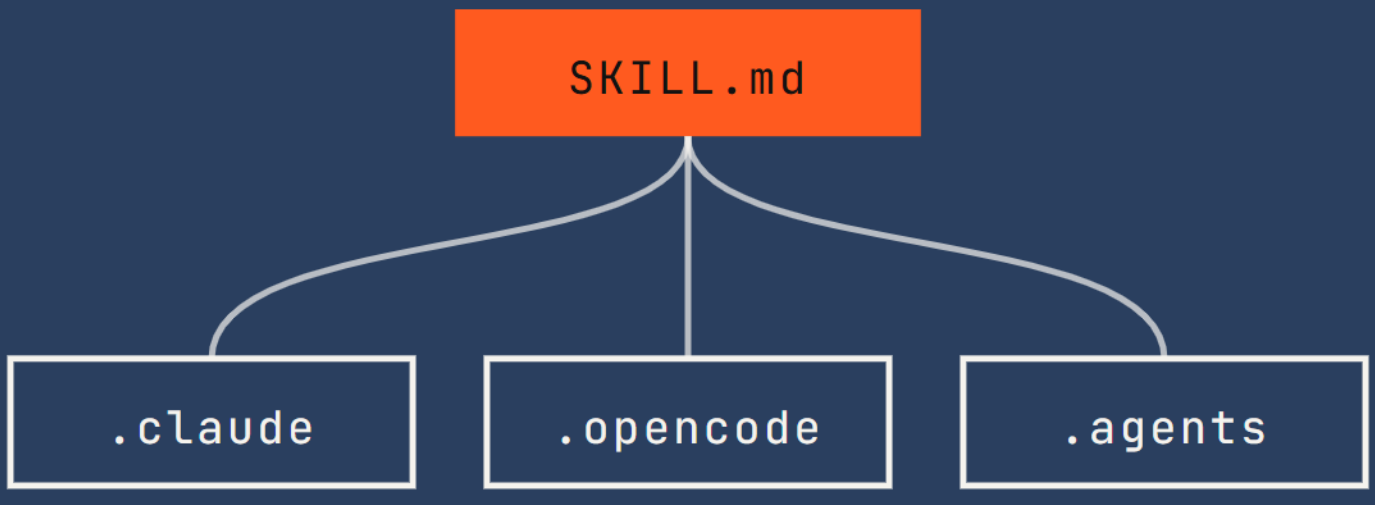
Tu stack de IA, sin suscripciones innecesarias



Tu stack de IA, sin suscripciones innecesarias

Skills, herramientas y automatizaciones elegidas mediante pruebas, no modas

Edición: manuscrito de trabajo, 2026-11-04. **Audiencia:** autónomos y equipos pequeños que ya utilizan varias aplicaciones de IA y necesitan un sistema pequeño, mantenible y reversible. **Estado de las prácticas:** prácticas ejecutadas en laboratorio el 04/11/2026 (calibración de herramientas del 03/11/2026), con Claude Code 2.1.317, OpenCode 1.19.5, Codex CLI 0.161.0, Kimi CLI 2.4.1 y Ollama 0.40.14 en Windows 11. Quedan por ejecutar la receta de correo a borrador (capítulo 07) y las mediciones propias de multimedia distintas del OCR (capítulo 06). Las condiciones comerciales de terceros se verifican el día de cada decisión.

Objetivo

Aprenderás a inventariar trabajo, distinguir componentes, escribir skills que carguen en varias herramientas, comparar alternativas con la misma batería, cerrar permisos y ensayar la salida. El objetivo no es reducir herramientas por principio, sino eliminar duplicidades que no aportan un resultado comprobado.

Reglas

- No damos importes ni recomendamos planes: las condiciones se consultan en la fuente oficial el día de decidir. Sí decimos el tipo de acceso que usamos en el laboratorio (suscripción, cuenta gratuita, API, local).
- Los ejemplos usan personas, empresas y datos ficticios.
- Todas las cifras proceden del laboratorio del libro; lo no medido se indica como «por medir».
- MCP se trata como un protocolo de conexión, no como un sistema de permisos.
- Toda integración se prueba primero sin efectos externos.

