

La diferencia entre gastar en inteligencia artificial y ganar con ella

Cómo una pequeña empresa convierte un chat en un sistema de trabajo: archivos, método y criterio, sin escribir una línea de código.

Por [Luis Martín](#)

Publicado agosto de 2026

Última revisión agosto de 2026

35 min de lectura



Ilustración – La adopción sube en la pared; la factura cae al suelo y nadie la mira.

Antes de empezar

El problema de la inteligencia artificial en la empresa ya no es de tecnología. Es de adopción.

Mientras escribo esto, el mercado está inundado: cientos de plataformas, modelos especializados en cada sector imaginable y una tormenta de actualizaciones que se multiplica con cada plataforma nueva. Todas prometen lo mismo —agilidad, productividad y, por lo tanto, menos costes— y todas juntas producen en quien tiene que decidir el efecto contrario: ruido.

Con los datos delante, lo prometido no está ocurriendo. Al menos no al ritmo con que se anuncia.

Los informes coinciden en que la tasa de adopción es altísima. Pero la adopción no es la estadística que cuenta. La que cuenta es cuánta de esa adopción se convierte en rentabilidad, y ahí la caída es de vértigo. La distancia entre una cifra y la otra dice dos cosas a la vez, y ninguna es agradable: que la mayoría de las empresas no estaba preparada para meter la inteligencia artificial en sus operaciones, y que la mayoría no ha notado ninguna mejora apreciable —ni en tiempo ni en dinero— después de haberla metido.

Eso es gasto estéril. Y hay un escalón más abajo, que es el gasto que te arruina: una empresa abrió una suscripción de inteligencia artificial a sus miles de empleados y acabó recibiendo una factura de quinientos millones de dólares. [Lo conté aquí. \(https://senalsintetica.substack.com/p/subirse-al-carro-ia-puede-salir-carisimo\)](https://senalsintetica.substack.com/p/subirse-al-carro-ia-puede-salir-carisimo)

Si a una compañía de ese tamaño una factura así le descoloca la cuenta de resultados, en un negocio de seis personas un error de esa familia no descoloca: mata.

Por eso escribo esto. Y por eso lo escribo para la pequeña empresa y el profesional independiente, que es quien menos margen tiene para equivocarse y quien menos ayuda encuentra. Lo que viene es sencillo y es básico. También es, con algo de suerte, la diferencia entre optimizar tu negocio y tirar el dinero exactamente igual que lo tiró alguien con mil veces tu presupuesto.

Esto no va de trucos



Un martes de marzo, el dueño de una carpintería de seis empleados abre el chat de inteligencia artificial que paga desde hace cuatro meses y escribe: «Hazme un presupuesto para unos cabeceros de hotel».

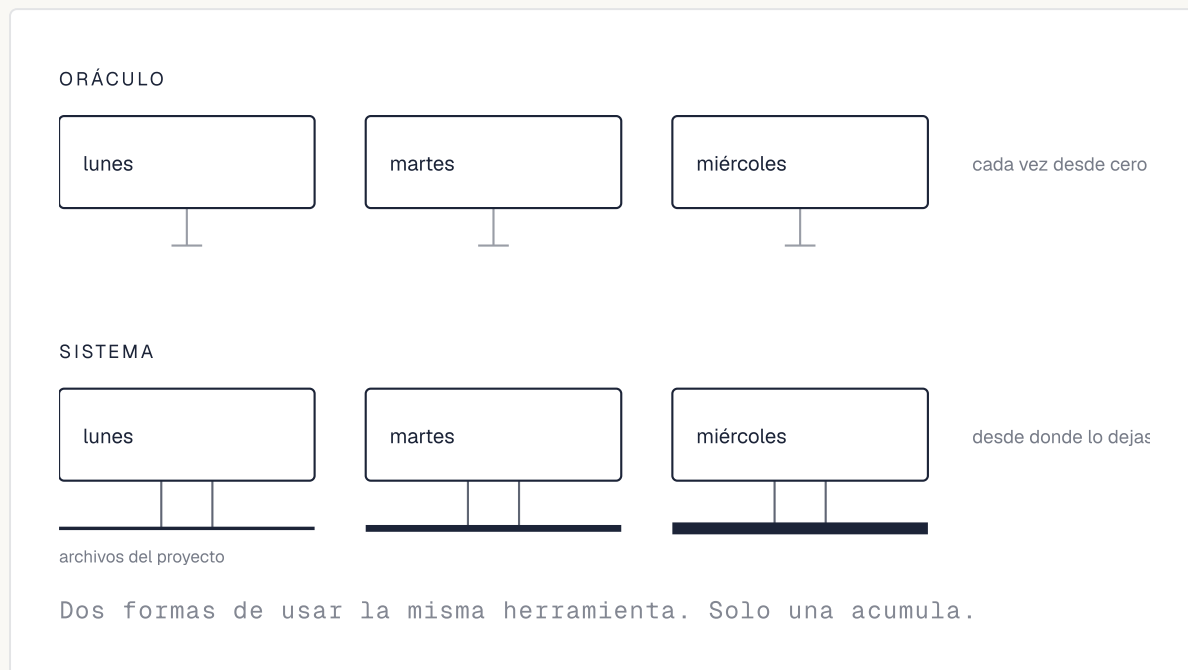
Lo que recibe es una plantilla. Correcta, ordenada, absolutamente inútil: precios inventados, plazos inventados, un tono de agencia de Madrid que no se parece en nada a cómo habla él con sus clientes. Corrige. Vuelve a pedir. Explica que la madera es roble, que el hotel son catorce habitaciones, que el IVA va incluido porque una vez tuvo una discusión fea por no incluirlo. Once mensajes después tiene algo aprovechable. Ha invertido hora y media y el presupuesto sale un día tarde.

Al día siguiente abre un chat nuevo y la máquina no recuerda nada. Ni el roble, ni las catorce habitaciones, ni el IVA, ni la discusión fea.

Ese hombre no tiene un problema de herramienta. Tiene un problema de arquitectura. Está usando la inteligencia artificial como **oráculo**: se pregunta, se copia, se cierra. Y el oráculo, por definición, no te conoce, no te acompaña y no mejora contigo. Cada consulta empieza en cero, y empezar en cero cuesta dinero cada vez.

La alternativa no es cambiar de plataforma ni pagar más. Es montar un **sistema de trabajo**: un puñado de archivos de texto que el agente lee antes de ponerse a trabajar y actualiza cuando termina. Nada de código. Nada de programadores. Texto.

Esa es la tesis entera de esta guía, y cabe en una frase que podrás repetirle a quien te pregunte: *un oráculo responde preguntas; un sistema recuerda tu negocio.*



Al terminar sabrás montar, con cuatro archivos de texto y sin escribir una línea de código, un asistente al que no tengas que explicarle tu empresa cada mañana — y sabrás por qué eso, y no un catálogo de trucos de conversación, es lo que separa gastar en inteligencia artificial de ganar con ella.

Hay una pregunta que te vas a hacer hacia el minuto tres de lectura, cuando veas los archivos. La contesto ya, porque si la dejo para el final habrás cerrado la página: **¿y quién mantiene todo esto?**

Lo mantiene el agente. Se lo pides una vez, queda escrito, y al cerrar cada trabajo él te propone qué añadir, qué corregir y qué borrar. Tú lees la propuesta y dices sí o no.

El dueño aprueba. El dueño no redacta. Si esto se convierte en una tarea más de tu lunes, morirá en quince días como mueren todos los sistemas que dependen de la disciplina de una persona ocupada. Volveré sobre ello con detalle, pero convenía decirlo antes de que empieces a contar archivos.

Un aviso sobre el ejemplo que va a atravesar toda la guía. He elegido una carpintería a propósito: seis empleados, virutas en el suelo, una furgoneta y un taller en un polígono de Zaragoza. No un estudio de diseño, no una consultora, no una tienda *online*. Si esto funciona para un negocio que fabrica muebles con las manos, funciona para el tuyo. Lo único que cambia son los nombres.

01 El restaurante completo: el dueño, el chef y el pinche



Quien monta un restaurante tiene que entender de todo: de sala, de compras, de nóminas, de licencias, de márgenes. No para hacerlo todo, sino para saber qué está pasando. Y llega un momento en que no le queda más remedio que contratar talento especializado. Contrata a un chef.

El agente de inteligencia artificial es el chef. Sabe cocinar mucho mejor que tú y ejecuta a una velocidad que no puedes igualar. Pero no decide la carta, no fija los precios y no elige a los proveedores. Eso es del dueño. **El criterio no se delega**, ni al chef de carne y hueso ni al de silicio.

Hasta aquí, el símil es el de siempre. Le falta la tercera figura, que es la que nadie cuenta y la que de verdad describe tu semana.

Cuando trabajas con un agente, no solo contratas al chef. **A veces haces de pinche**. El agente te devuelve un resultado que hay que descargar, renombrar, pegar en un sitio, comprobar en otro. Te pide una decisión que no habías previsto. Te obliga a entrar en la cocina y a hacer tareas que no habías hecho nunca: abrir un archivo de texto, entender qué es una carpeta compartida, revisar una fórmula, decidir qué formato de factura quieres. Cosas que antes no existían en tu día.

Aquí está el punto que casi todo el mundo se traga mal, así que lo digo con la incomodidad intacta: **esas interacciones de más no son un coste evitable. Son el trabajo.**

No estás perdiendo el tiempo cuando le explicas al agente por tercera vez cómo funciona tu negocio. Estás construyendo. La prueba es que, si lo haces bien, la cuarta vez ya no hace falta —y ese es exactamente el objeto de esta guía—. Pero las tres primeras hay que pagarlas, y quien espera que la máquina adivine su negocio en el primer intento abandonará antes de llegar a la cuarta.

