

The screenshot shows the GitHub repository page for 'thedotmack/claude-mem'. The repository is public and has 333 branches and 287 tags. The commit history is visible, showing recent updates to various files like .agentrules, .agents/plugins, .claude-plugin, .claude, .codex-plugin, .github, .plan, .windurf/rules, cursor-hooks, docker, docs, evals/webench, install, openclaw, plans, plugin, ragtime, and scripts. The right sidebar contains an 'About' section with a description of the project: 'Persistent Context Across Sessions for Every Agent - Captures everything your agent does during sessions, compresses it with AI, and injects relevant context back into future sessions. Works with Claude Code, OpenClaw, Codex, Gemini, Hermes, Copilot, OpenCode + More'. It also lists related tags like 'ai', 'rag', 'embeddings', 'artificial-intelligence', 'ai-agents', 'claude', 'memory-engine', 'long-term-memory', 'rag', 'anthropic', 'chromadb', 'ai-memory', 'mem0', 'claude-code', 'supermemory', 'openmemory', 'claude-agents', 'claude-agent-sdk', and 'claude-code-plugin', 'claude-skills'. The repository has 76.9k stars, 274 watching, and 6.6k forks. The latest release is v13.2.0.

# Claude-Mem

*Claude-Mem es un sistema avanzado de memoria persistente y compresión de contexto diseñado para agentes de IA y Claude Code. Esta herramienta permite a desarrolladores de software, ingenieros de datos y arquitectos técnicos capturar decisiones, acciones y herramientas utilizadas en sesiones de programación para reinyectarlas automáticamente en el futuro. Su arquitectura de divulgación progresiva optimiza el uso de tokens, permitiendo que el asistente recuerde contextos complejos sin saturar la memoria de trabajo.*

[Visitar Sitio Oficial](#) [Preguntar a ChatGPT](#) [Preguntar a Claude](#) [Preguntar a Grok](#)

## Contenido del Dossier

- [Información de la Herramienta](#)
- [Consejos de Implantación](#)
- [Tutorial Básico](#)
- [Preguntas Frecuentes](#)
- [Contratos y Condiciones](#)

## INFORMACIÓN DE LA HERRAMIENTA

### Qué y para quién es

Claude-Mem es un sistema de **memoria persistente y compresión de contexto** diseñado específicamente para agentes de IA, con un enfoque primario en Claude Code. Su función técnica es capturar cada acción, decisión y herramienta utilizada durante una sesión de programación, comprimirla mediante IA y reinyectar ese conocimiento en sesiones futuras de forma automática.

Está dirigido a desarrolladores de software, ingenieros de datos y arquitectos técnicos que utilizan asistentes de codificación basados en CLI o IDEs y sufren la "amnesia" del modelo al cambiar de rama, reiniciar la sesión o trabajar en proyectos de larga duración. Es ideal para entornos profesionales donde la trazabilidad de las decisiones técnicas y la continuidad del flujo de trabajo son críticas.

### Principal ventaja profesional

Desde una perspectiva técnica, lo que hace a Claude-Mem indispensable es su **arquitectura de "Divulgación Progresiva"**. A diferencia de otros sistemas RAG (Generación Aumentada por Recuperación) que inundan el contexto con información irrelevante agotando la ventana de tokens, Claude-Mem permite a la IA buscar primero un índice compacto y solo "desplegar" los detalles profundos de una decisión o código cuando es estrictamente necesario. En mis pruebas, esto permite que el agente "recuerde" por qué se tomó una decisión de arquitectura hace tres meses sin consumir miles de tokens de entrada innecesarios.

### Para quién no es

No es una herramienta para usuarios de perfiles no técnicos o "prompt engineers" que solo usan la interfaz web de Claude. Requiere manejo de terminal y comprensión básica de cómo funcionan los servidores MCP (Model Context Protocol). No es apto para entornos con políticas de privacidad extremadamente restrictivas que prohíban el uso de bases de datos locales tipo SQLite o bases de datos vectoriales (ChromaDB) para almacenar trazas de actividad del desarrollador, a menos que se configure cuidadosamente.

