GUÍA PASO A PASO

5 repos de GitHub para agentes de IA

Los cinco repositorios de la semana 1 con su enlace, qué hacen, cómo se instalan según su README, su licencia y qué ejecutan en tu computadora. Antes, dos comandos de solo lectura para revisar cualquier repo sin clonarlo.

PARA	Claude Code y otros agentes de programación
PROBADO CON	GitHub CLI (gh api) · PowerShell 5.1 · Windows 11
ACTUALIZADA	30 de octubre de 2026
DOCUMENTACIÓN	code.claude.com/docs/en/plugins/security
ARTÍCULO	ecomlabs.dev/g/repos
AUTOR	Luis Morales · @luismoralesdev

Índice

Antes de empezar	Cómo usar esta lista
Paso 01	Revisa licencia y actividad
Paso 02	Lee qué corre solo
Repo 01	obra/superpowers
Repo 02	thedotmack/claude-mem
Repo 03	addyosmani/agent-skills
Repo 04	pbakaus/impeccable
Repo 05	manaflow-ai/cmux
Final	Checklist

ANTES DE EMPEZAR

Cómo usar esta lista

Estos son los cinco repositorios de la semana 1. Los datos (licencia, último push, versión y estrellas) son del 30 de octubre de 2026 y salen de la API de GitHub. Los comandos de instalación son los que publica cada README: no instalamos ni ejecutamos ninguno de estos proyectos para escribir la guía.

Lo único que corrimos son los dos comandos de solo lectura de los pasos 01 y 02, con GitHub CLI 2.96 en PowerShell y en Git Bash. Sirven para revisar cualquier repo antes de instalarlo.

PASO 01

Revisa licencia y actividad

```
gh api repos/obra/superpowers --jq "{license: .license.spdx_id, archived, pushed_at, stars:
.stargazers_count}"
```

Respuesta real:

```
{"archived":false,"license":"MIT","pushed_at":"2026-10-29T23:35:34Z","stars":309369}
```

- `license`: identificador SPDX. Si sale `NOASSERTION`, GitHub no pudo clasificarla: lee el archivo `LICENSE`.
- `archived`: si es `true`, el dueño lo congeló y no habrá arreglos.
- `pushed_at`: fecha del último push. Meses sin cambios en un proyecto de agentes es mala señal.
- `stars`: popularidad, no calidad ni seguridad.

Cambia `obra/superpowers` por el repo que quieras revisar. Necesitas `gh` con sesión iniciada.

PASO 02

Lee qué corre solo

Un plugin de Claude Code puede ejecutar código con tus permisos, fuera del sandbox, mediante hooks y servidores MCP. Lee sus archivos sin clonar nada:

```
gh api -H "Accept: application/vnd.github.raw" repos/obra/superpowers/contents/hooks/hooks.json
```

El encabezado `Accept` devuelve el archivo tal cual. Busca qué evento dispara cada hook y qué comando corre, y abre ese script. Haz lo mismo con `.mcp.json` y la carpeta `bin/`. Rutas en estos cinco:

REPO	HOOKS	SERVIDOR MCP
obra/superpowers	`hooks/hooks.json`	No trae
thedotmack/claude-mem	`plugin/hooks/hooks.json`	`plugin/.mcp.json`
addyosmani/agent-skills	No registra (scripts opcionales en `hooks/`)	No trae
pbakaus/impeccable	`plugin/hooks/hooks.json`	No trae
manaflow-ai/cmux	No es plugin: es una app	No aplica

REPO 01

obra/superpowers

`github.com/obra/superpowers` · Licencia MIT · v7.0.1 · unas 309 mil estrellas.

Un método de desarrollo hecho de skills: antes de programar pregunta qué quieres, te muestra el diseño por partes, escribe un plan de tareas cortas y lo ejecuta con TDD y revisiones entre tareas. Las skills se activan solas.

```
/plugin install superpowers@claude-plugins-official
```

Qué corre: un hook `SessionStart` (al iniciar, limpiar o compactar) que ejecuta un script con bash. Su compañero visual opcional carga un logo desde su web con la versión; se apaga con `SUPERPOWERS_DISABLE_TELEMETRY`.