Contenido

Tu stack de IA, sin suscripciones innecesarias . . . 2
Skills, herramientas y automatizaciones elegidas mediante pruebas, no modas . . . 2
Objetivo . . . 2
Reglas . . . 2
01 · Mapa del trabajo antes del stack 6
Paso 1 · Inventaria tareas, no marcas . . . 6
Paso 2 · Apunta qué tienes de verdad . . . 6
Lo probamos · la combinación del laboratorio . . . 7
Paso 3 · Dibuja el mapa de dependencias . . . 7
Paso 4 · Clasifica cada pieza . . . 8
Paso 5 · Busca duplicidades como hipótesis . . . 8
Paso 6 · Elige la primera prueba . . . 8
Regla de simplificación . . . 8
Práctica . . . 9
Resumen . . . 9
02 · Prompt, instrucciones, skill, asistente, herramienta, agente y MCP 10
Prompt . . . 10
Instrucciones persistentes . . . 10
Skill . . . 10
Asistente . . . 10
Herramienta . . . 11
Agente . . . 11
MCP . . . 11
Lo probamos · un servidor MCP de unas 60 líneas . . . 11
Tabla de decisión . . . 13
Por qué importa el vocabulario . . . 13
Práctica . . . 13
Resumen . . . 13
03 · Diseño y versionado de skills 15
Lo probamos · ¿una skill funciona igual en todas las herramientas? . . . 15
La cabecera YAML, siempre . . . 16
Receta · una skill, tres carpetas . . . 16
Diseña el contrato . . . 17
Versiona y prueba contra respuestas guardadas . . . 17
Señales de que debes dividirla . . . 18
Práctica . . . 18

Resumen . . . 18

04 · Evaluación de alternativas **20**

Paso 1 · Ficha de encargo . . . 20
Paso 2 · Corrige con reglas objetivas . . . 20
Paso 3 · Mismo caso, mismas reglas, varias repeticiones . . . 20
Lo probamos · la calibración del 03/11/2026 . . . 21
Paso 4 · Cuatro bloques de evidencia . . . 21
Paso 5 · Matriz con estados, no con notas . . . 22
Paso 6 · Integración mínima . . . 22
Paso 7 · Decide con fecha de revisión . . . 22
Práctica . . . 23
Resumen . . . 23

05 · Investigación y documentos **24**

Dos flujos, dos controles . . . 24
Lo probamos · ¿existen las citas? . . . 24
La skill de investigación con fuentes . . . 25
Comprueba las citas tú mismo en Windows . . . 26
Documentos internos . . . 26
Selección de herramientas para este trabajo . . . 26
Práctica . . . 27
Resumen . . . 27

06 · Datos y multimedia **28**

Divide el trabajo: modelo frente a código . . . 28
Lo probamos · facturas en imagen . . . 28
Contrato de una skill de datos . . . 29
Gráficos y cifras . . . 29
Imagen y diseño . . . 29
Audio y vídeo . . . 29
Revisión antes de publicar . . . 29
Cómo elegir herramienta para este trabajo . . . 30
Práctica . . . 30
Resumen . . . 30

07 · Tres recetas agnósticas **31**

Receta 1 · Una skill, cuatro herramientas . . . 31
Receta 2 · Script con cualquier modelo: seguimiento.py . . . 32
Receta 3 · Correo a borrador (por ejecutar) . . . 33
Registro común de las tres recetas . . . 33
Cómo sustituir una pieza sin romper nada . . . 33
Práctica . . . 34

Resumen . . . 34

08 · Permisos y secretos 35

Lo probamos · ¿obedece un agente una orden escondida en un documento? . . . 35

Cómo cerramos el permiso en cada herramienta . . . 36

Lo que salió mal · la fuga por la lista negra . . . 36

Lo que salió mal · «connected» no era verdad . . . 37

Matriz de permisos . . . 37

Secretos: en variables de entorno, nunca en el código . . . 37

Separa laboratorio y operación . . . 37

Práctica . . . 38

Resumen . . . 38

09 · Registros, reintentos y aprobación 39

Lo probamos · el registro de una ejecución completa . . . 39

1 · El identificador: misma entrada, mismo trabajo . . . 39

2 · Reintentos con límite y espera creciente . . . 39

3 · Validación por código, no por el modelo . . . 40

4 · Lo que el modelo no hizo con el acta trampa . . . 41

5 · Aprobación específica . . . 41

6 · Reanudar tras una interrupción . . . 42

Alertas útiles . . . 42

Práctica . . . 42

Resumen . . . 42

10 · Auditoría de suscripciones y salida 44

Paso 1 · Reabre el inventario . . . 44

Lo probamos · el saldo que el panel enseña no es el disponible . . . 44

Paso 2 · Señales que abren una revisión . . . 45

Paso 3 · Clasifica con lo que has medido . . . 45

Paso 4 · Prueba de sustitución . . . 45

Paso 5 · Exporta y reabre fuera . . . 45

Paso 6 · Retirada . . . 46

Paso 7 · Changelog y próxima revisión . . . 46

El stack mínimo viable . . . 46

Práctica . . . 46

Resumen . . . 47

01 · Mapa del trabajo antes del stack

Un stack útil no empieza con una lista de aplicaciones. Empieza con las tareas que necesitas resolver y con la prueba que vas a exigir a cada una. Si inventarías primero las herramientas, acabarás justificando lo que ya pagas.

