GUÍA · INSTALACIÓN + LAS 10 REGLAS

Haz que Claude responda al *grano* con i-have-adhd.

Una skill que instalas en dos minutos. Claude parte por el comando, numera los pasos y corta el relleno.

AUTOR

Jorge Riquelme

Fundador · Otherway Labs

CONTENIDO

Instalación + 10 reglas

Antes y después, cómo encenderla, apagarla y dejarla fija

LECTURA

8 minutos

Nivel base. Para quien ya usa Claude a diario.

Por qué te responde *largo*.

Le escribes "sé conciso" y la siguiente respuesta baja de tamaño. Tres mensajes después vuelven los cuatro párrafos con contexto, alternativas y despedida. Pasa por tres razones concretas.

01 **Corto no significa listo para actuar**

Un párrafo puede quedar corto y aun así dejar el comando en la última línea. Tú lo necesitas en la primera. Lo que te falta es **orden**, no menos texto.

02 **La instrucción dura dos o tres turnos**

Cuando pides el formato dentro del chat, esa instrucción compite con todo el texto que viene después. A los pocos mensajes el modelo vuelve a su estilo por defecto y tú ni te das cuenta.

03 **El relleno siempre tiene la misma forma**

Abre con "buena pregunta", anuncia lo que va a hacer, hace el trabajo, resume lo que hizo y cierra con "espero que te sirva". De esas cinco partes hay tres que no ocupas nunca.

ÍNDICE

01 Por qué te responde largo
02 La misma pregunta, dos respuestas
03 Instálala en dos minutos
04 Las diez reglas que le cambias
05 Encenderla, apagarla, dejarla fija
06 Sus límites y qué hacer hoy

La misma pregunta, dos *respuestas*.

Este ejemplo sale del repositorio de la skill. Le preguntas por un error de autenticación en tu código. Arriba la respuesta por defecto, abajo la misma respuesta con la skill encendida.

ANTES · RESPUESTA POR DEFECTO

Buena pregunta. Déjame pensarlo. Tu flujo de autenticación tiene varias partes: el middleware, la verificación del token y el manejo de cookies. Mirando `src/auth.ts`, la función `verifyToken` (alrededor de las líneas 42 a 58) parece estar usando una API antigua de `jsonwebtoken`. Una opción sería actualizar el paquete y reescribir esa función. Después del cambio, convendría correr los tests de autenticación para confirmar que nada se rompe. De paso, quizás quieras revisar las versiones de tus dependencias en general. ¡Espero que te sirva! Avísame si quieres profundizar.

DESPUÉS · CON LA SKILL ENCENDIDA

Corre `npm install jsonwebtoken@latest` y edita `src/auth.ts:42`.

1. Abre `src/auth.ts`
2. Reemplaza `verifyToken` (líneas 42 a 58) por el snippet de abajo
3. Corre `npm test -- auth.spec.ts`

Siguiente: si un test falla, pega la primera línea del error.

LARGO TOTAL	
Antes	87 palabras
Después	39 palabras

HASTA LA PRIMERA INSTRUCCIÓN	
Antes	46 palabras
Después	1 palabra

Instálala en dos *minutos*.

La skill vive en un repositorio público de GitHub y es gratis. Tienes dos caminos para Claude Code: le pegas la instrucción y se instala sola, o corres los dos comandos a mano.

OPCIÓN 1 · PEGAR ESTO EN CLAUDE CODE

```
Install the i-have-adhd skill/plugin from
https://github.com/ayghri/i-have-adhd, refer to
the repo's AGENTS.md for instructions.
```

Va en inglés a propósito: así viene en el repositorio y Claude lee las instrucciones del autor sin traducciones de por medio.

OPCIÓN 2 · LOS DOS COMANDOS

```
claude plugin marketplace add ayghri/i-have-adhd
claude plugin install i-have-adhd@i-have-adhd
```

01 **Reinicia Claude Code**

El índice de plugins se lee al arrancar. Si no reinicias, `/i-have-adhd` no aparece en el autocompletado.

02 **Confirma que quedó instalada**

Corre `claude plugin list` y busca `i-have-adhd` en la lista.

03 **Enciéndela**

Escribe `/i-have-adhd` al principio del chat. Las reglas quedan puestas por toda esa sesión.

TIP

Instalada y encendida son dos cosas distintas. La skill trae `disable-model-invocation`, así que Claude nunca la activa por su cuenta. Si no escribes `/i-have-adhd`, te responde igual que siempre.

Las diez reglas que le *cambias*.

Esto es lo que la skill le instala a Claude como estilo de salida. Vale la pena leerlas aunque no la instales, porque son las mismas instrucciones que puedes pegar a mano en tu archivo de reglas permanentes.

