— GUÍA TÉCNICA

Crea tu *propio* MCP para tu plataforma interna.

→ Conecta tu CRM o sistema propio a la IA.

→ Sin reescribir tu app. Con Claude Code.

El paso a paso real para exponer tu plataforma interna a la inteligencia artificial — desde decidir qué automatizar hasta dejarlo corriendo.

AUTOR

Jorge Riquelme

Fundador · Otherway Labs

MÉTODO

Claude Code

Audita · Genera · Conecta

HERRAMIENTAS

127

— POR QUÉ UN MCP PROPIO

El *USB-C* entre tu sistema y la IA.

Un MCP es un conector estándar. Uno solo — y cualquier IA puede leer y actuar sobre tu plataforma, sin importar el modelo.

Sin él, cada modelo nuevo te obliga a programar una integración a medida. No escala. Todo se hace dos veces.

Con un MCP propio conectas una vez. Desde ahí, automatizar procesos con IA deja de ser un proyecto y pasa a ser una **conversación**.

— LOS 6 PASOS

01 Decide qué quieres que la IA haga 03
02 Activa la skill que lo construye 04
03 Deja que Claude audite tu plataforma 05
04 Genera el servidor MCP 06
05 Conéctalo y pruébalo en local 07
06 De local a servidor 08

— Y ANTES DE CERRAR

— Seguridad mínima no negociable + errores comunes.

Pasos	Lectura	Nivel
6	12min	Medio

→ LÉELO ASÍ

Esta es una guía **general**. En la práctica, cada paso se vuelve mucho más específico — y todo depende del código de tu plataforma: qué tan complejo es, en qué lenguaje está hecho y qué funcionalidades tiene. Hay matices que pueden complicar bastante el proceso. Tómalo como el mapa: dentro de estos 6 pasos, tu caso particular define el detalle fino.

→ PASO 01

Decide qué quieres que la *IA haga.*

Antes de escribir una línea, define el trabajo. No pienses en tablas ni en endpoints. Piensa en lo que tu equipo hace a mano todos los días y querrías pedirle a la IA en una frase.

Piensa en trabajos, no en botones

Cada herramienta del MCP es un trabajo concreto del negocio. Ejemplos reales:

- Buscar un cliente por nombre y traer su ficha.
- Crear un proyecto nuevo con su estado inicial.
- Subir un recurso y publicarlo en la web.
- Listar los tickets abiertos de la semana.

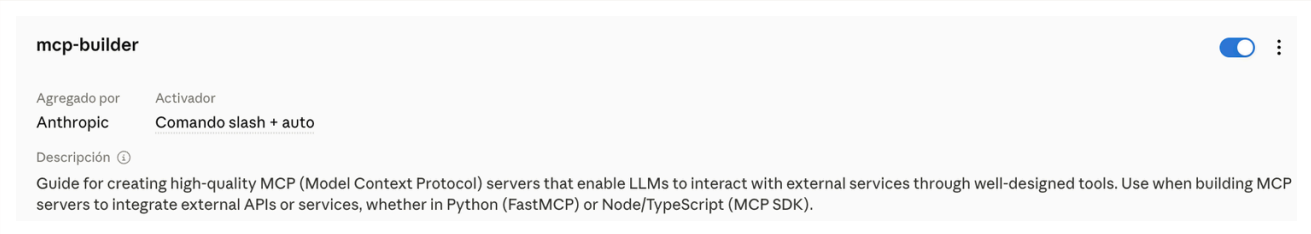
→ CÓMO LO HICIMOS

En Otherway partimos por 4 trabajos. Hoy son **127 herramientas**. Empieza por los 5 que más repites — el resto se agrega solo cuando lo necesitas.

→ PASO 02

Activa la *skill* que lo construye.

No instalas nada raro. En Claude entra a **Explorar** → **Skills** y busca la que construye servidores MCP. Viene nativa, creada por Anthropic, y se activa con un comando o de forma automática.



— La skill `mcp-builder` en Explorar → Skills. Nativa de Anthropic, activador "comando slash + auto".

Qué hace la skill por ti

- Te interroga sobre el caso antes de tocar código.
- Elige el tipo de despliegue (local o remoto).
- Define las herramientas, separando lectura de escritura.
- Arma el esqueleto del servidor y te dice cómo probarlo.

Pídeselo así

> *"Usa tu skill para construir servidores MCP. Quiero exponer mi plataforma interna a Claude. Hazme las preguntas que necesites antes de escribir una sola línea de código."*

→ PASO 03

Deja que Claude *audite* tu plataforma.

Tu plataforma ya sabe todo lo que el MCP necesita: qué datos guarda y qué acciones permite. El trabajo de Claude es leer ese mapa y proponer las herramientas — sin que tú se lo dictes.

