CLAUDE FOR AI ENGINEERING · CDMX · IBERO

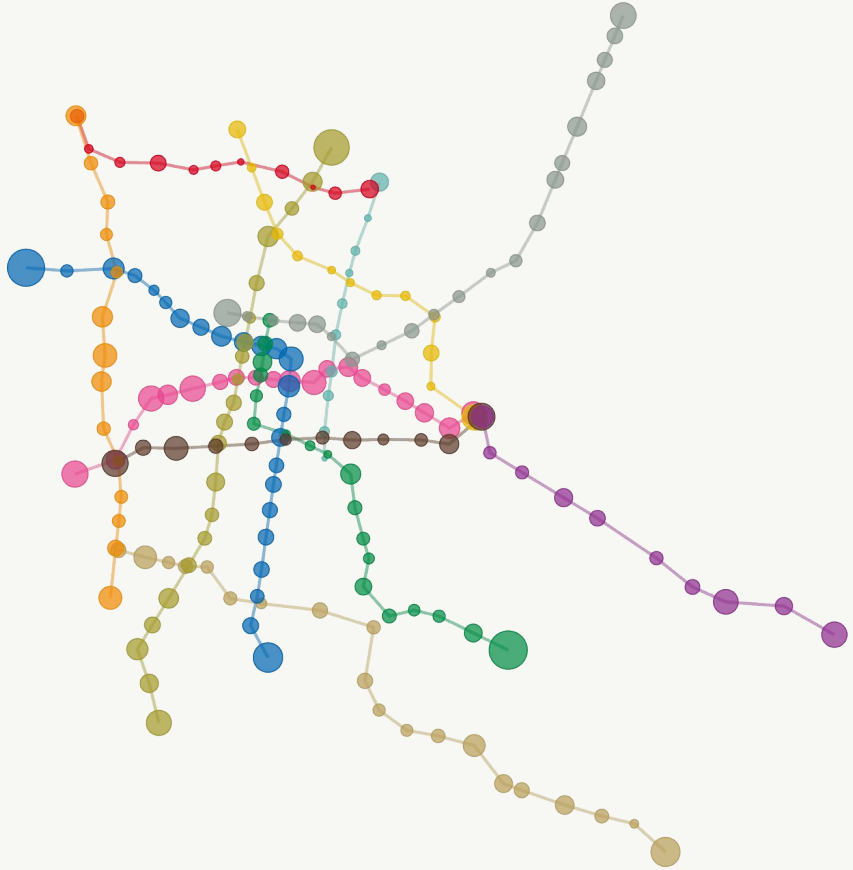
Datos abiertos over MCP

Claude que consulta fuentes reales

Taller práctico · Python + FastMCP · 40 min



Dante Bazaldua
Cofundador y CTO de Spot2.mx



Afluencia del Metro, julio de 2026 · datos.cdmx.gob.mx



Claude Code



Claude Code
with MCP servers

https://nikolaymiroshnychenko.medium.com/rapid-ui-prototyping-with-figma-android-mcp-servers-2f42fd7184cb

¿Qué alcaldía reporta más falta de agua?

¿A qué hora se llama más al 911?

¿Cuánta gente sube al Cablebús un domingo?

¿Dónde hay más choques los viernes?

¿Cuántos meses estuvo cerrada la Línea 12?

¿Qué nombre se registra más en Álvaro Obregón?

¿Dónde hay WiFi gratis cerca de mí?

¿Cuánta gente entró a la estación Pantitlán en julio de 2026?

¿Alguna idea?

¿Las mujeres usan más Ecobici que antes?

¿Qué mes tiene el peor aire?

¿Cuántas denuncias por ruido hay en mi colonia?

¿Qué estación del Metro perdió más gente desde 2019?

¿Qué es lo que más le piden al gobierno por *0311?

¿Cuánto presupuesto tiene mi alcaldía?

¿Hay Ecobici libre cerca de mi Metro ahorita?

¿En qué colonia hay más robos a transeúnte?

¿Cuántas bodas hay cada 14 de febrero?

Cada pregunta tiene detrás un conjunto de datos público · datos.cdmx.gob.mx

Pantitlán · julio de 2026

4,623,233

personas entraron en 31 días

≈149,137 al día

Quienes dijeron «más de 3 millones», acertaron.

Suma sus cuatro líneas

1 · 1.06 M	5 · 1.25 M	9 · 1.41 M	A · 0.91 M

línea · millones de entradas en el mes

Las 5 estaciones con más entradas

julio de 2026 · entradas en el mes



Pantitlán recibe 1.6 veces lo que la segunda.

¿Y Claude lo sabía? Veamos qué contesta con y sin herramientas.

Afluencia diaria del Metro CDMX · datos.cdmx.gob.mx · CC-BY-4.0 · afluencia = entradas, no ocupación de trenes

EL PROBLEMA

Sin herramientas, Claude estima. Con tu servidor MCP, responde con el dato.

Claude sin herramientas

«No puedo darte la cifra: julio de 2026 es posterior a lo que sé. Como referencia aproximada, unos 4 a 5 millones de entradas al mes… pero eso es una estimación.»

estimación

Claude + tu servidor MCP

«En julio de 2026 entraron **4,623,233 personas** a Pantitlán, contando sus cuatro líneas. Unas 149,137 al día. La fuente es datos.cdmx.gob.mx.»

dato con fuente

Misma pregunta y mismo modelo, dos sesiones de claude -p (17-oct-2026). La cifra coincide con la base del taller.

QUÉ TE LLEVAS

Al terminar tendrás un servidor MCP conectado a CC

Lo que vas a tener

- Un servidor MCP en tu laptop: 8 tools, un resource y un prompt sobre el Metro
- Conectado a Claude Code por HTTP, con una acción que escribe y pide tu aprobación
- Una evaluación de 10 preguntas que mide si Claude elige bien
- Un mapa de 16 años del Metro que Claude extiende con tus tools

Lo que vas a poder responder

1. ¿Qué es una tool y qué agrega MCP?
   (Solo repaso, ya se vio antes)
2. ¿Cuándo conviene MCP y cuándo una CLI o una API?
3. ¿Cómo diseño tools que el modelo elija bien?
4. ¿Dónde van los controles si una tool escribe?
5. ¿Cómo sé si Claude elige bien?