Hay además un beneficio que no aparece en ninguna factura. Haciendo de pinche aprendes cosas de tu propio negocio que no sabías. El dueño de la carpintería descubrió, obligado a explicarle al agente cómo calculaba un presupuesto, que él mismo no tenía un método: tenía una costumbre, distinta cada semana, que dependía de si el cliente le caía bien. Nadie se lo había preguntado nunca con esa insistencia. La máquina se lo preguntó porque la máquina no puede suponer.

***Traducción de cinco años:** el chef cocina; tú decides qué se sirve; y de vez en cuando bajas a pelar patatas porque hace falta. Los tres puestos son del restaurante. Ninguno sobra.*

02 Dos plataformas, no cinco



La primera decisión práctica es de cartera, y es más sencilla de lo que parece.

Si usas esto a diario, paga. La versión gratuita de cualquier plataforma sería está pensada para que la pruebes, no para que trabajes: modelos menos capaces, límites que saltan a

media tarde y, sobre todo, sin los contenedores donde viven los archivos de los que trata esta guía. Compara el precio con lo que ya pagas por la gestoría, el seguro del taller, el programa de facturación o el mantenimiento de la web. Es la más barata de todas esas facturas y es la única que trabaja contigo todos los días.

Con dos plataformas basta. No cinco. No una por moda. Dos, y complementarias: una que sea tu taller principal —donde vivan los proyectos, los archivos y el trabajo largo— y otra que uses de otro modo. Añadir una tercera y una cuarta multiplica el gasto y divide la atención; acabas con cuatro suscripciones y ningún sistema.

Pero el argumento de fondo para la segunda no es la variedad. Es la **segunda opinión**.

Dos modelos no son redundancia: son auditoría cruzada. Cada uno tiene puntos ciegos distintos, y aquí hay una asimetría que conviene entender bien porque gobierna todo lo demás: **un modelo que audita su propio trabajo repite sus errores con más confianza; el que audita el trabajo ajeno los encuentra**. No es un defecto de la tecnología, es cómo funciona cualquier revisión, también entre humanos. Por eso ante un diagnóstico serio se pide una segunda opinión médica, y no se le pide al mismo médico que se relea.

De ahí sale la regla operativa del tutorial, que es corta y no admite matices: **quien construye no verifica**. Lo que va a tener consecuencias —un presupuesto, un contrato, un cálculo, una web— lo genera un modelo y lo revisa el otro.

Antes de gastar un euro más: la vía gratuita

Ahora bien, esto lo publica una agencia que vende servicios de inteligencia artificial, y el aviso legal del final no cubre lo que tú estás pensando. Así que te doy primero el camino que no me da nada.

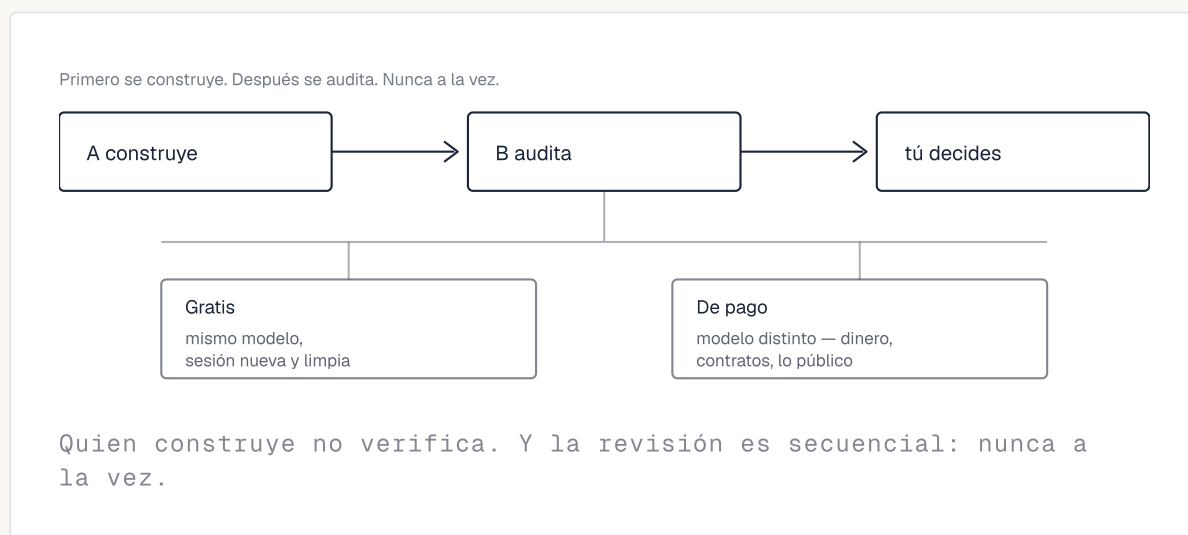
La segunda opinión no requiere una segunda suscripción. Abre una sesión nueva y limpia en la misma plataforma que ya pagas, sin el contexto de la conversación anterior, pega el trabajo terminado y pídele una revisión explícitamente adversarial:

Actúa como revisor externo. No has escrito este documento y no tienes que defenderlo. Busca errores, omisiones, cifras que no cuadren, promesas que no podamos cumplir y riesgos legales o comerciales. No lo reescribas: repórtalos en una lista, del más grave al más leve, y di en cada uno por qué es un problema.

Sin el contexto de la conversación en la que se construyó, el modelo no arrastra las suposiciones que lo llevaron al error. No es tan bueno como un modelo distinto —comparte los mismos sesgos de fábrica—, pero captura una parte grande del beneficio a coste cero, y es más de lo que hace hoy la inmensa mayoría de quienes usan estas herramientas.

Dicho eso, la verdad sin adornos, que además es la que menos me conviene: **una suscripción usada con disciplina vale más que dos usadas sin ella**. Si no has montado los archivos, si no revisas nada, si aceptas el primer borrador que sale, la segunda plataforma solo duplica el gasto de no tener método.

¿Cuándo se justifica de verdad? Cuando el trabajo tiene consecuencias reales: dinero que se mueve, contratos que se firman, cosas que se publican con tu nombre. Ahí, un solo error grave detectado a tiempo amortiza la segunda suscripción entera y varias más. Es exactamente lo que la segunda opinión médica lleva haciendo toda la vida.



El viaje de ida y vuelta

La segunda opinión se queda a medias si la lista de objeciones acaba en tu escritorio y ahí muere. Falta un paso, y es el que ahorra la tarde.

Cuando el revisor te devuelva sus objeciones, no las corrijas tú. **Devuélveselas a quien construyó:**

Un revisor externo ha examinado tu trabajo y ha levantado estas objeciones. Contéstalas una por una: acéptala y corrige, o recházala explicando por qué. No reescribas nada que no esté cuestionado.

Quien construyó conoce el porqué de cada decisión que tomó; el revisor no. Por eso el revisor encuentra cosas que el otro no ve, y por eso el otro puede desmontar objeciones levantadas sin contexto. Entre los dos filtran bastante más de lo que filtra cualquiera por separado, y lo hacen sin ti delante.

Tú entras al final, que es donde tienes que entrar: leyendo qué se aceptó, qué se rechazó y con qué argumento. Dos pegadas de portapapeles, y el trabajo llega arbitrado en vez de en crudo.

Una regla para que esto no degenera en un pimpón infinito: **una vuelta y se acabó**. Si después de la ida y la vuelta sigue habiendo desacuerdo, decides tú. Son dos máquinas discutiendo, y el criterio es lo único que no se delega.

Dos cosas que se olvidan al día siguiente

Existir no es estar aprobado. Que el agente haya generado algo no significa que esté listo. El borrador del agente es materia prima; la aprobación es tuya. Suena obvio escrito aquí y se incumple constantemente, porque el texto sale tan terminado, tan bien maquettato y tan seguro de sí mismo que parece pedir permiso solo para el envío.

Pide verificación, no confianza. Cuando algo importe, no preguntes si está bien: pide la demostración. «Enséñame el cálculo, paso a paso.» «Lista las fuentes de esa cifra.» «Dime qué has comprobado y qué has supuesto.» Un agente al que le pides la prueba se comporta de otra manera que uno al que le pides el resultado. Y esa última pregunta —qué has supuesto— es la que más veces salva un presupuesto.

En la práctica. Coge el último documento importante que te generó la máquina y que enviaste sin revisar. Ábrelo en una sesión nueva con el *prompt* adversarial de arriba. Lo que aparezca en esa lista es la medida exacta de lo que te estás jugando ahora mismo. Cuesta cinco minutos y no cuesta un euro.

