

GUÍA DE INSTALACIÓN

GHL Pro MCP

Conecta tu GoHighLevel con tu IA

De cero a funcionando en Windows, macOS o Linux.

Compatible con Claude Desktop, Claude Code, Codex, Cursor, Cline y Windsurf

TIEMPO ESTIMADO

~10 minutos

SISTEMAS

**Windows · macOS ·
Linux**

NECESITAS

GoHighLevel y Chrome

Tus credenciales se guardan únicamente en tu ordenador.

Lo primero que tienes que entender

No hay nada que instalar del programa. Nunca.

Lo que se instala en tu ordenador:	uv	(el motor, una sola vez)
Lo que NO se instala:	el MCP	(se descarga solo cada vez)

Y eso es a propósito. Significa que **siempre tienes la última versión**, sin hacer nada: cuando publiquemos un arreglo, **se recibe solo la próxima vez que arranques tu IA.**

No hay actualizaciones que hacer, ni versiones que comprobar, ni comandos que recordar.

El orden importa

Sigue los pasos **en este orden**. El 2 va antes que el 4 por una razón concreta: cuando el programa te pida las credenciales, **abre una ventana de 5 minutos** y espera a que la extensión se las mande. Si la extensión no está instalada todavía, esa ventana se cierra y hay que repetir el paso.

- 1 • Instalar uv (el motor – lo ÚNICO que se instala)
- 2 • Instalar la extensión ← ANTES del paso 4
- 3 • Sacar tus 2 datos de GoHighLevel
- 4 • Configurar
- 5 • Conectarlo con tu IA
- 6 • Comprobar que funciona

1 Instalar uv

uv es el motor que descarga y ejecuta GHL Pro MCP. **Es lo único que se instala en tu ordenador, y se instala una sola vez en la vida.**

Elige tu sistema:

Windows — abre **PowerShell** y pega:

```
winget install astral-sh.uv
```

macOS — abre **Terminal** y pega:

```
brew install uv
```

Linux — abre tu terminal y pega:

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

Comprueba que quedó bien. Cierra la terminal, ábrela otra vez y escribe:

```
uv --version
```

Tiene que responder algo como `uv 0.x.x`. Si dice «no se reconoce el comando», cierra la terminal y ábrela de nuevo — casi siempre es eso.

⚠ Lo que `winget install astral-sh.uv` **NO** hace

No instala GHL Pro MCP. Instala `uv`, el motor. Y eso es todo lo que hay que instalar.

El MCP no se instala nunca. Se descarga al vuelo, cada vez, con este comando:

```
uvx --refresh ghl-pro-mcp@latest
```

Vas a ver esa línea varias veces en esta guía, y siempre significa lo mismo: «bájate la última versión y ejecútala». **No hay que memorizarla** — está copiada en cada paso.

2 Instalar la extensión de Chrome

Este paso va antes del 4. Si lo dejas para después, tendrás que repetir el paso 4.

La extensión «**GHL Token Bridge**» lee de tu propio navegador las credenciales de sesión de GoHighLevel y **se las entrega al programa por `127.0.0.1`** — o sea, de tu navegador a tu ordenador, **sin pasar por internet**.

2.1 · Cargarla

1. Abre **Chrome** y escribe en la barra de direcciones: `chrome://extensions`
2. Arriba a la derecha, activa el **Modo de desarrollador**.
3. Pulsa «**Cargar descomprimida**» (o *Load unpacked*).
4. Elige la carpeta `token fire v8_4` que te hemos entregado.
5. Debe aparecer una tarjeta que dice **GHL Token Bridge (refresh token)** versión **8.4**.

No borres ni muevas esa carpeta. Chrome lee la extensión de ahí cada vez que arranca. Si la mueves, se desactiva y la renovación de sesión deja de funcionar.

Si tenías una versión anterior, desactívala para no tener las dos a la vez.