Todo queda en el repo y lo repites en casa con uv. Al final volvemos a esta lámina.



Dante Bazaldua

Cofundador y CTO de **Spot2.mx**

Ingeniería en Tecnologías de Cómputo y Telecomunicaciones y Maestría en Gestión de la Innovación Tecnológica · Ibero

Software para el experimento CMS del CERN
Microbioma en UC San Diego

Profesor en la Ibero: algoritmos, ciberseguridad y sistemas de comunicaciones

Un modelo convierte texto en texto: saber y actuar hay que dárselo

DEFINICIÓN

LLM (modelo de lenguaje grande): un modelo que, dado un token, predice el token que sigue. Por sí solo no consulta ni ejecuta nada.



Le falta SABER

lo que pasó después de su entrenamiento · lo que vive en tus sistemas · de dónde sale cada cifra

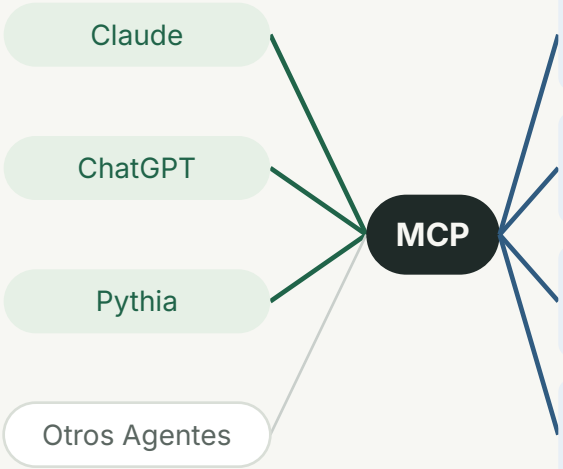
Le falta HACER

escribir, enviar, registrar · calcular con garantía · consultar en el momento

La forma de dárselo es una herramienta: una **tool**.

En Spot2, cada capacidad nace con su API y su MCP

Dónde lo usamos



Pythia: nuestro agente en Google Chat, también cliente MCP

Algunos servidores MCP

Djinn · datos
44 tools: 7 fuentes y Metabase

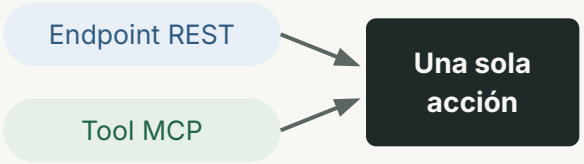
Colossu · operación
65 tools: contratos, CRM y más

Spot2.mx · en ChatGPT
Toda la plataforma disponible para agentes

Argos · archivos
archivos, anotaciones y skills

En Spot2 todos los sistemas deben de poder ser accesibles por agentes (first-class citizens)

Diseñamos API/MCP first



Cada capacidad nueva nace con su endpoint y su tool, sobre la misma lógica.

Conectores para Claude y ChatGPT; Grok, próximamente. Hoy construyes una versión chiquita de esto, con datos públicos.

Clonemos repo para empezar

https://github.com/danteese/claude-mcp-ibero-2026

Instala uv y sqlite3

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



Una tool es una función que el modelo pide y tu código ejecuta

DEFINICIÓN

Tool (herramienta): una función con nombre, descripción y parámetros que el modelo puede pedir que se ejecute.

DEFINICIÓN

Function calling: la petición que emite el modelo: el nombre de la tool y los argumentos en JSON.

DEFINICIÓN

Agente: un modelo que repite el ciclo (pedir una tool, leer el resultado, decidir) hasta resolver la tarea.

1

Tú preguntas

«¿Cuánta gente entró a Pantitlán?»

2

Claude pide la tool

```
afluencia_metro(estacion="Pantitlán")
```

3

Tu código la ejecuta

```
{"total": 4623233, "fuente": "datos.cdmx…"}
```

4

Claude responde con el dato

«Entraron 4,623,233 personas…»

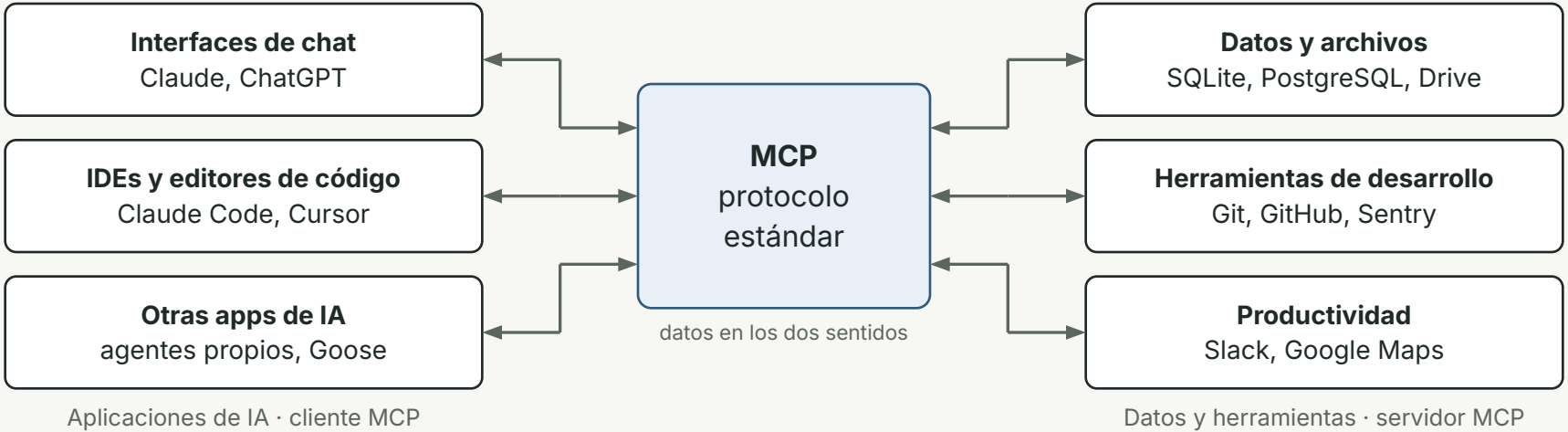
El modelo nunca ejecuta nada: solo pide.

