

— INFRAESTRUCTURA Y ENTREGA · BUILDER LAB —

El Stack de los 2 Días

— *Builder Lab*

Cómo entrego soluciones de IA que se cobran sobre los mil dólares en uno o dos días de trabajo. No es una lista de herramientas: es la decisión que se toma antes de abrir el editor, y el agente que escribe todo lo demás.

INFRAESTRUCTURA CERRADA

INFRAESTRUCTURA ABIERTA

COOLIFY

CLAUDE CODE

1-2

DÍAS DE ENTREGA

+\$1.000

POR PROYECTO

0

LÍNEAS QUE ESCRIBO YO

10 min

LECTURA

00

LA PREGUNTA EQUIVOCADA

Cada vez que muestro algo que le entregué a un cliente, la primera pregunta es siempre la misma: *"¿qué herramienta usaste?"*

Es la pregunta equivocada, y es la que te tiene atrapado.

Lo que hace que un proyecto se entregue en dos días o se arrastre dos meses no es la herramienta. Es **haber elegido bien la infraestructura antes de empezar**. Cuando esa decisión está mal tomada, no hay velocidad que la salve: terminas peleando contra la plataforma, forzando integraciones que no existen y explicándole al cliente por qué algo que parecía simple lleva tres semanas.

Esta guía no es un tutorial. Es el criterio que uso para decidir, y las dos configuraciones a las que llego casi siempre.

ANTES DE SEGUIR

No vas a encontrar acá un paso a paso de instalación. Vas a encontrar el mapa: qué se decide, cuándo, y por qué esa decisión es la que define si cobras mil dólares o discutes el precio.

La velocidad no es talento. Es *no reinventar nada*.

Cuando alguien tarda dos meses en entregar algo que yo entrego en un fin de semana, casi nunca es porque escriba peor código. Es porque está construyendo desde cero piezas que ya existen y funcionan.

Yo no construyo el CRM. No construyo el inbox. No construyo la mensajería de WhatsApp ni el sistema de plantillas ni el panel donde el cliente ve sus conversaciones. Todo eso ya está resuelto por gente que lleva años en ello.

Lo único que construyo es la parte que nadie tiene: la lógica específica del negocio de ese cliente. El resto se ensambla.

Ese es el cambio mental completo. Dejas de ser alguien que construye software y pasas a ser alguien que ensambla una solución. Y el cliente, curiosamente, paga más por lo segundo, porque lo que compra es que su problema desaparezca rápido.

QUIÉN ESCRIBE EL CÓDIGO

Acá está la parte que a mucha gente le cuesta creer: **yo no escribo el código. Lo dirijo.**

Claude Code es el superagente que programa. Yo defino qué tiene que hacer el sistema, cómo se conecta, qué pasa en cada caso y dónde están los límites — y él lo construye. Mi trabajo es entender el problema del cliente y dar indicaciones precisas. El suyo es escribir.

Por eso la velocidad no depende de cuántos años lleve programando. Depende de qué tan claro tengo el problema antes de empezar a pedir. Un builder que sabe exactamente qué necesita el negocio del cliente le saca más a Claude Code que un programador que no entiende el negocio.

LO QUE ESTO SIGNIFICA PARA TI

No necesitas ser programador para entregar esto. Necesitas entender el problema, saber decidir la infraestructura, y aprender a dar instrucciones precisas. Esas tres cosas se aprenden mucho más rápido que un lenguaje de programación.

POR QUÉ N8N QUEDÓ EN EL PASADO

Durante un tiempo hice todo con n8n. Hoy no lo uso, y no es una moda: es que dejó de tener sentido para mí.

Con n8n estás limitado a lo que los nodos permiten. Cuando el cliente pide algo que ningún nodo hace — y siempre llega ese pedido — terminas metiendo un nodo de código igual, pero peor: dentro de una caja, sin control de versiones decente y difícil de depurar. El flujo crece, se llena de cajas y flechas, y a los tres meses nadie entiende qué hace, ni tú.

Con Claude Code escribo directamente el servicio que hace falta. Sin techo, sin nodos, sin pelear con el lienzo. Se versiona, se modifica en minutos y se despliega donde yo quiera.

QUE NO SE MALINTERPRETE

n8n no es malo, y si estás empezando te enseña a pensar en flujos. Pero es un piso, no un techo. El día que quieras entregar algo que el cliente pueda mantener y hacer crecer, la limitación te va a alcanzar — y es mejor que te alcance antes de haber cobrado.

LA TRAMPA DEL BUILDER

Construir desde cero se siente más profesional. Se siente como que estás haciendo el trabajo de verdad. Pero el cliente no ve la diferencia entre tu backend artesanal y una plataforma ensamblada — solo ve cuánto tardaste y si funciona.