### Funcionalidades clave

- **Persistencia Multi-Sesión:** El contexto sobrevive a cierres de terminal y cambios de proyecto.
- **Workflow de 3 Capas:** Herramientas integradas (search, timeline, get\_observations) que optimizan el gasto de tokens hasta en un 10x.
- **Web Viewer UI:** Interfaz visual local en puerto 37777 para monitorizar en tiempo real qué está "pensando" y recordando el agente.
- **Endless Mode (Beta):** Arquitectura de memoria biomimética para sesiones de codificación extremadamente largas.
- **Privacidad Granular:** Permite usar etiquetas de exclusión para evitar que datos sensibles se guarden en la memoria persistente.
- **Integración con Claude Desktop:** Permite consultar la memoria de sesiones de terminal desde la aplicación de escritorio de Claude.

### Precios

Claude-Mem es un proyecto **Open Source** bajo licencia Apache 2.0.

- **Versión Gratuita:** Completa y funcional. No hay costes de software por la herramienta en sí.
- **Costes asociados:** El usuario debe asumir los costes de la API de Anthropic (u otros proveedores como Gemini) por el procesamiento de las observaciones y las consultas de búsqueda, aunque la herramienta está optimizada para minimizarlos.

### Perfil del usuario

- **Empresas de desarrollo de software:** Equipos que mantienen repositorios grandes y complejos donde el contexto histórico se pierde fácilmente.
- **DevOps e Ingenieros de SRE:** Para mantener un historial de intervenciones y soluciones aplicadas en el pasado.
- **Desarrolladores Solopreneurs:** Que saltan entre múltiples proyectos y necesitan que su asistente de IA mantenga el contexto de cada uno de forma independiente.

### Nivel técnico requerido

- **Uso:** Medio. Requiere familiaridad con CLIs (Claude Code, Gemini CLI).
- **Instalación/Configuración:** Medio-Alto. Instalación vía npx y configuración de variables de entorno y

archivos JSON.

- **Conocimientos necesarios:** Node.js (v18+), gestión de paquetes npm/npx y nociones de Model Context Protocol (MCP).

Ejemplos de uso profesional

- **Continuidad en Bug Fixing:** Un desarrollador trabaja en un error complejo un viernes tarde. El lunes, al iniciar Claude Code, el agente ya tiene inyectado el resumen de las pruebas fallidas y las hipótesis descartadas del viernes.

- **Onboarding Asistido:** Un nuevo miembro del equipo puede "preguntar" a la memoria del proyecto (vía MCP) por qué se eligió una librería específica en lugar de otra, consultando las observaciones guardadas por sus compañeros.

- **Auditoría de Decisiones:** Generar un "corpus" de conocimiento (función build\_corpus) sobre todas las refactorizaciones realizadas en un módulo durante el último trimestre.

Uso y distribución

- **Línea de comandos (CLI):** Uso nativo a través de Claude Code y Gemini CLI.

- **Integraciones IDE:** Soporte para Cursor, VS Code (vía Roo Code/Copilot CLI), Warp y Goose.

- **Versión Visual:** Web Viewer local (<http://localhost:37777>).

Open source

El código es totalmente abierto bajo **Apache License 2.0**, lo cual es excelente para empresas que deseen auditar el código o extender su funcionalidad sin bloqueos de proveedor.

Integraciones

- **MCP (Model Context Protocol):** Es su núcleo de comunicación. Dispone de un servidor MCP nativo.

- **IDEs y Herramientas:** Integración nativa documentada para Copilot CLI, Gemini (Antigravity), Roo Code, Goose y Warp.

- **Bases de Datos:** Utiliza SQLite para almacenamiento estructurado y ChromaDB para búsqueda vectorial/semántica.

Notas finales

Veredicto técnico

Como profesional, considero que Claude-Mem es la pieza que le faltaba a los agentes de programación modernos. Mientras que otras herramientas intentan meter todo el repositorio en el contexto (RAG masivo), Claude-Mem es **selectivo y eficiente**. Vale la pena el esfuerzo inicial de configuración por la enorme ganancia en productividad que supone no tener que explicarle de nuevo al modelo qué estábamos haciendo hace dos horas. Es una herramienta de gran utilidad para el desarrollador que busca una relación simbiótica real con la IA.

Información legal, licencias y contratos

- **Licencia:** Apache 2.0. Permite uso comercial, modificación y distribución.

- **Propiedad Intelectual:** El software reside localmente; el usuario mantiene el control de sus datos, aunque estos se procesan a través de los proveedores de LLM configurados.