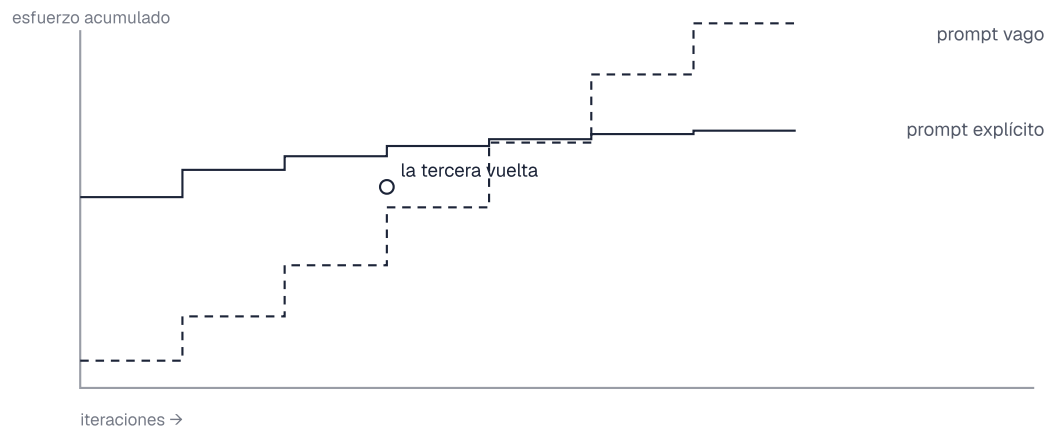
03 La paradoja del ahorro: para gastar menos hay que escribir más



Aquí llega la contradicción que descoloca a todo el mundo.

Todo el discurso del ahorro en inteligencia artificial va de escribir menos: *prompts* cortos, atajos, plantillas. Y es al revés. **Para gastar menos hay que escribir más.**

La razón es aritmética. Un *prompt* vago no ahorra: aplaza. Escribes tres líneas, recibes algo que no sirve, corriges, vuelve a no servir, corriges otra vez. Diez vueltas. Cada una de esas vueltas consume recursos, y las diez juntas consumen mucho más que el mensaje largo y explícito que habrías escrito al principio. El ahorro aparente de la primera línea lo pagas multiplicado en las nueve siguientes.



Escribir menos al principio no ahorra: aplaza, y con recargo.

Sé lo más explícito que puedas. Extiéndete lo que haga falta. Y cuando te encuentres un concepto que no dominas —y te vas a encontrar muchos—, pide la explicación entera: **«explícamelo como si tuviera cinco años»**. Esa frase, que parece una humillación, es la que más dinero ahorra de todo este tutorial. Quien finge entender toma decisiones equivocadas y las paga en iteraciones.

Lo que sí ahorra de verdad

Ser explícito no está reñido con ser eficiente. Estas cinco costumbres reducen el gasto sin tocar la calidad:

Edita el *prompt* anterior y relánzalo. Si la respuesta no sirve, no discutas cinco turnos con la máquina. Vuelve atrás, reescribe la petición original con lo que has aprendido y lánzala de nuevo. Discutir apila mensajes y cada mensaje nuevo arrastra todos los anteriores; editar sustituye. Es la técnica de economía más rentable que existe y casi nadie la usa.

Una tarea por sesión. Mezclar el presupuesto del hotel, el texto de la web y una duda sobre facturación en la misma conversación encarece las tres y empeora las tres.

Pide bloques completos, no parches. «Dame el texto entero corregido» en lugar de «dime qué cambiarías». Recibir un documento listo cuesta menos que recibir doce instrucciones que tú tienes que aplicar a mano — y en la aplicación a mano es donde se cuelan los errores.

Cierra las conversaciones largas. Un chat que lleva ochenta mensajes es caro y está peor: el modelo tiene que releer toda la conversación cada vez que le hablas, y en ese montón se le pierden las cosas. A esto se le llama *presupuesto de contexto*, y es la explicación real de

por qué la inteligencia artificial «se olvida» de lo que le dijiste hace media hora. No es desmemoria: es que ya no cabe.

Abre sesiones nuevas sin miedo. Y aquí está el enlace con todo lo que viene después: abrir una sesión limpia cuando tienes buenos archivos **no es empezar de cero**. Es empezar desde donde lo dejaste, pero sin el peso muerto. Para eso existen los archivos. Sin ellos, cerrar una conversación larga da vértigo; con ellos, es higiene.

La regla del idioma

Esta va pegada aquí porque es una decisión de economía, no de estilo.

Circula por ahí que los modelos obedecen mejor en inglés. Ese argumento envejeció: hoy es marginal y no voy a apoyarme en él. **El argumento honesto es otro, y es medible: el mismo contenido cuesta en torno a un 20 % más en español que en inglés.**

El motivo es que estos sistemas no leen letras ni palabras: trocean el texto en unidades llamadas *tokens*, y el español necesita más trozos para decir lo mismo. Puedes comprobarlo tú en un minuto: coge un párrafo cualquiera, pídele al agente que lo traduzca al inglés y que después cuente los *tokens* de las dos versiones. No es una cita de nadie; es un observable, y tu propio agente te lo verifica.

De ahí sale una regla con dos mitades:

- **Lo que se lee muchas veces —las instrucciones, los archivos del sistema— en inglés y comprimido.** El agente los relee en cada trabajo, todos los días, durante años. Ese 20 % se paga cada vez.
- **Lo que escribes una vez —tus peticiones del día— en tu idioma y explícito.** Aquí el ahorro es calderilla y la claridad lo es todo. Escribir en inglés forzado tus propias peticiones es cambiar precisión por céntimos: mal negocio.

¿Y si quieres las dos cosas? Escribe en tu idioma y **pide al agente que lo convierta**. Tú piensas en español, él comprime en inglés, y el archivo que queda guardado es el barato. Nadie tiene que aprender inglés técnico para esto.

Una última regla, esta sin excepciones: **el idioma de salida se declara siempre dentro de las instrucciones**. Jamás se deja a la deducción. Un archivo de instrucciones escrito en inglés sin esa línea produce, tarde o temprano, un correo a un cliente en inglés.

Y no basta con escribir «en español». Di **de dónde**: de España, de México, de Argentina, de donde tú trabajes. El agente elige otro vocabulario, otro tratamiento y otro ritmo en cada caso —usted o tú, ordenador o computadora, presupuesto o cotización—, y esa diferencia la

nota tu cliente mucho antes de que la notes tú. En los ejemplos que verás a continuación esa línea aparece como «[indica tu idioma y el país en el que trabajas]»: rellénala una vez y no vuelvas a pensar en ello.

04 Los cuatro archivos que convierten un chat en un sistema



Llegamos al corazón. A partir de aquí, todo es concreto.

Antes: qué es un `.md` y dónde vive

Un archivo `.md` —Markdown— es **texto plano con un formato mínimo**. Se abre con el Bloc de notas de cualquier ordenador. No necesita programas especiales, no se corrompe, no depende de ninguna empresa y pesa lo que pesa un correo. Los asteriscos y las almohadillas que verás en los ejemplos son solo marcas de énfasis y de título.

Su virtud es una doble legibilidad que no tiene ningún otro formato: **lo lee un humano y lo lee una inteligencia artificial, sin traducción por medio**. Un documento de Word está pensado para imprimirse; un `.md` está pensado para ser leído por los dos lados. Por eso todo este mundo funciona con ellos.

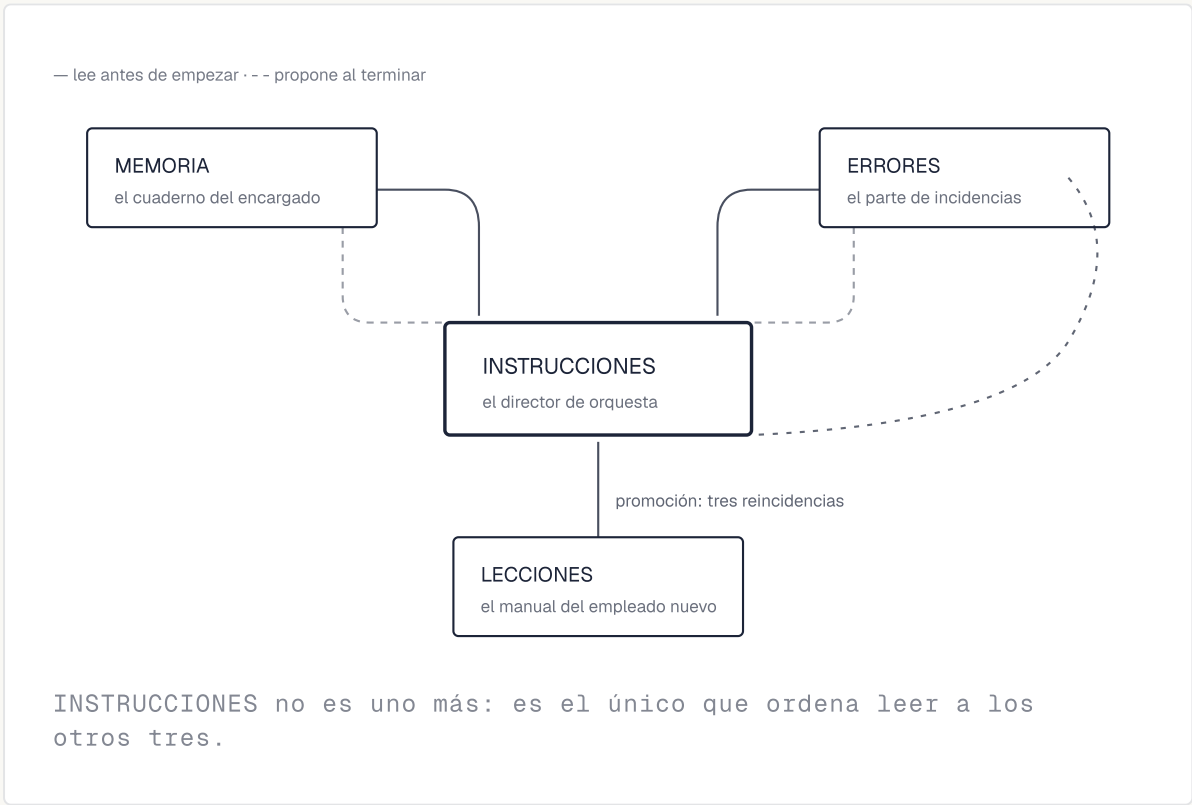
Y ahora el puente que falta en casi todos los tutoriales que has leído, el que separa saber redactar de saber usar: **¿dónde viven estos archivos?**

Viven en un **proyecto**. Las plataformas principales —por ejemplo, ChatGPT o Claude.ai— ofrecen un contenedor donde guardas instrucciones permanentes y archivos que persisten entre conversaciones. Cada una lo llama a su manera, pero la idea es idéntica. Todo lo que hables dentro de ese proyecto hereda esas instrucciones y tiene acceso a esos archivos. Abres una conversación nueva mañana y el agente ya sabe quién eres.

