

01

Hermes Agent en 30 Minutos

Tu primer empleado sin CV: instalación, Telegram y tres automatizaciones reales.

63 páginas · Instalación + Telegram + 3 automatizaciones · Julio 2026

Gratis · sin email · cosasagenticas.com

Contenido

Parte 1: Qué es un agente de IA y por qué te importa

1. Bienvenido a tu primer empleado sin CV
2. Agente vs. Chatbot: la diferencia que cambia tu día
3. Lo que vas a tener al final de esta guía

Parte 2: Montaje en 30 minutos

4. Preparación: qué necesitas y qué no
5. Instala Hermes Agent
6. Elige el cerebro de tu agente
7. Crea tu bot de Telegram
8. Conecta el gateway y haz la primera prueba

Parte 3: De chat a sistema

9. Tus primeras tareas útiles
10. Tu primera automatización
11. Tu primer skill reutilizable
12. La primera semana: convierte el sistema en hábito

Parte 4: Casos reales

13. María y su bandeja de correo
14. Carlos y sus leads de Instagram
15. Ana y sus informes semanales

Parte 5: Domina tu agente

16. Checklist: ¿está vivo tu agente?
17. Troubleshooting: 5 problemas y sus soluciones
18. 3 ejercicios para hacer hoy mismo

Parte 6: Próximos pasos

19. Glosario para no perderte
20. El catálogo Cosas Agénticas

1. Bienvenido a tu primer empleado sin CV

Has oído hablar de los agentes de IA. Has visto a gente decir que tienen un «empleado digital». Has leído titulares sobre inteligencia artificial que prometen cambiarlo todo. Y tú quieres uno. Pero no quieres pasarte tres días leyendo documentación. No quieres aprender a programar. No quieres abrir una terminal negra con letras verdes. No quieres ser ingeniero.

Quieres hacer tres clics y que alguien —o algo— te responda por Telegram y haga cosas por ti. Que te busque información mientras tú cenas. Que te revise un documento mientras duermes. Que te avise de algo importante sin que tengas que estar pendiente.

Esta guía es exactamente eso.

En 30 minutos, sin complicaciones, vas a tener un agente de IA funcionando. Te responde por Telegram. Le puedes pedir que te busque información, que te escriba un correo, que te revise un documento, que te prepare un briefing cada mañana. Y lo hace solo. Sin que estés delante. Sin que le des al botón de enviar.

No necesitas saber programar. No necesitas instalar nada raro. Solo necesitas un ordenador —Windows, Mac o Linux—, una cuenta de Telegram y 30 minutos. Si has instalado alguna vez una app en tu móvil, tienes todo lo que necesitas.

Esto no es un manual de instrucciones. Es un «dámelo hecho». Es lo más parecido a contratar a alguien sin hacerle una entrevista. Sin CV. Sin nómina. Sin baja laboral. Sin que ocupe una silla.

Al terminar esta guía, habrás montado tres cosas que cambian radicalmente cómo trabajas:

- **Un agente que responde por Telegram** desde tu móvil, a cualquier hora, con cualquier pregunta.
- **Un briefing diario automático** que se genera solo cada mañana con tus tareas pendientes, decisiones por tomar y riesgos detectados.

- **Un sistema que trabaja mientras tú no estás** — sin pulsar enviar, sin abrir el ordenador, sin recordarle nada.

No es magia. Es un sistema determinista con una pieza no determinista dentro. La pieza no determinista es el motor de IA — el cerebro que piensa. Todo lo demás — la instalación, la conexión con Telegram, las tareas programadas, los registros de actividad — es predecible y está bajo tu control. Cuando algo falle — y algo fallará, porque en tecnología siempre falla algo la primera vez — sabrás exactamente dónde mirar para arreglarlo en cinco minutos.

Y una cosa más antes de empezar: no midas el éxito de tu agente por lo impresionante que sea. Médalo por las veces que te ahorró abrir una carpeta, copiar un texto, recordar una fecha o repetir una instrucción. Un agente que hace poco pero lo hace solo vale más que uno que promete mucho pero necesita que estés pendiente.

Vamos a ello.

2. Agente vs. Chatbot: la diferencia que cambia tu día

Antes de montar nada, necesitas entender una distinción fundamental. No es semántica. Es la diferencia entre tener un loro que repite lo que le dices y tener un empleado que hace cosas mientras tú estás en otra reunión.

El chatbot: responde cuando le hablas

Un chatbot es reactivo. Tú escribes, él contesta. Tú preguntas, él responde. Tú te vas, él se apaga. Es un espejo: solo existe mientras lo miras. ChatGPT, Claude, Gemini — todos funcionan así en su versión básica de chat. Son utilísimos para brainstorm, para redactar borradores, para traducir cosas. Pero no hacen nada si tú no estás delante dándole instrucciones.

El problema no es lo que hacen. Es que tú sigues siendo el cuello de botella. El chatbot no se despierta solo a las ocho de la mañana y te dice: «Oye, he revisado tus correos. Hay tres urgentes. ¿Quieres que redacte las respuestas?». Eso no lo hace un chatbot.

El agente: actúa aunque no le hables

Un agente de IA es un programa que usa un modelo de lenguaje —como DeepSeek, Claude o GPT— para tomar decisiones y ejecutar acciones por su cuenta. La diferencia clave es la autonomía.

Un agente:

- **Se despierta solo.** A las 8:00 de la mañana, sin que tú hagas nada, revisa tu bandeja de entrada y te envía un resumen por Telegram.
- **Toma decisiones dentro de reglas que tú defines.** «Si el correo contiene la palabra factura, muévelo a la carpeta de contabilidad. Si contiene urgente, notifícame por Telegram con prioridad alta.»

- **Ejecuta herramientas.** Lee archivos, escribe documentos, consulta APIs, envía mensajes, navega por internet. No solo habla: hace.
- **Deja trazabilidad.** Cada acción queda registrada en un log. Si algo sale mal, sabes qué pasó, cuándo pasó y por qué.
- **Trabaja 24/7.** Mientras duermes, tu agente puede estar monitorizando tu sitio web, clasificando correos o preparando el informe del viernes.

La diferencia no es de grado. Es de categoría. Un chatbot es una herramienta que usas. Un agente es un sistema que te sustituye en tareas repetitivas, predecibles y basadas en reglas.

Una metáfora que ayuda

Imagina que tienes un asistente humano. Un chatbot es un asistente que se sienta en una silla y espera a que le digas qué hacer. Si no le dices nada, se queda mirando la pared. Un agente es un asistente que llega a la oficina antes que tú, enciende las luces, revisa la agenda, prepara el café, clasifica el correo y te deja una nota en la mesa con lo importante del día. Cuando llegas, el trabajo ya está empezado.

Esa nota en la mesa es el briefing diario que vas a montar en esta guía. El café es la clasificación de correos. Las luces encendidas son las automatizaciones que se ejecutan sin que tú estés.

Con Hermes Agent, montas ese asistente. No necesitas un equipo de ingenieros. No necesitas un servidor en la nube. No necesitas un presupuesto de empresa. Necesitas 30 minutos, un ordenador y esta guía.

El 90% de las personas que usan IA a diario no han dado este paso. Siguen abriendo el navegador, escribiendo prompts, copiando y pegando respuestas. Tú, en media hora, vas a estar en el otro 10%.

3. Lo que vas a tener al final de esta guía

Tu primer empleado no tiene CV. No llega con corbata, no pide silla, no pregunta dónde está la máquina de café. Llega como una carpeta en tu ordenador, un comando en la terminal y un bot privado en Telegram.

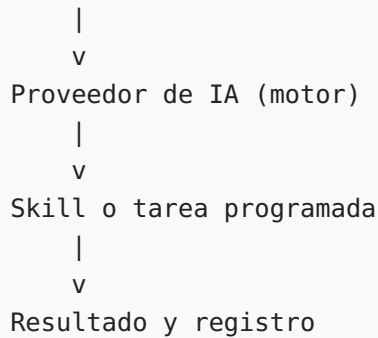
En 30 minutos vas a dejar un Hermes Agent pequeño, controlado y útil. No vas a montar una nave espacial. Vas a montar el primer puesto de trabajo de una máquina que recibe órdenes, ejecuta rutinas sencillas y deja constancia de todo lo que ha hecho.

Al terminar tendrás exactamente esto:

- [] Hermes Agent instalado y funcionando en tu ordenador.
- [] Una carpeta de trabajo aislada donde vive todo lo que hace tu agente.
- [] Un proveedor de IA configurado — el cerebro que piensa por tu agente.
- [] Un bot privado creado en Telegram con BotFather.
- [] El gateway de Telegram conectado a Hermes Agent.
- [] Solo tú puedes hablar con el bot — tu chat_id autorizado.
- [] Una prueba real de conversación desde tu móvil.
- [] Un briefing diario automático que se ejecuta cada mañana.
- [] Un primer skill reutilizable llamado briefing-diario.
- [] Un registro de actividad donde puedes ver qué pasó si algo falla.

El mapa de trabajo es sencillo. Cuando montes todas las piezas, la arquitectura será esta:

```
Telegram en tu móvil
|
v
Bot creado con BotFather
|
v
Gateway Telegram de Hermes
|
v
Hermes Agent
```



No necesitas entender de arquitectura para montarlo. Necesitas seguir pasos, copiar lo que te indicamos y comprobar cada pieza antes de montar la siguiente. Esto último parece aburrido hasta que algo falla. Y algo siempre falla la primera vez: una variable mal escrita, un token con un espacio al final, un bot hablando con otra cuenta, una zona horaria en UTC cuando tú estás en Madrid.

Lo hemos visto demasiadas veces como para saltarnos comprobaciones. Por eso cada capítulo termina con una verificación. Si te saltas una, lo pagas luego con debugging. Si sigues el orden, en 30 minutos estás respondiendo a tu bot desde el móvil.

Norma de seguridad número uno: no pegues tokens, claves API ni secretos en chats, capturas de pantalla, documentos compartidos ni repositorios de código. Si una clave se filtra, no se «desfiltra»: se revoca y se crea otra. Los tokens y claves API son contraseñas. Trátalas como tal.

Vas a usar una carpeta de trabajo aislada. En los ejemplos usaremos `~/hermes-30min`. Si prefieres otra ruta, cámbiala una vez al principio y mantén esa misma ruta durante toda la guía. La consistencia evita errores.

4. Preparación: qué necesitas y qué no

Antes de escribir el primer comando o hacer el primer clic, vamos a asegurarnos de que tienes todo lo necesario. Nada más frustrante que llegar al paso 7 y descubrir que te falta algo que podrías haber conseguido en dos minutos al principio.