El prompt que lo dispara

> *"Audita mi plataforma: revisa la capa de datos y las acciones que ya existen. Propón una lista de herramientas MCP, separando las de lectura de las de escritura. No generes código todavía."*

Qué va a mapear

- Las tablas y cómo se relacionan.
- Las acciones que tu sistema ya permite.
- Qué es lectura y qué modifica datos.

→ NIVEL PRO Apunta a Claude a tu **capa de datos** y a tu lógica de autenticación. Ahí está el 90% de lo que el MCP necesita entender para no romper nada.

→ PASO 04

Genera el *servidor.*

Con el mapa listo, la skill genera el servidor. Dos caminos: TypeScript con el SDK oficial, o Python con FastMCP. Ambos válidos — elige el que ya usa tu equipo.

Pídele que lo escriba

> *"Genera el servidor MCP con las herramientas que propusiste. Usa esquemas estrictos, descripciones claras y manejo de errores en cada una."*

↳ Cómo se ve una herramienta

```
server.tool(
  "buscar_cliente",
  "Busca un cliente por nombre y devuelve su ficha.",
  { nombre: z.string() },
  async ({ nombre }) ⇒ {
    const data = await db.clientes.buscar(nombre);
    return { content: [{ type: "text", text: JSON.stringify(data) }] };
  }
);
```

→ TIP Una herramienta = un trabajo. Descripción clara, inputs validados, y que devuelva **solo** lo que el modelo necesita leer. Nada más.

→ PASO 05

Conéctalo y *pruébalo* en local.

El MCP corre en tu máquina y se conecta a Claude por un archivo de configuración. La regla de oro: las credenciales van por variables de entorno, nunca escritas en el código.

5.1 Conéctalo a Claude

```
{
  "mcpServers": {
    "mi-plataforma": {
      "command": "node",
      "args": ["/ruta/a/mi-mcp/index.js"],
      "env": { "DB_URL": "···", "API_KEY": "···" }
    }
  }
}
```

5.2 Pruébalo con el Inspector

Antes de usarlo en Claude, ábrelo con `MCP Inspector`. Lista tus herramientas y te deja ejecutarlas a mano para confirmar que responden bien.

→ TIP Instala el servidor en una carpeta estable de tu home (ej: `~/.mi-mcp/`), no en Descargas ni en temporales. Evitas problemas de permisos y que el sistema lo borre.

→ PASO 06

De local *a servidor.*

Local funciona perfecto para una persona. El momento de subirlo a un servidor llega cuando lo necesita alguien más que tú.

¿Solo tú? → Quédate local

Transporte stdio. El MCP corre en tu equipo, cero infraestructura, cero costo. Ideal para empezar y para uso personal.

¿Tu equipo o clientes? → Remoto

Transporte HTTP con autenticación (OAuth). Lo montas una vez y queda accesible desde cualquier Claude, para todos.

Dónde hostearlo

Vercel, Railway o Cloudflare lo levantan en minutos. Agregas la capa de autenticación y queda en línea. No necesitas un servidor propio para partir.

→ TIP No te adelantes. Quédate en local hasta que el dolor de compartirlo sea real. Migrar de stdio a HTTP es un cambio de transporte, **no una reescritura**.

— SEGURIDAD NO NEGOCIABLE

Lo que *nunca* debes exponer.

01 La llave que salta los permisos vive en el backend. Nunca viaja al cliente ni al modelo.

02 Credenciales por variables de entorno. Jamás escritas dentro del código.

03 Separa lectura de escritura. Marca las herramientas que borran o modifican.

04 Devuelve solo lo necesario. El modelo no necesita ver la tabla entera.

— 3 ERRORES QUE CUESTAN CARO

→ Darle al MCP una llave que salta todos los permisos "para que funcione".

→ No validar inputs y dejar que el modelo mande cualquier cosa a tu base.

→ Exponer las 127 herramientas el día uno. Empieza por 5 y crece.



— SOBRE EL AUTOR

Jorge *Riquelme.* Fundador · Otherway Labs

Conecto plataformas internas con IA para empresas. En **Otherway Labs** construimos los MCPs que dejan a Claude operar sobre los sistemas propios de cada negocio. Esta guía nace de hacerlo en casa primero.

— PRÓXIMOS PASOS

Tres formas *de seguir avanzando.*

— PARA EMPRESAS
Sesión gratuita

¿Hay procesos en tu empresa que *deberían correr solos*?

Agenda un diagnóstico. Vemos qué automatizar con IA y por dónde partir.

Agenda en otherwaylabs.cl →

— PARA EQUIPOS
Capacitación

Forma a tu equipo para que *use IA todos los días*.

Capacitaciones prácticas en Claude para que el equipo automatice su propio trabajo.

Escríbenos en otherwaylabs.cl →

— ACCESO ANTICIPADO
Grupo WhatsApp · Gratis

Recibe estos recursos *antes que nadie*.

Grupo exclusivo donde publico cada guía, MCP y skill antes de subirla a YouTube o Instagram. Sin esperar a que salgan los videos.

Únete en chat.whatsapp.com →