Otros

Es importante mencionar que el creador ha "abrazado" un token de comunidad (\$CMEM), lo cual es inusual en proyectos técnicos serios, pero no afecta a la funcionalidad técnica ni a la calidad del código del repositorio principal.

Fuentes consultadas:

- [Sitio web oficial](#)

- [Repositorio Github](#)

- [Documentación técnica](#)

## CONSEJOS DE IMPLANTACIÓN

### Aplicación profesional

Según mi experiencia, esta herramienta es un cambio de juego para consultoras de software, agencias de desarrollo y departamentos de IT internos con alta deuda técnica. El tipo de empresa que más provecho extraerá es aquella que gestiona múltiples repositorios o microservicios donde las decisiones se quedan en el aire y no se documentan adecuadamente. Al usarlo te das cuenta de que no solo es un asistente de código, sino un sistema de gestión del conocimiento tácito del equipo. El presupuesto necesario es prácticamente nulo más allá del consumo de API de Anthropic, lo cual lo hace extremadamente eficiente frente a soluciones Enterprise cerradas.

### Madurez digital requerida

- Desarrolladores senior o mid-level habituados a flujos de trabajo basados en terminal (CLI) y que ya utilicen IA en su día a día. Se requiere que el equipo entienda el concepto de "contexto de ventana" y cómo gestionarlo para no disparar costes.
- Departamentos con una cultura de documentación ágil. La empresa debe permitir el uso de herramientas locales que interactúen con APIs externas bajo el protocolo MCP.

### Plan orientativo de implantación

#### Pasos necesarios y estimaciones

- Tiempos estimados de despliegue: De 1 a 3 días para una configuración estable en un equipo de desarrollo.
- Evaluación técnica de la infraestructura local (Node.js v18+, espacio en disco para SQLite/ChromaDB) y generación de claves API necesarias para el procesamiento de memoria.
- Instalación mediante npx y configuración del servidor MCP. Creación de un proyecto piloto en un repositorio de complejidad media para validar la calidad de las "observaciones" generadas.
- Sesión técnica de 2 horas para explicar la lógica de las tres capas (search, timeline, observations) y cómo evitar el almacenamiento de datos sensibles mediante etiquetas de exclusión.
- Revisión semanal de la calidad de los resúmenes generados por la IA y ajuste del presupuesto de tokens tras el primer mes.

#### Necesidades de formación del equipo

Es vital formar al equipo en el uso de los comandos específicos del servidor MCP y en la interpretación del Web Viewer UI. Los desarrolladores deben aprender a "etiquetar" mentalmente qué acciones quieren que la memoria persista y cuáles no.

#### Perfiles necesarios

- Perfiles técnicos necesarios: Un Lead Developer o Arquitecto que supervise la configuración del servidor MCP y las reglas de privacidad global.
- Personal externo recomendado: No es estrictamente necesario, aunque un consultor en IA puede ayudar a optimizar los prompts de compresión de memoria para reducir costes de API.
- Otros: Un responsable de seguridad (CISO) para validar el flujo de datos hacia Anthropic/Google Gemini.

#### Retorno de la inversión

- La reducción del tiempo de "re-contextualización" tras interrupciones o cambios de tarea se estima en un 20-30% a partir de la segunda semana.
- Se mide a través de la reducción de tokens de entrada en sesiones largas (eficiencia de contexto) y el tiempo que tarda un desarrollador en retomar un bug tras varios días de inactividad.

#### Otros

Mi experiencia en implantaciones me lleva a pensar que la mayor barrera no es técnica, sino de confianza en el almacenamiento de trazas de actividad. Es fundamental destacar que la base de datos es local. Lo que más me gusta es el modo "Endless", ya que permite trabajar en refactorizaciones masivas sin que la IA empiece a "alucinar" por saturación de contexto. Ten en cuenta que si el equipo usa hardware muy limitado, la ejecución local de ChromaDB junto con el IDE y otros servicios podría afectar al rendimiento de la máquina.

## TUTORIAL BÁSICO

### Instalación

La instalación de Claude-Mem es automatizada, pero requiere seguir los canales oficiales para que los hooks de memoria funcionen correctamente.