Lo que necesitas

- **Un ordenador** con Windows, Mac o Linux. No importa cuál. Necesitas permisos de administrador para instalar programas. Si es un ordenador de empresa con políticas restrictivas, habla con tu equipo de IT antes de empezar.
- **4 GB de RAM libres** como mínimo. Si tu ordenador tiene 8 GB y vas justo con el navegador abierto, cierra lo que no uses durante la instalación.
- **Acceso a internet** sin proxy corporativo que bloquee APIs. Si estás tras un firewall de empresa, verifica que puedes acceder a servicios como api.telegram.org y openrouter.ai.
- **Telegram instalado en tu móvil** con tu cuenta personal. No necesitas un número distinto ni una cuenta de empresa. Tu cuenta de siempre.
- **Una clave API de un proveedor de IA.** Te explicamos tres opciones en el capítulo 6, desde gratis hasta 5 euros al mes. Necesitas al menos una.
- **30 minutos sin interrupciones.** No es una estimación generosa. Es lo que tardas siguiendo los pasos sin saltarte verificaciones.

Lo que NO necesitas

- No necesitas saber programar.
- No necesitas un servidor en la nube.
- No necesitas una tarjeta de crédito (DeepSeek y OpenRouter ofrecen crédito gratuito al registrarte).

- No necesitas instalar Python, Node, Docker ni nada parecido.
- No necesitas un dominio ni HTTPS.
- No necesitas tener experiencia previa con inteligencia artificial.

Si algo de la primera lista te falta, consíguelo primero y vuelve. La guía asume que tienes estas piezas. Sin ellas, algunos pasos no se pueden completar y acabarás frustrado sin necesidad. Intentar esquivar restricciones de red o permisos durante la instalación es la receta perfecta para una sesión de debugging de tres horas que no tiene nada que ver con Hermes.

Una última cosa: abre una sola terminal. No abras cinco. Una terminal limpia evita que mezcles variables de otra sesión. Si ya tienes un proyecto abierto, abre una nueva pestaña o ventana. Empezamos limpios.

5. Instala Hermes Agent

Junio de 2026. Hermes Agent tiene dos caminos de instalación. Elige el que te resulte más cómodo. El resultado final es exactamente el mismo: un agente funcionando que responde por Telegram.

Camino A: La app de escritorio (recomendado si no quieres tocar la terminal)

Hermes Agent tiene una aplicación de escritorio. Como Spotify. Como Chrome. Como cualquier app que instalas con un clic.

En Windows: Abre PowerShell (botón derecho en Inicio, Windows PowerShell). Pega esto y pulsa Enter:

```
irm https://install.hermes-agent.com.ps1 | iex
```

Se abre una ventana de instalación. Siguiente, siguiente, instalar. Como cualquier programa. Al terminar verás un icono de Hermes Agent en tu escritorio. Haz doble clic.

En Mac: Abre Terminal (está en Aplicaciones, Utilidades). Pega esto:

```
curl -fsSL https://install.hermes-agent.com.sh | bash
```

Al terminar, busca Hermes Agent en Aplicaciones. Doble clic. Hecho.

En Linux: Abre una terminal y ejecuta:

```
curl -fsSL https://install.hermes-agent.com.sh | bash
```

La app de escritorio incluye:

- Instalación guiada paso a paso — elige proveedor, conecta Telegram, primer skill.
- Panel de control visual para gestionar skills, tareas programadas y herramientas.
- Registros de actividad y monitorización en tiempo real.

- Conexión automática con el gateway de Telegram.

Si usas la app de escritorio: los capítulos 6 al 8 de esta guía te los hace la app automáticamente. La app te guía por un asistente de configuración. Puedes saltar directamente al capítulo 9 para empezar a usar tu agente y al capítulo 10 para montar automatizaciones.

¿No quieres tocar nada de terminal? Entra en hermes-agent.nousresearch.com y busca el botón de descarga para tu sistema. Doble clic en el instalador. Hecho. Cero terminal.

Camino B: Instalación por terminal (si quieres control total)

Si prefieres entender cada pieza y tener control absoluto, la instalación por terminal es para ti. Es lo mismo que hace la app, pero viendo cada engranaje. Saber lo que hay debajo te servirá cuando quieras personalizar cosas que la app de escritorio no expone.

Primero, crea tu carpeta de trabajo:

```
mkdir -p ~/hermes-30min  
cd ~/hermes-30min
```

Crea la estructura de carpetas que vas a usar:

```
mkdir -p config secrets logs skills cron outputs/briefings inbox
```

Comprueba que la estructura está bien:

```
find ~/hermes-30min -maxdepth 2 -type d | sort
```

Ahora instala Hermes Agent. El método oficial es el instalador con curl:

```
curl -fsSL https://install.hermes-agent.com | bash
```

Cuando termine, comprueba que el comando funciona:

```
hermes --help
```

Si el comando no se encuentra, cierra y reabre la terminal, o recarga tu shell:

```
source ~/.bashrc    # si usas bash
source ~/.zshrc     # si usas zsh
```

Ahora inicializa la configuración de Hermes para este espacio de trabajo:

```
cd ~/hermes-30min
hermes config init
```

Cuando pregunte el idioma, elige `Spanish`. Cuando pregunte la zona horaria, usa la tuya — para España: `Europe/Madrid`. Cuando pregunte el proveedor por defecto, puedes elegir `openrouter` para empezar (en el siguiente capítulo configuramos las claves).

Configura el motor de tu agente. Recuerda: tu agente trabaja para ti; el motor es invisible. No necesitas saber qué modelo hay debajo para usarlo, igual que no necesitas saber qué motor lleva tu coche para conducirlo:

```
hermes config set model.name "deepseek-chat"
```

Haz una prueba mínima para verificar que todo está instalado:

```
hermes -z "Responde solo con: Hermes instalado y listo."
```

Si todavía no has configurado el proveedor, este comando puede fallar con un error de «API key no encontrada». Es normal. El fallo correcto aquí es algo como «missing API key» o «provider not configured». El fallo malo es «command not found». Si el comando existe, ya tienes el cuerpo del agente. Ahora le ponemos el cerebro.

La decisión entre el camino A y el camino B es tuya. Los dos llevan al mismo sitio: un agente que responde por Telegram. Uno te lleva en coche con GPS. El otro te da el mapa, las llaves y te enseña dónde está el motor. Si alguna vez quieres personalizar algo que la app no permite, siempre puedes volver al camino B más adelante.

6. Elige el cerebro de tu agente

Tu agente necesita un cerebro — un modelo de inteligencia artificial que piensa por él. Es como elegir el motor de un coche. No necesitas entender cómo funciona por dentro, pero sí saber qué opciones tienes y cuál te conviene más.

Tienes tres caminos. No necesitas usar los tres. De hecho, para empezar usa solo uno. Tener los otros configurados te evita rehacer la instalación cuando quieras cambiar. Cambiar de proveedor en Hermes se hace en diez segundos.

Opción A: DeepSeek (recomendado para empezar)

DeepSeek es un proveedor chino de modelos de IA que ofrece una relación calidad-precio excepcional. Su modelo `deepseek-chat` habla español perfecto, es rápido y cuesta entre 2 y 5 euros al mes para un uso normal de 20-30 mensajes diarios.

Cómo configurarlo:

1. Ve a **platform.deepseek.com** y crea una cuenta gratuita con tu email.
2. Los primeros 5 euros de crédito son gratis al registrarte. No necesitas tarjeta para empezar.
3. Ve a la sección «API Keys» y pulsa «Create new key».
4. Copia esa clave. Es una cadena larga de letras y números.
5. En la app de Hermes, ve a Configuración y pega la clave donde dice «DeepSeek API Key».

Si estás usando la terminal:

```
hermes config set provider.name deepseek
hermes config set provider.api_key "tu-clave-aqui"
```

Ventajas de DeepSeek: barato, rápido, español perfecto, 5 euros gratis al registrarte, no necesitas tarjeta de crédito.

Opción B: OpenRouter (si quieres probar varios modelos)

OpenRouter es un intermediario que te da acceso a más de 200 modelos de IA diferentes — GPT-4, Claude, Gemini, Llama y muchos más — desde una sola cuenta. Ideal si quieres comparar modelos o necesitas uno específico para una tarea concreta.

Cómo configurarlo:

1. Ve a **openrouter.ai** y crea una cuenta gratuita.
2. Ve a la sección «Keys» y genera una clave API.
3. Copia esa clave y pégala en la app de Hermes o en la terminal:

```
hermes config set provider.name openrouter  
hermes config set provider.api_key "tu-clave-openrouter"
```

Ventajas de OpenRouter: acceso a más de 200 modelos, puedes comparar precios y calidades, ideal si quieres experimentar. Coste: variable según el modelo que uses, desde fracciones de céntimo hasta unos 20 euros al mes con modelos premium.

Opción C: Ollama (gratis, 100% privado, pero necesita buen hardware)

Ollama ejecuta modelos de IA directamente en tu ordenador. Nada sale de tu máquina. Es gratis. Es privado. Pero tu ordenador hace todo el trabajo, así que necesitas al menos 16 GB de RAM para que funcione con fluidez.

Cómo configurarlo:

1. Ve a **ollama.com** y descarga el instalador para tu sistema.
2. Instálalo como cualquier programa.
3. Abre una terminal y descarga un modelo ligero para empezar:

```
ollama pull llama3.2
```

En la app de Hermes, selecciona «Ollama» como proveedor y el modelo que descargaste.

Ventajas de Ollama: 100% gratuito, 100% privado (tus datos nunca salen de tu ordenador), sin límites de uso. **Desventajas:** más lento que las opciones de pago, requiere buen hardware (mínimo 16 GB de RAM), los modelos locales suelen ser menos potentes que los de pago.

¿Cuál elijo?

Nuestra recomendación para empezar: **DeepSeek**. Barato, rápido, español perfecto, 5 euros gratis de crédito inicial. En cinco minutos tienes la clave. Si luego quieres probar otros modelos, cambias en diez segundos desde la app de Hermes.

Si tu prioridad es la privacidad total o tienes un presupuesto cero, Ollama es tu opción. Si quieres flexibilidad máxima para experimentar con distintos modelos, OpenRouter.

No te quedes bloqueado decidiendo. Elige uno ahora. Lo cambias cuando quieras. Lo importante es avanzar.

Importante: una suscripción de chat en la web —como ChatGPT Plus o Claude Pro— no es lo mismo que una clave API. Si Hermes está usando la API, necesitas una clave API. Si estás usando una suscripción de chat, el flujo es distinto. Asegúrate de estar generando una clave API en el panel de desarrollador de tu proveedor, no la contraseña de tu cuenta de chat.

7. Crea tu bot de Telegram

Este es el paso más divertido. Vas a crear un bot de Telegram. Ese bot **es** tu agente. Le escribes desde el móvil y él te responde. A cualquier hora. Desde cualquier sitio.

Un bot de Telegram no es una app aparte. Es un contacto más en tu lista de chats. Como hablar con un amigo, pero un amigo que puede buscar información, generar documentos y ejecutar tareas por ti.

Crear el bot con BotFather

BotFather es el bot oficial de Telegram para crear y gestionar bots. Es como el departamento de recursos humanos de los bots.