2.2 · Anclarla a la barra

Pulsa el icono de **piezas de puzzle** (a la derecha de la barra de direcciones) y fija **GHL Token Bridge**. Así la tienes a un clic.

Nota honesta sobre el modo de desarrollador: Chrome avisa a veces de que tienes extensiones en modo desarrollador. **Es normal en una extensión que se carga así**, y no significa que haya nada malo — pero es un aviso que verás. Está en camino publicarla en la tienda de Chrome para que desaparezca; hasta entonces, se instala así.

La extensión no hace nada por su cuenta. Solo lee tus credenciales de GoHighLevel cuando **tú pulsas su botón**, y solo se las manda a `127.0.0.1:3847` — tu propio ordenador.

3 Sacar tus dos datos de GoHighLevel

Tenlos a mano antes de empezar el paso 4, porque el programa te los va a pedir de seguido.

3.1 · El Private Integration Token

1. Entra en **GoHighLevel** con tu cuenta.
2. **Ajustes** → **Private Integrations** → **Create new**.
3. Ponle un nombre (por ejemplo *GHL Pro MCP*) y **marca los permisos** que quieras darle. Cuantos más marques, más podrá hacer — si dudas, márcalos todos.
4. Pulsa guardar y **cópialo EN ESE MOMENTO**.

 **GoHighLevel solo lo muestra una vez.** Si cierras esa ventana sin copiarlo, tendrás que crear otro.

Empieza por `pit-`.

3.2 · El Location ID

Es el identificador de tu subcuenta, y **está en la barra de direcciones**:

```
app.gohighlevel.com/v2/location/zcyTQVSM93prhmkQqWpL/dashboard
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
esto es el Location ID
```

Cópialo tal cual. **No hace falta ir a ningún menú.**

4 Configurar

Abre **PowerShell** (Windows) o **Terminal** (macOS y Linux) y pega **esta línea entera**:

```
uvx --refresh ghl-pro-mcp@latest setup
```

La primera vez tarda unos segundos mientras se descarga. Las siguientes es inmediato.

Te guía por tres pasos. No hay que saberse nada.

1/3 · El Private Integration Token

Pega el `pit-...` del paso 3.1.

Verás un asterisco por cada carácter mientras escribes. No es un fallo: el token no se mostrará nunca en pantalla, pero **sí ves que se está escribiendo** y puedes contar si el pegado salió entero. La tecla de borrar funciona con normalidad.

El programa lo comprueba contra GoHighLevel de verdad antes de guardarlo. Si es inválido, te lo dice ahí mismo y no lo escribe.

2/3 · El Location ID

Pega el del paso 3.2.

Y te dirá el nombre de la subcuenta que ha encontrado. Compruébalo: si no es la tuya, es que has mezclado el token de una cuenta con el id de otra.

3/3 · Workflows, embudos y encuestas

Aquí te sale un menú:

```
[1] Con la extension de Chrome    (recomendado, un clic)
[2] Pegando las credenciales aqui
[3] Omitir por ahora
```

Elige **1** — y esto es lo que has preparado en el paso 2. El programa dice:

```
Receptor escuchando en el puerto 3847.  
Pulsa el boton de la extension...
```

Ahora, en este orden:

1. Ten **GoHighLevel** abierto y con la sesión iniciada en Chrome.
2. Pulsa el **icono de la extensión**.
3. Pulsa **«Enviar a GHL Pro MCP»**.

Verás **Listo. Credenciales guardadas**. **La terminal continúa sola** y termina.

🕒 **Tienes 5 minutos.** Si tardas más, la ventana se cierra — y se arregla repitiendo solo este paso, sin tocar nada más: `bash uvx --refresh ghl-pro-mcp@latest setup --solo-firebase`

Elige **2** si prefieres pegar las credenciales a mano. **Pulsa antes el botón «Copiar» de la extensión:** deja en el portapapeles **las dos líneas juntas**, listas para pegar:

```
GHL_FIREBASE_REFRESH_TOKEN=...  
GHL_FIREBASE_API_KEY=...
```

Elige **3** si ahora no puedes. Todo lo demás funciona igual; solo te quedarás sin workflows, embudos ni encuestas. **Se añade después sin repetir nada**, con el comando de arriba.

