entender necesidades

Tus primeros clientes son oro puro. No solo están pagando; están invirtiendo su tiempo en un producto no probado. A cambio, debes extraer todo el aprendizaje posible de ellos.

Preguntas que deberías hacer regularmente:

- "¿Qué problema específico te llevó a probar nuestro producto?"
- "¿Qué alternativas consideraste antes de elegirnos?"
- "¿Qué es lo más frustrante de usar el producto ahora mismo?"
- "Si pudieras cambiar una sola cosa, ¿cuál sería?"
- "¿Nos recomendarías a un colega? ¿Por qué sí o por qué no?"
- "¿Qué funcionalidad usas más? ¿Cuál nunca usas?"

Cómo recoger feedback sin intimidar

Muchos clientes no quieren dar feedback porque sienten que es una molestia o que serán juzgados. Hazlo fácil para ellos:

1. **Normaliza el feedback negativo:** "Estamos en fase temprana y necesitamos saber qué no funciona para mejorar."
2. **Ofrece múltiples canales:** Algunos prefieren email, otros chat, otros llamada. Adapta tu método.
3. **Sé específico:** "¿Cómo fue tu experiencia configurando X?" funciona mejor que "¿Qué opinas?"
4. **Agradece y actúa:** Cuando implementes algo que un cliente sugirió, avísale. Crea una relación.

Capítulo 12 – Iterar según feedback real

Ajustes del MVP

El feedback de clientes reales es muy diferente al de amigos o testers. Ahora estás viendo cómo se usa el producto en condiciones reales, con problemas reales.

Tipos de ajustes comunes:

- **Clarificación de UX:** Botones que no se entienden, flujos confusos, texto que no comunica bien.
- **Bugs y casos edge:** Situaciones que no contemplamos y que causan errores.
- **Funcionalidades faltantes críticas:** Cosas sin las cuales ciertos usuarios no pueden usar el producto.
- **Ajustes de pricing:** Quizás tu precio sea demasiado alto o demasiado bajo para el valor percibido.

Cuándo añadir funcionalidades y cuándo no

No todas las peticiones de funcionalidades merecen ser implementadas. Usa este framework:

Añade si: Múltiples clientes piden lo mismo, es necesario para retener clientes existentes, o encaja con tu visión del producto.

No añadas si: Solo un cliente lo pide, te desvía de tu propuesta de valor core, o requiere un esfuerzo desproporcionado. **Si es tu único clientes pues si xd**

Recuerda: es más fácil añadir funcionalidades que quitarlas. Sé conservador.

Capítulo 13 – Señales de éxito inicial

Primeras métricas que importan

Al final de los 90 días, necesitas evaluar si tu experimento ha funcionado. No te obsesiones con métricas de vanidad (seguidores, visitas). Enfócate en las que realmente importan:

Retención: ¿Los usuarios que prueban el producto vuelven? Si tienes una prueba gratuita, ¿qué porcentaje convierte a pago? Una retención mayor al 30% es una buena señal.

Engagement: ¿Con qué frecuencia usan el producto? Un SaaS que se usa diariamente tiene más potencial que uno que se usa una vez al mes.

Feedback cualitativo: ¿Los clientes expresan entusiasmo? ¿Te recomiendan a otros? ¿Te piden funcionalidades en lugar de quejarse de bugs?

Cómo saber si puedes escalar

Puedes empezar a pensar en escalar cuando:

1. Tienes un canal de adquisición que funciona de forma repetible
2. La retención es estable o mejorando
3. El producto está estable y no te pasas el día apagando fuegos
4. Entiendes bien quién es tu cliente ideal y por qué compra

Si no cumples estos criterios, sigue iterando. Escalar un producto que no funciona solo amplifica los problemas.

Capítulo 14 – Qué haría distinto si empezara hoy

Errores que cometí

Mirando atrás, hay patrones de error que se repiten. Si pudiera volver atrás, estos son los errores que evitaría:

Construir demasiado antes de validar: Mi primer SaaS tuvo 6 meses de desarrollo antes de hablar con un solo cliente.

Perfeccionismo técnico: Mucha sobre ingeniería desde el principio, microservicios, kubernetes, docker.

Ignorar señales negativas. Cuando el feedback no era positivo, buscaba excusas en lugar de escuchar. El mercado siempre tiene razón.

No cobrar desde el principio. Ofrecer el producto gratis "para conseguir usuarios" solo atrae a gente que nunca pagará.

Cómo usar esto para planear el siguiente SaaS

Tu primer SaaS probablemente no será el último. Cada proyecto es un laboratorio de aprendizaje. Documenta todo: qué funcionó, qué no, qué harías diferente.

Cuando empieces tu siguiente proyecto, tendrás una ventaja enorme: experiencia real. Sabrás qué preguntas hacer, qué señales buscar, qué trampas evitar.

El viaje de emprender en SaaS es una maratón, no un sprint. Cada intento te acerca más a tu producto exitoso.

Palabras finales

Has llegado al final de este roadmap. Ahora tienes un mapa claro de cómo pasar de una idea a un producto con clientes reales en 90 días.

Pero un mapa no es el territorio. El verdadero aprendizaje vendrá de la ejecución: de los errores que cometerás, de los clientes que te enseñarán lo que realmente necesitan, de las madrugadas resolviendo bugs.

No esperes a sentirte "listo". Nunca lo estarás. Empieza hoy con la fase 1: identifica un problema real, habla con personas que lo tienen, valida antes de construir.

El mundo necesita más productos útiles creados por personas como tú. No dejes que el miedo al fracaso te paralice. El único fracaso real es no intentarlo.

¡Buena suerte y a construir!