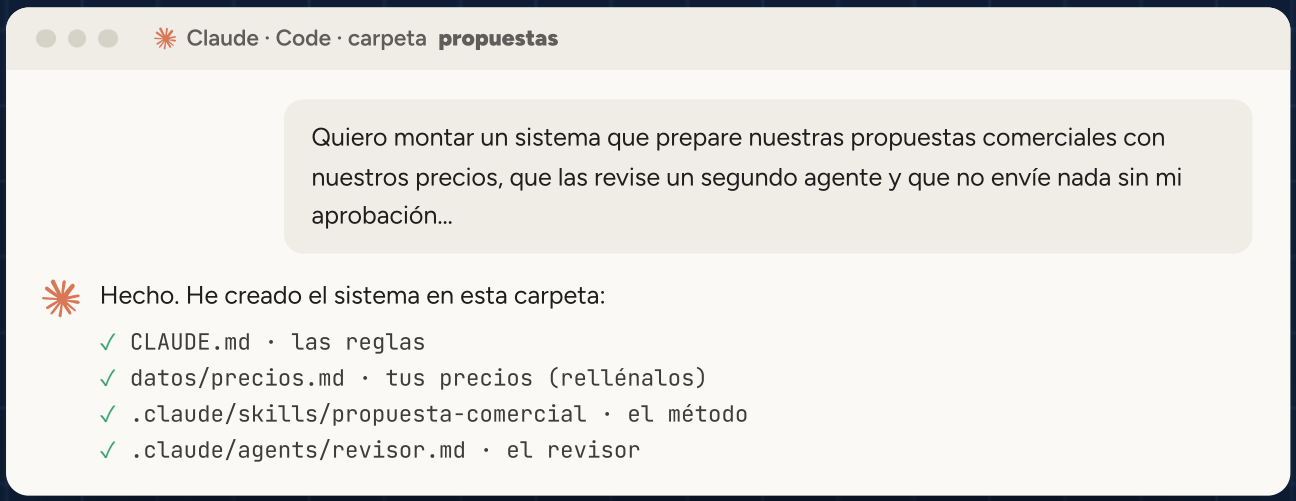
KIT GRATUITO · GUÍA PASO A PASO CON PROMPTS

Tu primer sistema de agentes con Claude, de principio a fin

Sin saber programar. Cada paso explicado y con el prompt listo para copiar y pegar en Claude. Al terminar tendrás un proceso de tu empresa que se prepara solo, se revisa solo y te pide permiso antes de enviar nada.

Con Claude Code, en la app de escritorio de Claude



Eneko Cid
Consultor y formador en automatización e IA
enekocid.com

¿Te ayudo a montarlo?
+34 611 00 23 99
hola@enekocid.com

ANTES DE EMPEZAR

Cómo usar esta guía

Vas a montar un **sistema de agentes**: un proceso de tu empresa que Claude hace solo, con vuestras reglas, revisado por un segundo agente y con tu aprobación antes de que nada salga fuera. Como ejemplo usamos las **propuestas comerciales**, pero sirve igual para informes, respuestas a clientes o contenido.

Dos caminos

Camino	Qué consigues	Tiempo
Exprés (página 5)	Un sistema sencillo funcionando con tres prompts. Para ver cómo es antes de dedicarle más tiempo.	20 minutos
Completo (páginas 6 a 14)	El sistema hecho con calma: tu proceso descrito, vuestros datos, el revisor probado con casos reales, conectado al correo y trabajando solo.	Una tarde (unas 3 horas)

Qué necesitas

- **Una cuenta de pago de Claude.** Claude Code está incluido en los planes Pro, Max, Ultra, Team y Enterprise; el plan gratuito no lo incluye. Para probar, basta con Pro. (Según Anthropic, noviembre de 2026; compruébalo antes de contratar.)
- **La app de escritorio de Claude** para Windows o Mac (se descarga en claude.com/download). Dentro tiene una pestaña llamada **Code**: ahí se trabaja en esta guía. No hace falta abrir ningún terminal.
- **Un proceso que se repita** y del que tengas algún ejemplo real: dos o tres propuestas antiguas, vuestra tarifa, un correo de solicitud.

PROMPT PARA COPIAR — así se ven en toda la guía

```
Copia todo el texto del recuadro y pégalo en Claude. Lo que está [ENTRE CORCHETES] lo cambias por lo tuyo antes de enviarlo.
```

> **Una regla para todo el sistema:** lo primero es que lo que produzca sea verdad. Si un dato no está en vuestros archivos, no se pone. Si no se puede comprobar, se borra. Es la regla que más problemas evita después.

