

03

GUÍA PRÁCTICA
aGo lab
Versión 1.0

Build vs Buy: cómo decidir sin conflicto de interés

Framework de 5 preguntas, tabla de decisión por proceso, plantilla TCO a 3 años y cláusulas exigibles si compras SaaS.

AUTOR

Sixto Valdés

FECHA

Mayo 2026

WEB

ago.cl

§ EN 30 SEGUNDOS

Construir o comprar.

Casi cualquiera que responde tiene conflicto de interés. El vendor de SaaS dirá comprar. La consultora que construye dirá construir. El integrador dirá ambas. Esta guía es para que la decisión la tome tu empresa con criterio propio. Cinco preguntas concretas, tabla de TCO a tres años, decisión por tipo de proceso y los tres errores más caros que vemos en empresas medianas. Sin venta. Sin «depende». La regla de fondo es de Spolsky (2001): construye lo que te distingue, compra lo que no.

§ TRES DATOS

80 %

empresas medianas operan con stack híbrido (Gartner, 2023)

3 años

horizonte mínimo para comparar TCO build vs buy

15-25 %

del costo inicial: mantenimiento anual realista de lo construido

§ CÓMO USAR ESTA GUÍA

Cinco minutos: responde las cinco preguntas de la página 3 y mira la tabla TCO de la página 4.

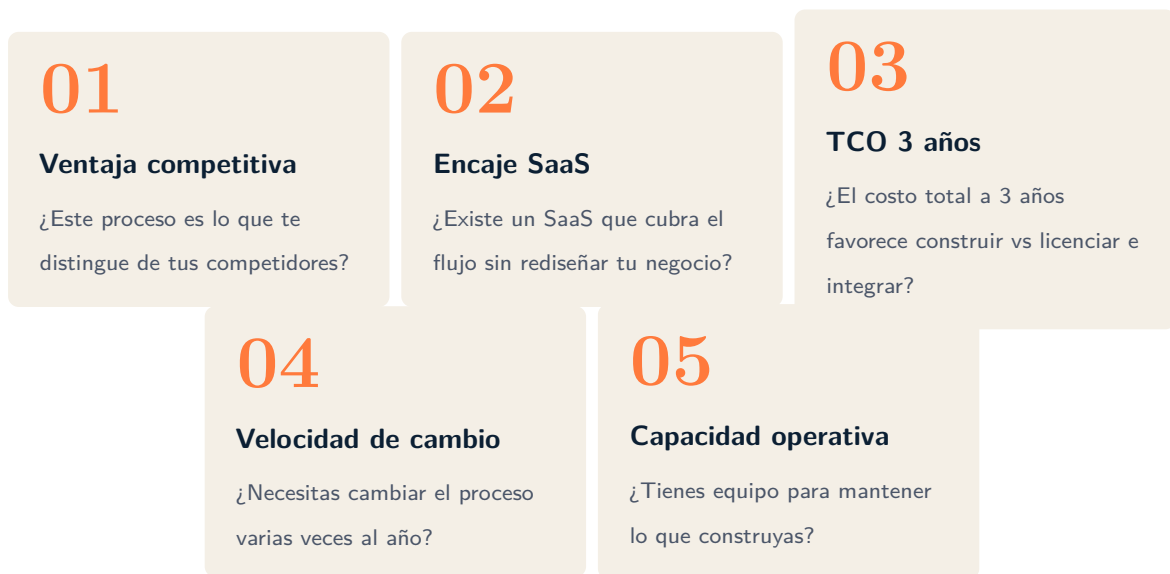
Treinta minutos: lee todo y llévalo a tu próxima reunión con el directorio o con tu equipo TI.

Para una conversación con vendor: imprime las páginas 5 y 6. Las cláusulas SaaS son lo que ningún vendor quiere ver al frente.

§ FRAMEWORK

Cinco preguntas.

Si respondes sí a tres o más, construir tiene sentido. Si respondes no a tres o más, comprar es lo correcto. Si quedas en el medio, considera híbrido planificado o custom-on-OSS.



“Construye lo que te distingue. Compra lo que no. La trampa es confundir lo segundo con lo primero.”

§ TCO 3 AÑOS

Construir vs. comprar.

Sistema de gestión interno para 20 usuarios, PyME mediana chilena, 2026. Rangos típicos, no universales. Pide cotizaciones reales con el mismo brief a 2-3 vendors SaaS y 2-3 proveedores de software a medida.

	CONSTRUIR A MEDIDA	COMPRAR SAAS ENTERPRISE
01	Año 1. 300-500 UF desarrollo + 50 UF infra y operación.	USD 12.000 licencias + USD 8.000 implementación.
02	Año 2. 60-100 UF mantenimiento más nuevas features según demanda.	USD 12.000-15.000 licencias (suben 10-15 % anual).
03	Año 3. 60-100 UF mantenimiento. Total acumulado ~600 UF.	USD 14.000-18.000 licencias. Total acumulado ~USD 50.000.
04	Propiedad. Código, datos y arquitectura son tuyos. Migrar es decisión interna.	Dependes del proveedor. Exportabilidad limitada. Vendor lock-in real.
05	Riesgo principal. Equipo se va o proveedor desaparece sin plan de transferencia.	Subida de precio, cambio de TOS, cierre de funcionalidades.

