

10

GUÍA PRÁCTICA · CLUB DE I.A.

CÓMO CREAR TU SOFTWARE PERSONALIZADO CON IA

10 buenas prácticas probadas construyendo software real
para negocios reales — **sin ser programador y sin contratar
un equipo de developers.**



Club de I.A.
La comunidad para
escalar con IA

skool.com/club-de-ia · [@luciano.spinelli.mkt](https://twitter.com/luciano.spinelli.mkt)

DE DÓNDE SALE ESTA GUÍA (Y PARA QUIÉN ES)

En los últimos meses construimos con IA **cuatro sistemas internos reales** para nuestros propios negocios: un ERP que reemplazó más de diez planillas, una plataforma de reportes para clientes, un tablero de monitoreo de toda la operación y un panel para gestionar un funnel de reclutamiento internacional.

Ninguno existía como producto comprable con la forma exacta que necesitábamos. Todos se construyeron en semanas —no meses— usando una IA como copiloto de desarrollo. Y lo más valioso no fue el código: fue el **método** que se repitió en los cuatro. Esta guía es ese método, destilado en 10 prácticas.

→ EL MITO QUE HAY QUE ROMPER

"Para tener software propio necesitás un equipo de developers y un presupuesto de seis cifras." **Falso.** Hoy necesitás tres cosas: un problema concreto, una IA que programe (como Claude Code) y —esto es lo que casi nadie tiene— **un método de trabajo**. Sin método, la IA genera código rápido que se rompe rápido. Con método, construís sistemas que siguen funcionando solos meses después.

ESTA GUÍA ES PARA VOS SI...

SOS DUEÑO DE NEGOCIO

Y tu operación vive en planillas, formularios y copy-paste que ya no escalan.

SOS MARKETER O AGENCIA

Y querés herramientas propias: reportes, paneles de clientes, dashboards.

YA USÁS IA

Pero tus proyectos quedan al 80%: funcionan en la demo y mueren en la semana dos.

¿Preferís aprenderlo acompañado?

En la comunidad construimos estos sistemas en vivo, paso a paso.

SUMATE → skool.com/club-de-ia

PRIMERO EL PROBLEMA, DESPUÉS EL CÓDIGO

01

PRÁCTICA

EMPEZÁ POR UNA PLANILLA QUE DUELE

No arranques queriendo "un ERP" o "un sistema integral". Arrancá por **una** planilla concreta que te roba horas cada semana: la de seguimiento de clientes, la de reportes, la del scorecard. Nuestro ERP interno empezó siendo un solo módulo; el resto se sumó después, bloque por bloque, cuando el primero ya funcionaba.

EN LA PRÁCTICA → Hacé la lista de tus 5 procesos más manuales. Elegí el que más tiempo consume por semana. Ese es tu primer software.

02

PRÁCTICA

DIAGNOSTICÁ ANTES DE CONSTRUIR

Antes de escribir una línea de código, escribí un diagnóstico: qué está roto hoy, dónde viven los datos, quién toma cada decisión y con qué información. En uno de nuestros módulos, el diagnóstico detectó 18 problemas y 3 fallas estructurales **antes** de implementar nada — cada uno hubiera sido un retrabajo carísimo descubierto tarde.

El truco: pedile a la IA que te **entreviste** sobre tu proceso antes de proponerte soluciones. Es mucho mejor auditora que adivina.

EN LA PRÁCTICA → Regla de oro: "nada implementado hasta que el diagnóstico esté escrito". Un documento de una página alcanza.

EL MÉTODO QUE HACE QUE LA IA NO SE PIERDA

03

PRÁCTICA

SPEC Y PLAN ESCRITOS: EL CONTRATO CON LA IA

Cada módulo nuevo deja dos documentos guardados junto al código: el **spec** (qué se construye y por qué, aprobado por vos) y el **plan** (la lista de tareas concretas). ¿Por qué importa? Porque la IA no recuerda las conversaciones de ayer. Los documentos son su memoria — y tu contrato: si en el medio aparece algo que contradice el spec, se vuelve al spec, no se improvisa.

EN LA PRÁCTICA → Antes de cada bloque pedile a la IA: "escribí el spec y el plan, y no implementes hasta que yo los apruebe".

04

PRÁCTICA

CONSTRUÍ POR BLOQUES Y CERRÁ CADA UNO EN VERDE

Ningún sistema nuestro se construyó de una vez. Todos crecieron por fases, y cada fase se cierra con una "**puerta de calidad**": el bloque funciona, las pruebas automáticas pasan y lo verificaste con datos reales. Recién ahí arranca el siguiente. Esto elimina el clásico "software 80% terminado en todo y 100% terminado en nada".

Bonus: las pruebas automáticas se acumulan de bloque en bloque. Nuestra primera suite tenía 60; hoy tiene casi el triple, y corre entera en segundos cada vez que algo cambia.