- npx (Recomendado): Ejecuta `npx claude-mem install`. El instalador detectará automáticamente tu IDE (Claude Code, Cursor, Windsurf, etc.) y configurará las dependencias necesarias como Bun y uv.
- Marketplace: Si usas Claude Code, puedes instalarlo directamente con `/plugin marketplace add thedotmack/claude-mem` seguido de `/plugin install claude-mem`.
- Evita el uso de `npm install -g claude-mem`, ya que según mi experiencia esto solo instala la librería SDK y no registra los procesos en segundo plano ni los hooks de sesión necesarios para que la persistencia funcione.
- Checklist de post-instalación:
  - Verifica que el worker esté activo en `http://localhost:37777`.
  - Asegúrate de tener `Node.js >= 20`.
  - En Windows, abre PowerShell como administrador si encuentras problemas con la ejecución de scripts.

### Uso en el día a día

- Automatización total: No necesitas hacer nada para "guardar". Al usarlo te das cuenta de que el sistema captura tus acciones, herramientas usadas y cambios en archivos de forma silenciosa al final de cada turno.
- Inyección de contexto: Al iniciar una nueva sesión, verás un bloque de contexto que resume lo que hiciste anteriormente. En mi opinión profesional, esto reduce drásticamente el tiempo de re-contextualización del modelo.
- Privacidad selectiva: Usa la etiqueta `<private>` en tus prompts para contenido sensible que no quieras que sea procesado o almacenado en la base de datos de memoria.
- Visor Web: Mantén abierta la pestaña de `http://localhost:37777`. Es extremadamente útil para ver en tiempo real qué "observaciones" está guardando el sistema y monitorizar el consumo de tokens.

### Trucos de experto

- Estrategia de búsqueda en 3 capas: Cuando necesites recuperar algo del pasado, usa el flujo optimizado:
  1. `search`: Para obtener un índice rápido de IDs.
  2. `timeline`: Para ver qué pasó antes y después de un evento específico.
  3. `get_observations`: Para traer el detalle completo solo de los IDs filtrados. Esto ahorra hasta un 90% de tokens comparado con volcar toda la memoria.
- Cambio de idioma: Si prefieres que las observaciones y memorias se generen en español, edita `~/claude-mem/settings.json` y cambia `"CLAUDE_MEM_MODE": "code-es"`.
- Proveedores alternativos: Si no quieres gastar créditos de Anthropic para el proceso de compresión de memoria, puedes configurar Gemini (que ofrece un tier gratuito generoso) editando el provider en la configuración.
- Personalización de inyección: Desde el visor web, puedes ajustar el número de observaciones (default 50) que se inyectan al inicio. Mi experiencia me lleva a pensar que bajarlo a 20-30 es suficiente para proyectos pequeños y ahorra latencia.

### Posibles problemas/incidencias

- Puerto bloqueado: El puerto 37777 es el predeterminado. Si está ocupado, el worker no arrancará. Cámbialo en el archivo de settings si tienes conflictos con otras herramientas de desarrollo.
- Desincronización de versiones: Si tras una actualización notas que no guarda datos, ejecuta `npx claude-mem repair` para refrescar los marcadores de versión y reinstalar dependencias críticas.
- Incompatibilidades: Aunque soporta múltiples IDEs, la integración con Cursor requiere un paso adicional de configuración manual de hooks (`bun run cursor:install -- user`) para que sea global.

### Otros

- El archivo de configuración principal se encuentra siempre en `~/claude-mem/settings.json`.
- Para usuarios de Windows: Si los scripts no se ejecutan, aplica `Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser` en PowerShell.
- Puedes usar el comando `/mem-search` (o similar según el IDE) para interrogar directamente a tu base de datos de conocimientos pasados.

## PREGUNTAS FRECUENTES

---

### ¿Qué es Claude-Mem y cuál es su función principal en un entorno profesional?

Es un sistema de memoria persistente y comprensión de contexto diseñado para agentes de inteligencia artificial, específicamente optimizado para Claude Code. Su función es capturar, indexar y comprimir las acciones, decisiones y herramientas utilizadas durante las sesiones de desarrollo, permitiendo que el asistente de IA recupere ese conocimiento en interacciones futuras y evite la pérdida de contexto al reiniciar sesiones o cambiar de proyecto.

