

4 HERRAMIENTAS PARA
HACER VIBE CODING CON CLAUDE CODE

Links oficiales, instalación paso a paso y cómo comprobar que cada una quedó funcionando

Herramienta	Link oficial
1. Ponytail	github.com/DietrichGebert/ponytail
2. OmniRoute	github.com/diegosouzapw/OmniRoute
3. Graphify	github.com/Graphify-Labs/graphify
4. Agent Skills	github.com/addyosmani/agent-skills

ANTES DE EMPEZAR

Necesitas Claude Code instalado y abierto en la carpeta de tu proyecto. Los comandos que empiezan con / se escriben dentro de Claude Code. Los demás se escriben en la terminal.

Ponytail y Agent Skills: solo requieren Claude Code. OmniRoute: Node.js. Graphify: Python 3.10 o superior y uv (o pipx). Agent Skills con acceso por Git: Git instalado.

Orden sugerido: Agent Skills y Ponytail primero (dos comandos cada una), después Graphify y al final OmniRoute, que lleva más pasos.

SEGURIDAD ANTES DE INSTALAR

Instala solo desde los links de arriba. Hay paquetes con nombres parecidos que no son los oficiales.

Los plugins y skills le dan instrucciones a tu agente y pueden ejecutar comandos en tu computadora. Dale una revisada rápida al repositorio antes de instalarlos y no los pruebes por primera vez en un proyecto de un cliente.

Nunca pegues claves de API en el chat ni las subas a un repositorio.

1 Ponytail: que escriba menos código

Es una instrucción (skill) que le pide al agente escribir solo el código que la tarea necesita, sin recortar validaciones, manejo de errores ni seguridad.

1. Abre Claude Code en tu proyecto.
2. Agrega el marketplace del plugin:

ESCRÍBELO EN CLAUDE CODE

```
/plugin marketplace add DietrichGebert/ponytail
```

3. Instala el plugin:

ESCRÍBELO EN CLAUDE CODE

```
/plugin install ponytail@ponytail
```

4. Comprueba que funciona: escribe /ponytail sin nada más. Te dice el nivel activo. Según su página, queda activo en cada sesión.

Niveles: /ponytail lite, /ponytail full, /ponytail ultra o /ponytail off para apagarlo. También tiene /ponytail-review (revisa tus cambios) y /ponytail-audit (revisa todo el repositorio).

SOBRE LAS CIFRAS

Lo del video sale de sus propias pruebas. Su página reporta cerca de 53% menos código y 45% menos tokens frente a trabajar sin la skill (39 tareas, modelo Opus 5.5, 5 ejecuciones por tarea). Tus resultados pueden variar según el proyecto y el modelo.

Si el comando no aparece, cierra Claude Code y ábrelo de nuevo. Para ver el panel de costos y tokens necesitas Claude Code 2.1.290 o posterior, en terminal o app de escritorio.

2 OmniRoute: cambia de modelo cuando se acaba tu límite

Es una pasarela gratuita y de código abierto. Tú conectas tus cuentas de proveedores de IA, tiene un solo endpoint local y, si a un modelo se le acaba la cuota o falla, pasa al siguiente disponible.

1. Instala Node.js desde nodejs.org (versión LTS). Su documentación no fija una versión mínima; verifícalo en el repositorio.
2. En la terminal, instala OmniRoute y arráncalo:

TERMINAL

```
npm install -g omniroute
omniroute
```

3. Abre el panel en **http://localhost:20128** (ese es el puerto por defecto; si quieres otro, usa omniroute --port 3000). En el primer inicio, su guía indica que la contraseña inicial por defecto es CHANGEME. Cámbiala en cuanto entres.
4. Conecta tus proveedores. En el panel: Providers, elige el proveedor y conecta tu cuenta. Por ejemplo, Gemini CLI se conecta con inicio de sesión de Google. Lee los términos de cada proveedor: los que su documentación marca con tos: avoid no se usan en el enrutamiento automático.
5. Crea un combo (la cadena de modelos entre los que cambia). En el panel: Combos, Create New, y agrega los modelos en el orden en que quieres que los use. También existe el modelo auto, que arma una ruta con los proveedores que conectaste.
6. Crea tu clave. En el panel: API Manager, New Key. Guárdala.
7. Conecta Claude Code. Lo más corto es lanzarlo desde OmniRoute, que ejecuta claude con la configuración activa:

TERMINAL (opción A)

```
omniroute launch
```

Opción B: configurarlo tú con variables de entorno antes de abrir Claude Code. La URL va sin /v1 y debes reiniciar Claude Code después de cambiarlas:

TERMINAL, macOS o Linux (opción B)

```
export ANTHROPIC_BASE_URL="http://localhost:20128"
export ANTHROPIC_AUTH_TOKEN="TU_CLAVE_DE_OMNIROUTE"
export CLAUDE_CODE_ENABLE_GATEWAY_MODEL_DISCOVERY="1"
claude
```

En Windows (PowerShell) es igual, pero cada línea empieza con $env: y lleva = en vez de export, por ejemplo: $env:ANTHROPIC_BASE_URL="http://localhost:20128"