02 LA BIFURCACIÓN

Todo se decide acá: *infraestructura cerrada o abierta*

Esta es la única decisión que importa de verdad, y hay que tomarla en la primera conversación con el cliente, no después de empezar.

CAMINO A

Infraestructura cerrada

Qué es: te apoyas en una plataforma de terceros que ya trae CRM, inbox, automatizaciones y mensajería.

Ventaja: velocidad máxima. El 80% ya está hecho el día uno.

Costo: el cliente queda atado a una mensualidad y a las reglas de esa plataforma.

Para quién: negocios que hoy no tienen nada y quieren todo resuelto.

CAMINO B

Infraestructura abierta

Qué es: piezas open source que tú despliegas y controlas por completo.

Ventaja: control total, sin techo de funcionalidades ni dependencia.

Costo: más trabajo de configuración y alguien tiene que mantenerlo.

Para quién: negocios con sistemas propios o que no quieren depender de terceros.

LAS CUATRO PREGUNTAS QUE DECIDEN POR TI

No lo decidas por gusto técnico. Decídelo con estas cuatro respuestas, que salen de la primera llamada:

1

¿QUÉ TIENE FUNCIONANDO HOY?

Si ya tiene un CRM, un ERP o un sistema propio del que no se va a mover, el camino cerrado te va a obligar a duplicar datos. Ahí vas a abierta.

¿ACEPTA PAGAR UNA MENSUALIDAD DE PLATAFORMA?

2

La cerrada siempre trae una suscripción encima de tu honorario. Si el cliente se resiste a ese costo recurrente, la conversación se rompe después, no antes. Pregúntalo el primer día.

¿QUIÉN LO VA A MANTENER DENTRO DE SEIS MESES?

3

Si nadie del lado del cliente puede tocar nada, la abierta te convierte en su soporte técnico permanente. Eso puede ser un negocio o una condena, según cómo lo cobres.

¿QUÉ TAN RARO ES LO QUE PIDE?

4

Si lo que necesita cabe en lo que la plataforma ya hace, cerrada. Si en la primera llamada aparecen tres cosas que la plataforma no soporta, abierta — porque si no, vas a pasar el proyecto peleando contra el producto.

03 CAMINO A — CERRADA

Acá la plataforma hace de columna vertebral: guarda los contactos, gestiona las conversaciones, dispara las automatizaciones y expone el panel que el cliente va a mirar todos los días. Yo solo enchufo la inteligencia.

01

GoHighLevel

La opción completa: CRM, embudos, calendarios, automatizaciones y multicanal en un solo lugar. Cuando el cliente necesita todo el aparato comercial, no solo mensajería.

02

YCloud

Proveedor oficial de la API de WhatsApp con capa de CRM, inbox compartido y campañas. Más simple y bastante más barato en la entrada. Cuando todo el negocio pasa por WhatsApp, sobra.

03

El agente, en Coolify

La lógica propia nunca vive dentro de la plataforma. Vive en mi servidor y se conecta por webhook. Así el cerebro es mío y portable.

LA REGLA QUE NO ROMPO

Aunque el camino sea cerrado, el agente jamás queda encerrado dentro de la plataforma. Si mañana el cliente quiere migrar de GoHighLevel a YCloud, o al revés, la parte inteligente sigue igual y solo cambio el conector. Eso convierte una migración de semanas en una tarde.

04 CAMINO B — ABIERTA

Cuando el cliente no puede o no quiere depender de una plataforma de terceros, la base cambia pero el principio es el mismo: ensamblar piezas maduras y construir solo lo propio.

01

Chatwoot

El inbox y la capa de atención.
Open source, multicanal, con panel para el equipo del cliente. Hace lo que hace la plataforma cerrada, pero corriendo en infraestructura propia.

02

El agente, en Coolify

Mismo lugar que en el camino cerrado. La lógica no cambia de casa según el proyecto: siempre vive donde yo la controlo.

03

Los datos, donde ya están

La ventaja real de este camino: te conectas al sistema que el cliente ya usa en vez de pedirle que migre. Nadie cambia de CRM para probar un agente.

Este camino tarda algo más en el arranque, pero se paga solo en el proyecto donde el cliente pide algo que ninguna plataforma cerrada soporta. Y ese pedido llega casi siempre.

05

LA CONSTANTE

Cambie lo que cambie por delante, *el backend siempre vive en Coolify*

Esta es la pieza que casi nadie enseña, y es la que más tiempo me ahorra.