REPO 02

thedotmack/claude-mem

`github.com/thedotmack/claude-mem` · Licencia Apache-2.0 · v15.7.0 · unas 106 mil estrellas.

Memoria entre sesiones: captura lo que hace el agente, lo comprime con un modelo de IA y lo inyecta en sesiones futuras. Usa un servicio local, SQLite, Chroma y un servidor MCP de búsqueda. Pide Node 20 o más reciente.

```
npx claude-mem install
```

Dentro de Claude Code: `/plugin marketplace add thedotmack/claude-mem` y `/plugin install claude-mem`. `npm install -g claude-mem` solo instala la librería.

Qué corre: hooks en 9 eventos (entre ellos `UserPromptSubmit`, `PreToolUse` y `PostToolUse`), un servidor MCP local y un servicio con visor web. Instala Bun y uv si faltan.

> **CUIDADO**
> El instalador ofrece iniciar sesión en el servicio alojado de claude-mem. Lo que comprime va al proveedor que elijas (su servicio, tu llave de OpenRouter o Gemini, o tu plan de Anthropic). Para saltarte el inicio de sesión: `--provider` o `CLAUDE_MEM_ONLINE_OPTIN=false`. Excluye datos sensibles con etiquetas `<private>`.

REPO 03

addyosmani/agent-skills

`github.com/addyosmani/agent-skills` · Licencia MIT · 0.6.14 · unas 112 mil estrellas.

25 skills de ingeniería y 9 comandos para todo el ciclo: `/spec`, `/plan`, `/build`, `/test`, `/constraints`, `/review`, `/webperf`, `/code-simplify` y `/ship`. Cada skill exige evidencia antes de dar algo por terminado.

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

Qué corre: nada por su cuenta; no registra hooks. Usa `/agent-skills:review` para no chocar con el `/review` de Claude Code. Si el `marketplace add` falla por SSH, su README propone la URL HTTPS del repositorio.

REPO 04

pbakaus/impeccable

`github.com/pbakaus/impeccable` · Licencia Apache-2.0 · cli-v4.8.0 · unas 97 mil estrellas.

Vocabulario de diseño para tu agente: 24 comandos (`critique`, `audit`, `polish`, `bolder`…) y un detector de 62 reglas deterministas, sin modelo ni llave de API.

```
npx impeccable install
```

Después, `/impeccable init` una vez por proyecto. En Claude Code también: `/plugin marketplace add pbakaus/impeccable` y luego `/plugin`.

Qué corre: hooks en `SessionStart`, `PostToolUse` (tras `Edit` o `Write`) y `Stop`, que ejecutan su propio binario; ese binario viene con la skill o se descarga una vez en `~/.impeccable/bin/`.

REPO 05

manaflow-ai/cmux

`github.com/manaflow-ai/cmux` · GPL-3.0-or-later (servidor en BUSL-1.1) · v0.65.2.2 · unas 28.6 mil estrellas.

Terminal nativa de macOS para correr varios agentes: pestañas verticales con rama y PR, avisos cuando un agente te espera, navegador integrado y SSH. Solo macOS.

```
brew tap manaflow-ai/cmux
brew install --cask cmux
```

Qué corre: es tu terminal, con acceso completo a tu shell. Su CLI y su socket pueden mandar teclas, leer la pantalla y manejar el navegador. Se actualiza sola con Sparkle.

FINAL

Checklist

- [ ] La licencia está clara; si dice `NOASSERTION`, leíste el `LICENSE`.
- [ ] El repo no está archivado y tiene pushes recientes.
- [ ] Leíste sus hooks y sabes qué corre en cada evento.
- [ ] Sabes si arranca un servidor MCP o un servicio local.
- [ ] Sabes a dónde manda datos y cómo apagarlo.
- [ ] Instalaste uno a la vez, en un proyecto que no es de producción.