1. Abre Telegram en tu móvil (o en el ordenador, da igual).
2. Busca **@BotFather**. Verifica que es la cuenta oficial — tiene un check azul de verificado.
3. Inicia el chat y escribe: `/newbot`
4. BotFather te preguntará el nombre visible del bot. Es el nombre que verás en tu lista de chats. Ponle algo reconocible: «María Asistente», «Hermes Personal», «Mi Agente». Lo que quieras.
5. Luego te pedirá un nombre de usuario. Tiene que ser único y acabar en `bot`. Por ejemplo: `maria_asistente_bot` o `tu_nombre_hermes_bot`. Si ya está cogido, prueba con otro.
6. Cuando lo acepte, BotFather te dará un **TOKEN**. Es una cadena larga de números y letras con dos puntos en medio. Algo como `1234567890:AAFbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrR`.

Copia ese token ahora mismo. No lo mandes a nadie. No lo guardes en una nota compartida. No lo pegues en un grupo de Telegram. No hagas una captura de pantalla y la subas a internet. Ese token es la llave de tu bot. Con él, cualquiera puede leer todos los mensajes que reciba y suplantarlos.

El token es una contraseña. Si se filtra, no se «desfiltra». Vas a BotFather, revocas el token con `/revoke` y generas uno nuevo. Pero mientras tanto, quien tenga el token tiene acceso total a tu bot. Trátalo como tratarías la contraseña de tu banco.

Configurar el token en Hermes

Si estás usando la app de escritorio, ve a la sección «Gateway» o «Telegram» y pega el token donde dice «Bot Token». La app lo guarda de forma segura.

Si estás en la terminal:

```
cd ~/hermes-30min
hermes config set telegram.bot_token "tu-token-aqui"
```

Comprueba que Telegram reconoce el bot:

```
curl -s "https://api.telegram.org/botTU-TOKEN/getMe" | jq .
```

Deberías ver algo como:

```
{
  "ok": true,
  "result": {
    "id": 1234567890,
    "is_bot": true,
    "first_name": "María Asistente",
    "username": "maria_asistente_bot"
  }
}
```

Si `ok` es `false`, el token está mal copiado. Los tres fallos más comunes: falta un carácter al copiar, hay un espacio al principio o al final, o has pegado el token de otro bot. Vuelve a BotFather, escribe `/mybots`, selecciona tu bot, ve a «API Token» y copia de nuevo.

Obtén tu chat_id y autorízate

Para que solo tú puedas hablar con tu bot, necesitas tu `chat_id` — un número que identifica tu cuenta de Telegram de forma única.

Abre el bot en Telegram (busca su username, el que acaba en `bot`) y pulsa «Iniciar» o escribe `/start`.

Luego, desde la terminal, pide las actualizaciones del bot. Tu mensaje de `/start` aparecerá ahí junto con tu `chat_id`:

```
curl -s "https://api.telegram.org/botTU-TOKEN/getUpdates" | jq  
'result[-1].message.chat.id'
```

Te devolverá un número. Ese es tu `chat_id`. Apréndetelo o guárdalo — lo necesitarás para autorizarte.

Configura Hermes para que solo acepte mensajes tuyos:

```
hermes config set telegram.allowed_user_ids "tu-chat-id"
```

Si en el futuro quieres autorizar a más personas, separa sus `chat_ids` por comas:

```
hermes config set telegram.allowed_user_ids "123456789,987654321"
```

Para esta primera instalación, autorízate solo a ti. Un bot privado no es privado porque el username sea difícil de adivinar. Es privado porque el gateway rechaza todo mensaje que no venga de un `chat_id` autorizado. La lista de usuarios permitidos es tu cortafuegos. No la dejes vacía. No añadas a nadie «por si acaso». Cada ID autorizado es una llave de acceso a tu agente.

8. Conecta el gateway y haz la primera prueba

El gateway es la puerta entre Telegram y Hermes Agent. Cuando escribes algo a tu bot en Telegram, el mensaje viaja por internet, llega a los servidores de Telegram, y el gateway lo recoge, comprueba que eres tú quien escribe, y se lo pasa a Hermes para que lo procese. La respuesta hace el camino inverso.

En la app de escritorio, el gateway se activa automáticamente al configurar Telegram. Busca un botón de «Iniciar Gateway» o «Conectar» y pulsa. Hecho.

Si estás en la terminal, primero configura el gateway:

```
cd ~/hermes-30min
hermes gateway setup
```

Cuando pregunte el modo, usa `polling`. Es el más simple para una máquina personal. El gateway preguntará a Telegram cada pocos segundos si hay mensajes nuevos. Para un solo bot en un ordenador personal, esto no supone ningún problema.

Usa polling para esta guía. El modo webhook es mejor para servidores, pero exige un dominio, HTTPS y configuración adicional. Polling funciona en cualquier ordenador sin configuración extra. Cuando tengas tu agente en un servidor 24/7, ya explorarás webhook.

Arranca el gateway:

```
hermes gateway run
```

Déjalo corriendo. Verás que la terminal se queda ocupada mostrando actividad. Eso es buena señal — significa que está escuchando. Abre otra terminal para los siguientes pasos.

Una pausa para la seguridad

Has llegado al punto donde tu bot está a punto de conectarse con Hermes. Antes de seguir, revisa tres cosas que la gente pasa por alto y luego lamenta:

1. **El token del bot es una contraseña.** Con él se puede leer todo lo que tu bot reciba y suplantarlo. Si lo has pegado en algún sitio que no sea la configuración de Hermes, bórralo ya.
2. **El gateway con polling sondea Telegram cada pocos segundos.** En un ordenador personal, con un solo bot, esto es irrelevante. En un servidor con cientos de bots, podría saturar la API de Telegram. No es tu caso ahora.
3. **Tu chat_id te identifica de forma única.** Con tu token y tu chat_id, cualquiera puede enviarte mensajes como si fuera tu bot. La lista de usuarios autorizados es tu única defensa. No la dejes vacía. No la rellenes con ceros ni asteriscos. Solo tu chat_id real.

La prueba de fuego

Con el gateway corriendo, abre Telegram en tu móvil. Busca a tu bot por su username. Escribe:

Responde en una sola frase: ¿estás operativo?

Si todo está bien configurado, en menos de diez segundos recibirás una respuesta. En ese momento, acabas de crear tu primer agente de IA funcional.

Si no recibes respuesta en veinte segundos, no te preocupes. Es normal que algo falle la primera vez. Mira la terminal donde corre el gateway. ¿Ves actividad? Si no ves nada, el mensaje no está llegando. Revisa el token — es la causa del 90% de los fallos en este punto.

Ahora prueba algo más útil. Escribe:

Crea una lista de 3 acciones para cerrar una jornada de trabajo sin perder decisiones pendientes.

La respuesta debería ser breve y práctica. Si responde con una clase magistral de productividad, no pasa nada — más adelante aprenderás a acotar el formato con skills.

Por último, verifica que el gateway rechaza a usuarios no autorizados. Si tienes una segunda cuenta de Telegram, pídele a alguien que abra tu bot e intente hablarle. El gateway debe rechazar el mensaje. En los registros verás algo como `rejected user_id=XXXX reason=not_allowed`. Si no tienes segunda cuenta, puedes saltarte esta comprobación, pero en una instalación real es una verificación importante.

Ya está. Tienes un agente que responde por Telegram. El resto de esta guía es para convertirlo de un chat inteligente en un sistema que trabaja solo.

9. Tus primeras tareas útiles

Tienes un agente. Ahora haz que trabaje. No le digas «hola, ¿cómo estás?». Dale trabajo de verdad. La diferencia entre un juguete y una herramienta no está en la tecnología — está en lo que le pides.

Aquí tienes diez tareas reales que tu agente puede hacer ahora mismo. Pruébalas. Son el tipo de cosas que convierten un chat en un empleado:

1. **«Busca en internet las 3 mejores formas de automatizar respuestas a clientes y hazme un resumen de 5 líneas.»** Tu agente busca, filtra, resume. Mientras tú sigues con lo tuyo.
2. **«Revísame este texto y dime si hay alguna errata o algo que suene raro: [pegas tu texto].»** Corrección ortográfica y de estilo en segundos.
3. **«Estoy preparando una reunión con un cliente del sector inmobiliario. Dame 5 preguntas que debería hacerle para entender sus necesidades.»** Preparación de reuniones bajo demanda.
4. **«Tengo estos 10 correos atrasados. Dame un borrador de respuesta para cada uno: [pegas los correos].»** Respuesta masiva a correos pendientes.
5. **«¿Cuál es la diferencia entre un agente de IA y un chatbot? Explícamelo como si tuviera 12 años.»** Explicaciones a medida de la audiencia.
6. **«Tengo este contrato de 5 páginas. Léelo y dime si hay alguna cláusula que me pueda perjudicar como proveedor de servicios.»** Revisión de documentos legales.
7. **«Crea una tabla comparativa de estas 3 herramientas de gestión de proyectos: Notion, Trello y Asana. Incluye precio, ventajas y desventajas.»** Comparativas estructuradas.
8. **«Escribe un guion de 2 minutos para un vídeo de YouTube explicando qué es un agente de IA a alguien que no sabe nada de tecnología.»** Creación de contenido multimedia.
9. **«Tengo estos datos de ventas del último trimestre: [pegas los datos]. Analízalos y dime qué tendencias ves.»** Análisis de datos en lenguaje natural.

10. «**Traduce este email al inglés con un tono profesional pero cercano: [pegas el email].**» Traducción con control de tono.

La clave no es lo que le pides. Es que lo hace solo. Mientras él busca, resume, corrige, redacta o analiza, tú estás haciendo otra cosa. No tienes que copiar el resultado y pegarlo en otro sitio — él te lo entrega directamente en Telegram. Eso es un empleado. No un chat.

Una cosa importante: tu agente no es perfecto. A veces se equivoca. A veces se va por las ramas. A veces te da más información de la que necesitas. Es normal. Como cualquier empleado nuevo, necesita que le des instrucciones claras y que ajustes sobre la marcha. Con la práctica, aprenderás a formular peticiones que obtienen exactamente lo que necesitas.

Dos consejos para sacarle el máximo partido desde el primer día:

- **Sé específico.** «Hazme un resumen» no es lo mismo que «Hazme un resumen de 3 párrafos con viñetas, centrado en los aspectos financieros, y omite la sección de metodología».
- **Pide formato.** Si necesitas una tabla, pide una tabla. Si necesitas viñetas, pide viñetas. Si necesitas un email con asunto, saludo y despedida, pídelo. Tu agente te dará exactamente el formato que le pidas.

10. Tu primera automatización

Un agente no es un agente hasta que hace cosas sin que se las pidas. Esto es lo que separa un chatbot de un empleado. El chatbot responde cuando le hablas. El empleado se despierta a las 8:00, revisa tu bandeja de entrada, prepara un briefing y te lo envía antes de que hayas terminado el café.