EN LA PRÁCTICA → Terminá cada sesión de trabajo con algo funcionando de punta a punta, aunque sea chico. Nunca dejes tres cosas a medias.

Estas 10 prácticas las aplicamos en vivo cada semana.

Mirá cómo se construye software real, sin saber programar.

ENTRÁ AL CLUB →

DECISIONES TÉCNICAS QUE NO TE VAN A DOLER

05

PRÁCTICA

STACK ABURRIDO, Y REPETIDO

El "stack" es el conjunto de herramientas con las que se arma tu software. La tentación es probar lo último de moda en cada proyecto. Error. Nuestras plataformas web usan **exactamente las mismas herramientas y versiones**: un framework web probado, una base de datos en la nube que ya trae login de Google incluido, y un hosting simple. La segunda plataforma se construyó al doble de velocidad porque heredó todas las decisiones de la primera.

Y la contracara: no todo necesita lo mismo. Nuestro tablero de monitoreo no usa framework — es una página estática más un script que corre cada tanto. **El stack se elige por el problema, no por moda.**

EN LA PRÁCTICA → Elegí una vez, documentá la elección y repetila hasta que duela. La velocidad viene de no re-decidir.

06

PRÁCTICA

SEPARÁ LOS CÁLCULOS DE LAS CONEXIONES

Pensalo como una cocina: una cosa son las **recetas** (cómo se calcula un puntaje de salud, un semáforo, un funnel) y otra los **proveedores** (las conexiones a Meta, Google, tu CRM). Si mantenés las recetas en archivos separados de las conexiones, la IA puede probarlas automáticamente miles de veces en segundos, sin depender de que las APIs externas respondan.

Esta separación es la diferencia entre "tengo cientos de pruebas que corren en un segundo" y "no pruebo nada porque probar es lento". Es la práctica más técnica de la guía — y la que más paga.

EN LA PRÁCTICA → Pedile a la IA: "separá la lógica de negocio en funciones puras testeables, aparte de las llamadas a APIs". Ella sabe hacerlo; vos tenés que pedirlo.

SEGURIDAD Y CRITERIO: DONDE SE SEPARAN LOS AMATEURS

07

PRÁCTICA

CERRADO POR DEFECTO, SEGURO POR CAPAS

Regla número uno: si el sistema no puede determinar qué puede ver un usuario, la respuesta es **nada** — nunca "todo". A esto se le llama *fail-closed*. Lo aprendimos con un susto real: un permiso que "ante la duda dejaba pasar" expuso datos financieros de un área a otra. Se corrigió en horas, pero la lección quedó.

Y la seguridad nunca es una sola barrera: login con Google restringido a tu dominio, permisos por rol, y un chequeo extra en **cada puerta** del sistema. Ojo especial con los datos personales: decidí explícitamente qué es público — los datos se filtran por los *resúmenes*, no solo por las pantallas principales.

EN LA PRÁCTICA → Pedile a la IA: "ante cualquier duda de permisos, negá el acceso" y "chequeá permisos en cada endpoint, no solo en el login". Desde el día uno, no después.

08

PRÁCTICA

LA IA REDACTA, LAS FÓRMULAS DECIDEN

El error más caro que vimos: dejar que el modelo de lenguaje tome decisiones numéricas. En nuestros reportes, la IA recomendaba "escalar" campañas con un costo por lead altísimo. La solución: los **juicios** (apagar, escalar, semáforo rojo o verde) se calculan con fórmulas simples y transparentes; la IA queda para lo que hace brillante — **redactar** el análisis, resumir, explicar.

Segunda regla de oro: si un humano editó un texto generado, el sistema lo marca y la regeneración automática **nunca lo pisa**.

EN LA PRÁCTICA → Preguntate siempre: "¿esto es una decisión o una redacción?"
Decisión → fórmula. Redacción → IA.

¿Querés que revisemos TU proyecto?

En el Club traés tu caso y lo destrabamos juntos en las sesiones en vivo.

skool.com/club-de-ia

QUE FUNCIONE SOLO (Y QUE AVISE CUANDO NO)

09 PRÁCTICA

AUTOMATIZÁ CON "FOTO PERIÓDICA" + ALERTAS

El patrón que repiten nuestros cuatro sistemas: un proceso automático junta los datos cada X horas, los pre-calcula y guarda una **foto** (snapshot). La app solo muestra la última foto. Resultado: pantallas instantáneas, APIs externas que no se saturan, y si una fuente se cae, el usuario ve el último dato bueno con su fecha — no una pantalla rota.

Y el complemento obligatorio: un **chequeo de salud** que mide la frescura de cada foto. Si tiene más de un día, alarma al celular. Reglas de diseño: silencio = todo bien (solo avisa cuando hay problema), y distinguir "**está roto**" (lo arregla la técnica) de "**falta un acceso**" (lo arregla una persona) — cada alerta dice exactamente qué hacer.