### ¿Cuál es el modelo de costes asociado a esta herramienta?

Claude-Mem es un proyecto de código abierto distribuido bajo la licencia Apache 2.0, por lo que su descarga y uso son gratuitos. No obstante, el usuario debe cubrir los costes operativos derivados de las llamadas a la API de los modelos de lenguaje (como Anthropic Claude o Google Gemini) necesarios para procesar las observaciones y realizar búsquedas semánticas.

### ¿Es Claude-Mem una tecnología de código abierto?

Sí, el proyecto es totalmente Open Source y su código fuente está disponible para auditoría y contribución en GitHub. Utiliza la licencia Apache 2.0, lo que facilita su adopción en entornos corporativos al permitir el uso comercial y la modificación del software sin restricciones de propiedad estrictas.

### ¿Cómo garantiza la seguridad y la privacidad de los datos sensibles?

La herramienta opera localmente utilizando SQLite y bases de datos vectoriales (ChromaDB) para el almacenamiento de trazas. Dispone de mecanismos de privacidad granular que permiten el uso de etiquetas de exclusión para evitar que información confidencial sea registrada en la memoria persistente. No obstante, los datos enviados para su procesamiento dependen de las políticas de privacidad de los proveedores de LLM configurados.

### ¿Qué requisitos técnicos son necesarios para su implementación?

Requiere un nivel técnico medio-alto. Es necesario contar con Node.js (v18 o superior), experiencia en el manejo de interfaces de línea de comandos (CLI) y conocimientos básicos sobre el protocolo de contexto de modelo (MCP). Su instalación se realiza generalmente a través de npx y requiere la configuración manual de variables de entorno y archivos JSON.

### ¿Con qué entornos de desarrollo e IDEs es compatible?

Se integra mediante servidores MCP con Claude Desktop y diversas herramientas de desarrollo como Cursor, VS Code (a través de extensiones como Roo Code), Warp, Goose y Gemini CLI. Además, ofrece una interfaz visual local conocida como Web Viewer UI accesible mediante el puerto 37777.

### ¿Cumple con la normativa española y europea de protección de datos?

Al ser una herramienta que se ejecuta localmente, el control de los datos reside en el usuario. Sin embargo, dado que interactúa con APIs externas para el procesamiento de IA, el cumplimiento del RGPD dependerá de los acuerdos de procesamiento de datos (DPA) que el usuario profesional tenga suscritos con los proveedores de los modelos de lenguaje (como Anthropic o Google).

### ¿Qué ventajas ofrece frente a un sistema RAG tradicional?

A diferencia de los sistemas RAG convencionales que pueden saturar la ventana de tokens con información irrelevante, Claude-Mem utiliza una arquitectura de divulgación progresiva. Esto permite buscar primero en un índice compacto y solo desplegar detalles profundos cuando son estrictamente necesarios, optimizando el consumo de tokens hasta en 10 veces.

### ¿Qué es el 'Endless Mode' y para qué sirve?

Es una funcionalidad en fase beta que implementa una arquitectura de memoria biomimética. Está diseñada para gestionar sesiones de codificación extremadamente largas, permitiendo que la IA mantenga la coherencia y el recuerdo de decisiones arquitectónicas tomadas meses atrás sin degradar el rendimiento del modelo.

## CONTRATOS Y CONDICIONES

### Opinión inicial

Tras verificar los contratos, condiciones y el repositorio oficial en GitHub, mi opinión técnica es que nos encontramos ante una herramienta de soporte a la programación con un impacto legal moderado. Desde la perspectiva de la normativa española y europea, el riesgo principal no reside en la herramienta en sí (que es local y de código abierto), sino en el flujo de datos hacia los proveedores de LLM (Anthropic/Google). Claude-Mem actúa como una "memoria intermedia" que automatiza la captura de actividad; si no se configura correctamente, podría enviar inadvertidamente fragmentos de código sensibles, secretos industriales o datos de carácter personal a servidores fuera de la UE para su procesamiento. En mi opinión profesional, es una herramienta excelente para productividad, pero exige una configuración estricta de filtros de exclusión para cumplir con el principio de minimización de datos del RGPD.