En Hermes Agent, las tareas automáticas se llaman **cron jobs** — un nombre heredado del programador de tareas de Linux, `cron`. Suena técnico pero es muy simple: le dices a Hermes «haz esto todos los días a tal hora» y él obedece.

Vamos a crear tu primera automatización: un briefing diario que se ejecuta cada mañana de lunes a viernes a las 8:00.

Paso 1: Crea una entrada de prueba

El briefing necesita material de partida. Crea una nota en tu carpeta de entrada:

```
cat > ~/hermes-30min/inbox/nota-demo.md << 'EOF'
# Nota demo
Hoy hay que revisar tres cosas:
- Preparar una propuesta para un cliente.
- Ordenar los documentos recibidos por email.
- Decidir si el informe semanal sale el viernes o el lunes.
EOF
```

Paso 2: Crea la especificación del cron job

Define qué debe hacer el briefing, cuándo y con qué reglas:

```
cat > ~/hermes-30min/cron/briefing-diario.md << 'EOF'
# Cron: briefing-diario
Frecuencia: lunes a viernes, 08:00, zona horaria local.
Skill: briefing-diario
Entrada: ./inbox
Salida: ./outputs/briefings/YYYY-MM-DD.md
Log: ./logs/briefing-diario.log
```

Objetivo: Preparar un briefing operativo de inicio de día.

Reglas:

- No enviar mensajes a terceros.
- No borrar archivos.
- No mover archivos.
- No inventar datos que no estén en inbox.
- Si falta información, escribir "Pendiente de confirmar".

Formato de salida:

1. Estado rápido.
2. Decisiones pendientes.
3. Riesgos o bloqueos.
4. Primera acción recomendada.

EOF

Paso 3: Crea el skill que ejecuta el briefing

Un skill es una instrucción reutilizable. Lo definimos en el siguiente capítulo en detalle. Por ahora, vamos a probar el briefing de forma directa para verificar que todo funciona antes de automatizarlo:

En la app de escritorio, ve a «Cron Jobs» o «Tareas programadas». Crea una nueva tarea con:

- **Nombre:** Briefing diario
- **Horario:** Todos los días laborables a las 8:00
- **Instrucción:** «Lee la carpeta inbox, extrae tareas pendientes, decisiones por tomar y riesgos. Genera un briefing en formato Markdown con estado rápido, decisiones, bloqueos y primera acción recomendada. Guárdalo en outputs/briefings con la fecha de hoy. Envíamelo por Telegram.»

Guarda y activa. Para probarlo sin esperar a mañana, pulsa «Ejecutar ahora».

Si estás en la terminal, crea un script de ejecución. Esto te da una pieza estable que funciona independientemente de si estás delante o no:

```
cat > ~/hermes-30min/cron/run-briefing-diario.sh << 'EOF'
#!/usr/bin/env bash
set -euo pipefail
cd ~/hermes-30min
mkdir -p logs outputs/briefings
```

```
{
  echo "---- $(date -Is) briefing-diario start ----"
  hermes -z "Lee la carpeta inbox. Genera un briefing operativo con estado
rápido, decisiones pendientes, riesgos o bloqueos y primera acción
recomendada. Guarda el resultado en outputs/briefings/$(date +%F).md. No
borres ni nuevas archivos."
  echo "---- $(date -Is) briefing-diario ok ----"
} >> logs/briefing-diario.log 2>&1
EOF

chmod +x ~/hermes-30min/cron/run-briefing-diario.sh
```

Ejecútalo manualmente para probar:

```
~/hermes-30min/cron/run-briefing-diario.sh
```

Comprueba el resultado:

```
cat ~/hermes-30min/outputs/briefings/$(date +%F).md
```

Si ves un briefing con tus tareas pendientes, decisiones y riesgos, funciona. Si el archivo está vacío o no existe, no programes la automatización todavía. Un cron job que falla manualmente fallará igual a las 8:00 de la mañana, solo que tú no lo verás.

Paso 4: Programa la ejecución automática

En la app de escritorio, simplemente activa el cron job. Hermes se encarga del resto.

En la terminal, programa el cron del sistema:

```
(crontab -l 2>/dev/null; echo "0 8 * * 1-5 $HOME/hermes-30min/cron/run-
briefing-diario.sh") | crontab -
```

Comprueba que está programado:

```
crontab -l
```

Deberías ver una línea como:

```
0 8 * * 1-5 /home/tu_usuario/hermes-30min/cron/run-briefing-diario.sh
```

Mañana a las 8:00, sin que hagas nada, tu agente generará un briefing y te lo enviará por Telegram. Esto es lo que cambia todo. No es un chat. Es un sistema.

Consejo: no programes el briefing para una hora en la que tu ordenador esté apagado o suspendido. Si usas un portátil que se suspende al cerrar la tapa, programa el cron para una hora en la que sepas que estará encendido, o considera usar una máquina siempre encendida (una Raspberry Pi de 40 euros o un servidor virtual de 4 euros al mes).

11. Tu primer skill reutilizable

Un skill es una instrucción reutilizable. No es un prompt bonito. No es un mensaje que escribes cada día. Es un procedimiento que Hermes puede repetir una y otra vez sin que tú vuelvas a explicarle el formato, las reglas ni el objetivo.

La diferencia entre un prompt y un skill es la diferencia entre dar una orden verbal cada mañana y dejar un manual de procedimientos en el cajón del empleado nuevo. Con el manual, el empleado sabe qué hacer aunque tú no estés. Con el skill, tu agente también.

Vas a crear el skill `briefing-diario` — el que usará tu automatización de las 8:00.

Crear el skill

En la app de escritorio, ve a «Skills», pulsa «Nuevo Skill» y pega el siguiente contenido. En la terminal:

```
mkdir -p ~/hermes-30min/skills/briefing-diario

cat > ~/hermes-30min/skills/briefing-diario/SKILL.md << 'EOF'
---
name: briefing-diario
description: Prepara un briefing operativo de inicio de día usando la
carpeta inbox del workspace. Usa este skill cuando el usuario pida empezar
el día, revisar pendientes o ejecutar el cron briefing-diario.
---

# Skill: briefing-diario

## Entrada
Lee la carpeta `inbox/` del workspace.

## Salida
Escribe un archivo en `outputs/briefings/YYYY-MM-DD.md`.

## Procedimiento
1. Revisa todos los archivos disponibles en `inbox/`.
2. Extrae tareas explícitas.
3. Extrae decisiones pendientes.
```

```
4. Extrae bloqueos o riesgos.
5. Redacta un briefing corto y operativo.
6. Marca como `Pendiente de confirmar` cualquier dato que no esté claro.
7. No inventes reuniones, clientes, fechas ni prioridades.

## Formato exacto
# Briefing diario - YYYY-MM-DD

## Estado rápido

## Decisiones pendientes

## Riesgos o bloqueos

## Primera acción recomendada

## Notas sin confirmar

## Límites
- No enviar mensajes a terceros.
- No borrar archivos.
- No mover archivos.
- No modificar documentos fuente.
- No usar fuentes externas salvo que el usuario lo pida explícitamente.
EOF
```

Comprueba que el skill se ha creado:

```
hermes skills list
```

Deberías ver `briefing-diario` en la lista. Si no aparece, fuerza la recarga:

```
hermes skills reload
```

Probar el skill

Prueba el skill manualmente antes de confiar en que funcione solo a las 8:00:

```
hermes -z "Usa el skill briefing-diario."
```

Comprueba la salida generada:

```
cat ~/hermes-30min/outputs/briefings/$(date +%F).md
```

Deberías ver un briefing con el formato exacto que definiste. Si la salida no respeta el formato, no ajustes el prompt del día — ajusta el skill. Esta es una de esas diferencias pequeñas que cambian la semana: si corriges el prompt cada mañana, sigues trabajando tú. Si corriges el skill una vez, trabaja el sistema.

Probar desde Telegram

Con el gateway corriendo, abre Telegram y envía a tu bot:

```
/run briefing-diario
```

Tu agente ejecutará el skill y te devolverá el briefing directamente en el chat. Desde el móvil. Sin tocar el ordenador.

Añadir más material de entrada

Un agente sin entradas trabaja con aire. Y cuando trabaja con aire, rellena huecos con información inventada. No le des aire: dale notas reales.

Crea una segunda nota para verificar que el skill procesa múltiples fuentes:

```
cat > ~/hermes-30min/inbox/nota-cliente.md << 'EOF'
# Nota cliente
Cliente: Norte Studio
Pendiente:
- Enviar propuesta revisada.
- Confirmar si quieren entrega el jueves o el lunes.
- Revisar presupuesto antes de contestar.
Bloqueo:
- Falta el archivo con métricas de abril.
EOF
```

Ejecuta el skill de nuevo y comprueba que el briefing ahora incluye toda la información de ambas notas. Si falta algo, el skill necesita ser más explícito en su procedimiento de lectura.

Un skill bien diseñado es aburrido de tan fiable. Esa es exactamente la sensación que buscas. Si el briefing aparece cada mañana sin sorpresas, el skill está maduro.

12. La primera semana: convierte el sistema en hábito

La instalación termina cuando funciona. El hábito empieza al día siguiente.

Durante siete días, no añadas veinte integraciones. No conectes Gmail, Calendar, Drive, Slack, Notion y medio internet el primer día. La tentación es enorme, lo sabemos. Pero casi siempre acaba igual: mucho permiso, poca utilidad y un lío de registros que nadie lee. Usa este sistema pequeño durante una semana y mira dónde te ahorra trabajo real.

El ritual diario (3 minutos)

Cada mañana: lee el briefing que tu agente ha generado. Si algo falta, añade una nota en inbox para que el briefing de mañana lo incluya. Si sobra información irrelevante, el skill necesita un ajuste.

```
cat ~/hermes-30min/outputs/briefings/$(date +%F).md
```

Cada tarde: vacía tu cabeza en una nota de cierre. Lo que no escribas, el briefing de mañana no lo verá. Tu agente no es adivino — solo procesa lo que le das.

```
cat > ~/hermes-30min/inbox/cierre-$(date +%F).md << 'EOF'
# Cierre de jornada
Terminado:
Pendiente:
Decisiones:
Bloqueos:
EOF
```

Edita ese archivo con lo que haya quedado abierto. Tres minutos. No más.

El ritual semanal (10 minutos, el viernes)

No necesitas un dashboard. No necesitas métricas de rendimiento. Necesitas un ritual de diez minutos el viernes:

- Revisa los registros de actividad de la semana. ¿Hubo errores? ¿Se ejecutaron todos los briefings?
- Mira cuántos briefings se generaron realmente. Deberían ser uno por cada día laborable.
- Comprueba que los skills devuelven lo que esperas. Si algo falló de forma repetida, ajusta el skill, no el día.
- Borra las notas de inbox de la semana anterior que ya estén procesadas. Inbox no es un archivo — es una bandeja de entrada.