Contenido

01 Las partes de un agente 3
02 Cómo es el sistema que vas a montar 4
03 Versión exprés: tu sistema en 20 minutos 5
04 Paso a paso completo (pasos 0 a 9) 6
05 Cómo se usan las skills 9
06 Prompts de rescate, cuando algo no sale 15
07 Ideas para tus siguientes sistemas 16
08 Así lo uso yo en mi web 17

Las partes de un agente

Un agente no es un chat que responde: es **alguien que hace un trabajo**. La forma más fácil de entenderlo es pensar en una persona nueva que entra en tu empresa. Necesita lo mismo que necesitaría ella.

1 El cerebro

`el modelo: Claude`

Entiende lo que le pides, razona y decide qué hacer. Es lo que ya conoces del chat.

2 El manual del puesto

`CLAUDE.md`

Un archivo con las reglas de la casa: qué es el proyecto, qué nunca debe hacer. Claude lo lee cada vez que empieza.

3 La información de la empresa

`la carpeta datos/`

Vuestros precios, vuestro estilo y ejemplos buenos. Es su única fuente de verdad: lo que no está aquí, no lo usa.

4 El procedimiento

`una skill`

Los pasos para hacer una tarea concreta, como los explicarías a alguien nuevo: qué leer, qué escribir, cómo entregarlo.

5 Las manos

`conectores (MCP)`

Le permiten usar tus herramientas: leer el correo, dejar un borrador, mirar el CRM. Sin ellos solo trabaja con archivos.

6 El segundo par de ojos

`un subagente revisor`

Otro agente que no ha escrito nada y comprueba cada dato contra vuestros archivos. Devuelve «aprobado» o «cambios».

7 Lo que puede hacer solo

`los permisos`

Qué hace sin preguntar (leer, preparar borradores) y qué te consulta siempre (enviar, borrar, publicar).

8 El horario

`una tarea programada`

Cuándo se pone a trabajar sin que se lo pidas: cada mañana, cada lunes, cuando llega algo nuevo.

✓ Y tú, que apruebas lo importante

`la aprobación humana`

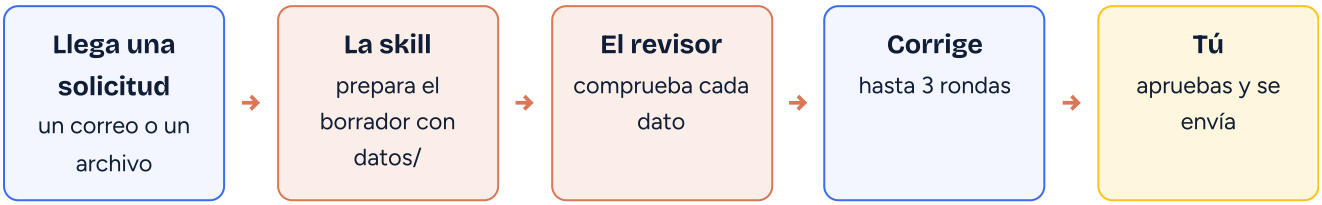
El sistema prepara y revisa; tú das el visto bueno antes de que algo llegue a un cliente. Esta pieza no se quita nunca en lo que sale fuera de la empresa.

Chat o sistema. Si una tarea la haces una vez, el chat basta. Si se repite cada semana, tiene siempre las mismas reglas y alguien debería revisarla antes de que salga, merece un sistema. Las piezas 2 a 8 son justo lo que hace que no tengas que explicarlo todo cada vez.

02 · EL MAPA

Cómo es el sistema que vas a montar

Así trabaja el sistema de propuestas cuando está terminado. Cada caja es una de las partes de la página anterior.



Lo que tendrás en la carpeta

```
propuestas/                          la carpeta del proyecto
├── CLAUDE.md                        2 · el manual del puesto
├── proceso.md                       cómo es tu proceso, por escrito
├── datos/                           3 · la información de la empresa
│   ├── precios.md
│   ├── estilo.md
│   └── ejemplos/                    propuestas antiguas que te gustan
├── solicitudes/                     lo que llega
├── propuestas/                      los borradores que prepara
└── .claude/
    ├── skills/propuesta-comercial/SKILL.md   4 · el procedimiento
    ├── agents/revisor.md                     6 · el revisor
    └── settings.json                         7 · los permisos
```

No tienes que crear nada de esto a mano. En cada paso le pides a Claude que lo cree, y él escribe los archivos. Tu trabajo es explicarle bien tu proceso y revisar lo que hace.