5 Conectarlo con tu IA

Elige **una** de estas dos formas.

Forma A · Claude Desktop — sin terminal

Te hemos entregado un archivo llamado `ghl-pro-mcp.mcpb` (unos 3 KB).

1. Abre **Claude Desktop**.
2. **Ajustes** → **Extensiones**.
3. **Arrastra el archivo** `.mcpb` a esa ventana.
4. Rellena los **4 campos**. **Dos los tienes del paso 3** (el token y el Location ID). **Los otros dos los da la extensión:** pulsa su botón **«Copiar»** y tendrás el refresh token y la API key en el portapapeles.
5. Reinicia Claude Desktop.

Y ya está. Aquí **no hay que escribir ningún comando, nunca**.

Forma B · Claude Code, Codex, Cursor, Cline, Windsurf

Hay que decirle a tu IA que use el servidor. **Copia este bloque** en su archivo de configuración:

```
{
  "mcpServers": {
    "ghl-pro": {
      "command": "uvx",
      "args": ["--refresh", "ghl-pro-mcp@latest"],
      "env": {
        "GHL_ENABLE_SESSION": "1"
      }
    }
  }
}
```

Fíjate: es **la misma línea** que usaste en el paso 4. **No hay nada nuevo que aprender.**

¿Dónde va?

Tu IA	El archivo
Claude Code	<code>.mcp.json</code> en la carpeta de tu proyecto (créalo si no existe)
Cursor	<code>~/.cursor/mcp.json</code>
Cline / Windsurf	el suyo, en Ajustes → MCP Servers
Codex	<code>~/.codex/config.toml</code> — pero en formato TOML, abajo

Codex usa TOML, no JSON:

```
[mcp_servers.ghl-pro]
command = "uvx"
args = ["--refresh", "ghl-pro-mcp@latest"]
env = { GHL_ENABLE_SESSION = "1" }
```

Para tu IA no hay que instalar nada. `uvx` se lo trae de PyPI la primera vez que arranca. Tarda unos segundos y es la única vez.

Si ya tenías otros servidores MCP, no los borres: añade el bloque `ghl-pro` junto a los que había.

Reinicia tu IA para que lo detecte.

6 Comprobar que funciona

Aquí hay un detalle que confunde a todo el mundo si no se sabe. La vía de sesión —la de workflows, embudos y encuestas— **está apagada por defecto**, y solo la enciende tu IA cuando arranca el servidor. **En la terminal hay que encenderla a mano**, o el diagnóstico dirá que está desactivada y parecerá que algo va mal.

Windows (PowerShell):

```
$env:GHL_ENABLE_SESSION='1'; uvx --refresh ghl-pro-mcp@latest doctor
```

macOS y Linux:

```
GHL_ENABLE_SESSION=1 uvx --refresh ghl-pro-mcp@latest doctor
```

Te dice, una a una, qué funciona y qué no:

```
Configuración
ok  Credenciales cargadas
ok  Subcuenta configurada

Vía pública (API V2)
ok  Leer la subcuenta  «Tu Clínica»
ok  Listar etiquetas   42 etiquetas
ok  Listar pipelines  3 embudos de venta
ok  Listar usuarios   1 usuarios

Vía de sesión
ok  Canjear el refresh token  id_token válido
ok  Listar workflows          29 workflows
ok  Leer disparadores         2 disparadores
ok  Leer un workflow          en draft, 9 acciones

Resultado: 11 correctas, 2 omitidas.
```

Si alguna sale **FALLO, el propio mensaje te dice qué hacer.** No hay que interpretar nada.

Si la vía de sesión sale como **-- desactivada**, es que te has dejado el `GHL_ENABLE_SESSION=1` delante del comando. **No es un fallo:** el diagnóstico solo está diciendo que no le has encendido esa puerta **a él**.