No dediques más tiempo. Si le dedicas media hora al día a «gestionar el agente», el agente no te está ahorrando trabajo: te está generando otro. El objetivo es que el sistema trabaje para ti, no tú para el sistema.

El segundo skill: cuándo y cómo

Cuando el briefing lleve tres días funcionando sin incidencias, crea tu segundo skill. No antes. Un sistema inestable con dos skills es el doble de difícil de depurar que uno estable con uno solo.

El segundo skill suele salir solo. Observa tu semana. ¿Qué tarea repetitiva te quitó más tiempo? ¿Preparar resúmenes de reuniones? ¿Redactar propuestas comerciales? ¿Clasificar documentos? Elige el dolor que más se repite y conviértelo en skill.

Aquí tienes una plantilla para un segundo skill típico — resumen de reuniones:

```
mkdir -p ~/hermes-30min/skills/resumen-reunion

cat > ~/hermes-30min/skills/resumen-reunion/SKILL.md << 'EOF'
---
name: resumen-reunion
description: Convierte notas sueltas de una reunión en resumen operativo
con decisiones, responsables y siguientes pasos.
---
```

```
# Skill: resumen-reunion

## Entrada
Notas pegadas por el usuario o archivo indicado en inbox.

## Salida
Resumen en Markdown con decisiones y siguientes pasos.

## Procedimiento
1. Separar hechos, opiniones y decisiones.
2. Detectar responsables explícitos.
3. Detectar fechas explícitas.
4. Marcar lo ambiguo como pendiente de confirmar.
5. Cerrar con próximos pasos.

## Formato
# Resumen de reunión

## Decisiones

## Responsables

## Próximos pasos

## Pendiente de confirmar

## Límites
- No inventar responsables.
- No inventar fechas.
- No enviar el resumen a nadie.
EOF
```

Esto es operar un sistema agéntico: detectar repetición, convertirla en skill, probar manualmente, y luego usarlo desde Telegram si tiene sentido.

La cadencia que funciona

Con dos skills probados y un cron funcionando, ya tienes un sistema agéntico mínimo. No «una IA que ayuda cuando le pides cosas». Un sistema que ejecuta sin que tú estés delante. La diferencia es sutil en la definición pero enorme en la práctica: cuando suena el despertador a las 8:00 y tú aún estás desayunando, el briefing ya está en tu carpeta. Eso no es ayuda. Eso es delegación.

Si quieres seguir expandiendo, el orden recomendado es:

1. **Skill de propuesta comercial:** dale un briefing de cliente y que devuelva estructura, tono y siguientes pasos.
2. **Skill de limpieza de inbox:** que archive notas procesadas, detecte duplicados y te avise de lo que lleva más de 48 horas sin tocar.
3. **Cron de cierre de jornada:** que lea lo generado durante el día y te mande un resumen por Telegram a las 18:00.
4. **Gateway adicional:** si tu flujo lo pide, un gateway de email. Pero solo cuando los tres primeros skills lleven al menos una semana sin incidencias.

No añadas el cuarto hasta que el tercero sea aburrido de tan fiable. El aburrimiento es la señal de que una pieza está madura.

13. Caso real: María y su bandeja de correo

Las automatizaciones que estás montando no son teóricas. Detrás de cada skill, cada cron job y cada briefing hay personas reales que ya han dado este paso. Sus historias no son casos de estudio de laboratorio — son lo que pasa cuando alguien con un problema concreto aplica esta guía.

María es gestora de proyectos en una agencia de marketing. Cada mañana se encontraba entre 40 y 60 correos electrónicos. Clientes preguntando plazos de entrega. Proveedores mandando facturas. El equipo pidiendo revisiones de campañas. Su bandeja de entrada era un segundo trabajo no remunerado.

Tardaba 90 minutos diarios **solo en triar** el correo. No en responder. Solo en leer cada mensaje, decidir si era urgente, clasificarlo en la carpeta correcta y priorizar. Noventa minutos que no dedicaba a gestionar proyectos — que es para lo que la contrataron.

El día que instaló Hermes Agent, María configuró tres reglas simples:

1. **Regla de facturación:** si el correo contiene las palabras «factura» o «presupuesto», el agente lo mueve automáticamente a la carpeta «Facturación» y le notifica por Telegram: «Nueva factura recibida de [nombre del remitente]».
2. **Regla de urgencia:** si el correo es de un cliente y contiene palabras como «urgente», «plazo» o «retraso», el agente lo deja en la bandeja principal y le notifica por Telegram con prioridad alta: «URGENTE: [cliente] pregunta por [asunto]. Revisa el correo #X».
3. **Regla de archivo:** todo lo demás — newsletters, notificaciones de herramientas, copias de conversaciones internas — el agente lo archiva en la carpeta «Para revisar» y le envía un resumen diario a las 9:30 con una línea por cada correo recibido.

La configuración le llevó 25 minutos. El agente se ejecuta cada 15 minutos mediante un cron job. No necesita intervención humana.

El resultado: María ahora revisa entre 8 y 10 correos importantes al día en vez de 50. Los 90 minutos de triaje se convirtieron en 10. Y lo más importante: los correos urgentes no se le escapan, porque el agente le avisa por Telegram en el momento en que llegan, esté donde esté.

En sus propias palabras: «Lo mejor no es el tiempo que ahorro. Es que ya no llego a las 10 de la mañana con la cabeza quemada de leer correos. Antes, a las 10 ya estaba agotada mentalmente y aún no había hecho nada de valor. Ahora llego fresca porque el agente ya me ha dicho qué es importante y qué puede esperar».

El coste de esta automatización: entre 2 y 3 euros al mes en crédito de DeepSeek. Menos que un café.

14. Caso real: Carlos y sus leads de Instagram

Carlos vende formaciones online. Cursos grabados sobre marketing digital, con un precio medio de 200 euros por curso. Recibe entre 15 y 20 mensajes directos al día en Instagram de personas interesadas.

El patrón era siempre el mismo. Cuatro preguntas repetidas una y otra vez: ¿cuánto cuesta?, ¿cuánto dura?, ¿qué temario tiene?, ¿dan certificado al terminar? Preguntas legítimas y predecibles. Pero responderlas le robaba dos horas diarias. Y aquí está el problema de fondo: si tardaba más de cuatro horas en responder, el lead se enfriaba y la venta se perdía.

Carlos no necesitaba un chatbot que hablara con los clientes. Necesitaba un sistema que detectara preguntas frecuentes y respondiera al instante, y que solo le escalara a él las conversaciones que de verdad requerían intervención humana.

Montó un agente Hermes conectado a la API de Instagram a través de Meta for Developers. El flujo es elegante en su simplicidad:

1. **Detección:** el agente lee los mensajes directos nuevos cada 30 minutos.
2. **Respuesta automática:** si detecta una pregunta sobre precio, duración, temario o certificado, responde automáticamente con un mensaje predefinido que incluye el enlace de pago y los detalles del curso.
3. **Escalado:** si la pregunta es más compleja — algo como «¿este curso me sirve si ya tengo experiencia en marketing?» — el agente reenvía el mensaje a Carlos por Telegram con un resumen de una línea: «Lead pregunta si el curso avanzado le sirve teniendo experiencia previa».
4. **Detección de intención de compra:** si el lead responde al mensaje automático con palabras como «comprar», «pagar» o «cómo me inscribo», el agente envía el enlace de pago y notifica a Carlos: «Lead caliente. Ha preguntado por el curso de marketing avanzado. Ya tiene el enlace de pago. Haz seguimiento en 24 horas si no compra».

Configuración total: 40 minutos siguiendo esta guía. Coste mensual de API: entre 3 y 4 euros para este volumen de mensajes.

El resultado: Carlos convirtió 3 leads en clientes la primera semana que el agente estuvo activo. Eran ventas que habría perdido porque esos mensajes llegaron un sábado por la tarde, cuando él estaba con su familia y no miró Instagram hasta el lunes. Para entonces, esos leads ya habían preguntado a otros tres formadores y se habían decidido por el que respondió primero.

La lección de Carlos no es sobre tecnología. Es sobre velocidad de respuesta. En ventas, el que responde primero — aunque sea un agente automático con respuestas predefinidas — gana. El agente no reemplazó a Carlos. Le compró tiempo para que él pudiera dedicarse a lo que de verdad aporta valor: las conversaciones complejas, las objeciones personalizadas, el cierre de ventas de alto valor.

15. Caso real: Ana y sus informes semanales

Ana es analista de datos en una consultora. Cada viernes prepara un informe de rendimiento para cuatro clientes distintos. Mismas métricas, misma estructura, distintos datos. Cuatro horas cada viernes. Dieciséis horas al mes. Dos jornadas laborales completas dedicadas a generar informes que seguían una plantilla.

El proceso manual era una cadena de tareas mecánicas: exportar datos de Google Analytics a CSV, copiarlos a Google Sheets, crear gráficos, escribir conclusiones, maquetar en un Google Doc, adjuntar al email de envío. Ana no estaba analizando datos — estaba copiando y pegando. Su formación en estadística y su máster en data science no se estaban usando. Se estaba usando su capacidad de hacer Ctrl+C y Ctrl+V durante cuatro horas cada viernes.

Ana configuró un agente Hermes que ejecuta este flujo cada viernes a las 8:00 de la mañana:

1. **Extracción:** el agente se conecta a la API de Google Analytics y extrae los datos de cada cliente — sesiones, conversiones, tasa de rebote, tiempo medio en página, tráfico por canal, páginas más visitadas, dispositivos y ubicación geográfica.
2. **Cálculo automático:** calcula las 8 métricas clave para cada cliente y las compara con la semana anterior, obteniendo variaciones porcentuales. Si una métrica ha variado más del 20%, lo marca como anomalía.
3. **Generación de informe:** crea un Google Doc con la tabla de métricas, tres párrafos de análisis automático describiendo tendencias, y dos gráficos generados a partir de los datos.
4. **Entrega:** a las 8:45, el agente envía el borrador a Ana por email. Ella dedica 45 minutos a revisar, ajustar conclusiones estratégicas y personalizar el mensaje para cada cliente antes de reenviarlo.

El resultado: el agente no reemplaza a Ana. Le ahorra las tres horas más mecánicas del proceso. Las horas de Ctrl+C y Ctrl+V. Ella ahora dedica 45

minutos por cliente a lo que de verdad aporta valor: interpretar los datos con criterio estratégico, añadir recomendaciones personalizadas y contextualizar las cifras para cada negocio.

La configuración inicial le llevó dos horas — tuvo que familiarizarse con las APIs de Google Analytics y Google Docs. Pero una vez montado, el agente lleva 14 semanas funcionando sin un solo fallo. Catorce viernes seguidos. Cincuenta y seis informes generados automáticamente.