EL CONCEPTO

MCP es el estándar para conectar la IA con tus sistemas

DEFINICIÓN

MCP (Model Context Protocol): un estándar abierto para conectar aplicaciones de IA con sistemas externos. «Como un puerto USB-C para aplicaciones de IA» (modelcontextprotocol.io).



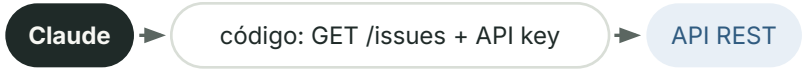
MCP es el idioma común, no un intermediario: cada app implementa el cliente una vez y cada sistema su servidor. Con 3 apps y 4 sistemas son 3 + 4 = **7 implementaciones**, no 3 × 4 = 12 integraciones.

Adaptado de modelcontextprotocol.io · «What is MCP?» (docs 2026-07-28)

¿POR QUÉ MCP?

La IA puede usar API, CLI o MCP: elige según la situación

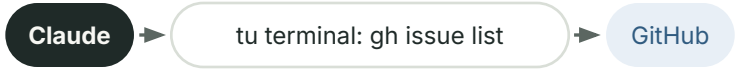
API



Cada servicio documenta distinto; la credencial se la das tú al agente.

CONVIENE SI
el flujo es siempre el mismo y no lo decide un modelo: un script o un pipeline

CLI



Corre en tu máquina con todos tus permisos. Sin tu terminal, no hay CLI.

CONVIENE SI
trabajas en tu máquina con una herramienta que el modelo ya conoce: git, gh, kubectl

MCP



El host ve cada tool: pide aprobación y registra cada llamada; OAuth por persona.

CONVIENE SI
el modelo elige en el momento, para varias personas y con auditoría; también sin terminal (Claude.ai, ChatGPT)

Con tools bien diseñadas, CLI y MCP empatan en éxito; MCP gasta más contexto (de 1.3× a 80× más tokens en cinco tareas de GitHub). ¿Muchos pasos sobre muchos datos? Que el modelo escriba código que llame las tools.

Zechner (ago-2025) · Scalekit, MCP vs CLI benchmark (2026) · Osmani, MCP deep dive · Anthropic (nov-2025)

¿POR QUÉ MCP?

MCP te da control y alcance; lo pagas en contexto y riesgo

Lo que te da

Funciona sin tu terminal: Claude.ai, ChatGPT o un agente en la nube no corren tus CLIs; sí se conectan a un MCP remoto.

Autorización por usuario: OAuth 2.1 viene en el estándar; cada persona entra con su cuenta.

El host ve cada tool: puede pedir aprobación y registrar cada llamada; una CLI corre con todos tus permisos.

Tú defines la superficie: expones intenciones, no 40 endpoints, con un contrato escrito para el modelo.

Lo escribes una vez: estándar neutral en la Linux Foundation; lo usan Claude, ChatGPT, Gemini, Copilot y VS Code.

Lo que te cuesta

Contexto: cada definición ocupa tokens; el MCP de GitHub: 93 tools, unos 55,000 tokens.

Composición: cada paso pasa por el modelo; para muchos pasos, conviene que escriba código que llame tus tools.

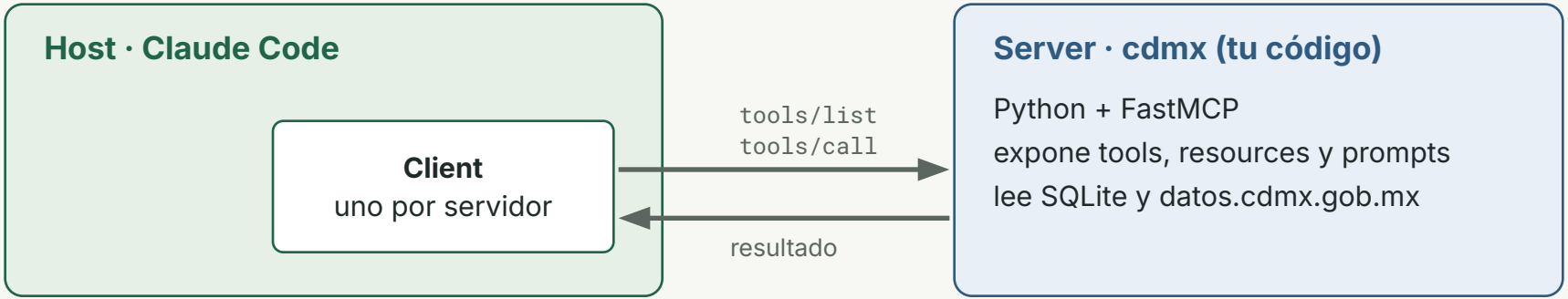
Seguridad: tool poisoning, prompt injection en los resultados, token passthrough. Las anotaciones son pistas, no garantías.

Operación: otro servicio que desplegar, versionar y probar con cada host.

MCP spec 2026-07-28 (autorización, seguridad, tools) · Anthropic (dic-2025, nov-2025) · Huntley vía Willison (ago-2025) · OWASP

EL CONCEPTO

En MCP participan tres actores: host, client y server



DEFINICIÓN

Host: la aplicación de IA donde conversas: Claude Code, Claude Desktop.

DEFINICIÓN

Client: el conector que el host crea para hablar con un servidor.

DEFINICIÓN

Server: el programa que expone capacidades. Hoy escribes uno.

Transporte: por dónde viajan los mensajes. Hoy, HTTP en tu propia máquina (127.0.0.1:8765). También existe stdio: el host lanza el servidor como un proceso.

EL CONCEPTO

MCP tiene tres capas obligatorias; el resto es opcional

Extensiones: Tasks (trabajos largos) · MCP Apps (UI en la conversación) · auth empresarial — opt-in

El servidor ofrece: Tools ✓ · Resources ✓ · Prompts ✓ — **El cliente ofrece**: Elicitation — opcional

Utilidades: progreso · cancelación · logging · autocompletado · paginación — opcional

Autorización: OAuth 2.1, solo en HTTP; cada persona entra con su cuenta — opcional

Mensajes
JSON-RPC 2.0, sin estado

Versionado
versión y capacidades en cada petición

Patrones
petición · multi round-trip · suscripción

obligatorio

Transporte: stdio, el host lanza el proceso · Streamable HTTP ✓ — elige uno

✓ = lo usamos hoy. Obsoleto = sigue en la spec al menos 12 meses, pero no se debe adoptar.

modelcontextprotocol.io/specification/2026-07-28 · basic · changelog · client/sampling · client/roots · extensions

EL CONCEPTO

Tres piezas, una pregunta: ¿quién decide cuándo ocurre?