### Principales recomendaciones

- Configurar estrictamente las etiquetas de exclusión para evitar que datos sensibles, contraseñas o datos personales sean indexados en la base de datos local de SQLite/ChromaDB.
- Establecer una política de retención de datos en la base de datos local para evitar la acumulación indefinida de trazas de actividad que puedan contener propiedad intelectual.
- Asegurar que los acuerdos de nivel de servicio (SLA) con los proveedores de LLM (Anthropic/Google) sean de nivel "Enterprise" para garantizar que los datos enviados a través de Claude-Mem no se utilicen para entrenar modelos globales.
- Realizar una Evaluación de Impacto (EIPD) si se va a utilizar en proyectos que traten datos personales a gran escala, dado que la herramienta realiza un "perfilado" de la actividad del desarrollador.

### Ley de Inteligencia Artificial (AI Act)

Según documentos consultados y el análisis de la arquitectura, Claude-Mem se clasifica como un sistema de IA de propósito general o una herramienta de soporte. Al no tomar decisiones automatizadas sobre personas y limitarse al entorno de programación, no entra en las categorías de "alto riesgo". No obstante, bajo el AI Act, la empresa usuaria es responsable de garantizar la transparencia: los desarrolladores deben ser conscientes de que su actividad está siendo monitorizada y resumida por un sistema de IA.

### Privacidad y protección de datos

- **Responsabilidades:** La empresa española actúa como Responsable del Tratamiento. Claude-Mem es un procesador local, pero los proveedores de API (Anthropic) actúan como Encargados del Tratamiento.
- **Ubicación de los datos:** Los índices (SQLite) y vectores (ChromaDB) residen en el equipo local del desarrollador o en servidores controlados por la empresa.
- **Transferencia internacional:** Existe transferencia internacional de datos hacia EE.UU. cada vez que la herramienta envía una "observación" para ser comprimida por el LLM. Es imprescindible contar con las Cláusulas Contractuales Tipo (SCC) firmadas con el proveedor del modelo.
- **Derechos ARCO:** Al ser una base de datos local y abierta (SQLite), la empresa puede cumplir fácilmente con el derecho de acceso o supresión, eliminando las entradas específicas de la base de datos o borrando el archivo de memoria del proyecto.

### Propiedad intelectual

- **Propiedad de datos:** Al ser una herramienta bajo licencia Apache 2.0, el usuario mantiene la propiedad total de los datos de entrada y la base de datos de memoria generada.
- **Propiedad del resultado:** Según la legislación española, el software o código generado con asistencia de IA pertenece a la empresa, pero el "resumen de memoria" generado por el modelo podría carecer de protección por derechos de autor al no haber una intervención humana directa en su redacción creativa, aunque se protege como secreto empresarial.

### Usos y prohibiciones

- **Usos admitidos:** Optimización de contextos en desarrollo de software, auditoría técnica de decisiones y soporte en la continuidad de proyectos.
- **Usos prohibidos:** No utilizar para monitorizar el rendimiento laboral de los empleados de forma encubierta (vulnera el Estatuto de los Trabajadores y el RGPD). No utilizar en proyectos con datos clasificados o de alta seguridad estatal sin una instancia de LLM local (ej. Ollama).

### Seguridad y certificaciones

- **Seguridad:** La seguridad recae en el entorno local del usuario. Almacena datos en texto claro o vectores en la ruta local de la aplicación.
- **Certificaciones:** Al ser un proyecto Open Source, no posee certificaciones ISO o SOC2 propias; la empresa debe auditar el código fuente disponible en GitHub para asegurar que no existan vulnerabilidades de inyección de prompts.

Otros

Es importante destacar que el uso del componente "Web Viewer UI" abre un puerto local (37777). En un entorno corporativo, este puerto debe estar protegido por el firewall del sistema para evitar que otros equipos en la misma red local puedan visualizar el historial de "pensamientos" y código del desarrollador.

Fuentes consultadas:

- [Repositorio oficial y código fuente](#)
- [Licencia Apache 2.0](#)
- [Documentación oficial de arquitectura](#)
- [Condiciones de uso de la API de Anthropic](#)

#### Para más información y herramientas:

Explora look4.tools para descubrir las mejores soluciones tecnológicas del mercado.

[Inicio](#) [Todas las herramientas](#) [Categorías](#)

Este documento ofrece recomendaciones generadas mediante análisis humano y sistemas de IA automatizados. La información tiene carácter meramente informativo y no constituye asesoramiento legal, profesional ni garantía de resultados. Las marcas, logotipos y nombres comerciales pertenecen a sus respectivos propietarios y se utilizan únicamente con fines identificativos.