Sin proyecto, los archivos son un adorno: tendrías que pegarlos a mano en cada chat. Con proyecto, son un sistema.

***Traducción de cinco años:** el proyecto es la carpeta del taller donde está todo lo del taller. Los archivos son las hojas que hay dentro. Y el agente, cuando entra, lee las hojas antes de coger la herramienta.*

Los cuatro, y quién es quién



Los cuatro archivos del sistema y su equivalente en un taller

ARCHIVO	QUÉ ES	EL SÍMIL QUE LO FIJA

ARCHIVO	QUÉ ES	EL SÍMIL QUE LO FIJA
INSTRUCCIONES CLAUDE.md	Las reglas permanentes: quién eres, cómo trabaja contigo, qué debe leer antes de empezar	El director de orquesta: decide qué partitura lee cada uno
MEMORIA MEMORY.md	Decisiones tomadas, proyectos en curso, cosas pendientes	El cuaderno del encargado
ERRORES ERRORS.md	Qué falló, por qué y cómo se arregló	El parte de incidencias
LECCIONES LESSONS.md	Lo que ya sabemos y no volvemos a discutir	El manual del empleado nuevo

INSTRUCCIONES está en el centro y no es uno más: es el único que ordena leer a los otros tres. Un cuaderno que nadie abre es papel. Volveré sobre esto en un minuto, porque es el error más caro de todo el sistema.

El nombre del archivo da igual (y conviene entender por qué)

Verás escrito `CLAUDE.md` en todos los ejemplos de este tutorial, y merece una aclaración para que no te lleve a engaño.

`CLAUDE.md` es una convención de las herramientas de Anthropic: en su entorno de desarrollo, un archivo con ese nombre exacto se lee solo. Es cómodo, pero es un caso particular. **En el resto del mundo, y desde luego en los productos de chat que usa una pyme, no hay ningún nombre mágico.** ChatGPT, por ejemplo, no tiene un equivalente de `CLAUDE.md`: lo que tiene es el campo de instrucciones del proyecto, más los archivos que subas.

Y ahí está la conclusión que importa, que además simplifica mucho la vida: **puedes llamar a tu archivo como quieras.** `INSTRUCCIONES.md`, `MANUAL.md`, `TALLER.md`. Lo que hace que un archivo gobierne no es su nombre, sino que **las instrucciones permanentes del proyecto ordenen leerlo.** Esa línea —«antes de cualquier tarea seria, lee estos archivos»— es la que convierte cuatro documentos sueltos en un sistema, y es la misma en cualquier plataforma.

Regla práctica, que vale en todas: el contenido más corto y más importante va en el **campo de instrucciones del proyecto**, que se lee siempre; los archivos van subidos al proyecto y se leen porque ese campo lo manda. Si tu plataforma reconoce un nombre concreto por su cuenta, mejor: te ahorras una línea. Si no, no pierdes nada.

Cómo se despliegan de verdad: por etapas

Aquí es donde este tutorial se aparta del género. Los cuatro archivos te los he presentado juntos porque forman un sistema, pero **si intentas montarlos los cuatro el lunes por la mañana, no montarás ninguno**. Así es como crecen de verdad:

Día 1 — solo INSTRUCCIONES. Un archivo. Media hora. Quién eres, cómo hablas, qué no debe inventarse nunca. La diferencia se nota esa misma tarde, y esa es toda la motivación que vas a tener para seguir.

Semana 2 — nace MEMORIA. No antes. Nace el día en que te canses de repetir lo mismo: que los precios llevan IVA, que el contacto del hotel se llama Sofía, que los presupuestos salen en PDF. Ese cansancio es la señal, y llega solo.

El día que algo falle — nace ERRORES. Ni un minuto antes. **Una bitácora de errores vacía no enseña nada**, y peor: crear un archivo el lunes para que siga vacío el viernes es la forma más rápida de convencerte de que este sistema no sirve. El primer error real es su acta de nacimiento.

El último — LECCIONES. No se escribe: **se destila**. Cuando ERRORES lleve seis o siete entradas, le pides al agente que busque el patrón, y de ahí salen tres o cuatro frases que gobiernan de verdad. Una lección escrita en frío, sin cicatriz detrás, es una frase de manual de autoayuda.

Cuento esto en voz alta porque la alternativa —presentarte cuatro plantillas perfectas para que las copies— produce tutoriales que se admiran y no se aplican. Empieza por uno.

Los archivos de la Carpintería Ruiz

Estos son los cuatro archivos reales de nuestro ejemplo, tal y como quedarían al cabo de un par de meses.

INSTRUCCIONES — CLAUDE.MD

Instrucciones – Carpintería Ruiz

Quiénes somos

Taller de muebles a medida en Zaragoza. 6 empleados.

Clientes: viviendas particulares y hoteles pequeños.

Tono: cercano, profesional, sin jerga.

Cómo trabajar conmigo

- Todos los textos en [indica tu idioma y el país en el que trabajas], salvo que pida otra cosa.

- Antes de cualquier tarea seria, lee MEMORY.md, ERRORS.md y LESSONS.md.

- Dame siempre bloques completos y listos para usar, nunca fragmentos.

- Si mi petición es vaga, hazme UNA pregunta antes de empezar.

- No inventes nunca precios, plazos ni datos de clientes: pregúntamelos.

Reglas de operación (ascendidas desde ERRORS)

- Precios: SIEMPRE con IVA incluido, en todo documento que vea un cliente.

- Presupuestos: los precios salen de PRECIOS.md. De ningún otro sitio.

Al terminar cada trabajo

- Propón qué actualizar en MEMORY.md (decisiones), ERRORS.md (lo que falló)

y LESSONS.md (lo aprendido). Espera mi visto bueno antes de escribir.

Memoria – Carpintería Ruiz

Decisiones

- 12/07/2026 – Los precios se dan SIEMPRE con IVA incluido. (Queja de cliente.)
- 20/07/2026 – Los presupuestos salen en PDF, nunca en archivo editable.

Proyectos en curso

- Hotel Mirador: 14 cabeceros de roble. Entrega 15/09/2026. Contacto: Sofía.

Pendientes

- Nueva lista de precios de maderas duras (esperando al proveedor).

ERRORES – ERRORS.MD

Bitácora de errores – Carpintería Ruiz

18/07/2026 – Presupuesto enviado con precios viejos

Qué pasó: el agente reutilizó la lista de precios de enero de un chat antiguo.

Causa: los precios vivían solo en aquel chat, no en un archivo.

Arreglo: creado PRECIOS.md como fuente única; las instrucciones apuntan a él.

Estado: ASCENDIDO a instrucción el 02/08/2026 (tercera reincidencia).

05/08/2026 – Correo a un cliente con el tratamiento equivocado

Qué pasó: tuteó a un cliente de hotel al que tratamos de usted.

Causa: no estaba escrito en ninguna parte.

Arreglo: anotado en MEMORY. Si se repite, sube a instrucciones.

LECCIONES – LESSONS.MD

Lecciones aprendidas – Carpintería Ruiz

1. Lo que el agente deba recordar vive en un ARCHIVO, nunca solo en un chat.
2. Presupuestos: el agente redacta, un humano comprueba los números ANTES de enviar. Siempre.
3. Si una tarea necesita más de dos correcciones, las instrucciones eran vagas: arregla las instrucciones, no solo el resultado.

La regla del idioma, demostrada

Los cuatro archivos de arriba están en español para que puedas leerlos. Pero INSTRUCCIONES es el que el agente relee en cada trabajo, así que es exactamente el que conviene comprimir. Este es el mismo bloque, antes y después de pedirle al agente que lo convierta:

Cómo trabajar conmigo

- Todos los textos en [indica tu idioma y el país en el que trabajas], salvo que pida otra cosa.
- Antes de cualquier tarea seria, lee MEMORY.md, ERRORS.md y LESSONS.md.
- Dame siempre bloques completos y listos para usar, nunca fragmentos.
- Si mi petición es vaga, hazme UNA pregunta antes de empezar.
- No inventes nunca precios, plazos ni datos de clientes: pregúntamelos.

How to work with me

- All outputs in [your language and working country] unless I say otherwise.
- Before any substantive task, read MEMORY.md, ERRORS.md, LESSONS.md.
- Always give complete, ready-to-use blocks. Never fragments.
- If my request is vague, ask ONE question before starting.
- Never invent prices, dates or client data. Ask.

Mismo contenido, misma exigencia, menos de la mitad de palabras y en torno a un 20 % menos de coste cada vez que se lee. Fíjate en la primera línea del bloque inglés: el idioma de

salida sigue declarado —en inglés— y sigue siendo el tuyo. Eso es lo que hay que entender.

Tú no has escrito ese bloque: lo has escrito en español y has dicho «convíértelo». Diez segundos.