01 Parte por la acción siguiente. El comando, la ruta o el snippet va en la primera línea.

02 Numera el trabajo de varios pasos. Una acción por paso, sin dos "y después" adentro.

03 Cierra con un solo paso concreto que puedas hacer en menos de dos minutos.

04 Termina un tema antes de abrir el siguiente. Lo demás se ofrece aparte, al final.

05 Te recuerda en qué punto van cada vez: "paso 3 de 5 listo, sigue el backfill".

06 Estima en minutos u horas. Se acabó el "esto va a tomar un poco de trabajo".

07 Después de cada cambio te muestra qué quedó funcionando y cómo probarlo.

08 Los errores van sin drama: dónde falla, por qué falla y cuál es el arreglo.

09 Máximo cinco ítems por lista. Si son más, los separa en "ahora" y "después".

10 Cero preámbulo, cero resumen de lo que acaba de hacer y cero despedida.

TIP La que más se nota en el día a día es la 10. Cuando desaparecen el "buena pregunta" del principio y el "espero que te sirva" del final, la respuesta arranca en el dato que fuiste a buscar.

Enciéndela, apágala o déjala *fija*.

Por defecto queda apagada hasta que la invocas y dura solo esa sesión. Estos son los tres estados y cómo mueves de uno a otro.

`/i-have-adhd`
dentro del chat

Encenderla por esta sesión

Lo escribes al principio del chat. Las reglas se quedan puestas aunque cambies de tema, y no se caen a los pocos turnos como pasa cuando pides el formato a mano.

`stop adhd mode`
o "normal mode"

Apagarla sin desinstalar nada

Claude confirma en una línea y vuelve a su estilo normal. Sirve cuando estás escribiendo un texto largo con él y quieres que se explaye.

`touch ~/.claude/.i-have-adhd-always`
en tu terminal

Dejarla encendida para siempre

Ese comando crea un archivo vacío que funciona como bandera. Con la bandera puesta, un hook de inicio carga las reglas desde el primer mensaje de cada sesión nueva. Para volver atrás, borras el archivo con `rm ~/.claude/.i-have-adhd-always`.

SI NO TRABAJAS EN CLAUDE CODE

El repositorio trae instrucciones de instalación para Cursor, Codex, Copilot, Gemini CLI, Zed, OpenCode y varios más. Para cualquier herramienta que lea reglas permanentes hay algo más simple todavía: el mismo repositorio publica las diez reglas como un bloque de texto listo para pegar en tu archivo de instrucciones, y con eso quedan activas sin instalar nada.

Cuándo se *salta* sus propias reglas.

La skill trae cuatro excepciones escritas. Conviene conocerlas para no creer que se rompió el día que te llegue un texto largo.

Le pides que explique	Se extiende lo que haga falta y agrega títulos para que puedas volver a buscar. Igual sigue sin preámbulo y sin despedida.
Viene algo destructivo	Un `rm -rf`, un force push o una migración de base de datos. Ahí pregunta antes de ejecutar, aunque le cueste una respuesta más larga.
Tres intentos fallidos	Si llevan tres turnos con "sigue malo", deja de tocar el código y te nombra el supuesto que puede estar equivocado. Esta es la excepción que más tiempo te ahorra.
Tu pedido es ambiguo	Hace una sola pregunta corta en vez de adivinar y entregarte algo que vas a mandar a rehacer.

Mi recomendación es que **no la dejes fija el primer día**. Úsala una semana a mano, escribiendo `/i-have-adhd` solo en las tareas donde de verdad te apura la respuesta. Si al cabo de esa semana te molesta tener que escribirlo cada vez, ahí creas el archivo bandera y listo.

PARA HACER HOY

- ☐ Pego la instrucción de instalación en Claude Code
- ☐ Reinicio y corro `claude plugin list` para confirmar
- ☐ Escribo `/i-have-adhd` al empezar mi próxima tarea real
- ☐ Comparo esa respuesta con las que recibía la semana pasada
- ☐ Si me sirvió, creo el archivo bandera y la dejo fija



SOBRE EL AUTOR

Jorge Riquelme

FUNDADOR · OTHERWAY LABS

Ayudo a pymes y profesionales a recuperar tiempo con automatizaciones de IA, con ROI claro y garantizado.

otherwaylabs.cl · LinkedIn: Jorge Riquelme

Tres formas de seguir *avanzando*.

PARA EMPRESAS
GRATIS

Sesión gratuita

¿Hay procesos en tu empresa que *deberían correr solos*? Los revisamos juntos y te digo cuáles se pueden automatizar y cuánto tiempo recuperas al mes.

→ otherwaylabs.cl

PARA EQUIPOS

Capacitación

Forma a tu equipo para que use IA *todos los días*. Sesiones prácticas sobre sus propias tareas, con Claude configurado a la medida del equipo.

→ otherwaylabs.cl/capacitaciones

PARA
PROFESIONALES
COMUNIDAD

Axia

La comunidad para *implementar IA de verdad*. Profesionales y empresas que pasan de probar herramientas sueltas a dejar procesos que corren solos.

→ axia.academy