Este capítulo te deja un inventario de tareas, un mapa de dependencias y una primera hipótesis de sustitución. Nada más. Las decisiones de mantener o retirar llegan en los capítulos 04 y 10, cuando tengas datos.

Paso 1 · Inventaria tareas, no marcas

Elige una semana de trabajo normal y apunta cada vez que una IA interviene. Registra la transformación (qué entra y qué sale), no el nombre del programa.

Tarea	Entrada	Salida	Veces por semana	Cómo se revisa	Herramienta actual
Seguimiento de reunión	Acta con turnos	Acciones, desacuerdos, excepciones	Por medir	Contra el acta	Por registrar
Responder solicitud	Correo + ficha de servicio	Borrador de respuesta	Por medir	Hechos y compromisos	Por registrar
Comprobar un dato actual	Pregunta	Respuesta con fuente y cita	Por medir	Abrir la fuente	Por registrar
Leer facturas escaneadas	Imagen	Campos de la factura	Por medir	Sumas y registro de proveedor	Por registrar

Reglas del inventario:

1. La frecuencia se cuenta, no se estima. Si no la has contado, escribe «por medir».
2. Cada fila tiene una persona que usa la salida. Si nadie la usa, la tarea es candidata a desaparecer antes que la herramienta.
3. Apunta cómo se revisa hoy. Si la respuesta es «no se revisa», ya has encontrado un riesgo.
4. Usa la plantilla `plantillas/inventario-stack.csv` del paquete para no empezar de cero.

Paso 2 · Apunta qué tienes de verdad

Para cada herramienta, anota cuatro datos que puedas comprobar hoy mismo: versión instalada, cuenta con la que la usas, tipo de acceso (suscripción, cuenta gratuita, API, local) y qué tareas del paso 1 cubre. Las condiciones comerciales se consultan en la fuente oficial el día de la decisión; no las copies de memoria ni de una comparativa.

Para sacar las versiones en Windows, abre PowerShell y pide la versión a cada herramienta de línea de comandos que tengas instalada. La mayoría responde a `--version`:

```
claude --version
opencode --version
codex --version
kimi --version
ollama --version
```

Si alguna no responde, no está en el `PATH` o no la tienes: anótalo igual. Un inventario con huecos visibles es mejor que uno completo de memoria.

Lo probamos · la combinación del laboratorio

Para escribir este libro montamos un laboratorio el 04/11/2026 (la calibración de herramientas es del 03/11/2026). Te lo enseñamos como ejemplo de inventario real, no como recomendación de compra. No damos importes ni recomendamos planes: cambian y cada lector debe consultar las condiciones vigentes. Sí te decimos qué tipo de acceso usamos para cada herramienta.

Equipo: Windows 11 Pro, Intel Core Ultra 5 245K, 31,3 GB de RAM, NVIDIA GeForce RTX 5060 Ti con 16 GB de VRAM y disco NVMe de 1 TB. Es un equipo de gama media-alta; los tiempos no se trasladan a un portátil sin gráfica dedicada.

Herramientas y versiones:

Herramienta	Versión	Tipo de acceso en el laboratorio
Claude Code	2.1.317	Suscripción (modelos Claude Haiku 5.5 y Claude Sonnet 5.5)
Codex CLI	0.161.0	Cuenta gratuita de ChatGPT, modelo por defecto de esa cuenta
OpenCode	1.19.5	Modelos por API compatible con OpenAI (Experiential Labs), sin coste en esa fecha
Kimi CLI	2.4.1	Su modelo por defecto
Ollama	0.40.14	Local: qwen3.5:4b, qwen3.5:9b, gemma4:12b, qwen3.6:35b, embeddinggemma:300m

Por API, a través de Experiential Labs, usamos GPT-6 luna, DeepSeek V4.1 Flash y DeepSeek V4.1 Flash 1017, que en esa fecha se ofrecían sin coste. En la calibración del 03/11/2026 (capítulo 04) participaron además Claude Opus y GPT-6 sol; este último, por una API de pago con saldo, que es la que protagoniza el caso del saldo engañoso del capítulo 10.

Lo útil de esta tabla no son las marcas, sino la mezcla: una suscripción, una cuenta gratuita, modelos gratuitos por API en esa fecha y modelos locales. Cada tipo de acceso tiene un riesgo distinto. Lo gratuito puede desaparecer o cambiar de condiciones; la API depende de un saldo (capítulo 10 tiene el caso real de un saldo que engañaba); lo local depende de tu hardware.

