

Loop de Verificación · el estándar que escribes y quién lo cierra

CATEGORÍA · EMPEZAR CON CLAUDE

Le pides algo a Claude, te lo entrega, lo revisas, no está como lo querías, se lo regresas. Ese ir y venir se llama loop de verificación, y hoy lo estás cerrando tú, a mano, cada vez. Esta es cómo se cierra solo, y no es un comando: son dos piezas. La primera es el estándar — la lista que corres en

QUÉ VAS A ENCONTRAR

- 01 Qué es y qué resuelve
- 02 Para qué sirve
- 03 Dónde aplicarlo
- 04 Cómo implementarlo paso a paso
- 05 Comprobación y errores frecuentes

FICHA DEL RECURSO

Qué es, para qué sirve y cómo se implementa

FUNCIÓN

Le pides algo a Claude, te lo entrega, lo revisas, no está como lo querías, se lo regresas. Ese ir y venir se llama loop de verificación, y hoy lo estás cerrando tú, a mano, cada vez. Esta es cómo se cierra solo, y no es un comando: son dos piezas. La primera es el estándar — la lista que corres en

PARA QUÉ SIRVE

Le pides algo a Claude, te lo entrega, lo revisas, no está como lo querías, se lo regresas. Ese ir y venir se llama loop de verificación, y hoy lo estás cerrando tú, a mano, cada vez. Esta es cómo se cierra solo, y no es un comando: son dos piezas. La primera es el estándar — la lista que corres en tu cabeza cuando revisas, sacada de ahí y escrita donde Claude la lea. La segunda es quién la cierra, y ahí van /goal y /loop combinados sobre la misma vara: /goal empuja el trabajo hasta cumplir el estándar y se limpia solo cuando lo cumple; /loop vuelve a levantar ese mismo estándar cada tanto, sobre lo que haya entrado nuevo. Va con el dato que rompe la mayoría de los loops y que casi nadie sabe: el modelo que juzga si tu meta se cumplió no llama herramientas.

DÓNDE APLICARLO

Nivel Avanzado. Temas: loops de verificación, stop.

CÓMO IMPLEMENTARLO

- 1 Confirma requisitos: plan de Claude, herramientas y accesos que la tarea necesita.
- 2 Prepara el material (archivos, cuentas, datos) antes de empezar.
- 3 Ejecuta el método sobre un caso real pequeño, no sobre el proyecto completo.
- 4 Verifica el resultado contra lo que esperabas antes de repetirlo o automatizarlo.
- 5 Cuando funcione dos veces seguidas, conviértelo en rutina o skill.

EL RECURSO EN DETALLE

Qué resuelve y cómo funciona

Le pides algo a Claude, te lo entrega, lo revisas, no está como lo querías, se lo regresas. Ese ir y venir se llama loop de verificación, y hoy lo estás cerrando tú, a mano, cada vez. Esta es cómo se cierra solo, y no es un comando: son dos piezas. La primera es el estándar — la lista que corres en tu cabeza cuando revisas, sacada de ahí y escrita donde Claude la lea.

La segunda es quién la cierra, y ahí van `/goal` y `/loop` combinados sobre la misma vara: `/goal` empuja el trabajo hasta cumplir el estándar y se limpia solo cuando lo cumple; `/loop` vuelve a levantar ese mismo estándar cada tanto, sobre lo que haya entrado nuevo. Va con el dato que rompe la mayoría de los loops y que casi nadie sabe: el modelo que juzga si tu meta se cumplió no llama herramientas.

No abre tus archivos, no corre tus pruebas, no consulta nada — solo lee lo que Claude ya dejó escrito en la conversación. Si no se ve, no cuenta, y de ahí sale la causa número uno de metas que dan vueltas para siempre.

Adentro: los tres ingredientes de una condición que aguanta, cinco pares de criterio flojo contra criterio con prueba, la tabla de los cuatro cerradores con qué dispara el siguiente turno en cada uno, los cuatro tipos de loop según el equipo de Claude Code, y la tercera capa que no está en ninguna otra de la bóveda: los hooks de Stop, tanto el de prompt —que es el mismo motor de `/goal` pero viviendo en tu configuración y aplicando a todas tus sesiones— como el de agente, el único que sí abre archivos y corre comandos antes de decidir, y por eso el único que tapa el punto ciego.

Cierra con siete prompts para copiar, cinco recetas completas con su estándar, su condición y su loop, los seis modos en que esto se rompe con la línea exacta que arregla cada uno, y el costo sin adornos. `#/goal + /loop hooks`

ANTES DE DARLO POR HECHO

Comprobación y errores frecuentes

- 1 Aplica el método sobre un caso pequeño y compara el resultado con lo que esperabas.
- 2 Si el resultado no sale, revisa primero los requisitos: plan, versión del modelo y accesos.
- 3 Anota lo que tuviste que corregir a mano: eso es lo que hay que añadir a tus instrucciones permanentes.
- 4 Cuando funcione dos veces seguidas, automatízalo como rutina o skill.

DATOS DEL RECURSO

Tipo: Guía paso a paso

Categoría: Empezar con Claude

Nivel: Avanzado

Temas: loops de verificación, stop

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)