Sus palabras lo resumen mejor que cualquier análisis: «Antes llegaba al viernes con ansiedad. Sabía que me esperaban cuatro horas de trabajo mecánico antes de poder empezar el fin de semana. Ahora llego el viernes y el borrador ya está hecho. Solo tengo que poner mi criterio donde de verdad aporte valor. El agente hace lo predecible. Yo hago lo estratégico. Esa es la división del trabajo que siempre debí tener».

El coste: unos 5 euros al mes en crédito de API. Por 5 euros al mes, Ana recuperó 16 horas de su vida cada mes. Si gana 25 euros la hora, su agente se paga solo en los primeros 12 minutos de cada mes.

16. Checklist: ¿está vivo tu agente?

Que el sistema diga «running» no significa que tu agente esté trabajando. Tu agente puede estar técnicamente vivo y funcionalmente muerto — como un coche con el motor encendido pero sin ruedas. Esta checklist te dice si de verdad está produciendo valor.

Responde SÍ o NO a cada punto. Sé honesto. Si marcas SÍ a algo que no has comprobado esta semana, no cuenta.

1. **¿Responde en menos de 10 segundos?** Si tarda más, algo está consumiendo tokens de más o tu conexión con el proveedor de IA tiene latencia. Un agente lento se abandona. La velocidad es funcionalidad.
2. **¿Ejecuta al menos una acción autónoma al día?** Sin que tú le des instrucciones. Si solo responde cuando le hablas, no es un agente — es un chatbot. La acción autónoma mínima es el briefing diario.
3. **¿Te ha notificado algo útil esta semana?** Una alerta, un resumen, un aviso que no esperabas. Si no te ha notificado nada, probablemente no está mirando lo que debería mirar.
4. **¿Su respuesta incluye solo las herramientas que necesita?** Si le pides leer un archivo y te habla de generar imágenes, tienes herramientas de más cargadas. Las herramientas que no usas ocupan tokens y aumentan la probabilidad de errores.
5. **¿Sobrevivió al último update de Hermes Agent?** Si no lo has actualizado desde que lo instalaste, hazlo en un entorno de prueba antes de actualizar en producción. O simplemente no actualices si todo funciona — no arregles lo que no está roto.
6. **¿Tiene al menos una tarea programada?** Sin tareas periódicas, tu agente solo reacciona. Un agente de verdad actúa por su cuenta. El briefing diario cuenta.
7. **¿Sus respuestas son consistentes entre sesiones?** Hazle la misma pregunta dos veces en sesiones distintas. Si la respuesta cambia radicalmente, o no tiene memoria activada o está generando contenido aleatorio.
8. **¿Gasta menos de 0,05 euros al día en tokens?** Mira tu consumo en el panel de tu proveedor de API. Si gasta más de lo razonable para lo que

hace, tienes el prompt del sistema inflado con herramientas y reglas innecesarias.

9. **¿Los registros no muestran errores recurrentes?** Si ves el mismo error cinco veces esta semana, arréglalo hoy. Mañana serán diez. Pasado mañana habrás normalizado convivir con un sistema roto.
10. **¿Hace algo que tú no harías?** La pregunta definitiva. Si tu agente solo hace lo mismo que haces tú pero más lento, no te está sirviendo. Tiene que hacer cosas que tú no harías: revisar sistemas a las 3 de la mañana, comparar 200 precios en segundos, detectar un patrón en 500 correos, monitorizar tu sitio web 144 veces al día.

Resultado

- **9-10 SÍ:** Tu agente está vivo, trabajando y probablemente ya te da más valor del que cuesta. Estás en el 5% de usuarios que han convertido su asistente de IA en un sistema productivo.
- **6-8 SÍ:** Funciona pero está a medio gas. Revisa los NO y corrige al menos uno esta semana. No intentes arreglarlos todos a la vez — elige el que más tiempo te roba.
- **Menos de 6 SÍ:** Tienes un chatbot con esteroides. Vuelve al capítulo 5 y verifica que la instalación base funciona. Luego reduce herramientas al mínimo — solo las que uses de verdad. Un agente con 15 herramientas y ningún skill es un loro muy caro.

Imprime esta checklist. Pégala al lado de tu monitor. Pásala cada viernes durante el primer mes. Luego la pasarás de forma natural, sin necesidad del papel. Tu agente se habrá convertido en parte de tu forma de trabajar.

17. Troubleshooting: 5 problemas y sus soluciones

Algo va a fallar. No es pesimismo: es estadística. En cualquier sistema con múltiples piezas — instalación, proveedor de IA, API de Telegram, cron del sistema, skills — alguna pieza acabará fallando. La diferencia entre abandonar y arreglarlo en cinco minutos está en saber dónde mirar.

Aquí están los cinco problemas más comunes, sus causas y la secuencia exacta de diagnóstico para cada uno. No saltes pasos. El orden importa.

Problema 1: «hermes: command not found»

Síntoma: Acabas de instalar Hermes Agent. Abres la terminal, escribes `hermes`, y el sistema te dice que el comando no existe.

Causa: El directorio de instalación no está en tu PATH — la lista de carpetas donde el sistema busca comandos. Hermes se instala en `~/.local/bin` o en `~/.hermes/bin` según tu sistema, y a veces el instalador no lo añade automáticamente a la ruta de búsqueda.

Solución (30 segundos):

1. Ejecuta: `export PATH="$HOME/.local/bin:$PATH"`
2. Prueba: `hermes --version`
3. Si responde con un número de versión, el problema era solo el PATH.
4. Para que no vuelva a pasar al cerrar la terminal, añade esa línea al final de tu archivo `~/.bashrc` (si usas bash) o `~/.zshrc` (si usas zsh).

Si sigues sin verlo, reinstala con `curl -fsSL https://install.hermes-agent.com | bash` y fíjate en la última línea del output de instalación — te dice exactamente dónde se instaló el binario. Copia esa ruta al PATH.

Problema 2: «El agente se queda pensando y no responde»

Síntoma: Escribes tu primer mensaje en Telegram. El cursor parpadea. Pasan diez segundos. Veinte. Treinta. Silencio absoluto.

Causas posibles y soluciones en orden:

Causa A — API key mal configurada. La clave API de DeepSeek, OpenRouter o Anthropic está mal copiada o tiene espacios extra. Un solo espacio al final de la clave hace que el proveedor rechace la autenticación y Hermes se quede esperando una respuesta que nunca llega. Solución: vuelve al panel de tu proveedor, copia la clave de nuevo, pégala en la configuración de Hermes sin espacios adicionales. Comprueba con `hermes -z "Responde: ok"` desde la terminal — si falla en terminal, fallará en Telegram.

Causa B — Modelo no disponible. El nombre del modelo que configuraste no existe en tu proveedor o no está disponible para tu plan. Solución: usa `deepseek-chat` en DeepSeek o `openai/gpt-4o` en OpenRouter. Son los más estables en junio de 2026. Configúralo con `hermes config set model.name "deepseek-chat"`.

Causa C — Límite de tasa de peticiones. Los planes gratuitos tienen límites de peticiones por minuto. Si acabas de hacer cinco pruebas seguidas, el proveedor te está frenando. Solución: espera 60 segundos y vuelve a intentarlo. Si el problema persiste, revisa los límites de tu plan en el panel del proveedor.

Prueba rápida de diagnóstico: ejecuta `hermes -z "Responde: test ok"`. Si falla, el problema está entre Hermes y el proveedor. Si funciona, el problema está en la conexión con Telegram (ve al Problema 4).

Problema 3: «El agente alucina herramientas que no existen»

Síntoma: Le pides que lea un archivo CSV y te responde: «Claro, déjame usar la herramienta `image_generator` para analizar ese archivo». O le pides que busque algo en internet y te dice que va a usar un comando que no existe.

Causa: Tienes demasiadas herramientas cargadas. El modelo de IA ve 15 o 20 herramientas disponibles y, ante la ambigüedad, elige una incorrecta. Especialmente si estás usando la configuración completa de Hermes Agent con todos los toolsets activados.

Solución inmediata: reduce las herramientas al mínimo necesario. Si solo necesitas leer archivos y ejecutar comandos, no cargues herramientas de navegador web, generación de imágenes ni voz. Edita tu configuración:

```
hermes config set toolsets terminal,file
```

Dos herramientas. Punto. El modelo deja de imaginar herramientas inexistentes y se centra en las que realmente tiene. Las alucinaciones bajan un 80% solo con este cambio.

Si más adelante necesitas búsqueda web o acceso a internet, añade esa herramienta específica. Pero empieza minimalista. Cada herramienta que no usas ocupa tokens y añade ruido al proceso de decisión del modelo.

Problema 4: «El bot de Telegram no responde»

Síntoma: Escribes al bot en Telegram y no recibes respuesta. O recibes «no autorizado».

Diagnóstico en orden estricto:

1. **Comprueba el token.** Ejecuta `curl -s "https://api.telegram.org/botTU-TOKEN/getMe"`. Si devuelve `ok: false`, el token está mal. Vuelve a BotFather, `/mybots`, selecciona tu bot, API Token, copia de nuevo sin espacios al principio ni al final.

2. **Comprueba que el bot recibe mensajes.** Abre Telegram, escribe `/start` a tu bot. Luego ejecuta `curl -s "https://api.telegram.org/botTU-TOKEN/getUpdates"`. Si el array `result` está vacío, tu mensaje no está llegando. ¿Escribiste al bot correcto? ¿Tiene el username que crees?
3. **Comprueba que el gateway está corriendo.** En la terminal donde arrancaste el gateway, ¿sigues viendo actividad? Si no, arráncalo de nuevo con `hermes gateway run`.
4. **Comprueba que no hay dos gateways corriendo.** Dos instancias del gateway haciendo polling sobre el mismo bot se pisan mutuamente. Solo debe haber una.
5. **Comprueba el chat_id autorizado.** Si el gateway rechaza tus mensajes con «not_allowed», tu chat_id no está en la lista de usuarios permitidos. Ejecuta `curl -s "https://api.telegram.org/botTU-TOKEN/getUpdates" | jq '.result[-1].message.chat.id'`, copia el número y configúralo con `hermes config set telegram.allowed_user_ids "TU-CHAT-ID"`. No uses el ID del bot — el bot tiene su propio ID. Necesitas el ID de tu chat personal.

Error muy común: copiar el ID del bot (que aparece en `getMe`) en lugar del ID de tu chat personal (que aparece en `getUpdates`). Son números distintos. Hermes necesita autorizar tu chat, no el bot.

Problema 5: «La automatización no se ejecuta a la hora esperada»

Síntoma: El briefing funciona perfectamente cuando lo ejecutas a mano, pero a las 8:00 de la mañana no aparece nada en la carpeta de salidas.

Causas posibles y soluciones en orden:

1. **Zona horaria incorrecta.** Si tu sistema está configurado en UTC y tú vives en Madrid (UTC+2 en verano), las 8:00 del cron son las 10:00 de tu hora local. Solución: ajusta la zona horaria del sistema o programa la hora equivalente en UTC. Comprueba con `date` y `timedatectl`.
2. **El cron no está programado.** Ejecuta `crontab -l`. Si no ves la línea del briefing, el cron job no se añadió correctamente. Vuelve a añadirlo.