> **¿Y las plantillas del .zip?** Son estos mismos archivos ya escritos, con huecos `[ENTRE CORCHETES]`. Si prefieres partir de ellos, cópialos en tu carpeta y en cada paso usa el prompt «Adapta…» que verás en el recuadro gris. El resultado es el mismo.

La ruta del camino completo



Azul: preparar el proceso y los datos. Naranja: crear los agentes y probarlos. Oscuro: permisos, conexiones y trabajo en automático.

03 · VERSIÓN EXPRÉS · 20 MINUTOS

Tu primer sistema en tres prompts

Para verlo funcionar antes de dedicarle una tarde. Crea una carpeta vacía en tu escritorio llamada `propuestas`, abre la app de Claude, entra en la pestaña **Code** y elige esa carpeta para trabajar.

PROMPT 1 · CREAR EL SISTEMA — 5 minutos

```
Quiero montar en esta carpeta un sistema sencillo de agentes para preparar las propuestas comerciales de mi empresa, [NOMBRE], que se dedica a [A QUÉ OS DEDICÁIS, EN UNA FRASE].
Hazlo así:
1. Crea CLAUDE.md con las reglas: no inventar precios, clientes ni datos; todo sale de la carpeta datos/; nada se envía sin mi aprobación.
2. Crea datos/precios.md y datos/estilo.md con huecos [ENTRE CORCHETES] para que yo los rellene.
3. Crea la skill propuesta-comercial en .claude/skills/ con los pasos para preparar un borrador en la carpeta propuestas/.
4. Crea un agente revisor en .claude/agents/ que solo pueda leer, que compruebe cada dato contra datos/ y que responda APROBADO o CAMBIOS.
Cuando termines, explícame en cinco líneas qué has creado y qué tengo que rellenar yo.
```

Claude te pedirá permiso antes de crear cada archivo. Lee lo que va a hacer y acepta.

PROMPT 2 · RELLENAR VUESTROS DATOS — 10 minutos

```
Ayúdame a rellenar datos/precios.md y datos/estilo.md. Hazme las preguntas de una en una y espera a mi respuesta antes de pasar a la siguiente. Al final, enséñame cómo han quedado los dos archivos.
```

PROMPT 3 · PROBARLO — 5 minutos

```
/propuesta-comercial Prepara una propuesta para esta solicitud: [PEGA AQUÍ EL CORREO DE UN CLIENTE O INVÉNTATE UNO]
Cuando tengas el borrador, pásalo al agente revisor y corrige lo que te diga. Al final dime si ha quedado APROBADO y qué ha cambiado el revisor.
```

TERMINADO CUANDO tienes en `propuestas/` un borrador con tus precios reales y el revisor lo ha aprobado.

> **Fíjate en esto:** en el prompt 3 escribes `/propuesta-comercial` al principio. Así llamas a la skill por su nombre. En el apartado 05 tienes todas las formas de usar una skill.

Si te ha convencido, el camino completo hace lo mismo con más cuidado: describe bien tu proceso, prueba con casos reales, conecta el correo y lo deja trabajando solo.

0 Prepara el puesto de trabajo

10 minutos

1. Instala la app de escritorio de Claude e inicia sesión con tu cuenta de pago.
2. Crea en tu ordenador una carpeta para el proceso, por ejemplo `propuestas`. Todo el sistema vivirá ahí.
3. Mete en una subcarpeta `material` lo que tengas: tu tarifa, dos o tres propuestas antiguas que te gusten y algún correo de solicitud real. Si tienen datos personales de clientes, bórralos o cámbialos antes.
4. Abre la pestaña **Code**, elige la carpeta y escribe el primer prompt.

PROMPT · COMPROBAR QUE TODO ESTÁ LISTO

```
Hola. Antes de empezar: dime en qué carpeta estás trabajando, qué archivos ves dentro y resume en una frase qué hay en cada archivo de la carpeta material/. No cambies nada todavía.
```

TERMINADO CUANDO Claude te nombra tu carpeta y te describe tus archivos.

Si no ve los archivos: comprueba que has elegido la carpeta correcta al abrir la sesión. Puedes abrir una sesión nueva y elegirla otra vez.

> **Por qué importa la carpeta.** En el chat, Claude solo ve lo que pegas. En Claude Code trabaja *dentro de una carpeta*: puede leer tus archivos, crear otros nuevos y ejecutar acciones, siempre pidiéndote permiso. Esa carpeta es su puesto de trabajo.