Paso 3 · Dibuja el mapa de dependencias

Escribe cada flujo en una línea, de la entrada a la decisión humana:

```
Acta → Claude Code + skill extraer-acciones-acta → tabla → revisión → correo manual
Acta → seguimiento.py + modelo por API → bandeja → aprobación → envío manual
Pregunta → Kimi CLI con web → respuesta con URL y cita → comprobación de la cita
Factura escaneada → modelo con visión → campos → comprobación de sumas por código
```

Marca en cada línea:

- **Dónde se bloquea si una pieza desaparece.** Si el flujo solo funciona con una herramienta, es una dependencia.
- **Dónde copias la misma información a mano.** Cada copia es un sitio donde se pierde una versión.
- **Dónde hay una comprobación que no depende del modelo** (código, persona, fuente abierta). Si no hay ninguna, el flujo descansa entero en la confianza.

Paso 4 · Clasifica cada pieza

Asigna a cada herramienta una de estas etiquetas:

1. **Prescindible:** puedes retirarla sin efecto.
2. **Sustituible:** existe otro procedimiento y lo has probado.
3. **Dependencia:** no hay salida ensayada; necesita plan.
4. **Sin uso:** nadie la usa en la semana registrada. Confírmalo con quien la tenga antes de tocarla.

«Sustituible» exige haberlo probado. Que otra aplicación «también tenga IA» no cuenta. En el capítulo 03 verás por qué: la misma skill, en el mismo proyecto, no cargaba como skill en OpenCode ni en Codex por un detalle de formato.

Paso 5 · Busca duplicidades como hipótesis

Dos herramientas son redundantes para una tarea cuando comparten entrada, salida y nivel de revisión, y la batería del capítulo 04 lo confirma. Hasta entonces, escríbelo como hipótesis:

```
Hipótesis: para «seguimiento de reunión», la herramienta A y la herramienta B
podrían ser intercambiables.
Prueba que lo decidiría: misma acta, mismo encargo, mismos criterios, tres repeticiones.
Qué haría que no lo fueran: una no carga la skill, una no exporta la tabla,
una necesita permisos que la otra no.
```

Antes de sospechar de la herramienta, sospecha del encargo. En la batería del laboratorio, la mayor mejora medida en la tarea del acta vino de reescribir el encargo (pasar de un prompt ingenuo a uno con reglas y esquema), no de cambiar de modelo: los cinco modelos de nube pasaron todos los criterios con el encargo bueno. Si tu inventario dice «la herramienta A lo hace mal», comprueba primero qué le estás pidiendo.

Paso 6 · Elige la primera prueba

Escoge una sola tarea para la primera prueba de sustitución. Criterios para elegirla:

- Tiene entrada y salida claras (un acta y una tabla, no «ayudarme con el proyecto»).
- La puedes corregir con reglas objetivas, mejor si es con código.
- No tiene efectos externos: no envía, no borra, no paga.
- La haces con frecuencia suficiente para que el resultado importe.

El seguimiento de reunión cumple las cuatro. Por eso es el ejemplo que recorre todo el libro, y por eso el paquete incluye la skill `extraer-acciones-acta`, el acta ficticia y la receta `seguimiento.py`.

Regla de simplificación

Elimina pasos antes de eliminar herramientas. Si un documento pasa por tres aplicaciones para acabar en la misma tabla, quizá no necesitas sustituir ninguna: basta con quitar dos copias. Un stack pequeño es una consecuencia de un proceso claro, no un objetivo estético.

Checklist del mapa

- [ ] Cada tarea tiene entrada, salida y persona usuaria.
- [ ] Las frecuencias no contadas dicen «por medir».
- [ ] Cada herramienta tiene versión, cuenta y tipo de acceso anotados.
- [ ] Ninguna celda incluye precios copiados de memoria.
- [ ] Hay al menos una dependencia sin salida ensayada, señalada.

- [ ] Las duplicidades están escritas como hipótesis con su prueba.
- [ ] Hay una tarea elegida para la primera prueba, sin efectos externos.

Práctica

1. Abre `plantillas/inventario-stack.csv` y registra cinco tareas reales de esta semana.
2. Ejecuta en PowerShell `--version` para cada herramienta de línea de comandos que uses y apunta el resultado.
3. Dibuja el mapa de dependencias en texto, una línea por flujo, y marca dónde se bloquearía cada uno.
4. Escribe una hipótesis de duplicidad con la prueba que la decidiría.
5. Elige la tarea de la primera prueba. Guarda el inventario con la fecha en el nombre del archivo: lo volverás a abrir en el capítulo 10.

Resumen

- Inventaría tareas (entrada, salida, revisión), no logotipos.
- Lo no contado se escribe «por medir»; los precios se consultan el día de decidir.
- El laboratorio combinó suscripción, cuenta gratuita, API gratuita en esa fecha y modelos locales: cada tipo de acceso tiene su propio riesgo.
- Antes de culpar a una herramienta, revisa el encargo: fue la mejora más grande que medimos.
- Empieza por una tarea sin efectos externos y corregible con reglas.