Actualizaciones: no hay nada que hacer

Esto es lo bueno de no instalar nada.

Tu IA	Con <code>--refresh</code> y <code>@latest</code> en su configuración, mira si hay versión nueva cada vez que arranca
La terminal	El comando <code>uvx --refresh ghl-pro-mcp@latest</code> se baja la última cada vez que lo escribes

Así que cuando publiquemos un arreglo:

```
Tú no haces nada.  
Reinicias tu IA, y ya lo tienes.
```

No hay versiones que comprobar, ni comandos de actualización, ni nada que se quede viejo.

Cuando caduque la sesión

El *refresh token* de GoHighLevel **caduca cada cierto tiempo** (semanas o meses). Cuando pase, verás errores de sesión al pedir workflows o embudos.

Se arregla con un comando, y no hay que repetir nada de lo anterior:

```
uvx --refresh ghl-pro-mcp@latest setup --solo-firebase
```

Ese comando **pide solo el par de Firebase** — no vuelve a pedirte el Private Integration Token ni el Location ID, porque **esos no caducan**.

Problemas comunes

«**no se reconoce el comando uv**» Cierra la terminal y ábrela otra vez. Si sigue:

```
uv tool update-shell
```

«**no se reconoce el comando uvx**» Es el mismo caso. `uvx` viene con `uv`: si `uv` funciona y `uvx` no, cierra la terminal y ábrela de nuevo.

En el paso 3/3, la extensión dice que no se pudo conectar La ventana de 5 minutos se cerró. Vuelve a ejecutar:

```
uvx --refresh ghl-pro-mcp@latest setup --solo-firebase
```

y pulsa el botón otra vez.

La extensión dice que no encuentra las credenciales Abre GoHighLevel, **inicia sesión**, espera a que cargue el panel y reinténtalo. La extensión las lee del almacenamiento del navegador, y ahí solo están con la sesión abierta.

Chrome ha desactivado la extensión Has movido o borrado la carpeta `token fire v8_4`. Vuelve a ponerla donde estaba, o cárgala otra vez desde `chrome://extensions`.

«GoHighLevel rechazó las credenciales (401)» El Private Integration Token ya no vale. Crea uno nuevo en **Ajustes** → **Private Integrations** y vuelve a ejecutar el paso 4.

Funciona todo menos workflows, embudos y encuestas Es la vía de sesión: está apagada o le faltan las credenciales de Firebase. Ejecuta el diagnóstico **con la variable delante** y te dirá exactamente cuál de las dos cosas es:

```
$env:GHL_ENABLE_SESSION='1'; uvx --refresh ghl-pro-mcp@latest doctor # Windows
```

```
GHL_ENABLE_SESSION=1 uvx --refresh ghl-pro-mcp@latest doctor # macOS y Linux
```

Si es que faltan las credenciales:

```
uvx --refresh ghl-pro-mcp@latest setup --solo-firebase
```

Mi IA no ve las herramientas nuevas No has reiniciado la IA después de cambiar su configuración. Ciérrala y ábrela.

Qué puedes pedirle una vez instalado

En lenguaje normal, sin comandos:

- «¿Cuántos contactos tengo con la etiqueta "cliente"?»
- «Créame un workflow que etiquete como "nuevo lead", espere 2 horas y mande un SMS de bienvenida»
- «Muéstrame los pipelines y sus etapas»
- «Crea un embudo llamado "Webinar de octubre"»
- «Hazme una encuesta que filtre a quien no pueda pagar y me pregunte qué quiere aprender»

- «¿Qué ha contestado la gente en la encuesta?»

Sobre las encuestas. No son un formulario: **filtran**. El que no encaja se descarta solo, y el que llega a la llamada ya viene caliente. Pídeselas en lenguaje normal; cuando estén, **te dará una dirección web para abrirlas** — ábrelas y míralas, porque es la única forma de comprobar que se ven bien.