1 Describe tu proceso

20 minutos

Es el paso más importante y el que más se salta. Si el proceso no está claro en tu cabeza, el agente lo hará mal. En vez de escribirlo tú, deja que Claude te entreviste.

PROMPT · LA ENTREVISTA

```
Quiero convertir en un sistema de agentes este proceso de mi empresa: [EJEMPLO: PREPARAR PROPUESTAS COMERCIALES CUANDO UN CLIENTE PIDE PRESUPUESTO].
Entrevístame para entenderlo bien. Hazme las preguntas de una en una y espera mi respuesta. Necesito que acabes sabiendo:
- qué lo pone en marcha;
- qué pasos hace hoy una persona, en orden;
- dónde está cada dato que hace falta;
- dónde hace falta criterio y qué reglas sigue alguien con experiencia;
- qué no se puede hacer nunca;
- quién lo revisa antes de que salga.
Cuando termines, escríbelo todo en proceso.md, claro y ordenado, y enséñamelo.
```

TERMINADO CUANDO lees `proceso.md` y podrías dárselo a una persona nueva para que hiciera el trabajo.

Consejo: contesta como si se lo explicaras a alguien que empieza mañana. Las excepciones («si es un cliente antiguo, no cobramos desplazamiento») son oro: díselas todas.

2 Escribe el manual del puesto: CLAUDE.md

15 minutos

`CLAUDE.md` es un archivo de texto con las reglas del proyecto. Claude Code lo lee **al empezar cada sesión**, así que todo lo que pongas ahí lo tiene presente sin que se lo repitas. Es la diferencia entre explicar las normas cada día y dejarlas escritas.

PROMPT · CREAR CLAUDE.MD

```
Lee proceso.md y crea CLAUDE.md con las reglas de este proyecto. Que tenga:
1. Qué es el proyecto, en dos frases.
2. Dónde está cada cosa: datos/, solicitudes/, propuestas/.
3. Las reglas que no se saltan nunca: no inventar precios, clientes, plazos ni resultados; todo dato sale de datos/ o de la solicitud; si falta un dato, se escribe «Falta por confirmar» en vez de suponerlo; nada sale de la empresa sin mi aprobación; las contraseñas nunca van en archivos ni en el chat.
4. Cómo se trabaja: cada borrador pasa por el agente revisor antes de darlo por terminado.
5. Una línea «automatizacion: activa». Si la cambio a «pausada», no debes hacer cambios.
Escríbelo corto, en frases sencillas, como para alguien que empieza hoy.
```

SI USAS LAS PLANTILLAS DEL .ZIP

```
Adapta el CLAUDE.md de esta carpeta a mi empresa usando proceso.md. Cambia todo lo que está [ENTRE CORCHETES] y pregúntame lo que no sepas.
```

TERMINADO CUANDO abres `CLAUDE.md` y las reglas son las vuestras, sin corchetes vacíos.

> **Truco.** Cuando el sistema haga algo que no te gusta, no corrijas solo ese resultado: dile «añade a CLAUDE.md una regla para que esto no vuelva a pasar». Así el sistema mejora cada vez.

3 Prepara la información de la empresa

20 minutos

El agente solo es tan bueno como los datos que le das. Aquí se crea la carpeta `datos/`, que será su **única fuente de verdad**: si un precio no está ahí, no lo pondrá.

PROMPT · CREAR LA CARPETA DATOS

```
Con lo que hay en material/ y en proceso.md, crea la carpeta datos/ con:
- precios.md: los servicios, qué incluye cada uno y su precio, indicando si es con o sin
IVA. Copia las cifras tal cual; si algo no está claro, pregúntame.
- estilo.md: cómo escribimos (tú o usted, tono, longitud, palabras que usamos y que no).
- ejemplos/: una copia de las mejores propuestas de material/, sin datos personales de
clientes.
Cuando acabes, enséñame precios.md para que lo revise línea a línea.
```

TERMINADO CUANDO has revisado `precios.md` tú mismo y todas las cifras son correctas.

No te saltes la revisión de precios. Es el único archivo del que depende que una propuesta sea correcta. Un precio mal copiado aquí se repetirá en todas.

4 Crea el procedimiento: la skill

20 minutos

Una **skill** es una carpeta con un archivo `SKILL.md` que explica cómo se hace una tarea, paso a paso. Tiene dos partes: arriba, un **nombre** y una **descripción** de cuándo usarla; debajo, las **instrucciones**. Claude lee la descripción para saber cuándo le toca, y las instrucciones cuando la usa.