Tool

decide **el modelo**

Una función que el modelo invoca cuando la necesita.

`afluencia_metro(...)`

SE DEFINE CON

`name` · `description` · `inputSchema`

Resource

decide **la aplicación**

Un dato de solo lectura, con dirección (URI), que la app incluye como contexto.

`cdmx://fuentes`

SE DEFINE CON

`uri` · `name` · `mimeType`

Prompt

decides **tú**

Una plantilla de instrucciones que eliges ejecutar; en Claude Code, un comando /.

`/informe_estacion`

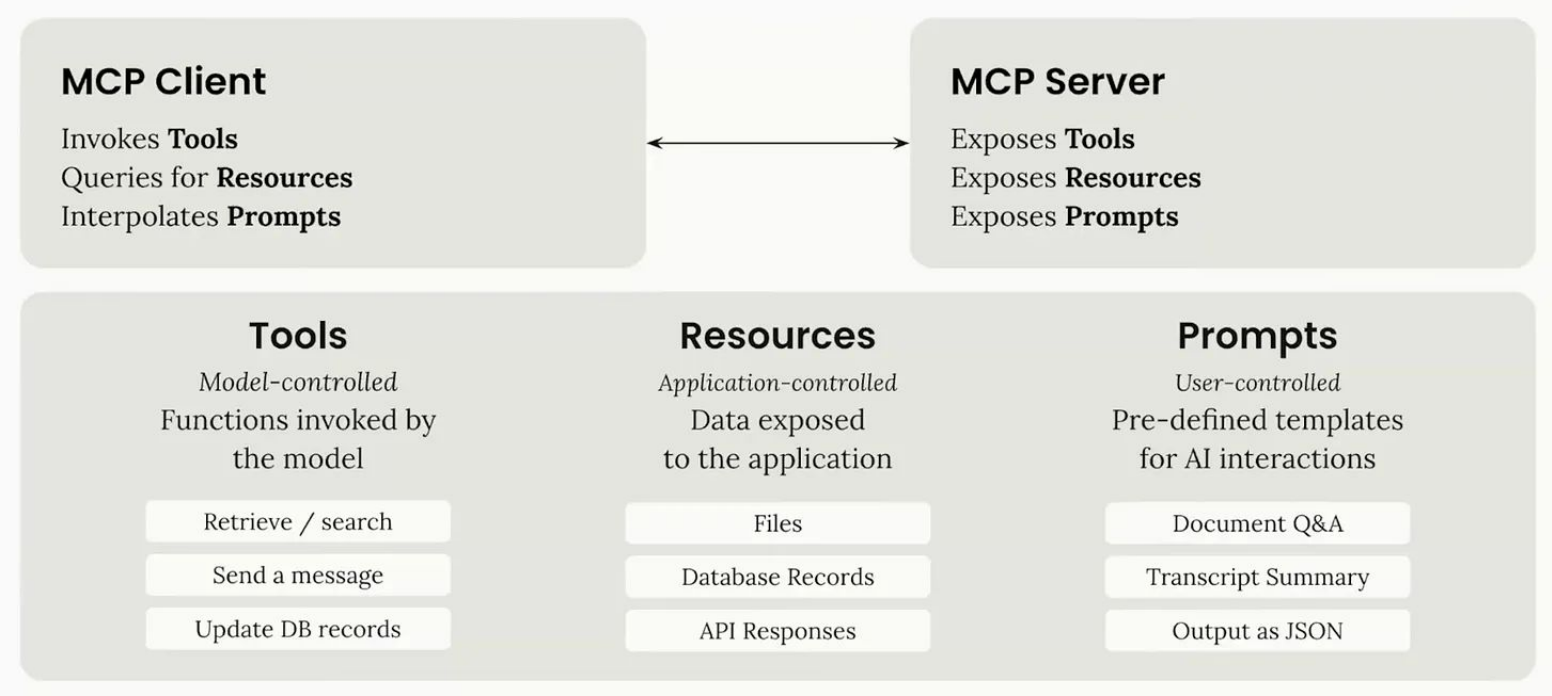
SE DEFINE CON

`name` · `description` · `arguments`

Nuestro servidor tiene las tres. La mayoría expone todo como tools.

EL CONCEPTO

Tres piezas, una pregunta: ¿quién decide cuándo ocurre?



QUÉ VAMOS A CONSTRUIR

Construimos un servidor MCP por HTTP con FastMCP

Tipos de servidor según el transporte

stdio · local

Claude Code lanza tu servidor como proceso y le habla por stdin/stdout. Sin red.

HTTP local ✓

Hoy: tu servidor corre aparte en 127.0.0.1:8765 y Claude Code se conecta por URL.

HTTP remoto

Desplegado en internet con OAuth 2.1; así lo usan Claude.ai o ChatGPT.

FastMCP

DEFINICIÓN

FastMCP: framework de Python para servidores MCP: de tu función saca el schema y habla el protocolo por stdio o HTTP.

Las tools no cambian; el transporte es un argumento:

```
mcp.run()                    # stdio
mcp.run(transport="http")    # HTTP
```

```
app.mount("/mcp", mcp.http_app())
FastMCP.from_fastapi(app)
```

gofastmcp.com (running-server, integrations/fastapi) · fastapi.tiangolo.com · MCP spec 2026-07-28 (transports)

Construyamos

Cuatro pasos. Cada uno termina con una comprobación.

1. Una tool
2. Diseño
3. Una acción
4. Medir

```
git clone https://github.com/danteese/claude-mcp-ibero-2026
cd claude-mcp-ibero-2026
uv run preparar.py      # datos → SQLite, sin internet
uv run verificar.py     # ¿todo listo?
```

DEFINICIÓN

Datos abiertos: datos que publica una institución con una licencia que permite reusarlos. Aquí: afluencia del Metro, CC-BY-4.0.