8. Comprueba que funciona. Con OmniRoute encendido, esta prueba debe devolver una respuesta (necesitas al menos un proveedor conectado):

TERMINAL

```
curl http://localhost:20128/v1/chat/completions -H "Authorization: Bearer TU_CLAVE_DE_OMNIROUTE" -H "Content-Type: application/json" -d '{"model":"auto","messages":[{"role":"user","content":"Hola"}]}'
```

LEE ESTO ANTES DE ACTIVARLO

Los 1,600 millones de tokens al mes son una estimación que suma las capas gratuitas de proveedores de terceros (unos 1,620 millones según su página). No son tokens que regale OmniRoute, cada proveedor pone sus propios límites y la cifra cambia. Lo de más de 300 proveedores sí coincide: su página habla de 375, con más de 150 con capa gratuita.

Tu código y tus prompts pasan por el proveedor que esté respondiendo. No lo uses con código confidencial de clientes sin revisar las políticas de ese proveedor. Y otros modelos pueden rendir distinto a Claude.

Si algo de estos pasos no coincide con lo que ves, sigue la documentación del repositorio: cambia con las versiones.

3 Graphify: tu código como un mapa

Convierte tu proyecto (código, documentos, configuraciones, PDFs) en un grafo de conocimiento que el agente puede consultar. El código se analiza en tu computadora, sin modelo de IA.

1. Revisa que tengas Python 3.10 o superior y que instales uv (recomendado) o pipx.
2. Instala el paquete. Ojo: se llama **graphifyy**, con doble y. Otros paquetes graphify en PyPI no son los oficiales.

TERMINAL

```
uv tool install graphifyy
```

Alternativas: pipx install graphifyy, o pip install graphifyy (puede requerir configurar el PATH). Si después no encuentra el comando graphify, ejecuta uv tool update-shell y abre una terminal nueva.

3. Registra la skill en tu asistente (se hace una vez):

TERMINAL

```
graphify install
```

4. Dentro de la carpeta de tu proyecto, deja la configuración para que Claude Code consulte el grafo:

TERMINAL

```
graphify claude install
```

Opcional: graphify hook install reconstruye el grafo solo cuando haces commit o cambias de rama.

5. Abre Claude Code en esa carpeta y construye el grafo:

ESCRÍBELO EN CLAUDE CODE

```
/graphify .
```

6. Comprueba que funciona: se crea la carpeta graphify-out con tres archivos: graph.html (el mapa interactivo, ábrelo en el navegador), GRAPH_REPORT.md (resumen) y graph.json (el grafo completo). También puedes preguntarle desde la terminal:

TERMINAL

```
graphify query "¿cómo funciona el inicio de sesión?"
```

SOBRE EL AHORRO DE TOKENS

Su página no mide cuántos tokens ahorra la versión gratuita. El 80% que menciona es de Graphify Cloud, su producto de pago. Lo que sí hace es darle al agente un mapa del proyecto para que no tenga que releerlo todo. Compruébalo en tu propio proyecto comparando el consumo antes y después.

4 Agent Skills: un proceso para cada etapa

Es un conjunto de 25 skills de ingeniería de software de Addy Osmani (licencia MIT): 24 del ciclo de vida más una que explica cómo usarlas. Cubren definir, planear, construir, verificar, revisar y publicar.

1. En Claude Code, agrega el marketplace e instala el plugin:

ESCRÍBELO EN CLAUDE CODE

```
/plugin marketplace add addyosmani/agent-skills
/plugin install agent-skills@addy-agent-skills
```

2. Si falla con un error de SSH (Permission denied publickey), usa la dirección HTTPS:

ESCRÍBELO EN CLAUDE CODE

```
/plugin marketplace add https://github.com/addyosmani/agent-skills.git
/plugin install agent-skills@addy-agent-skills
```

Si sigue fallando, ejecuta esto una vez en la terminal y repite el paso 1:

TERMINAL

```
git config --global url."https://github.com/".insteadOf
git@github.com:
```

3. Comprueba que funciona: escribe /plugin y confirma que agent-skills aparece instalado. Luego escribe / y busca los comandos de la tabla.

Comando	Para qué sirve
/spec	Definir qué vas a construir antes de escribir código
/plan	Dividir el trabajo en tareas pequeñas
/build	Construir paso a paso (/build auto implementa todo tras una aprobación)
/test y /ship	Verificar con pruebas y preparar el despliegue

Importante: /review choca con un comando que ya trae Claude Code. Usa la forma con nombre: /agent-skills:review. Hay más comandos en su página.

✅ CHECKLIST FINAL

- Ponytail: /ponytail muestra el nivel activo.
- OmniRoute: el panel abre en localhost:20128 y la prueba con curl responde.
- Graphify: existe la carpeta graphify-out con graph.html.
- Agent Skills: /plugin lista agent-skills y aparecen /spec y /plan.

¿Quieres llevar la IA más allá del código y automatizar
la operación de tu negocio, pero no sabes por dónde empezar?
Escríbenos directamente a nuestro agente de WhatsApp
y agendemos una llamada.
+52 81 2188 4401