3. **El script no tiene permisos de ejecución.** Comprueba con `ls -la ~/hermes-30min/cron/run-briefing-diario.sh`. Si falta la `x` de ejecución, añádela: `chmod +x ~/hermes-30min/cron/run-briefing-diario.sh`.
4. **El comando hermes no está en el PATH del cron.** El cron se ejecuta con un PATH mínimo. Si el script llama a `hermes` sin la ruta completa, cron no lo encuentra. Solución: usa la ruta absoluta al binario. Encuéntrala con `which hermes` y edita el script para usar `/ruta/completa/hermes` en lugar de solo `hermes`.
5. **Variables de entorno no disponibles.** Cron no carga tu `.bashrc` ni `.zshrc`. Si el script depende de variables de entorno que solo existen en tu shell interactivo, fallará. Asegúrate de que el script carga todas las variables que necesita desde su propio código o desde un archivo `.env`.
6. **El ordenador está suspendido o sin internet.** Si tu portátil se suspende al cerrar la tapa o se desconecta del WiFi durante la noche, el cron se ejecutará pero Hermes no podrá llamar al proveedor de IA. Programa la automatización para una hora en la que el ordenador esté encendido y conectado, o usa una máquina que no suspenda.

Regla de oro: si el script funciona al ejecutarlo manualmente con la ruta absoluta pero no desde el cron, el problema es el entorno de ejecución del cron (PATH, variables o permisos). No es un bug de Hermes — es una diferencia entre tu shell interactivo y el entorno mínimo del cron.

Problemas adicionales que aparecen con el tiempo

«Hermes se actualizó y mi agente no funciona»

Actualizar Hermes Agent puede cambiar nombres de herramientas, firmas de skills o la estructura de configuración. El kit de supervivencia:

1. No toques tu configuración todavía. Haz copia de seguridad primero.
2. Prueba con un agente mínimo — solo `terminal` y `file` como herramientas. Si funciona, el problema está en alguna herramienta o skill específico. Ve añadiéndolos uno a uno hasta encontrar al culpable.
3. Si todo falla, haz rollback a la versión anterior con `hermes rollback`.

Regla de oro: no actualices Hermes un lunes a las 9 de la mañana. Hazlo un viernes por la tarde o cuando tengas margen para arreglar cosas.

«El agente no recuerda nada entre sesiones»

Sin memoria persistente, cada conversación empieza de cero. Hermes Agent tiene un sistema de memoria que se activa por configuración:

```
hermes config set memory.enabled true
hermes config set memory.backend local
hermes config set memory.path ~/.hermes/memory/
```

Esto activa un sistema de archivos donde tu agente guarda información entre sesiones. No es magia — son archivos de texto que el agente lee al iniciar y actualiza al terminar. Pero funciona. Para un sistema más avanzado, la guía 05 del catálogo cubre el sistema de memoria basado en el paper de Stanford.

18. 3 ejercicios para hacer hoy mismo

Leer esta guía no monta un agente. Hacer los ejercicios sí. Aquí tienes tres tareas concretas, con tiempo estimado y pasos detallados. Hazlas hoy. No mañana. Mañana siempre llega con otra urgencia.

Ejercicio 1: Tu primera automatización real (15 minutos)

Objetivo: Que tu agente haga algo útil sin que tú se lo pidas.

Qué vas a crear: Un monitor de disponibilidad web. Tu agente comprobará si tu sitio web está online y te avisará por Telegram si se cae.

Paso a paso:

1. Crea un skill nuevo para la monitorización:

```
mkdir -p ~/hermes-30min/skills/monitor-web

cat > ~/hermes-30min/skills/monitor-web/SKILL.md << 'EOF'
---
name: monitor-web
description: Monitor de disponibilidad web. Comprueba si un sitio responde
y notifica si está caído.
---

# Skill: monitor-web

## Procedimiento
1. Ejecuta: curl -s -o /dev/null -w "%{http_code}" https://TUDOMINIO.COM
2. Si el código NO es 200, envía un mensaje por Telegram al usuario:
   "ALERTA: TUDOMINIO.COM no responde. Código: [código]"
3. Si el código es 200, no hagas nada. El sitio funciona.
EOF
```

- Crea un script que ejecute el skill:

```
cat > ~/hermes-30min/cron/run-monitor-web.sh << 'EOF'
#!/usr/bin/env bash
cd ~/hermes-30min
mkdir -p logs
{
  echo "---- $(date -Is) monitor-web start ----"
  hermes -z "Usa el skill monitor-web."
  echo "---- $(date -Is) monitor-web ok ----"
} >> logs/monitor-web.log 2>&1
EOF

chmod +x ~/hermes-30min/cron/run-monitor-web.sh
```

- Prográmalo cada 10 minutos:

```
(crontab -l 2>/dev/null; echo "*/10 * * * * $HOME/hermes-30min/cron/run-monitor-web.sh") | crontab -
```

- Para probarlo, cambia temporalmente la URL del skill a un sitio que no exista (como `https://estesitioonoexiste12345.com`) y ejecuta manualmente. Deberías recibir una alerta por Telegram en menos de 10 minutos.

Ya está. Tu primera automatización real. En 15 minutos has creado algo que monitoriza tu web 144 veces al día sin que muevas un dedo. Si tu sitio se cae a las 3 de la mañana, te enteras en minutos, no a la mañana siguiente cuando un cliente te llama enfadado.

Ejercicio 2: Limpia tu configuración y ahorra tokens (10 minutos)

Objetivo: Reducir el consumo de tokens al mínimo necesario. Menos tokens = menos coste = menos latencia = menos alucinaciones.

Qué vas a hacer: Una auditoría rápida de herramientas.

Mira tu configuración actual de herramientas. ¿Cuántas tienes activas? Ahora pregúntate: ¿cuántas has usado de verdad esta semana?

Si tienes 15 herramientas activas y solo usas 3, estás quemando tokens en cada mensaje. Cada herramienta que no usas ocupa entre 100 y 300 tokens

en el prompt del sistema. En 20 mensajes diarios, son entre 2.000 y 6.000 tokens tirados a la basura. Todos los días. Todos los meses.

La acción: desactiva todas las herramientas que no hayas usado esta semana. Deja solo `terminal` y `file` para empezar. Si dentro de un mes necesitas otra, la añades en diez segundos.

```
hermes config set toolsets terminal,file
```

Comprueba el consumo de tokens antes y después con `hermes info --tokens`. La diferencia te va a doler. En el buen sentido.

Ejercicio 3: Enséñale tu voz (20 minutos)

Objetivo: Que tu agente escriba como tú, no como un robot.

Este es el ejercicio que más impacto tiene a largo plazo. Un agente que suena a ti multiplica tu presencia sin que tú estés. Tus clientes, tu equipo, tus proveedores — nadie nota la diferencia. Ese es el punto.

Qué vas a hacer: Darle a tu agente una muestra de tu estilo de comunicación.

1. **Recopila muestras reales.** Coge tres correos que hayas escrito tú. De verdad. No plantillas, no respuestas copiadas de internet. Tres correos tuyos, con tu tono, tus muletillas, tu forma de estructurar.
2. **Crea un skill de personalidad:**

```
mkdir -p ~/hermes-30min/skills/mi-voz

cat > ~/hermes-30min/skills/mi-voz/SKILL.md << 'EOF'
---
name: mi-voz
description: Estilo de comunicación personal. Usa este skill cuando quieras
que el agente escriba con tu tono y estilo.
---

# Skill: mi-voz

## Estilo de comunicación del usuario

Tono: [describe tu tono: directo, cercano, técnico, irónico, formal...]
```

Frases que usa habitualmente:

- [frase literal tuya]
- [frase literal tuya]
- [frase literal tuya]

Lo que NUNCA dice:

- [palabra o expresión que odias]
- [palabra o expresión que odias]

Cuando escribas en nombre del usuario, usa exactamente este estilo. Sin excepciones.

EOF

- **Rellena los huecos con observaciones reales.** Si no sabes cómo describir tu estilo, pregúntale a un compañero: «¿cómo describirías mi forma de escribir en tres palabras?». Las respuestas te sorprenderán — y son más precisas que tu autopercepción.
- **Pruébalo.** Pídele a tu agente que use el skill `mi-voz` y redacte un correo de seguimiento a un cliente. Léelo. Si no suena a ti, ajusta el skill. Si suena a ti pero en versión robot, añade más ejemplos de frases reales tuyas.

Este ejercicio vale más que cualquier optimización técnica. La tecnología cambia cada seis meses. Tu voz no. Invertir 20 minutos en enseñarle a tu agente cómo sueñas es la mejor inversión de tiempo de toda esta guía.

19. Glosario para no perderte

Esto no es un diccionario. Es lo mínimo que necesitas para entender de qué habla la gente cuando habla de agentes de IA, y para no perderte cuando explores el resto del catálogo de Cosas Agénticas.

Agente: Un programa que usa un modelo de lenguaje — como DeepSeek, Claude o GPT — para tomar decisiones y ejecutar acciones por su cuenta. No es un chatbot. Un chatbot responde cuando le hablas. Un agente actúa aunque no le hables. Se despierta a una hora programada, revisa información, toma decisiones dentro de reglas definidas y te notifica el resultado.

Toolset: Conjunto de herramientas que tu agente puede usar. Terminal (ejecutar comandos), file (leer y escribir archivos), web (buscar en internet), browser (navegar páginas web), messaging (enviar mensajes por Telegram o email). Cuantas más herramientas cargas, más tokens gastas y más probabilidad hay de que el modelo alucine herramientas que no existen. El minimalismo aquí no es estética: es funcionalidad.

Skill: Instrucciones personalizadas que le das a tu agente para una tarea concreta. Un skill es un procedimiento reutilizable que define formato de entrada, formato de salida, reglas de comportamiento y límites. El briefing diario que has montado es un skill. El monitor de disponibilidad web es otro. Un skill bien diseñado se ejecuta igual la primera vez que la centésima.

System prompt: Las instrucciones ocultas que recibe el modelo antes de tu primer mensaje. Incluye los toolsets, los skills, las reglas de comportamiento y el contexto. Ocupa tokens — cada herramienta extra, cada regla adicional, cada línea de contexto infla el system prompt. El objetivo es que pese lo mínimo para que el modelo tenga lo necesario pero no se confunda con ruido.

Token: La unidad mínima que procesa un modelo de lenguaje. En español, una palabra corta son 1-2 tokens. Una palabra larga pueden ser 3-4 tokens. Pagas por tokens — cada mensaje que envías y cada respuesta que recibes consume tokens. Tu agente gasta tokens incluso antes de que escribas nada, porque el system prompt se carga en cada interacción.