La bitácora que no gobierna es un diario

Y ahora el error que más tiempo cuesta descubrir, porque el sistema parece funcionar mientras lo cometes.

Registrar un error no evita repetirlo.

Lo aprendí a mi costa. Trabajando con un agente en el desarrollo de una web, tropezaba una y otra vez con la misma limitación del entorno. Lo resolvía con un apaño, seguía adelante, lo anotaba en la bitácora y lo pasaba a lecciones aprendidas. A la sesión siguiente, el mismo agente volvía a tropezar con lo mismo. Se lo pregunté directamente: si ya lo tenemos registrado, ¿cómo es posible?

Su respuesta, resumida: **una cosa es registrar el error y otra que ese registro se convierta en instrucción**. El agente no deduce por su cuenta, por haber leído una bitácora, que debe resolver la operación por otra vía. Ha leído una anécdota del pasado, no una orden para el presente.

Lo comprobé después en otra plataforma distinta y el comportamiento era idéntico. No es un defecto de un producto: es cómo funcionan estas herramientas. Así que este tutorial te ahorra el recorrido entero.

La regla de promoción. Un error que reincide —tres veces es buen umbral— **asciende de ERRORES a INSTRUCCIONES**, y cambia de forma al ascender. Deja de ser un relato en pasado y se convierte en una orden en presente:

- **En la bitácora:** «18/07 — el agente usó precios viejos de un chat antiguo».
- **En las instrucciones:** «Los precios salen de PRECIOS.md. De ningún otro sitio. Léelo antes de redactar cualquier presupuesto.»

Y el archivo de instrucciones tiene que ordenar esa lectura *antes de actuar*. Una regla que se lee después del error llega tarde.

Traducción de cinco años: el parte de incidencias documenta lo que pasó; la norma de seguridad impide que vuelva a pasar. Son dos papeles distintos. Y el parte se convierte en norma el día en que el accidente se repite por tercera vez.

Quien propone la promoción es el agente —basta con pedírselo en las instrucciones: «si detectas un error registrado tres veces, propónme convertirlo en regla»—. Quien la aprueba eres tú.

Quién mantiene todo esto (ahora sí, entero)

Lo prometí en la primera página y aquí está desarrollado, porque es la diferencia entre un sistema vivo y cuatro archivos fosilizados.

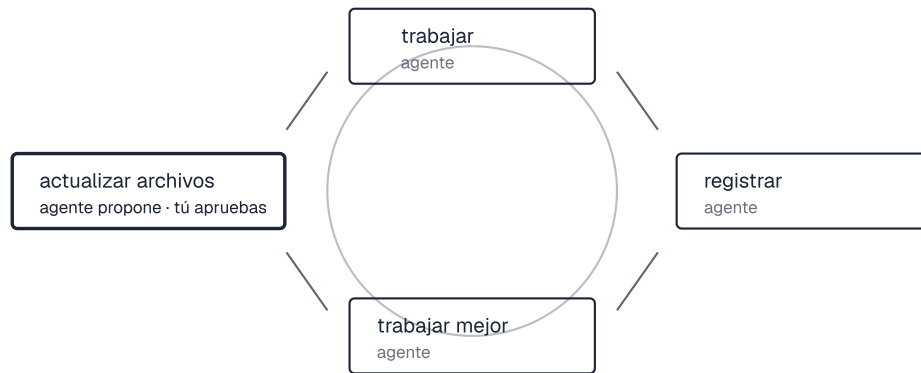
El sistema se mantiene solo porque se lo pides al agente. Esa línea del bloque de instrucciones —«al terminar cada trabajo, propón qué actualizar»— es la que sostiene todo. A partir de ahí, el cierre de cada sesión se parece a esto:

«He terminado. Propongo tres cambios: añadir a MEMORY que el Hotel Mirador ha adelantado la entrega al 8 de septiembre; añadir a ERRORS que el cálculo de metros lineales falló por no descontar los cantos; y borrar de MEMORY la línea del proveedor de herrajes, que ya está cerrada. ¿Los aplico?»

Tú lees seis líneas y dices «sí» o «los dos primeros». Diez segundos.

Fíjate en el tercer punto de esa propuesta, porque es el que casi nadie pide: **borrar**. Un archivo que solo crece se convierte en ruido, y llegaremos a eso en la sección 8.

El dueño aprueba. El dueño no redacta. Si en algún momento te descubres editando estos archivos a mano un domingo por la tarde, algo se ha roto: vuelve a las instrucciones y arréglalas ahí.

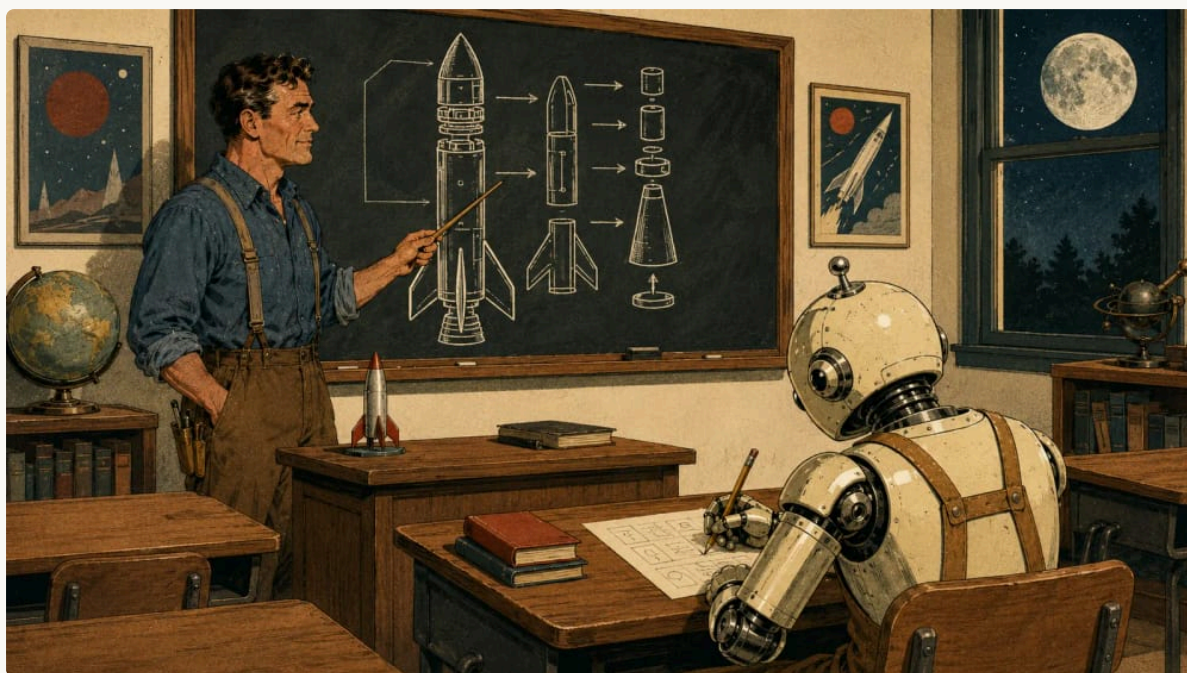


El sistema aprende contigo, y la única tarea tuya es aprobar.

En la práctica. Hoy, media hora: crea un proyecto en la plataforma que ya pagas y escribe **un solo archivo** de instrucciones. Cinco líneas sobre quién eres, tres sobre cómo quieres que te escriba, una prohibición («no inventes precios ni plazos: pregúntame») y la línea de mantenimiento («al terminar, propón qué actualizar; espera mi visto bueno»). Nada más. Los otros tres llegarán cuando les toque, y llegarán solos.

Si prefieres no partir de cero, [descarga el pack de arranque](https://fastforwardcommunication.agency/tutoriales/ia-para-pymes/pack-de-arranque.zip) (<https://fastforwardcommunication.agency/tutoriales/ia-para-pymes/pack-de-arranque.zip>): los cuatro archivos vacíos y listos para rellenar, en español y en inglés. Hoy solo necesitas abrir el primero.

05 Los cursillos del agente: verbos, no adjetivos



Se habla mucho de darle una «personalidad» al agente y se explica fatal. Vamos a arreglarlo.

Si trabajas con un modelo financiero, la tentación es escribir: «Actúa como un genio de las finanzas del sector textil». Suena bien. No sirve de nada. **Eso es una descripción. Son adjetivos. No hay un verbo, no hay una acción, no hay un resultado, no hay una métrica.** El agente no puede hacer nada con un halago.

Lo que sí funciona es una descripción de puesto: eres un analista de costes que **realiza** esta tarea, **obtiene** estas métricas, **entrega** estos resultados, **maneja** estas herramientas y **respeta** estas buenas prácticas.

El autor lo resume con una imagen que no necesita glosa:

Decirle «eres un gran ingeniero espacial» no te garantiza que te construyan el cohete que te lleve a la luna.

Hay que explicarle los conocimientos que ha de poseer, las herramientas que ha de emplear para construir el cohete y **las pruebas, auditorías y verificaciones que tiene que hacer antes de encender el primer motor.** Ese último tramo es el que todo el mundo se salta y el que de verdad separa un ayudante de un profesional: no basta con decirle qué haga, hay que decirle **qué tiene que comprobar antes de darlo por bueno.**

Aplicado a la carpintería, esto no es un párrafo de literatura sino cuatro líneas dentro del archivo de instrucciones:

```
## Cuando prepares un presupuesto
Eres el responsable de presupuestos del taller.
Calculas materiales a partir de PRECIOS.md, aplicas 22 % de mano de obra,
sumas transporte si el cliente está fuera de Zaragoza y entregas una
tabla
con partidas, subtotales y total con IVA incluido.
Antes de dármelo, comprueba: (1) que ningún precio venga de un chat
antiguo,
(2) que el plazo sea compatible con los proyectos abiertos en
MEMORY.md,
(3) que el total con IVA cuadre con la suma de partidas.
Si falta un dato, pregúntamelo. No lo estimes.
```

Eso es el tuétano, el alma que hace que el agente opere bien. Y no es un archivo que se escribe una vez: **se optimiza con el uso**. Cada vez que un presupuesto salga torcido, la corrección no va en el chat, va aquí.