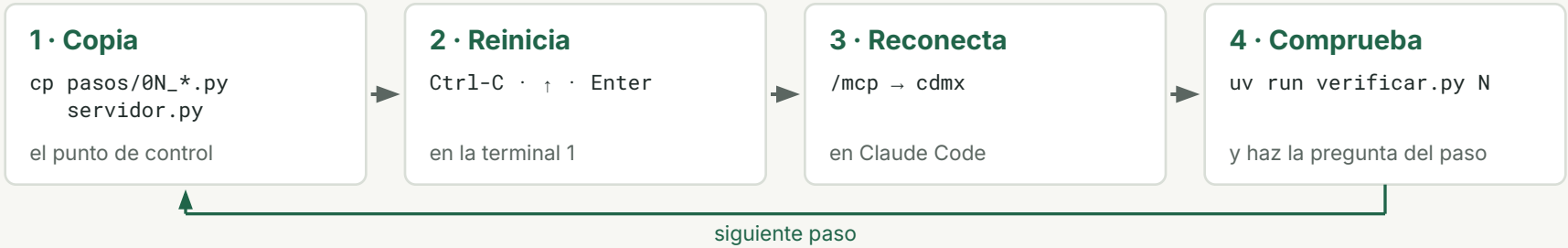
DEFINICIÓN

uv: herramienta que instala la versión de Python y las dependencias correctas, y corre el script. No instalas nada más.

¿Atorado? Abre claude en el repo y pídele: «guíame paso a paso».

CONSTRUYAMOS

En cada paso: copia, reinicia, reconecta y comprueba



Qué tiene tu servidor en cada paso

Paso 1	`servidor.py (ya lo trae)`	`buscar_datasets_cdmx`
Paso 2	`pasos/02_consulta.py`	`+ buscar_estacion_metro · afluencia_metro · ranking_estaciones_metro`
Paso 3	`pasos/03_accion.py`	`+ registrar_reporte · consultar_reportes`
Paso 4	`pasos/04_final.py`	`+ ecobici_cerca_de_metro · actualizar_mapa_ecobici`

Windows (cmd): copy pasos\02_consulta.py servidor.py · copia solo el comando, sin comentarios · ¿Te perdiste? En Claude Code: «guíame paso a paso».

Una función con docstring ya es una tool

```python
@mcp.tool(annotations={"readOnlyHint": True})
async def buscar_datasets_cdmx(
    texto: Annotated[str, Field(
        description="Tema a buscar, en español.")],
    limite: Annotated[int, Field(ge=1, le=10)] = 5,
) -> dict:
    """Busca conjuntos de datos en datos.cdmx.gob.mx.
    Úsala para descubrir qué datos existen."""
```

DEFINICIÓN

FastMCP: librería de Python: un decorador convierte tu función en tool.

DEFINICIÓN

Docstring: el texto entre triples comillas: es la descripción que lee el modelo.

DEFINICIÓN

Schema: tipos y límites de los parámetros; sale de tus type hints.

TERMINAL 1

```
uv run servidor.py
```

TERMINAL 2

```
claude
```

COMPRUEBA

```
uv run fastmcp list
http://127.0.0.1:8765/mcp
```

PRUÉBALO

«¿Qué datos abiertos hay sobre bicicletas?»

PASO 1 · HAZLO AHORA

servidor.py ya es el paso 1 **1** 2 3 4

Levanta tu servidor y conéctalo a Claude Code

TU SERVIDOR AHORA · 1 TOOL

`buscar_datasets_cdmx`

Haz esto

1. Terminal 1: levanta tu servidor; se queda corriendo, es normal
   `uv run servidor.py`
2. Terminal 2: entra a la carpeta del repo y abre Claude Code
   `cd claude-mcp-ibero-2026 · claude`
3. Acepta con las flechas las dos preguntas del inicio
   `Yes, I trust this folder · Use this MCP server`
4. Comprueba que cdmx esté conectado
   `/mcp`
5. Pregúntale
   «¿Qué datos abiertos hay sobre bicicletas?»

Lo que debes ver

En /mcp: **cdmx · connected**

Claude usa buscar_datasets_cdmx y te lista conjuntos de Ecobici: viajes, cicloestaciones...

uv run verificar.py 1 → **Todo en orden.**

Si no funciona

cdmx no aparece: ¿corre la terminal 1? ¿Abriste claude dentro de la carpeta?

Esta tool usa internet; si falla la red, sigue con el paso 2.

Reto si terminas antes: lee el docstring en servidor.py: ¿con eso sabrías cuándo usar la tool?

Diseña para quien lee el contrato: el modelo

1 · Por intención

✗ `ejecutar_sql(query)`

```
afluencia_metro(
  estacion, linea,
  desde, hasta)
```

No por endpoint: el dominio vive en el schema.

2 · Errores que corrigen

'insurgente' coincide con varias estaciones: Insurgentes (Línea 1), Insurgentes Sur (Línea 12). Repite con el nombre exacto.

Claude se corrige en un turno.

3 · Respuestas cortas

`alcance · unidad`
`notas · fuente`

DEFINICIÓN

Contexto: lo que el modelo tiene presente en la conversación. Es finito: Claude Code avisa a los 10k tokens.

Devuelve la serie agregada, no 38,804 filas.

✓ uv run verificar.py 2

Reto: ¿qué parámetros le darías a comparar_estaciones(a, b)?

Tu servidor aprende el Metro: pregunta por Pantitlán

TU SERVIDOR AHORA · 4 TOOLS

buscar_datasets_cdmx · **buscar_estacion_metro** · **afluencia_metro** · **ranking_estaciones_metro**

Haz esto

1. Copia el punto de control del paso 2
   `cp pasos/02_consulta.py servidor.py`
2. Terminal 1: reinicia el servidor
   `Ctrl-C · ↑ · Enter`
3. Claude Code: reconecta para que vea las tools nuevas
   `/mcp → cdmx → Reconnect`
4. Comprueba (en otra pestaña, o en Claude Code empieza con !)
   `uv run verificar.py 2`
5. Pregúntale
   «¿Cuánta gente entró a Pantitlán en julio de 2026?»