§ CRUCE TÍPICO

Construir cuesta más al inicio y menos después. SaaS cuesta menos al inicio y más después. El punto de cruce suele estar entre el año 2 y el año 4 según volumen de uso y proveedor.

A 5 años, construir gana en costo en la mayoría de procesos con uso intensivo.

SaaS gana en procesos con uso esporádico o bajo número de usuarios.

§ DECISIÓN

Qué corresponde según el proceso.

La misma empresa tiene procesos donde construir es correcto y procesos donde comprar es correcto. La trampa es aplicar la misma regla a todo.

- 01 Construye**
Motor central del negocio. Plataforma de matching, lógica de precio dinámico, operación específica del rubro, certificación con trazabilidad legal.
- 02 Compra**
Procesos commodity. Email, CRM básico, contabilidad, facturación SII, helpdesk, gestión documental. Time-to-value importa más que diferenciación.
- 03 Híbrido**
El caso común: 80 % de empresas medianas opera así. Compras commodity, construyes lo central. Lo que falta es planificarlo, no acumularlo.
- 04 Custom-on-OSS**
Punto medio. Construyes sobre open source (Django, Saleor, PocketBase, n8n). Reduces costo inicial, mantienes propiedad y portabilidad.
- 05 SaaS con escape plan**
Compras SaaS pero exiges desde día uno: DPA, exportabilidad documentada, plan de salida con fecha. Si el proveedor se resiste, bandera roja.

§ PRUEBA DE HONESTIDAD

Pregunta 1: si describes el proceso a un competidor, ¿lo identifica como único tuyo?

Pregunta 2: ¿el proceso cambia con frecuencia o es estable?

Pregunta 3: si renuncia el dev principal mañana, ¿qué pasa? (gobernanza antes de build).

§ ERRORES CAROS

Tres errores y siete cláusulas.

§ TRES ERRORES MÁS CAROS

01 Construir lo commodity. Sistema de facturación propio: dos años para descubrir que el SII cambió las reglas.

02 Comprar lo central. Plataforma estándar para metodología propia: la diferenciación se diluye.

03 Construir sin mantenimiento. A los 18 meses, deuda técnica fase 3 y dependencia de una persona.

§ SI COMPRAS SaaS, EXIGE

- a DPA firmado con jurisdicción definida.
- b Exportabilidad completa de datos, sin costo extra al salir.
- c Plazo de notificación de brechas (72 h ideal).
- d Acceso a logs de auditoría.
- e Compromiso de continuidad (qué pasa si cierran).
- f Cláusula de revisión de precio.
- g Plan de salida con fecha máxima garantizada.

§ CONVERSACIÓN TIPO CON UN VENDOR DE SAAS

Tres preguntas que delatan si el proveedor está preparado para una relación seria:

1. «¿Cuál es el plazo máximo en que me notifican de una brecha?» Si la respuesta es «lo conversamos», no firmes.
2. «¿Cómo recibo mis datos si decido salir en 3 años?» Si no hay procedimiento documentado con formato y plazo, no firmes.
3. «¿Qué pasa si suben el precio 30% en la próxima renovación?» Si no hay cláusula que lo limite, asume que pasará.

§ PARA PROFUNDIZAR

Lecturas y casos.

§ LECTURA RECOMENDADA

Joel Spolsky, In Defense of Not-Invented-Here
Marc Andreessen, Why Software Is Eating the World
MIT CISR, What Makes Companies Successful at IT

§ CASOS aGo PÚBLICOS

Bioaudita: construir es correcto (trazabilidad legal específica)
TCultura: híbrido planificado (Flow, Transbank, OSS más build core)
Hub Nubiq: integra Stripe para billing, construye lo agéntico

§ VEINTE MINUTOS SIN COMPROMISO

¿Quieres una segunda opinión build vs buy, sin que te vendamos lo que no necesitas?

Trabajamos los dos lados. Si lo correcto es comprar, te lo decimos. Si es construir, también.

La honestidad sale del contexto: vivimos los dos modelos.

hola@ago.cl • +56 9 7744 7331 • ago.cl

§ BIBLIOGRAFÍA

Spolsky, J. (2001). In Defense of Not-Invented-Here Syndrome. <https://www.joelonsoftware.com/2001/10/14/in-defense-of-not-invented-here-syndrome/>

§ aGo Lab

Sistemas que cumplen desde la arquitectura.

Construimos software a medida con cumplimiento Ley 21.719, GDPR y ARCO+ integrado desde el modelo de datos, no como capa agregada después. Operamos desde Chillán, Concepción y Santiago para clientes en Chile, Latinoamérica y proyectos internacionales.

WEB

ago.cl

CORREO

hola@ago.cl

WHATSAPP

+56 9 7744 7331

LINKEDIN

linkedin.com/company/ago-lab

INSTAGRAM

[@agolab.cc](https://instagram.com/agolab.cc)

GITHUB

github.com/sixtovaldese

§ VEINTE MINUTOS, SIN COMPROMISO

Si tu equipo está evaluando construir o reformar un sistema, conversemos.

Lleva este documento a tu próxima reunión de TI o legal y nos contactas si quieres profundizar.