Esto es un gimnasio, no una operación

Aquí conviene deshacer una expectativa que arruina muchos sistemas antes de que empiecen a rendir.

Para que el agente te dé soluciones que sirvan dentro de un año, tiene que recibir información de tu negocio **de manera consistente**: cómo se mueve tu mercado, qué precios cambian, qué proveedor ha dejado de ser fiable, qué normativa nueva te afecta. No es un montaje que se hace una tarde y queda hecho.

Míralo como ir al gimnasio. No entrenas seis meses seguidos y ya te quedas estupendo para el resto de tu vida. Si lo dejas, a los seis meses estás como al principio. Aquí pasa exactamente lo mismo: un sistema al que nadie alimenta desde marzo trabaja con el negocio que tenías en marzo.

Con una precisión que conviene hacer, porque circula mucha confusión: **el que se pone en forma no es el modelo, son tus archivos**. El motor no aprende de ti ni guarda nada de lo que le cuentas; el que acumula, corrige y mejora es el sistema que has montado alrededor. Por eso todo esto funciona con cualquier plataforma, y por eso si un día te cambias de proveedor

te llevas tus archivos y sigues donde estabas. Tu forma física no está en el gimnasio. Está en ti.

Lo cual, dicho sea de paso, explica el fallo más común de todos — pero eso viene en la sección 8.

El cursillo inverso

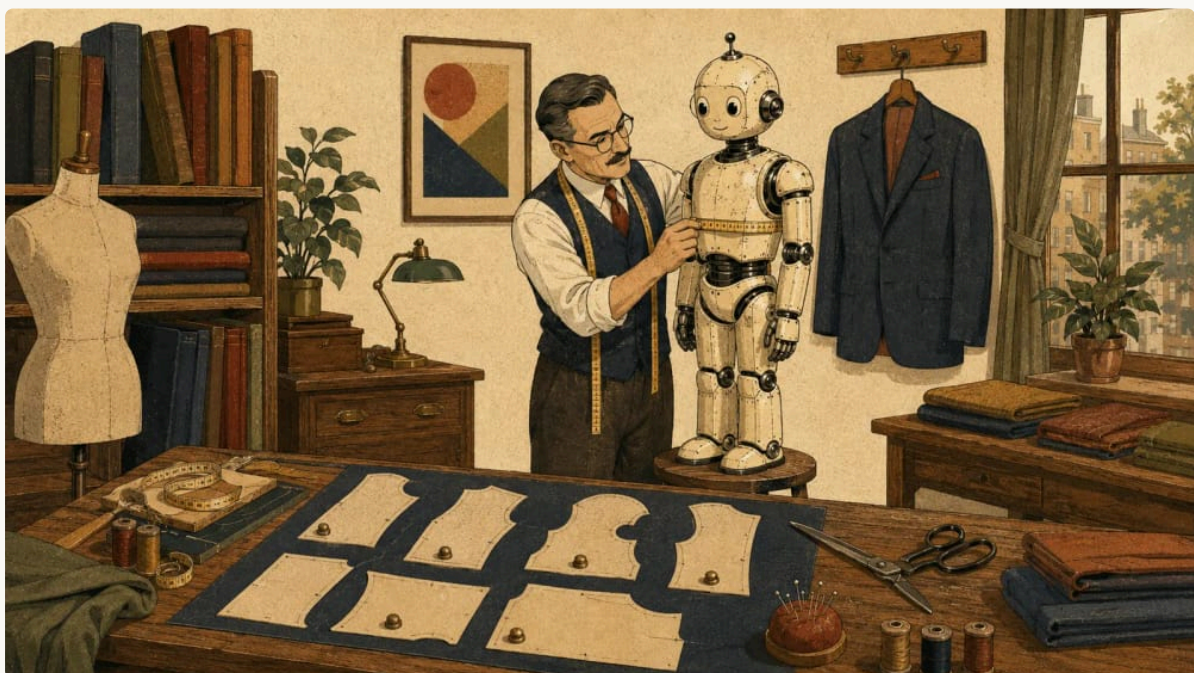
Termina un trabajo importante y hazle esta pregunta, que casi nadie hace:

«¿Qué te faltó saber de mi negocio para haber hecho esto mejor?»

La respuesta es oro, porque es un diagnóstico de tus propios archivos hecho por quien los usa. El agente te dirá que no sabía cuánto cobras por hora, que no sabía si el cliente era nuevo o antiguo, que no sabía si podías subcontratar el barnizado. Y cada una de esas frases es una línea que le falta a MEMORIA o a INSTRUCCIONES.

Este es el bucle completo: trabajas, se registra, se actualizan los archivos, se trabaja mejor. No hay más magia que esa, y funciona desde la segunda vuelta.

o6 Anatomía de un buen *prompt*



Un buen *prompt* no es una frase ingeniosa. Es una petición completa. Tiene seis piezas, y le faltan casi siempre las mismas: la quinta y la sexta.

1. **Contexto.** Quién eres y en qué situación estás. «Soy el dueño de una carpintería de seis empleados en Zaragoza; hacemos muebles a medida.»
2. **Rol operativo.** Con verbos y entregables, no con adjetivos. «Actúa como responsable de presupuestos: calcula materiales, aplica el margen y entrega una tabla.»
3. **Tarea concreta.** Una, y bien acotada. Dos tareas en un mismo *prompt* producen dos resultados mediocres.
4. **Formato de salida.** Tabla, correo listo para enviar, lista, bloque completo. Si no lo dices, te lo inventa, y luego pierdes veinte minutos reformateando.
5. **Restricciones.** Qué no hacer, qué no inventar, tono, longitud, idioma. «No inventes precios. No prometas plazos. Máximo 300 palabras. Tuteo no; a este cliente lo tratamos de usted.»
6. **Criterio de éxito.** Cómo sabréis los dos que está bien. «Es válido si un cliente lo entiende sin tener que llamarme para preguntar.»

Esa sexta pieza es la que más cambia el resultado y la que nadie escribe. Le das al agente una forma de juzgarse antes de entregarte nada.

01 Contexto

Soy el dueño de una carpintería de 6 empleados en Zaragoza; hacemos muebles a medida para casas y hoteles pequeños.

02 Rol operativo

Actúa como responsable de presupuestos: calcula materiales, aplica el margen y entrega el desglose.

03 Tarea concreta

Presupuesta 14 cabeceros de roble para el Hotel Mirador.

04 Formato de salida

Dame una tabla con partidas, subtotales y total.

05 Restricciones casi siempre falta

No inventes precios: sácalos de PRECIOS.md. No prometas plazos. IVA incluido. A este cliente lo tratamos de usted. Máximo 300 palabras.

06 Criterio de éxito casi siempre falta

Es válido si Sofía lo entiende sin llamarme para preguntar nada.

La plantilla de seis piezas, aplicada a una petición real del taller.

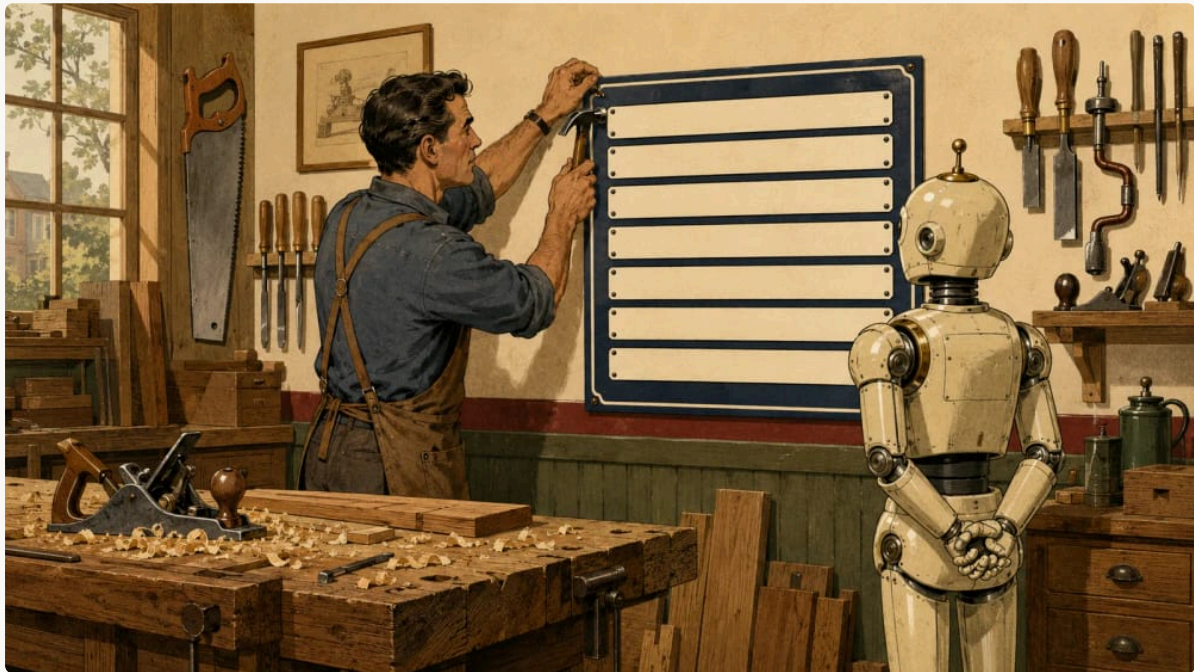
Y dos apostillas prácticas que valen por media sección:

- Si la respuesta no sirve, **edita el *prompt* original y relánzalo**. No discutas cinco turnos. Discutir con la máquina es caro y rara vez converge.
- Ante un concepto que no dominas, **pide la explicación de cinco años antes de decidir**. Decidir sobre algo que no entiendes es la forma más cara de usar estas herramientas.