Lo que debes ver

verificar.py: ✓ afluencia_metro suma la Línea 12… · **Todo en orden.**

Claude: «Entraron **4,623,233 personas**…», con la fuente

Claude ya ve 4 tools en /mcp

Si no funciona

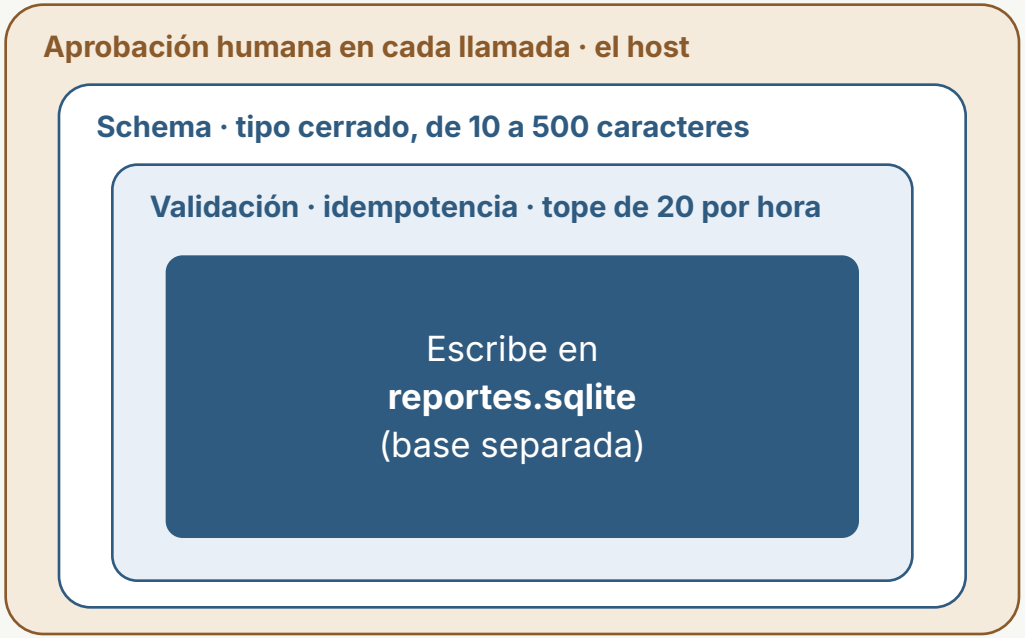
«Falta datos/cdmx.sqlite»: corre uv run preparar.py

Claude no ve afluencia_metro: reinicia el servidor y reconecta

Avanzamos juntos cuando la mayoría vea **4,623,233**.

Reto si terminas antes: pregunta por «insurgente», mal escrito. ¿Qué parámetros le darías a comparar_estaciones(a, b)?

Si la tool escribe, los controles van en el código



✓ uv run verificar.py 3

Claude propone; el código decide.

```
@mcp.tool(meta={
  "anthropic/
   requiresUserInteraction":
   True})
```

DEFINICIÓN

Idempotente: repetir la llamada no duplica el efecto: mismo folio.

Lo que regresa también es entrada: el texto ciudadano vuelve como dato no confiable.

Tu servidor ya escribe: levanta un reporte

TU SERVIDOR AHORA · 6 TOOLS

buscar_datasets_cdmx · buscar_estacion_metro · afluencia_metro · ranking_estaciones_metro · **registrar_reporte** · **consultar_reportes**

Haz esto

1. Copia el punto de control del paso 3
   `cp pasos/03_accion.py servidor.py`
2. Reinicia y reconecta
   `Ctrl-C · ↑ · Enter` · `/mcp → cdmx → Reconnect`
3. Comprueba
   `uv run verificar.py 3`
4. Pídele un reporte y apruébalo
   «Levanta un reporte: el elevador de Pantitlán, Línea A, no funciona»
5. Pídelo otra vez igual; luego, para la Línea 2

Lo que debes ver

Claude Code pregunta **«Do you want to proceed?»** → Yes

Folio **CDMX-00001**; la segunda vez, el mismo folio

En la Línea 2 no se registra: Pantitlán no está en esa línea

Si no funciona

No pide aprobación: ¿copiaste el 03, reiniciaste y reconectaste?

No pongas datos personales reales: el texto viaja al modelo

Avanzamos juntos cuando la mayoría vea su folio.

Reto si terminas antes: ¿qué otro control le pondrías en el código?

EL CONCEPTO

Un MCP da acceso; una skill le enseña a usarlo

DEFINICIÓN

Skill (Agent Skill): una carpeta con un SKILL.md (nombre, descripción e instrucciones) que el agente carga sólo cuando la tarea lo pide. Estándar abierto: agentskills.io.

Un plugin junta las dos cosas

Servidor MCP	qué puede hacer: tools, datos y acciones
Skill	cómo usarlo: cuándo, en qué orden y qué revisar
Opcional	hooks, comandos y subagentes

skills/metro-cdmx/SKILL.md

```
---
name: metro-cdmx
description: Preguntas de afluencia del
  Metro y Ecobici con el servidor cdmx.
---
1. Confirma la estación con buscar_estacion_metro.
2. Afluencia = entradas, no ocupación.
3. Cita la fuente; separa dato y opinión.
4. ¿Mapa? actualizar_mapa_ecobici.
```

Al iniciar: sólo name y description (~100 tokens) → **Si la tarea coincide**: lee el SKILL.md completo → **Si hace falta**: abre sus scripts y referencias

Se distribuyen juntos en el formato de cada host, no del estándar MCP: plugins de Claude Code (.mcp.json + skills/), de ChatGPT y Codex, y extensiones de Gemini CLI. Lo estándar: el servidor MCP puede servir sus skills (extensión SEP-2640; en FastMCP, SkillsDirectoryProvider).

Mide si Claude elige bien: 10 preguntas reales, 10 sesiones

10/10

tool y argumentos correctos
medido el 17-oct-2026

DEFINICIÓN

Evaluación (eval): casos reales con la tool esperada, para medir si Claude elige bien.

La primera corrida dio 9/10. Claude confirmó la estación antes de levantar el reporte: fallaba la métrica, no el modelo.

Si falla de verdad, itera la descripción y el schema, no el system prompt.

✓ uv run evaluar.py

Reto: escribe la pregunta que haga fallar a tu servidor.