Coolify es donde despliego todo lo que es mío: el agente, los servicios, las conexiones, las bases de datos. Un panel, un servidor, y desde ahí levanto lo que necesite sin depender de la infraestructura de nadie. Y lo que corre ahí lo escribió Claude Code siguiendo mis indicaciones.

Lo importante no es Coolify en sí. Es **tener un único lugar donde vive tu lógica, sea cual sea el proyecto**. El cliente de arriba cambia — a veces GoHighLevel, a veces YCloud, a veces Chatwoot — pero por debajo siempre es el mismo entorno, con las mismas convenciones y los mismos despliegues.

Esa constancia es la que hace posible el segundo día. El proyecto número diez no empieza de cero: empieza reutilizando la estructura del nueve.

LO QUE CASI NADIE MIRA

El mundo de la IA habla todo el tiempo de modelos y prompts, y casi nunca de dónde corre lo que construyes. Es justo la parte que separa una demo bonita de algo que un cliente puede usar en producción y pagar todos los meses.

06

POR QUÉ ESTO SE COBRA SOBRE LOS MIL DÓLARES

La objeción que todos tienen en la cabeza: "*si lo hago en dos días, ¿cómo cobro mil dólares?*"

Al revés. Lo cobras **porque** lo haces en dos días.

El cliente no está comprando tus horas. Está comprando que un problema que lleva meses costándole dinero desaparezca esta semana. Si le entregas eso el viernes, el precio es lo de menos. Si tardas dos meses, el precio empieza a parecerle caro aunque sea el mismo, porque durante esos dos meses siguió perdiendo.

CÓMO SE PRESENTA PARA QUE NO SE DISCUTA

- Nunca cotices por horas. Cotiza por el problema resuelto y por la fecha de entrega.
- No menciones las herramientas en la propuesta. Al cliente le da lo mismo, y nombrarlas invita a que las busque y calcule cuánto cuestan.
- Deja el alcance por escrito y con nombre y apellido. Lo que no está listado, no está incluido.
- Separa siempre tu honorario del costo de plataforma. Que quede claro que la mensualidad es de él, no tuya.

LA FRASE QUE CAMBIA LA CONVERSACIÓN

Deja de decir "te construyo un agente de IA". Empieza a decir "esto te está costando X al mes y lo puedo tener funcionando el viernes". La primera se compara con precios. La segunda se compara con la pérdida.

07

LO QUE ESTIRA UN PROYECTO DE DOS DÍAS

La velocidad no se pierde construyendo. Se pierde en decisiones que se toman mal al principio.

LO QUE MANTIENE LOS DOS DÍAS

- ✓ Elegir infraestructura en la primera llamada, no después de empezar.
- ✓ Un solo problema resuelto de punta a punta, no cinco a medias.
- ✓ Reutilizar la estructura del proyecto anterior.
- ✓ Entregar algo funcionando y mejorarlo después, con el cliente viéndolo.
- ✓ Dirigir a Claude Code con el problema ya entendido, no descubrirlo escribiendo.

LO QUE LO CONVIERTE EN DOS MESES

- ✗ Alcance abierto: "y ya que estamos, ¿podría también...?".
- ✗ Elegir la infraestructura equivocada y descubrirlo a mitad.
- ✗ Construir desde cero lo que la plataforma ya trae hecho.
- ✗ Perseguir la herramienta nueva de la semana en medio de un proyecto.
- ✗ Forzar un flujo de nodos a hacer algo para lo que no está pensado.

El último es el más caro y el menos evidente. Cambiar de herramienta a mitad de proyecto se siente como estar mejorando. No lo es: es empezar de nuevo con otro nombre.

Ya tienes el criterio: qué decidir, cuándo y por qué. Con esto dejas de coleccionar herramientas y empiezas a elegir las con una razón.

Lo que no está acá, porque no se puede escribir de forma genérica, es la parte de *tu* caso: cómo se ve el primer proyecto que deberías vender con tu nivel actual, qué precio ponerle, cómo se redacta el alcance para que no se te abra, y qué hacer cuando el cliente pide algo que la plataforma no soporta y ya firmaste.

Eso es lo que vamos a trabajar en Builder Lab. Y como ya estás en la lista, no tienes que hacer nada para entrar.

Ya estás dentro. *Ahora lo único que te pido*

Respóndeme por privado y cuéntame qué estás construyendo ahora mismo y dónde te atascaste. Estoy diseñando Builder Lab con lo que me contesten las primeras personas de esta lista, y quiero que resuelva tu caso concreto y no un caso promedio.

En este grupo escribo solo yo para que no se llene de ruido — pero mi privado está abierto y lo leo todo.

BUILDER LAB · APERTURA DE PLAZAS PRÓXIMAMENTE