

Memoria infinita para Claude Code · 2 herramientas open source

CATEGORÍA · EMPEZAR CON CLAUDE

Claude Code arranca cada sesión con contexto vacío y eso es el dolor universal de todos los que lo usamos en serio: avanzas un proyecto durante horas, cierras la terminal, abres al día siguiente y Claude no se acuerda ni del stack ni de las decisiones que tomaron anteayer.

QUÉ VAS A ENCONTRAR

- 01 Qué es y qué resuelve
- 02 Para qué sirve
- 03 Dónde aplicarlo
- 04 Instalación en 3 rutas (con comandos)
- 05 El código en GitHub
- 06 Comprobación y errores frecuentes

FICHA DEL RECURSO

Qué es, para qué sirve y cómo se implementa

FUNCIÓN

Claude Code arranca cada sesión con contexto vacío y eso es el dolor universal de todos los que lo usamos en serio: avanzas un proyecto durante horas, cierras la terminal, abres al día siguiente y Claude no se acuerda ni del stack ni de las decisiones que tomaron anteayer.

PARA QUÉ SIRVE

Claude Code arranca cada sesión con contexto vacío y eso es el dolor universal de todos los que lo usamos en serio: avanzas un proyecto durante horas, cierras la terminal, abres al día siguiente y Claude no se acuerda ni del stack ni de las decisiones que tomaron anteayer. Hay dos sistemas open-source que arreglan eso desde ángulos opuestos: Claude-Mem (thedotmack · Apache 2.0 · ~75k★) es un plugin de Claude Code que se instala con `npx claude-mem install`, se mete en cinco hooks (SessionStart, UserPromptSubmit, PostToolUse, Stop, SessionEnd) y captura observaciones automáticamente con dashboard local en `localhost:37777`, almacenamiento local en `~/.claude-mem/` (SQLite + Chroma) y soporte también para Gemini CLI y OpenCode; MemPalace (mempalaceofficial · MIT · 96.6% R@5 raw y 98.4% hybrid R@5 en LongMemEval) va por el camino del control total con CLI puro y 100% local — le picas `uv tool inst`

DÓNDE APLICARLO

Nivel Intermedio. Temas: memoria, claude-code. Incluye el código de thedotmack/claude-mem, mempalace/mempalace, xiaowu0162/LongMemEval.

CÓMO IMPLEMENTARLO

- 1 Ruta rápida: pégale a Claude Code el enlace del repositorio con el prompt de la página siguiente — instala y comprueba solo.
- 2 Ruta manual: clona el repositorio y corre los comandos de instalación de tu lenguaje.
- 3 Revisa el README antes de instalar: requisitos, versión y licencia.
- 4 Pruébalo en un caso pequeño antes de usarlo en trabajo real.
- 5 Revisa qué permisos y llaves pide antes de conectarlo a tus cuentas.

EL RECURSO EN DETALLE

Qué resuelve y cómo funciona

Claude Code arranca cada sesión con contexto vacío y eso es el dolor universal de todos los que lo usamos en serio: avanzas un proyecto durante horas, cierras la terminal, abres al día siguiente y Claude no se acuerda ni del stack ni de las decisiones que tomaron anteayer.

Hay dos sistemas open-source que arreglan eso desde ángulos opuestos: Claude-Mem (thedotmack · Apache 2.0 · ~75k★) es un plugin de Claude Code que se instala con `npx claude-mem install`, se mete en cinco hooks (SessionStart, UserPromptSubmit, PostToolUse, Stop, SessionEnd) y captura observaciones automáticamente con dashboard local en `localhost:37777`, almacenamiento local en `~/.claude-mem/` (SQLite + Chroma) y soporte también para Gemini CLI y OpenCode; MemPalace (mempalaceofficial · MIT · 96.6% R@5 raw y 98.4% hybrid R@5 en LongMemEval) va por el camino del control total con CLI puro y 100% local — le picas `uv tool install mempalace`, luego `mempalace init` en tu proyecto, y tú decides cuándo indexar (mine), cuándo buscar (search), cuándo cargar el wake-up de contexto al arrancar y cuándo barrer transcripts viejos (sweep), con scope per-project vía `--wing` y 29 herramientas MCP listas para que Claude las llame.

La cubre las 7 secciones — el dolor de la amnesia entre sesiones, tabla comparativa de 10 criterios lado a lado (instalación, prerequisites, modo de trabajo, cuándo guarda, cómo carga, dónde viven los datos, privacidad, observabilidad, licencia, mejor para), Claude-Mem paso a paso con sus 3 prerequisites (Node 18+, Claude Code instalado, Bun autoinstalado) y comando único `npx claude-mem install + dashboard` en `localhost:37777`, MemPalace paso a paso con sus 3 prerequisites (Python 3.9+, uv de Astral, Claude Code) y la secuencia `uv → uv tool install mempalace → mempalace init + warning crítico de auto-save hook` (sin él las sesiones de Claude Code expiran en 30 días), decision helper de 3 perfiles para que escojas (me quiero olvidar del tema → Claude-Mem · quiero todo local sin nube → MemPalace · soy beginner → Claude-Mem), warning explícito de que correr las dos al mismo tiempo hace que los hooks choquen y se duplique la captura, 5 prompts copy-paste en español natural sin pa' para que Claude Code haga todo el trabajo (diagnóstico de qué memoria tengo activa hoy, instalar Claude-Mem y verificar que arrancó, instalar uv si no lo tengo y luego MemPalace en este proyecto, migrar de Claude-Mem a MemPalace exportando lo que ya tengo, desinstalar limpio cualquiera de las dos con backup de settings.json) y checklist final en 3 fases (antes de empezar, durante el install, después del reinicio) con 9 ítems en total.