EN LA PRÁCTICA → Una automatización sin alerta de frescura es una automatización que va a fallar en silencio. Nos pasó: un proceso murió de madrugada y nadie lo notó hasta que existió este chequeo.

10 PRÁCTICA

DOCUMENTÁ EL PORQUÉ Y VERIFICÁ EN REAL

Cada sistema mantiene una **bitácora viva**: qué decisiones se tomaron, qué trampas tiene, qué queda pendiente. Es lo que permite retomar el proyecto en un mes (vos, tu equipo o la propia IA) sin arqueología. Y en el código, los mejores comentarios explican el **porqué** — el "qué" ya se lee solo.

Última regla, la más importante: nada está terminado hasta verlo funcionar **con datos reales, en el entorno real**. Y nunca confíes en el resumen de la IA de que "ya está listo": abrí la pantalla, tocá el botón, mirá el resultado con tus ojos.

EN LA PRÁCTICA → Creá un archivo de bitácora desde el día uno y pedile a la IA que lo actualice al final de cada sesión de trabajo.

7 ERRORES QUE YA COMETIMOS POR VOS

Todos estos pasaron de verdad construyendo nuestros sistemas. Cada uno costó horas (o plata). Te los regalamos.

1

Permisos que "ante la duda, dejan pasar"

Una lógica fail-open expuso datos financieros entre áreas. **Ante la duda, el sistema niega.**

2

Datos privados que se escapan por los resúmenes

Excluimos una sección con datos sensibles de la vista pública... pero el resumen ejecutivo la citaba. Revisá los agregados, no solo las pantallas.

3

Dejar que la IA decida plata

El modelo recomendaba escalar campañas caras. Las decisiones numéricas van por fórmula; la IA redacta.

4

Automatizaciones que mueren en silencio

Un proceso nocturno fue interrumpido y varios tableros quedaron con datos viejos días. Sin chequeo de frescura, los fallos son invisibles.

5

Servicios pagos por consulta, sin caché ni tope

Un scraper re-consultaba lo mismo una y otra vez y vació el crédito mensual en días. Todo lo que cobra por llamada necesita caché y límite de gasto.

6

El mismo dato viviendo en cinco lugares

La relación equipo-clientes estaba en cinco tablas sin dueño, todas distintas. Elegí una fuente única de verdad y jubilé el resto.

7

Confiar en el "ya está listo" de la IA

Los agentes a veces resumen con optimismo. Verificá los archivos y resultados reales, no el relato.

Evitá los próximos 7 errores antes de cometerlos.

En el Club compartimos cada lección nueva apenas la aprendemos.

SUMATE HOY →

CHECKLIST PARA ARRANCAR ESTA SEMANA

Imprimila o marcala en tu notas. Si completás los 10 puntos, tenés tu primer software personalizado andando — en serio.

- ☐ **Elegí UNA planilla que duele.** El proceso manual que más horas te roba por semana.
- ☐ **Escribí el diagnóstico** (1 página): qué está roto, dónde viven los datos, quién decide qué.
- ☐ **Pedile a la IA el spec y el plan**, y no dejes que implemente hasta aprobarlos vos.
- ☐ **Elegí un stack aburrido:** framework web probado + base de datos en la nube con login incluido.
- ☐ **Construí el bloque 1** — solo ese — y cerralo en verde: funcionando, probado, verificado con datos reales.
- ☐ **Pedí lógica separada de conexiones** y pruebas automáticas desde el primer bloque.
- ☐ **Configurá el acceso "cerrado por defecto"** antes de invitar al primer usuario.
- ☐ **Automatizá con foto periódica** + una alerta al celular si los datos tienen más de un día.
- ☐ **Usá la IA solo para redactar;** toda decisión numérica va por fórmula transparente.
- ☐ **Abrió la bitácora del proyecto** y actualizala al final de cada sesión de trabajo.

→ LA REGLA DE LAS REGLAS

La IA multiplica lo que le des: si le das caos, te devuelve caos más rápido. Si le das método — diagnóstico, spec, bloques, verificación — te devuelve **software que funciona meses después de que te olvidaste de cómo lo hiciste**. ¿El paso 0 de la checklist? Rodearte de gente que ya está construyendo: skool.com/club-de-ia

¿Y AHORA QUÉ?

ESTO ES LO QUE CONSTRUIMOS **EN VIVO** EN EL CLUB DE I.A.

Cada semana nos juntamos a construir agentes, automatizaciones y software real para negocios reales de LATAM — paso a paso, sin asumir que sabés programar.

Si esta guía te sirvió, imaginate **verlo construirse en pantalla** y salir con el tuyo andando.

ENTRÁ AHORA → [SKOOL.COM/CLUB-DE-IA](https://skool.com/club-de-ia)

Sesiones en vivo semanales · comunidad LATAM · sin requisitos técnicos

También estamos en [@luciano.spinelli.mkt](https://twitter.com/luciano.spinelli.mkt)



Club de I.A.
La comunidad para
escalar con IA

www.skool.com/club-de-ia