Tu IA te pedirá confirmación antes de cambiar nada, y nunca publica ni borra nada sin que se lo digas explícitamente.

Varias cuentas: los perfiles

Si trabajas con **más de una subcuenta** de GoHighLevel —una agencia con varios clientes, por ejemplo— no hace falta instalar nada más ni tocar archivos. **Cambias de cuenta hablando con tu IA, al momento, sin reiniciar.**

Crear una cuenta (una sola vez por cuenta)

Dile a tu IA que cree un **perfil**, con los datos de esa cuenta:

«Crea un perfil llamado "clínica" con su Private Integration Token, su Location ID, su refresh token y su API key»

Tu IA **valida los datos contra GoHighLevel** y los guarda como un perfil: un archivo aparte en tu ordenador. **Esto se hace una sola vez por cuenta.** Después, esos datos no se vuelven a pedir nunca.

Cambiar de cuenta (cuando quieras)

A partir de ahí, basta con decirlo:

«Cámbiate a la clínica» «Vuelve a la cuenta principal»

Sin reiniciar la IA, sin comandos, sin tocar archivos. Contactos, workflows, embudos y citas siguen funcionando igual; solo cambian a qué cuenta apuntan.

Lo que conviene saber

- **Cada perfil es un archivo local**, igual de privado que `credentials.env`. Nada sale de tu ordenador.
- Pregunta «¿qué cuentas tengo configuradas?» para verlas y saber cuál está activa.
- Los datos de un perfil se dan **una sola vez**; por privacidad, conviene entregarlos como en el paso 4 (terminal o archivo) en lugar de pegarlos en el chat.

Dónde se guardan tus credenciales

En un único archivo, en tu ordenador. Nada sale de ahí.

```
Windows    %LOCALAPPDATA%\ghl-pro-mcp\credentials.env
macOS      ~/Library/Application Support/ghl-pro-mcp/credentials.env
Linux      ~/.config/ghl-pro-mcp/credentials.env
```

Cada perfil de otra cuenta se guarda al lado, en su propia carpeta:

```
Windows    %LOCALAPPDATA%\ghl-pro-mcp\profiles\<<nombre>.env
macOS      ~/Library/Application Support/ghl-pro-mcp/profiles/<nombre>.env
Linux      ~/.config/ghl-pro-mcp/profiles/<nombre>.env
```

Por qué la configuración se hace en la terminal y no en el chat

Lo que se escribe en una conversación con una IA entra en el contexto del modelo, en el historial de la sesión y en cualquier transcripción. Un Private Integration Token da acceso completo a tu CRM. **Por eso no pasa por ahí.**

En la terminal, en cambio, **va de tu teclado a tu disco. Ni un carácter en pantalla.**

Lo que la extensión NO hace

- **No manda tus credenciales a ningún sitio remoto.** Solo habla con `http://127.0.0.1:3847`, que es tu propio ordenador.
- **No guarda nada fuera de tu navegador**, salvo lo que tú le pidas al programa.
- **No toca tu sesión de GoHighLevel.** Solo lee.

Resumen en 6 líneas

```
1 • winget install astral-sh.uv      (o brew, o curl – según tu sistema)
   ← es lo ÚNICO que se instala. El MCP no se instala nunca.
2 • chrome://extensions → modo desarrollador → cargar «token fire v8_4»
3 • De GoHighLevel: el Private Integration Token y el Location ID
4 • uvx --refresh ghl-pro-mcp@latest setup      (y pulsar el botón de la extensión)
5 • El bloque de configuración en tu IA
6 • GHL_ENABLE_SESSION=1 uvx --refresh ghl-pro-mcp@latest doctor
```

Y cuando caduque la sesión, dentro de unas semanas:

```
uvx --refresh ghl-pro-mcp@latest setup --solo-firebase
```

No hay nada más. Ni actualizaciones, ni versiones, ni mantenimiento.