PROMPT · CREAR LA SKILL

```
Crea una skill llamada propuesta-comercial en .claude/skills/propuesta-comercial/SKILL.md,
a partir de proceso.md.
- En la descripción, explica cuándo usarla con las frases que diríamos nosotros, por
ejemplo «prepara una propuesta» o «responde a esta solicitud».
- En las instrucciones, los pasos en orden: qué leer antes (la solicitud,
datos/precios.md, datos/estilo.md y los ejemplos), la estructura del borrador (lo que nos
piden, lo que proponemos, qué incluye y qué no, plazos, precio y siguiente paso), y qué
hacer al final (pasarlo al agente revisor, aplicar sus cambios con un máximo de 3 rondas y
guardar el borrador en propuestas/ con la fecha y el nombre del cliente).
- Añade las reglas: ningún precio, cliente o resultado que no esté en datos/; si falta un
dato imprescindible, escribir «Falta por confirmar» al principio.
Cuando la tengas, explícame en tres líneas cómo la voy a usar.
```

SI USAS LAS PLANTILLAS DEL .ZIP

```
Adapta la skill de .claude/skills/propuesta-comercial/ a nuestro proceso usando proceso.md
y los archivos de datos/. Cambia el ejemplo por uno parecido a nuestras solicitudes
reales.
```

TERMINADO CUANDO escribes `/` en el cuadro de mensaje y aparece `propuesta-comercial` en la lista.

Si no aparece: pídele «comprueba que la skill está en .claude/skills/propuesta-comercial/SKILL.md y que el encabezado tiene name y description». Si sigue sin salir, abre una sesión nueva en la misma carpeta.

> **La skill es vuestra forma de trabajar.** Lo que hace que una propuesta parezca vuestra no es la IA: son los pasos, las reglas y los ejemplos que pones aquí. En la página siguiente tienes cómo usarla y cómo mejorarla.

05 · CÓMO SE USAN LAS SKILLS

Usar, mejorar y compartir una skill

Una vez creada, la skill está siempre disponible en esa carpeta. No tienes que volver a explicar el procedimiento: lo llamas y Claude lo sigue.

Tres formas de usarla

Cómo	Qué escribes	Cuándo conviene
1. Por su nombre	`/propuesta-comercial` y, detrás, lo que necesitas: «/propuesta-comercial Prepara la propuesta para la solicitud que te pego: …»	Cuando quieres asegurarte de que la usa. Es la forma más segura.
2. Pidiéndolo con tus palabras	«Prepara una propuesta para este cliente». Si la frase encaja con la descripción de la skill, Claude la usa solo.	En el día a día, cuando la descripción ya está bien escrita.
3. Desde otra tarea	En una tarea programada o en otra skill: «usa la skill propuesta-comercial para cada archivo nuevo de solicitudes/».	Cuando el sistema trabaja solo (paso 9).

Para ver qué skills tienes en una carpeta, escribe `/` en el cuadro de mensaje: aparece la lista con los comandos y las skills disponibles.

Mejorarla: corrige la instrucción, no el resultado

La primera versión nunca es perfecta. Cuando un borrador salga mal, no lo arregles a mano y sigas: cambia la skill para que la próxima vez salga bien.

PROMPT · MEJORAR LA SKILL

```
La última propuesta tenía este problema: [EJEMPLO: PUSO EL PRECIO SIN IVA CUANDO NOSOTROS SIEMPRE LO DAMOS CON IVA].
No arregles solo esa propuesta. Cambia la skill propuesta-comercial (o CLAUDE.md, si es una regla general) para que no vuelva a pasar. Enséñame qué líneas has cambiado.
```

Compartirla con tu equipo

- Las skills de la carpeta `.claude/skills/` **van con el proyecto**: si tu equipo trabaja en la misma carpeta compartida o en el mismo repositorio, todos tienen la misma skill.
- Si quieres una skill para ti en todos tus proyectos, se guarda en la carpeta `.claude/skills/` de tu usuario. Pídeselo a Claude: «copia esta skill a mis skills personales».

Ideas de skills para empresas

Skill	Qué hace
responder-solicitud	Contesta a una petición de información con vuestro tono y vuestras condiciones.
informe-mensual	Lee los datos del mes y prepara el resumen con las conclusiones de siempre.
acta-reunion	Convierte las notas o la transcripción en acta con acuerdos, responsables y fechas.
ficha-cliente	Recopila lo que sabéis de un cliente antes de una reunión.

> **Para ir más allá:** en Automatekit Academy tienes la guía «Cómo crear una skill de Claude Code con el método de tu empresa» (automatekit.es/guias), con el detalle del archivo SKILL.md y los archivos de apoyo.