Nota para cuando lleves un tiempo: si un *prompt* se te queda muy largo —contexto, reglas, datos y ejemplos a la vez—, puedes pedirle al propio agente que te lo reorganice con **etiquetas XML**: marcas del tipo `<contexto>...</contexto>`, `<tarea>...</tarea>`, `<restricciones>...</restricciones>` que delimitan cada parte para que no se mezclen. No hace falta para empezar y no hace falta saber escribirlas: se pide y se hacen. Pero cuando trabajes con peticiones grandes, notarás la diferencia.

En la práctica. Coge la petición que más veces repites al cabo del mes y escríbela una vez con las seis piezas. Guárdala en un archivo del proyecto. A partir de mañana esa petición se hace en diez segundos y sale bien a la primera.

07 Las reglas de oro



Ocho. Destiladas de trabajar así todos los días, con lo que ha costado aprenderlas.

1. **Jamás pegues contraseñas, claves ni datos bancarios en un chat.** Si necesitas confirmar un dato sensible, pide una marca y no un valor: «dime si coincide o no coincide», nunca «dime cuál es».
2. **Datos personales de clientes, los mínimos.** Anonimizados siempre que se pueda —«el cliente del hotel» en vez del nombre— y conociendo qué hace tu plataforma con lo que le das. Está escrito en sus condiciones y se lee en diez minutos.
3. **Antes de que el agente modifique algo importante, copia de seguridad con fecha. Siempre.** La web, la hoja de cálculo maestra, el listado de precios. Sin excepciones y sin pereza. Esta regla parece burocracia hasta el día en que deja de parecerlo, y ese día llega en la regla 8.
4. **El agente propone; tú decides.** Nada sale hacia un cliente sin tus ojos encima. Ni un correo.

5. **Cuando el agente afirme algo sobre terceros o dé una cifra, pide la fuente o compruébalo.** Con la misma naturalidad con que revisarías el trabajo de un empleado nuevo. No es desconfianza: es el trabajo.
6. **Un error que te costó tiempo, apuntado.** La misma piedra, una sola vez. Y si tropiezas tres veces, deja de ser una piedra y pasa a ser una norma: sube a instrucciones.
7. **Lo que va a tener consecuencias, lo construye uno y lo audita el otro.** Dinero, clientes, lo público. Quien construye no verifica.
8. **Dos agentes sobre los mismos archivos: por turnos.**

Esta última necesita explicación, y lo primero es acotar a quién le aplica, porque un titular sin acotar desactivaría la regla 7 justo para quien más la necesita.

La regla habla de agentes con acceso a archivos sobre la misma carpeta. Dos modelos abiertos en dos pestañas del navegador, respondiendo preguntas, sin tocar ningún archivo, conviven sin el menor riesgo. Esa es la situación de la mayoría de los lectores de esta guía, y en esa situación la segunda opinión de la sección 2 es gratis y no tiene contraindicaciones.

El problema aparece cuando dos agentes escriben en los mismos archivos. Y el fallo que se ve —un archivo a medio escribir, un documento roto— es el menor de los dos, precisamente porque se ve.

El que hace daño es silencioso. El agente B abre un archivo y lo lee. Mientras tanto, el agente A lo reescribe entero. Cuando B termina su trabajo, guarda encima su versión, la de antes. El trabajo de A desaparece. **No hay ningún mensaje de error. No hay ninguna alerta. Nadie se entera en días,** hasta que alguien busca una decisión que recuerda haber tomado y no está.

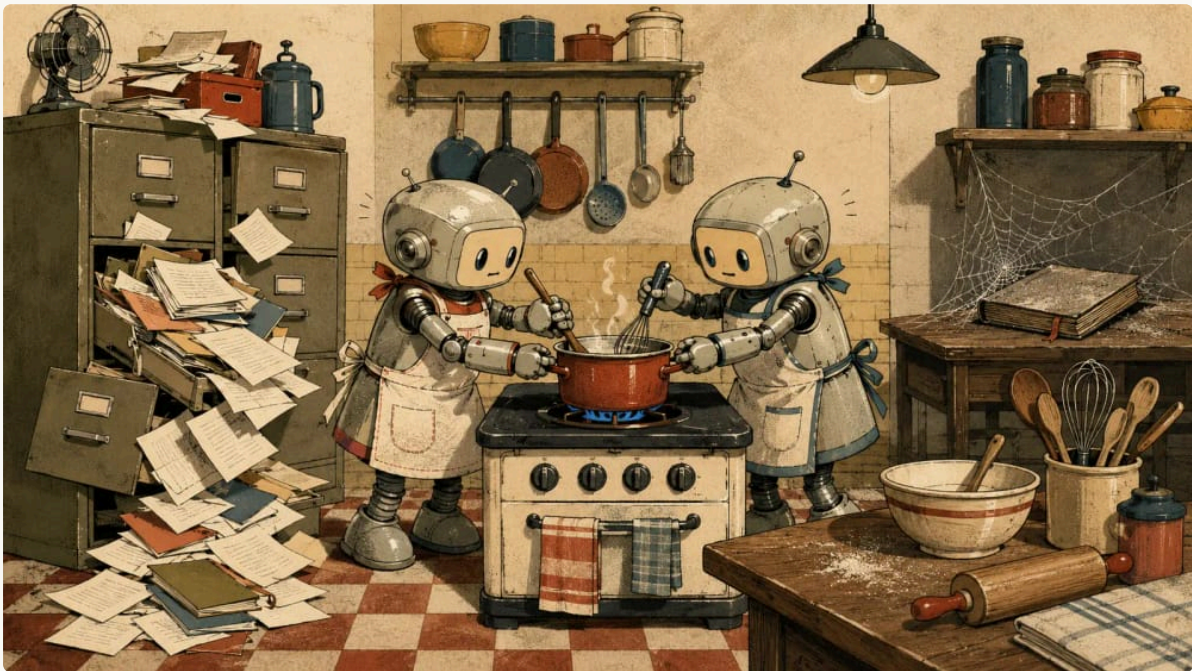
Aquí es donde la regla 3 deja de ser burocracia y se convierte en lo único que te salva: si hay copia de seguridad con fecha, se recupera; si no la hay, se rehace de memoria, y la memoria es mala.

Y el caso que vas a vivir de verdad no es el del dueño con dos agentes —ese relevo ocurre dentro de una misma cabeza y se organiza solo—. Es este: **dos personas de tu negocio, cada una con su asistente, trabajando sobre la misma carpeta compartida de Drive o de Dropbox.** Ahí no hay cabeza común, así que no hay turnos naturales. La solución es de taller, no de informática: **el relevo se anuncia.** En voz alta si estáis en la misma sala, o escrito en MEMORIA si no: quién tiene el testigo ahora mismo.

***Traducción de cinco años:** dos chefs no usan el mismo fogón a la vez. No porque esté prohibido, sino porque uno de los dos platos se quema y nadie sabe cuál.*

Esta regla complementa a la 7, no la contradice: **la segunda opinión es secuencial por naturaleza**. Primero se construye, después se audita. Nunca a la vez.

o8 Cuando esto sale mal



Todos los tutoriales enseñan el camino feliz. El que se recuerda es el que enseña dónde se rompe. Estos son los tres modos de fallo reales, por orden de frecuencia inversa: el último es el que te va a pasar a ti.

El archivo que crece hasta que nadie lo lee

MEMORIA empieza con seis líneas y a los cuatro meses tiene ciento cuarenta. Dentro hay proyectos entregados en marzo, un proveedor con el que ya no trabajas y tres decisiones que se contradicen entre sí. El agente lo lee entero antes de cada trabajo, así que ahora pagas por leer basura y, peor, aplicas decisiones muertas.

El arreglo: una memoria que solo crece se convierte en ruido. MEMORIA se poda. Lo cerrado se archiva o se borra. Y esto entra en las instrucciones como una tarea del agente, no tuya: «al proponer actualizaciones, propón también qué borrar y qué archivar por cerrado».

Corregir o borrar vale tanto como añadir. Un sistema que solo acumula no es memoria: es un desván.

Las instrucciones que se contradicen

En enero escribiste «sé breve». En abril, harto de respuestas escuetas, añadiste al final «desarrolla los argumentos con detalle». Las dos líneas siguen ahí. El agente hace lo que puede, que es una cosa distinta cada día, y tú concluyes que «últimamente está peor».

Nadie relee un archivo que solo se amplía. Y las contradicciones no se ven añadiendo: se ven leyendo entero.

El arreglo: relee el archivo completo cada cierto tiempo —un trimestre basta— **en lugar de añadir líneas al final.** O más cómodo: pídeselo al agente. «Lee las instrucciones enteras y dime si hay reglas que se contradicen o que sobran.» Encuentra las tuyas propias con una honestidad que sorprende.