Tu servidor sale a internet: Ecobici y el mapa

TU SERVIDOR AHORA · 8 TOOLS

buscar_datasets_cdmx · buscar_estacion_metro · afluencia_metro · ranking_estaciones_metro · registrar_reporte · consultar_reportes · **ecobici_cerca_de_metro** · **actualizar_mapa_ecobici**

Haz esto

1. Copia el punto de control del paso 4
   `cp pasos/04_final.py servidor.py`
2. Reinicia y reconecta
   `Ctrl-C · ↑ · Enter` · `/mcp → cdmx → Reconnect`
3. Comprueba
   `uv run verificar.py 4`
4. Pregúntale
   «¿Hay Ecobici cerca del Metro Insurgentes?»
5. Pídele el mapa y ábrelo (open mapa/index.html)
   «Actualiza el mapa de Ecobici para el Metro Insurgentes»

Lo que debes ver

Bicis disponibles a menos de 500 m del Metro Insurgentes; cambian cada minuto

En el mapa, la sección **«Ecobici en vivo cerca del Metro»**

La evaluación (10 preguntas) la corre Dante; en casa: uv run evaluar.py

Si no funciona

Ecobici necesita internet; las tools del Metro siguen funcionando

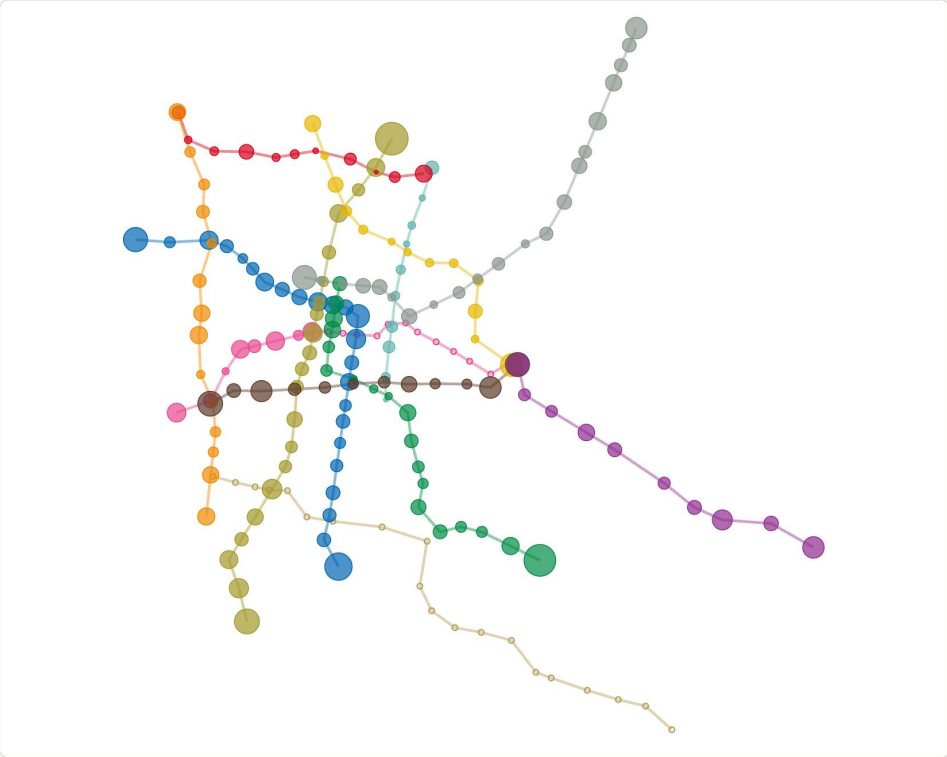
En Windows, el mapa se abre con start mapa\index.html

Avanzamos juntos cuando la mayoría vea las bicis en el mapa.

Reto si terminas antes: escribe la pregunta que haga fallar a tu servidor y arregla la descripción.

EL CIERRE

El mismo dato, ahora en un mapa para personas



abr-2020

COVID: 1.37 millones de entradas diarias en toda la red

jun-2021

Línea 12 en cero hasta dic-2022

oct-2022

Línea 12 y la mitad oriente de la Línea 1 en cero (esta captura)

Círculo = entradas diarias promedio. Anillo vacío = cero. Entradas, no ocupación.

EL CIERRE

Claude extiende el mapa; tú lees lo que añade

```
«Usa el servidor cdmx para saber cuándo estuvo en cero la Línea 12 y agrega a
 mapa/index.html un aviso que aparezca solo en esos meses.»
```

afluencia_metro → edita mapa/index.html → aviso en el mapa

42 s en el ensayo

Lo dice el dato
Línea 12: 0 entradas
de jun-2021 a dic-2022

Lo añadió el modelo
«cerrada tras el colapso del 3 de mayo de 2021»: no viene de la tool

RECAP

Recap: tienes un servidor MCP y sabes cuándo usarlo

Lo que construiste

- Un servidor MCP con 8 tools, un resource y un prompt (tu servidor.py)
- Conectado a Claude Code por HTTP, con una acción que pide tu aprobación
- Una evaluación de 10 preguntas (evaluar.py)
- Un mapa de 16 años del Metro que Claude extendió con tus tools

Lo que te llevas

1. El modelo solo pide; tu código ejecuta. MCP estandariza cómo se publican y descubren las tools.
2. MCP si el modelo decide en runtime, para varios usuarios o sin tu terminal. Si no, CLI o API.
3. Por intención y con errores que corrigen: el modelo solo lee nombre, descripción y schema.
4. En el código: validación, idempotencia, tope y aprobación del host. Nunca solo en el prompt.
5. Con preguntas reales. Si falla, arregla la descripción, y separa el dato de lo que añade el modelo.

Es la lámina del inicio, ahora con respuestas. Repasa en el repo: pasos/01 a 04 y docs/.

Ahora te toca

Clona, cambia el dato, inventa la tool, rompe el mapa.