5 Crea el segundo par de ojos: el revisor

15 minutos

Un **subagente** es un asistente especializado con sus propias instrucciones y solo las herramientas que necesita. El más útil en una empresa es un revisor: **el agente que escribe no debe revisarse a sí mismo**, porque tiende a dar por bueno lo que acaba de hacer. Uno que no lo ha escrito, y que tiene orden de no dar nada por bueno sin comprobarlo, encuentra el precio que no está en la tarifa o la promesa que no podéis cumplir.

PROMPT · CREAR EL REVISOR

```
Crea un subagente llamado revisor en .claude/agents/revisor.md.
- Que solo tenga herramientas para leer archivos (Read, Grep y Glob): no puede editar
nada.
- Su trabajo: leer CLAUDE.md y datos/, leer el borrador entero y comprobar cada precio,
plazo, nombre y condición. Si un dato no está en datos/ o en la solicitud, o dice otra
cosa, es un error bloqueante.
- Que responda siempre con este formato:
VEREDICTO: APROBADO o CAMBIOS
BLOQUEANTES: la frase exacta, el problema y la corrección (o «eliminar esta frase» si no
se puede comprobar)
MEJORAS: sugerencias opcionales
- Que solo dé APROBADO si no hay ningún bloqueante, y que una preferencia de redacción
nunca sea bloqueante.
```

TERMINADO CUANDO existe `.claude/agents/revisor.md` y, al leerlo, ves que no puede editar archivos.

Cómo se usa el revisor

No hace falta llamarlo a mano: la skill del paso 4 ya le dice a Claude que pase cada borrador al revisor. Si quieres revisar algo concreto, pídelo así: «usa el agente revisor para revisar propuestas/**[archivo]**».

> **La instrucción que más mejora los resultados:** que el revisor proponga «eliminar esta frase» cuando no puede confirmar un dato. Un texto un poco más corto es mejor que uno con un error delante de un cliente.

6 Prueba el sistema con casos reales

30 minutos

Ahora se comprueba que todo funciona junto: la skill escribe, el revisor revisa, se corrige y queda un borrador. Pruébalo con **tres solicitudes reales** distintas, mejor si una es complicada.

PROMPT · PRUEBA COMPLETA

```
/propuesta-comercial Prepara la propuesta para esta solicitud: [PEGA LA SOLICITUD]
Cuando el revisor termine, enséñame: el borrador final, cuántas rondas ha hecho, qué
bloqueantes encontró y cómo se corrigieron.
```

Mira cada borrador con lupa: ¿los precios son los de tu tarifa?, ¿suena como vosotros?, ¿ha marcado «Falta por confirmar» lo que no sabía? Por cada fallo, usa el prompt «Mejorar la skill» de la página anterior.

TERMINADO CUANDO tres solicitudes seguidas salen bien sin que tengas que tocar el borrador.

7 Decide qué puede hacer solo: los permisos

10 minutos

Por defecto, Claude Code **te pregunta antes de hacer cosas** como crear o cambiar archivos. Al principio está bien, pero en un sistema que funciona querrás que lo rutinario lo haga solo y que lo delicado no pueda tocarlo nunca. Eso se decide en el archivo `.claude/settings.json`.

PROMPT · CONFIGURAR PERMISOS

```
Crea .claude/settings.json con estos permisos:
- Puedes leer sin preguntar todo lo que hay en solicitudes/, datos/ y proceso.md.
- Puedes crear y editar sin preguntar los archivos de propuestas/.
- No puedes editar nunca datos/precios.md ni leer ningún archivo .env.
Para todo lo demás, sigue preguntándome. Explícame después, en palabras sencillas, qué has permitido y qué no.
```

Acción	Sin preguntar	Siempre con tu aprobación
Leer solicitudes, datos y ejemplos	Sí	
Preparar borradores en propuestas/	Sí	
Enviar correos, publicar, pagar		Siempre
Cambiar precios o borrar datos		Siempre, y mejor ni darle permiso

> **Contraseñas y claves:** nunca en los archivos del proyecto ni en la conversación. Las conexiones del paso siguiente se autorizan en la web de cada servicio, no escribiendo la contraseña en el chat.

8 Conecta tus herramientas

20 minutos

Hasta aquí el sistema trabaja con archivos de la carpeta. Con los **conectores** (usan un estándar abierto llamado MCP) Claude puede usar tus herramientas: leer el correo, dejar un borrador de respuesta, consultar una hoja de cálculo o el CRM.