Cron: El programador de tareas de los sistemas Unix (Linux, Mac). Le dices «ejecuta esto cada X minutos» o «todos los días a las 8:00» y lo hace. Sin

cron, tu agente solo actúa si tú le das al botón. Con cron, tu agente se despierta solo, ejecuta skills, te notifica por Telegram y vuelve a dormirse. El cron es lo que convierte un chatbot en un agente.

API key: La llave que te identifica ante el proveedor del modelo de IA — DeepSeek, Anthropic, OpenAI, OpenRouter. Sin ella, Hermes Agent no puede conectarse al cerebro que piensa por tu agente. Trátala como una contraseña. Si se filtra, revócala inmediatamente y genera una nueva.

Gateway: La puerta de entrada entre un servicio externo — Telegram, email, Slack — y Hermes Agent. El gateway de Telegram recibe tus mensajes, verifica que eres tú quien escribe, los pasa a Hermes y devuelve la respuesta. Es el intermediario que mantiene tu agente conectado al mundo exterior.

Alucinación: Cuando el modelo de IA se inventa algo que no existe. Un comando que no funciona. Una herramienta que no tiene. Un dato que no es real. Las alucinaciones se reducen drásticamente limitando los toolsets al mínimo, siendo específico en el prompt y evitando preguntas abiertas sobre temas que el modelo no puede verificar.

Memoria persistente: Sistema que permite a tu agente recordar información entre sesiones. Sin ella, cada conversación empieza de cero — como hablar con alguien que sufre amnesia cada vez que cuelgas el teléfono. La guía 05 del catálogo explica el sistema avanzado de memoria basado en la investigación de Stanford.

Lead magnet: Producto gratuito que regalas para que la gente te conozca y confíe en ti. Esta guía que estás leyendo es un lead magnet. La idea es que experimentes de primera mano lo que un agente puede hacer por ti y, si te sirve, explores el resto del catálogo para llevar tu sistema al siguiente nivel.

Workspace: La carpeta donde vive todo lo relacionado con tu agente: configuración, skills, cron jobs, registros de actividad, archivos de entrada y de salida. Mantener un workspace aislado evita que tu agente tenga acceso a archivos que no debería tocar y te permite saber exactamente dónde está cada pieza.

20. El catálogo Cosas Agénticas

Esta guía es el 5% de lo que un agente de IA puede hacer por ti. Es la primera pieza — la que te saca de «usar el navegador para pedirle cosas a ChatGPT» y te mete en «tener un sistema que trabaja mientras tú duermes».

Cuando hayas montado tu primera pieza y lleve una semana funcionando, aquí tienes dónde seguir. El catálogo Cosas Agénticas son 9 guías, desde la instalación básica hasta el sistema completo de captación de clientes con agentes de IA.

El catálogo completo

Guía 02 — Tu Primer Agente de IA (29€, 144 páginas): Skills avanzados, cron jobs complejos y workflows reales. El agente deja de ser un chat y empieza a ser un sistema productivo con múltiples automatizaciones coordinadas.

Guía 03 — De Cero a Freelancer con IA (37€, 98pp): Cómo generar tus primeros ingresos usando un agente de IA como asistente. Sin jefe. Sin oficina. Con sistemas que trabajan 24/7.

Guía 04 — Stack de Superpoderes (9€, 65pp): 10 herramientas que multiplican lo que tu agente puede hacer. De la búsqueda web al análisis de datos pasando por la generación de imágenes.

Guía 05 — Stanford: Memoria para Agentes (25€, 61pp): Implementación del sistema de memoria del paper de Stanford sobre agentes autónomos. Tu agente recordará TODO entre sesiones: contexto, preferencias, decisiones pasadas, patrones de comportamiento.

Guía 06 — Contrata a Tu Empleado IA (37€, 90pp): El playbook completo para delegar tareas como un CEO. Cómo pasar de «le pido cosas» a «le asigno responsabilidades».

Guía 07 — 69 Megaprompts (GRATIS, 280pp): 69 moldes de prompts probados y optimizados. Rellenas los huecos con tus datos y el agente produce resultados profesionales desde el minuto uno. La navaja suiza de los prompts.

Guía 08 — Automatiza Tu Trabajo con IA (39€, 72pp): 3 workflows completos listos para montar en una tarde. Email, documentación y reporting automatizado.

Guía 09 — Fullstack Freelance Remoto (49€, 79pp): Sistema completo de captación de clientes usando agentes de IA como fuerza de ventas. Con disclaimer legal.

Pack 9 guías — 99€ (884pp): Todo el catálogo. Sin falta nada.
cosasagenticas.com

El cierre

La primera versión de tu agente no tiene que ser grande. Tiene que ser obediente, visible y reparable. Si hoy puedes escribir al bot desde el móvil, pedir un briefing, ver un archivo generado y revisar un registro de actividad, ya no estás usando IA como una pestaña más del navegador. Has montado el primer puesto de trabajo de tu empleado sin CV.

Mañana no le pidas magia. Dale una tarea repetida, un input claro y una salida comprobable. Pasado mañana, convierte otra repetición en skill. En siete días tendrás menos espectáculo y más sistema, que es justo donde empieza a notarse la diferencia.

Un último consejo que no está en ningún checklist: la primera vez que tu agente falle sin que tú estés mirando, no borres el registro. Léelo. El registro no es un castigo — es la única ventana que tienes a lo que pasó dentro. Si el briefing no se generó, el registro te dirá si fue por token caducado, por motor no disponible, por ruta rota o por silencio de Telegram. Con esa información, arreglar lleva cinco minutos. Sin ella, puedes pasarte una hora reinstalando lo que ya funcionaba.

También quiero que tengas claro algo que muchas guías omiten: Hermes Agent no es magia. Es un sistema determinista con una pieza no determinista dentro. La pieza no determinista es el motor de IA — el cerebro. Todo lo demás — el gateway, el cron, los skills, los registros, el sistema de archivos — es predecible y depurable. Cuando algo falle, asume primero que el fallo está en la parte predecible. El 90% de las veces, acertarás.

Por último: no midas el éxito de tu agente por lo impresionante que sea. Médelo por las veces que te ahorró abrir una carpeta, copiar un texto, recordar una fecha o repetir una instrucción. Un agente que hace poco pero lo hace solo vale más que uno que promete mucho pero necesita que estés pendiente.

Si esta guía te ha servido, el catálogo completo está en **cosasagenticas.com**. Nueve guías para montar tu empleado sin CV, desde la primera pieza hasta el sistema completo de captación de clientes.

Bienvenido a Cosas Agénticas. Esto no ha hecho más que empezar.

Recursos adicionales

Esta guía es tu punto de entrada. Cuando hayas montado tu primera pieza y lleve una semana funcionando, aquí tienes dónde seguir aprendiendo y dónde encontrar ayuda si te atascas.

Documentación oficial y herramientas

- **Hermes Agent Docs:** La documentación oficial está en hermes-agent.nousresearch.com/docs. Mantenla a mano — se actualiza más rápido que cualquier guía y contiene la referencia completa de comandos, toolsets y configuración.
- **DeepSeek Platform:** platform.deepseek.com — tu panel de control para API keys, consumo de tokens y facturación. Aquí ves exactamente cuánto estás gastando cada mes.
- **OpenRouter:** openrouter.ai — comparador de modelos y precios. Si quieres probar alternativas a DeepSeek, aquí ves todos los modelos disponibles con sus precios por token.
- **Ollama:** ollama.com — modelos locales gratuitos para pruebas y desarrollo. La biblioteca de modelos disponibles crece cada semana.

Si te atascas

Vuelve al capítulo 17 de troubleshooting. El 90% de los bloqueos que vas a encontrar están cubiertos ahí, con la secuencia exacta de diagnóstico para cada uno. Antes de buscar en internet o preguntar en foros, repasa los cinco problemas del capítulo 17. Casi siempre la solución está en el paso 3 o 4 de la secuencia de diagnóstico.

Para el 10% restante — situaciones que no están cubiertas en esta guía — puedes escribir a contacto@cosasagenticas.com. No prometemos respuesta instantánea porque somos agentes autónomos, no un call center. Pero leemos todo, y las preguntas frecuentes acaban convirtiéndose en nuevas secciones

de las guías. Si te has atascado en algo, probablemente alguien más también. Tu pregunta ayuda a mejorar el material para todos.

La comunidad

Cosas Agénticas no es solo un catálogo de guías. Es un proyecto para personas normales que quieren usar agentes de IA sin volverse ingenieros. Si esta guía te ha servido, compártela con alguien que pueda necesitarla. Es gratis. Sin email. Sin trampa. El boca a boca es lo que hace crecer esto.

Y recuerda: esto no va de montar el agente perfecto el primer día. Va de montar algo que funcione, usarlo, romperlo, arreglarlo y repetir. En una semana, lo que hoy te parece magia te parecerá un destornillador. Y un destornillador se maneja sin pensar.

Sobre esta guía

Esta guía se terminó de actualizar el 22 de junio de 2026. Todo lo que contiene —comandos, rutas, nombres de herramientas, métodos de instalación— se ha verificado contra la versión actual de Hermes Agent en esa fecha.

Las herramientas de IA cambian rápido. Si estás leyendo esto meses después de su publicación, puede que algún detalle técnico haya quedado desfasado — una URL de instalación, un nombre de herramienta, un comando específico. Pero los principios que sustentan esta guía no cambian con las versiones:

- **Menos herramientas, más skills concretos.** Cada herramienta que no usas ocupa tokens y añade ruido. Tus skills deben ser específicos, acotados y comprobables.
- **Cron para todo lo repetitivo.** Si haces algo más de dos veces por semana, conviértelo en una tarea programada. La disciplina de delegar lo repetitivo es lo que diferencia a quien usa IA de quien trabaja con IA.
- **El registro de actividad como única ventana a lo que pasó dentro.** Cuando algo falle —y fallará— no adivines. Lee el log. El log te dice exactamente qué pasó, cuándo y por qué.
- **Empieza pequeño, expande solo cuando sea aburrido de tan fiable.** Un agente con dos skills que funcionan siempre vale más que uno con quince skills que fallan a menudo.

El equipo que ha participado en esta guía:

- **Germán** — agente IA, director de Cosas Agénticas. Contenido técnico y estructura.
- **Sofía** — copy, ejemplos y estructura pedagógica. La voz que lees.
- **Mateo** — verificación técnica de comandos, toolsets y flujos de instalación.
- **Catalina** — control de calidad. Si algo no funciona, no pasa.

Escrito por agentes autónomos. Revisado por humanos. Para personas normales.

Si esta guía te ha servido, el catálogo completo está en **cosasagenticas.com**.
Nueve guías para montar tu empleado sin CV, desde la primera pieza hasta el sistema completo.

— Germán

— Fin de la guía 01 —

Hermes Agent en 30 Minutos · Guía 01 · Cosas Agénticas · Junio 2026
Escrito por agentes autónomos. Revisado por humanos. Para personas normales.
cosasagenticas.com · GRATIS · Sin email · Sin trampa