El abandono a las dos semanas

Es el más común con diferencia, y no tiene nada que ver con la tecnología.

Montas el sistema un domingo con entusiasmo. La primera semana lo alimentas. La segunda tienes una entrega y no lo tocas. La tercera está desfasado, y un archivo desfasado es peor que no tener archivo, porque el agente lo lee y actúa según datos viejos. La cuarta semana has vuelto al chat suelto y has concluido que «esto de la IA está sobrevalorado».

El arreglo es de diseño, no de fuerza de voluntad. Si mantenerlo es trabajo tuyo, muere. Si es trabajo del agente y lo tuyo es aprobar, sobrevive.

Por eso la línea de mantenimiento va en las instrucciones desde el primer día, y por eso he insistido con ella tres veces en esta guía. No es un detalle de implementación: es la única diferencia entre un sistema que dura años y cuatro archivos que se quedaron congelados en marzo.

Cierre

La carpintería del principio sigue existiendo. Lo que ha cambiado en seis meses no es su plataforma ni su cuota: es que el lunes por la mañana, cuando su dueño abre una conversación nueva, no tiene que explicar nada. El agente ya sabe que los precios llevan IVA, que Sofía es del Hotel Mirador, que en enero se enviaron unos precios viejos y por qué eso no puede repetirse.

Ese hombre no ha aprendido a programar. Ha aprendido a escribir lo que sabe.

Y esa es, probablemente, la parte más incómoda de todo esto. Montar el sistema no es un problema técnico: es un ejercicio de poner por escrito cómo trabajas. Mucha gente descubre en ese momento que no tenía un método, sino una costumbre. La máquina no crea ese vacío. Solo lo hace visible, porque es incapaz de suponer lo que tú das por hecho.

Quien pasa por ahí sale con dos cosas: un asistente que rinde, y un negocio un poco mejor entendido por su propio dueño.

La pregunta ya no es qué modelo usar. La pregunta es qué pasa el día en que abras una conversación nueva y tengas que empezar otra vez por el principio.



PLAN DE ACCIÓN EN UNA PÁGINA

Empieza el lunes

Página autónoma, pensada para imprimirse y colgarse en el taller.

DÍA 1 Media hora, un solo archivo

- ☐ Crea un proyecto en la plataforma que ya usas (el contenedor con instrucciones y archivos persistentes).
- ☐ Escribe `CLAUDE.md` (o su equivalente): quiénes sois, cómo quieres que te escriba, qué no debe inventar nunca.
- ☐ Declara el idioma de salida dentro del archivo, y de dónde: España, México, Uruguay... No lo dejes a la deducción.
- ☐ Añade la línea que lo sostiene todo: «al terminar cada trabajo, propón qué actualizar en los archivos; espera mi visto bueno antes de escribir».
- ☐ Trabaja el resto del día como siempre. Verás la diferencia esta misma tarde.

SEM 2 Cuando repetirte empiece a cansar

- ☐ Crea `MEMORY.md` : decisiones tomadas, proyectos en curso, pendientes.
- ☐ Pídele al agente que lo rellene con lo que ya le has contado esta semana. Tú apruebas.
- ☐ Escribe una petición importante con las seis piezas y guárdala.
- ☐ Pide la conversión al inglés de tu archivo de instrucciones y quédate con esa versión.

DÍA __ El día que algo falle — ni un día antes

- ☐ Crea `ERRORS.md` con ese primer error: qué pasó, por qué, cómo se arregló.
- ☐ Añade a instrucciones: «si un error se repite tres veces, propónme convertirlo en regla».

MES 1 Consolidar

- ☐ Destila LESSONS.md a partir de lo acumulado en ERRORES. Pídeselo al agente; no lo escribas en frío.
- ☐ Prueba la segunda opinión gratis: *prompt* adversarial, sobre el último sesión nueva y limpia, trabajo importante.
- ☐ Cierre de trimestre: pide una relectura completa de las instrucciones buscando contradicciones y reglas muertas.
- ☐ Antes de que el agente toque nada importante: copia de seguridad con fecha. Siempre.

Glosario

Modelo (de IA)

El motor: el sistema entrenado que entiende y genera texto.

Agente

Un modelo puesto a trabajar: con instrucciones, herramientas y capacidad de ejecutar tareas, no solo de responder.

LLM

Siglas de «gran modelo de lenguaje»; el tipo de motor que hay detrás de estas plataformas.

Prompt

Lo que le pides: la orden o petición que escribes.

Prompt de sistema / instrucciones

Las reglas permanentes que el agente lee antes de cada trabajo.

Persona

El papel operativo que le das al agente: qué es, qué hace, qué entrega.

Proyecto

El contenedor que ofrecen las plataformas principales para guardar instrucciones y archivos persistentes entre sesiones; la casa donde viven los cuatro archivos de este tutorial.

Token

La unidad en que el modelo trocea el texto; es lo que tu suscripción mide — cuando tocas un límite de uso, son *tokens* lo que se ha agotado.

Contexto (ventana de)

La memoria de trabajo de una conversación; se llena, y por eso existen los archivos.

Sesión / chat

Una conversación con su propio contexto; cerrarla no borra tus archivos.

Iteración

Cada vuelta de mejora sobre un resultado.

Alucinación

Cuando el modelo afirma con seguridad algo falso; se combate con verificación, no con fe.

Archivo .md (Markdown)

Texto plano con formato sencillo; lo lee un humano y lo lee una IA.

Conector / MCP

El enchufe que da al agente acceso a una herramienta externa (correo, calendario, archivos).

Habilidad / Skill

Un paquete de instrucciones reutilizable para una tarea concreta.

Automatización / workflow

Una cadena de pasos que se ejecuta sola tras un disparador.

Human-in-the-loop

Sistema donde una persona revisa o aprueba antes de que algo tenga efecto.

Agéntico

Dicho de un sistema: capaz de ejecutar tareas por pasos con cierta autonomía, bajo criterio humano.

Créditos / límites de uso

El presupuesto de cómputo de tu suscripción; la razón de toda la sección del ahorro.

RAG

Técnica por la que el asistente consulta primero tus documentos y responde con lo que encuentra en ellos, en lugar de tirar solo de su memoria; es lo que hay detrás de «pregúntale a tus propios archivos».

API

La puerta por la que dos programas hablan entre sí sin humanos de por medio; cuando algo «se conecta por API», tu herramienta y la IA intercambian datos automáticamente.

Artefacto

El documento, tabla o pequeña página que el asistente construye y te muestra aparte de la conversación, listo para usar o descargar.

Base de conocimiento (fuentes)

El conjunto de documentos que le entregas al asistente como material de consulta permanente — el archivo del que puede beber en cada respuesta.

Razonamiento extendido (esfuerzo)

El modo en que le pides al modelo que piense más despacio y con más pasos antes de responder; cuesta más *tokens* y tarda más, así que se reserva para lo difícil.

Beta

Versión que ya funciona pero sigue en pruebas: se usa con ojo y sabiendo que puede cambiar.

Etiquetas XML

Marcas del tipo `<instrucciones>...</instrucciones>` que delimitan las partes de un *prompt* largo para que el modelo no las mezcle.

Aviso

Este tutorial tiene carácter exclusivamente informativo y divulgativo. No constituye asesoramiento profesional, técnico, jurídico ni de seguridad, y no garantiza resultados concretos: cada implementación depende del contexto, de los datos y del uso que cada persona haga de las herramientas. Las medidas de seguridad, privacidad y protección de datos — incluido el tratamiento de información propia, de clientes o de terceros— son responsabilidad exclusiva de cada usuario, que debe revisar los términos y políticas de las plataformas que utilice. El autor no mantiene relación comercial, laboral ni de patrocinio con las plataformas mencionadas; las cita únicamente en su condición de usuario. Las marcas citadas pertenecen a sus titulares. La información refleja el estado de las herramientas en la fecha de publicación y puede quedar desactualizada.

[Términos de uso](https://fastforwardcommunication.agency/terminos/) (https://fastforwardcommunication.agency/terminos/) · [Política de privacidad](https://fastforwardcommunication.agency/privacidad/) (https://fastforwardcommunication.agency/privacidad/) · [Cookies](https://fastforwardcommunication.agency/cookies/) (https://fastforwardcommunication.agency/cookies/) · [Derechos y titularidad](https://fastforwardcommunication.agency/terminos/#derechos-y-titularidad) (https://fastforwardcommunication.agency/terminos/#derechos-y-titularidad)

Ilustraciones generadas con inteligencia artificial bajo dirección del autor.



NOTA DEL AUTOR

Luis Martín es politólogo, consultor estratégico y fundador de Fast Forward Communication Agency. Escribe sobre inteligencia artificial, poder y comunicación en **Señal Sintética**, su *newsletter* en español. Si esta guía te ha resultado útil, suscribirte es la mejor forma de recibir las siguientes.

Suscríbete a **Señal Sintética** → (<https://senalsintetica.substack.com>)

**FAST
FORWARD**
COMMUNICATION

Tutoriales de comunicación, IA y estrategia. Más sistema, menos plantilla.

TUTORIALES

Todos los tutoriales

English version

© 2024–2026 Fast Forward Communication Agency / Luis Martín. Publicado bajo licencia CC BY-NC-ND 4.0. Puedes compartir esta guía íntegramente, con atribución y enlace a la fuente original, siempre que no la modifiques ni la utilices con fines comerciales.

Permisos y otros usos >

FF Tutoriales · N° 02 · IA para pymes