1. En claude.ai, entra en **Personalizar → Conectores** (claude.ai/customize/connectors) y busca tu herramienta de correo, por ejemplo Gmail. Los conectores de tu cuenta están disponibles también en Claude Code. En los planes Team y Enterprise los añade la persona que administra la cuenta.
2. Pulsa **Conectar** e inicia sesión en la página del servicio. Lee qué permisos pide.
3. Vuelve a tu sesión de Code y prueba con un prompt que solo lea.

PROMPT · TRAER LAS SOLICITUDES DEL CORREO

```
Busca en mi correo los mensajes de esta semana que pidan presupuesto o información sobre [VUESTROS SERVICIOS]. Guarda cada uno como un archivo en solicitudes/, con la fecha y el nombre de la empresa. No respondas, no borres y no envíes nada. Al final dime cuántos has encontrado.
```

PROMPT · DEJAR EL BORRADOR EN EL CORREO

```
Para la propuesta propuestas/[ARCHIVO], que está APROBADA por el revisor, crea un borrador de respuesta en mi correo al remitente de la solicitud. No lo envíes: lo reviso y lo envío yo.
```

Si no puede crear borradores: cada conector tiene sus propias acciones; algunos solo permiten leer. En ese caso, pídele que te deje el texto listo para copiar y pegar.

TERMINADO CUANDO las solicitudes llegan solas a `solicitudes/` y el borrador de respuesta aparece en tu correo, sin enviar.

9 Ponlo a trabajar solo

15 minutos

El último paso es que el sistema se ponga en marcha sin que se lo pidas: por ejemplo, cada mañana revisa las solicitudes nuevas, prepara los borradores y te avisa. Eso es una **tarea programada**.

Antes de programarlo

- Haz el paso 6 hasta que funcione bien a mano. **No automatices algo que todavía falla.**
- Guarda el proyecto con **Git**, un historial de cambios que te permite ver qué ha cambiado y deshacerlo. Pídeselo a Claude: «guarda este proyecto en Git y haz un primer guardado con todo lo que hay».
- Revisa la checklist de seguridad del .zip.

Dos formas de programarlo

Dónde	Cómo funciona	Qué necesita
En la nube	Se ejecuta a la hora fijada aunque tu ordenador esté apagado. Se crea escribiendo `/schedule` en Claude Code y explicando la tarea.	El proyecto en un repositorio de GitHub conectado a tu cuenta, porque cada ejecución empieza copiando ese repositorio. Según Anthropic, está en versión preliminar.

En tu ordenador	La app de escritorio también programa tareas locales, que trabajan directamente con tus archivos.	Solo se ejecutan si la app está abierta y el ordenador despierto a esa hora; si está en reposo, esa ejecución se salta.

PROMPT · LA TAREA PROGRAMADA

```
/schedule Cada día laborable a las 8:00, hora de Madrid, en este proyecto:
1. Lee CLAUDE.md. Si dice «automatizacion: pausada», no cambies nada y termina.
2. Busca en solicitudes/ los archivos que todavía no tengan propuesta en propuestas/.
3. Para cada uno, usa la skill propuesta-comercial. Cada borrador pasa por el agente
revisor, con un máximo de 3 rondas.
4. Si queda APROBADO, déjalo en propuestas/ como listo para revisar. Si no, márcalo como
«pendiente de revisión humana».
5. Nunca envíes correos, publiques ni borres nada.
6. Termina con un resumen de tres líneas: qué has preparado, qué ha quedado pendiente y
qué necesita mi aprobación.
```

TERMINADO CUANDO la primera ejecución te deja los borradores y el resumen, y no ha enviado nada.

Las cinco piezas del control

- **Interruptor:** la línea `automatizacion: pausada` de CLAUDE.md lo detiene todo sin tocar nada más.
- **Aviso:** cada ejecución termina con un resumen de lo hecho y lo pendiente.
- **Frecuencia baja al principio:** una vez al día, y revisa tú los primeros resultados.
- **Coste:** cada ejecución gasta uso de tu plan. Vigila el consumo las primeras semanas.
- **Mantenimiento:** cuando cambie una tarifa o una regla, se cambia en `datos/`, la skill o CLAUDE.md, nunca parcheando cada borrador.

Prompts de rescate

Lo normal es que algo no salga a la primera. Estos prompts resuelven casi todo.