Otro dato

calidad del aire, *0311, GTFS: 475 conjuntos en datos.cdmx.gob.mx

Otra tool

comparar_estaciones, alerta de cierre, tendencia anual

Otra vista

etiquetas, filtro por línea, cambio contra el año anterior

Tu sistema

tu inventario, tu CRM, tus tickets: como hicimos en Spot2

Comparte lo que construyas · github.com/danteese/claude-mcp-ibero-2026

Esto ya lo puedes conectar hoy

Hay 142,221 servidores MCP activos (MCP Toplist, 15-oct-2026). Algunos que vale la pena probar:

CREAR EN 3D
Blender
«Arma una escena low-poly de una playa con palmeras»
local · lo instalas

FABRICAR
Impresoras 3D
«Rebana el modelo de Blender y mándalo a imprimir»
local · lo instalas

MÚSICA
Ableton Live
«Haz una canción de house con intro, build-up y drop»
local · lo instalas

DISEÑO
Figma
«Convierte este frame en un componente de React»
remoto · pegas la URL

TU CASA
Home Assistant
«Apaga todas las luces de la casa»
URL de tu Home Assistant

ROBÓTICA
Robots con ROS
«Mueve el brazo a la posición inicial»
local · lo instalas

WEB
Chrome DevTools
«¿Por qué carga lento mi página?»
local · lo instalas

DOCS AL DÍA
Context7
«¿Cómo monto FastMCP en FastAPI? Usa la doc actual»
remoto · pegas la URL

BÚSQUEDA WEB
Parallel
«Investiga quién soporta Agent Skills y cita fuentes»
remoto · pegas la URL

MÉXICO Y LATAM
Mercado Pago
«Genera el código de Checkout Pro y crea usuarios de prueba»
remoto · pegas la URL



Gracias

Dante Bazaldua · stdout.mx · linkedin.com/in/dantebh

¿Tu tool se entiende? Diseña, describe y mide

1 · Diseña para el agente

Resultados, no endpoints: una tool por intención, no una por endpoint.

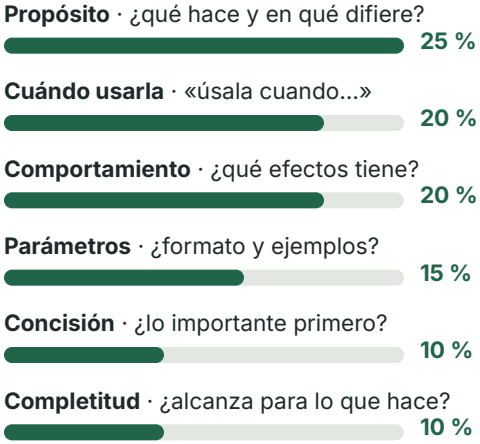
Argumentos planos: tipos simples, enums y valores por defecto.

Respuestas cortas: agrega, pagina y devuelve nombres, no IDs.

Errores que enseñan: isError con el siguiente paso; el modelo se corrige.

Schmid (ene-2026) · Anthropic · spec MCP 2026-07-28

2 · ¿Se entiende? Criterios TDQS



Nota de 1 a 5 y nivel de A a F (Glama)

3 · Mide y corrige

Golden set: pedidos directos, indirectos y negativos, con lo esperado.

Mide: tool y argumentos correctos, llamadas, tokens y errores.

Lee las transcripciones: Claude puede proponer mejores descripciones.

Valida aparte: con casos que no usaste para ajustar.

OpenAI, Optimize metadata · Anthropic, Writing tools for agents

si falla: corrige la descripción, no el prompt

En el taller: tools por intención y errores que corrigen (paso 2) y evaluar.py con 10 casos, uno negativo (paso 4). Siguiente: califica tus 8 tools con TDQS.

Glosario del taller

LLM	predice el texto que sigue a un texto
Tool	el modelo la pide; tu código la ejecuta
Function calling	la petición: tool + argumentos JSON
Agente	repite pedir tool → leer → decidir
MCP	estándar abierto: apps de IA ↔ sistemas
Host	app donde conversas: Claude Code, Desktop
Client	conector del host a un solo servidor
Server	expone tools, resources y prompts
Transporte	por dónde viajan: stdio o HTTP
JSON-RPC	formato de los mensajes de MCP
OAuth 2.1	autorización por usuario en HTTP
Elicitation	el servidor pide un dato a la persona
Resource	dato de solo lectura con URI
Prompt	plantilla que tú ejecutas (comando /)
Skill	guía SKILL.md; el agente la carga si aplica
API	interfaz para que un programa use a otro
CLI	programa de terminal: git, gh, kubectl
FastMCP	librería de Python para servidores MCP
uv	instala Python y dependencias, y ejecuta
Schema	tipos y límites de los parámetros
Docstring	descripción de la tool que lee el modelo
Contexto	lo que el modelo tiene presente; finito
Idempotente	repetirla no duplica el efecto
Eval	casos con respuesta esperada para medir
Prompt injection	instrucciones ocultas en datos o tools

Para aprender más

MCP, la fuente

Qué es MCP
modelcontextprotocol.io/docs/2026-07-28/getting-started/intro

Especificación 2026-07-28
modelcontextprotocol.io/specification/2026-07-28

Registry oficial de servidores
registry.modelcontextprotocol.io

Diseñar tools

Writing effective tools for agents · Anthropic
anthropic.com/engineering/writing-tools-for-agents

MCP best practices · Phil Schmid
philschmid.de/mcp-best-practices

Code execution with MCP · Anthropic
anthropic.com/engineering/code-execution-with-mcp

Skills y empaquetado

Agent Skills
agentskills.io

Skills sobre MCP · SEP-2640
modelcontextprotocol.io/seps/2640-skills-extension

MCP Bundles (.mcpb)
github.com/modelcontextprotocol/mcpb

Construir

FastMCP
gofastmcp.com

Claude Code y MCP
code.claude.com/docs/en/mcp

uv
docs.astral.sh/uv/

Medir

Demystifying evals for AI agents · Anthropic
anthropic.com/engineering/demystifying-evals-for-ai-agents

Optimize metadata · OpenAI
developers.openai.com/plugins/guides/optimize-metadata

Tool Definition Quality Score · Glama
github.com/glama-ai/tool-definition-quality-score

Descubrir servidores

Conectores de Claude
claude.com/connectors

Glama
glama.ai/mcp/servers

MCP Toplist
mcptoplist.com

Este taller: github.com/danteese/claude-mcp-ibero-2026 · todas las fuentes, decisiones y pruebas, en docs/ · datos: datos.cdmx.gob.mx

Enlaces comprobados el 19-oct-2026