CTAs primarios al de Claude-Mem en GitHub y al de MemPalace en GitHub; 3 related guides al ecosistema de memoria y workflow.

INSTALACIÓN

claude-mem

Plugin de Claude Code que captura observaciones por hooks con dashboard local y storage SQLite.

EL CÓDIGO ESTÁ AQUÍ

<https://github.com/thedotmack/claude-mem>
Licencia Apache-2.0

Ruta 1 · La más simple. Abre Claude Code, pégale este mensaje y él hace el resto:

PÉGASELO A CLAUDE CODE

Instala esto por mi y explicame lo que haces:
<https://github.com/thedotmack/claude-mem>

Antes de correr nada, leete el README y dime:

1. que hace exactamente y para que sirve
2. que requisitos necesito instalados
3. que permisos o llaves va a pedir

Luego instalalo paso a paso, avisame de cada comando antes de ejecutarlo, y al final comprueba que quedo funcionando.

Ruta 2 · Manual, si prefieres controlar cada paso desde la terminal:

INSTALACIÓN MANUAL · TERMINAL

```
# 1. Clona el repositorio
git clone https://github.com/thedotmack/claude-mem.git
cd claude-mem
```

```
# 2. Sigue el apartado de instalacion del README
cat README.md
```

```
# 3. Instalalo como plugin (dentro de Claude Code)
/plugin marketplace add thedotmack/claude-mem
```

```
# 4. Comprueba la lista de plugins
/plugin
```

Ruta 3 · Sin instalar nada. Entra al enlace de arriba y lee el README y los ejemplos para decidir si te sirve antes de bajarlo.

INSTALACIÓN

mempalace

CLI 100% local para indexar, buscar y mantener memoria persistente — control total sobre cuándo guarda.

EL CÓDIGO ESTÁ AQUÍ

<https://github.com/mempalace/mempalace>
Licencia MIT

Ruta 1 · La más simple. Abre Claude Code, pégale este mensaje y él hace el resto:

PÉGASELO A CLAUDE CODE

Instala esto por mi y explicame lo que haces:
<https://github.com/mempalace/mempalace>

Antes de correr nada, leete el README y dime:

1. que hace exactamente y para que sirve
2. que requisitos necesito instalados
3. que permisos o llaves va a pedir

Luego instalalo paso a paso, avisame de cada comando antes de ejecutarlo, y al final comprueba que quedo funcionando.

Ruta 2 · Manual, si prefieres controlar cada paso desde la terminal:

INSTALACIÓN MANUAL · TERMINAL

```
# 1. Clona el repositorio
git clone https://github.com/mempalace/mempalace.git
cd mempalace

# 2. Sigue el apartado de instalacion del README
cat README.md

# 3. Pruebalo
./mempalace --help    # o el comando que indique el README
```

Ruta 3 · Sin instalar nada. Entra al enlace de arriba y lee el README y los ejemplos para decidir si te sirve antes de bajarlo.

INSTALACIÓN

LongMemEval

Benchmark para evaluar memoria de largo plazo en LLMs — usado por MemPalace para validar.

EL CÓDIGO ESTÁ AQUÍ

<https://github.com/xiaowu0162/LongMemEval>

Ruta 1 · La más simple. Abre Claude Code, pégale este mensaje y él hace el resto:

PÉGASELO A CLAUDE CODE

Instala esto por mi y explicame lo que haces:

<https://github.com/xiaowu0162/LongMemEval>

Antes de correr nada, leete el README y dime:

1. que hace exactamente y para que sirve
2. que requisitos necesito instalados
3. que permisos o llaves va a pedir

Luego instalalo paso a paso, avisame de cada comando antes de ejecutarlo, y al final comprueba que quedo funcionando.

Ruta 2 · Manual, si prefieres controlar cada paso desde la terminal:

INSTALACIÓN MANUAL · TERMINAL

1. Clona el repositorio

```
git clone https://github.com/xiaowu0162/LongMemEval.git
cd LongMemEval
```

2. Sigue el apartado de instalacion del README

```
cat README.md
```

3. Sigue el ejemplo de uso del README para tu caso

Ruta 3 · Sin instalar nada. Entra al enlace de arriba y lee el README y los ejemplos para decidir si te sirve antes de bajarlo.

ANTES DE DARLO POR HECHO

Comprobación y errores frecuentes

- 1 No quedó instalado si no lo has probado: corre el ejemplo del README y comprueba que devuelve lo esperado.
- 2 Si un comando falla, casi siempre falta un requisito previo (versión de Node, Python o git). Pégale el error a Claude Code y pídele que lo resuelva.
- 3 Si pide una llave o token, créala en la cuenta correspondiente y guárdala fuera del código, en una variable de entorno.
- 4 Antes de conectarlo a cuentas reales, pruébalo con datos de prueba o una cuenta secundaria.
- 5 Comprueba la licencia si lo vas a usar en trabajo de cliente: define qué puedes hacer con el código.

DATOS DEL RECURSO

Tipo: Guía paso a paso

Categoría: Empezar con Claude

Nivel: Intermedio

Licencia: Apache 2.0

Lenguaje: Python

Temas: memoria, claude-code

Código: <https://github.com/thedotmack/claude-mem>

Código: <https://github.com/mempalace/mempalace>

Código: <https://github.com/xiaowu0162/LongMemEval>

El ecosistema de IA cambia rápido: confirma en el repositorio que el proyecto sigue activo y que los comandos no han cambiado desde la fecha de esta ficha.

SIGUIENTE PASO

Instálalo hoy, en pequeño.

Elige la ruta 1, pégale el enlace a Claude Code y pruébalo en un caso real chico. Un recurso instalado y probado vale más que diez guardados para después.

MÁS GUÍAS GRATIS EN [377CREATIVE.COM/RECURSOS](https://377creative.com/recursos)