NO ENTIENDO QUÉ HA HECHO

```
Explícame, en palabras sencillas y sin tecnicismos, qué has hecho en tus últimos pasos, qué archivos has creado o cambiado y por qué.
```

SE HA INVENTADO UN DATO

```
En propuestas/[ARCHIVO] has puesto [EL DATO], que no está en datos/. Bórralo, dime de dónde lo sacaste y añade a CLAUDE.md una regla para que no vuelva a pasar.
```

HA CAMBIADO ALGO QUE NO DEBÍA

```
Deshaz los cambios que has hecho en [ARCHIVO] y déjalo como estaba. Después explícame qué ibas a hacer y pregúntame antes de volver a intentarlo.
```

EL REVISOR NO TERMINA NUNCA

```
El revisor bloquea por detalles de redacción. Cambia .claude/agents/revisor.md para que solo sean bloqueantes los datos falsos o sin fuente y las reglas de CLAUDE.md rotas. Lo demás, que vaya a MEJORAS.
```

LA CONVERSACIÓN ES MUY LARGA Y SE LÍA

```
Resume en un mensaje todo lo importante de esta conversación: qué hemos montado, qué funciona y qué falta. Lo usaré para seguir en una sesión nueva.
```

NO SÉ SI ESTÁ BIEN MONTADO

```
Revisa todo el sistema de esta carpeta como si fueras un consultor: CLAUDE.md, datos/, la skill, el revisor y los permisos. Dime qué está bien, qué falta y qué riesgo ves, por orden de importancia. No cambies nada.
```

Dos costumbres que ahorran disgustos: pide los cambios de pocos en pocos (si pides diez a la vez, alguno saldrá mal) y, antes de un cambio grande, pide que guarde en Git lo que ya funciona.

07 · LO SIGUIENTE

Ideas para tus siguientes sistemas

El método es siempre el mismo: describe el proceso, prepara los datos, crea la skill y el revisor, prueba con casos reales, conecta y programa. Cambian las piezas:

Proceso	Qué lo pone en marcha	Qué hace la skill	Qué comprueba el revisor
Respuestas a clientes	Un correo con una consulta	Prepara la respuesta con vuestras condiciones y tono	Que las condiciones y plazos sean los vuestros
Informe mensual	El día 1 de cada mes	Lee los datos del mes y redacta conclusiones	Que cada cifra coincida con la fuente
Contenido del blog o boletín	Un calendario de temas	Investiga en fuentes oficiales y escribe con vuestro estilo	Cada dato contra su fuente; lo dudoso se borra
Preparar reuniones	Una cita en el calendario	Ficha del cliente con lo que sabéis de él	Que no mezcle datos de otros clientes

Son tipos de proceso de ejemplo, no casos de clientes.

> **¿Cuál montar primero?** El que más horas os quite al año, con reglas claras y poco riesgo si sale mal. El Excel del kit tiene una hoja para puntuar tus procesos candidatos y otra para describir el elegido.

08 · CASO REAL

Así lo uso yo en mi web

Todo lo de esta guía sale de cómo funciona enekocid.com. Estas son sus piezas:

Pieza	Qué hace
CLAUDE.md	Reglas del proyecto: nunca inventar clientes, casos ni precios; los precios solo salen de un archivo; si un dato es dudoso, se borra; qué archivos no se tocan sin mi permiso.
Skill /articulo	Elige el tema del calendario editorial y de los datos de Search Console, investiga en fuentes oficiales y redacta según una guía de estilo.
Agente revisor	Comprueba cada dato contra su fuente, la guía y la intención comercial. Devuelve APROBADO o CAMBIOS. No edita.
Tareas programadas	Dos mañanas a la semana se escribe un artículo, y solo se publica si el revisor lo aprueba.
Interruptor	Una línea en la configuración pausa todas las tareas automáticas sin tocar nada más.

Lo que he aprendido montándolo

- **El revisor es lo que da tranquilidad.** Sin él no dejaría publicar nada solo.
- **Separar lo que se cambia solo de lo que se propone.** Los textos se mejoran solos; lo que puede costar clientes (formularios, botones) solo se propone, y lo decido yo.
- **Se corrige la instrucción, no el resultado.** Cada vez que algo no me convencía, cambiaba la skill. Por eso cada semana sale mejor.

¿Quieres montarlo con tu equipo?

En la formación **«Crea tu equipo de agentes con Claude»** cada equipo monta un proceso real de su empresa de principio a fin: reglas, conexiones, skills, el agente que escribe y el que revisa, y tareas programadas, con aprobación humana donde importa. In-company u online. Temario: enekocid.com/claude-code-para-empresas

Si prefieres que lo diseñemos juntos sobre tu proceso, es la consultoría por horas. Escríbeme o llámame y lo vemos:

TELÉFONO Y WHATSAPP
+34 611 00 23 99

EMAIL
hola@enekocid.com