/goal para Claude Code

CATEGORÍA · PROMPTS Y TÉCNICAS

Como darle a Claude Code una sola instrucción y hacer que trabaje solo hasta terminar. El comando que convierte un asistente en un agente de verdad.

QUÉ VAS A ENCONTRAR

FICHA DEL RECURSO

Qué es, para qué sirve y cómo se implementa

FUNCIÓN

Como darle a Claude Code una sola instrucción y hacer que trabaje solo hasta terminar. El comando que convierte un asistente en un agente de verdad.

PARA QUÉ SIRVE

La mayoría usa Claude Code como un chat: pregunta, respuesta, pregunta, respuesta. /goal cambia eso. Le das una condición de llegada y Claude trabaja solo, turno tras turno, hasta que la cumple. Sin que le mandes un solo mensaje más durante el proceso. Así funciona internamente: después de cada turno, un modelo evaluador revisa si tu condición ya se cumplió. Si no, Claude arranca otro turno automáticamente. Si sí, para solo y registra el logro. Mientras eso pasa, tú puedes estar haciendo otra cosa completamente distinta. Requiere Claude Code v2.1.139 o superior.

DÓNDE APLICARLO

Cubre: Qué es /goal y como funciona el loop de ejecución autónoma; Como escribir condiciones que Claude entienda y cumpla; 10 prompts de /goal para emprendedores que construyen con Claude Code; Los errores más comunes y como evitarlos desde el primer uso.

CÓMO IMPLEMENTARLO

1. 01 / El Comando: La diferencia entre asistente y agente.
2. 02 / Las Condiciones: Como escribir condiciones que funcionan.
3. 03 / Prompts De Desarrollo: 5 prompts para construir sin parar.
4. 04 / Prompts De Negocio: 5 prompts para operar sin interrupciones.
5. 05 / Lo Que Debes Saber: Errores comunes y como evitarlos.

01 / EL COMANDO

La diferencia entre asistente y agente.

La mayoría usa Claude Code como un chat: pregunta, respuesta, pregunta, respuesta. /goal cambia eso. Le das una condición de llegada y Claude trabaja solo, turno tras turno, hasta que la cumple. Sin que le mandes un solo mensaje más durante el proceso. Así funciona internamente: después de cada turno, un modelo evaluador revisa si tu condición ya se cumplió. Si no, Claude arranca otro turno automáticamente. Si sí, para solo y registra el logro. Mientras eso pasa, tú puedes estar haciendo otra cosa completamente distinta. Requiere Claude Code v2.1.139 o superior.

COMANDOS ESENCIALES

```
/goal [condición]         Arranca el loop autónomo
/goal                   Ver estado, turnos y tokens
/goal clear             Detenerlo manualmente
Alias de clear: stop, off, reset, cancel
```

02 / LAS CONDICIONES

Como escribir condiciones que funcionan.

Una buena condición tiene tres elementos: el estado final que quieres (qué debe ser verdad cuando Claude pare), como Claude puede verificarlo en pantalla (un test que pase, un archivo que exista, un output visible), y opcionalmente un límite de seguridad para que no corra infinito. Las condiciones vagas no funcionan. 'Mejora el código' o 'termina el proyecto' no le dan a Claude un criterio claro de parada. Las condiciones verificables sí funcionan: 'no pares hasta que el endpoint responda 200 en todos los casos de prueba' o 'no pares hasta que los 47 archivos estén procesados y el log no tenga errores'. Cuanto más específica, más preciso el resultado.

ESTRUCTURA DE UNA BUENA CONDICIÓN

Estado final claro:
qué debe ser verdad cuando Claude pare
Verificación en pantalla:
un test, un archivo, un output visible
Límite opcional:
o para después de 20 turnos

03 / PROMPTS DE DESARROLLO

5 prompts para construir sin parar.

Estos prompts funcionan para proyectos de software: hunting de bugs, construcción de features, refactorización y conexión con APIs. En todos los casos, /goal mantiene a Claude en loop hasta que el resultado es verificable. No hasta que 'parece que funciona': hasta que funciona.

PROMPTS LISTOS PARA USAR

```
/goal encuentra el bug que causa [error].
No pares hasta que lo reproduzcas y resuelvas.
/goal construye [feature] completo. No pares
hasta que funcione en móvil y desktop sin errores.
/goal conecta la API de [servicio]. No pares
hasta que una transacción de prueba confirme.
/goal refactoriza [archivo] sin romper nada.
No pares hasta que todos los tests pasen.
/goal encuentra todos los problemas de seguridad
y corrígelos. No pares hasta que el scan
no reporte vulnerabilidades críticas.
```

04 / PROMPTS DE NEGOCIO

5 prompts para operar sin interrupciones.

No todos los usos de /goal son de código. Cualquier tarea con una condición verificable puede correr en loop autónomo: organización de archivos, generación de contenido, análisis de datos, construcción de reportes. Si puedes describir cuándo está terminado, /goal puede manejarlo.

PROMPTS LISTOS PARA USAR

```
/goal revisa los archivos en [carpeta], corrige
duplicados e incompletos. No pares hasta que
el log no tenga errores.
/goal escribe las [N] secciones del reporte.
No pares hasta que cada una tenga al menos
[X] palabras y esté completa.
/goal construye la landing: hero, beneficios,
precios y contacto. No pares hasta que cargue
limpia en móvil y desktop.
/goal organiza [carpeta] en subcarpetas por
categoría y genera un índice. No pares hasta
que el índice esté completo.
/goal escribe el plan de lanzamiento de
[producto]: estrategia, canales y calendario.
No pares hasta que cada sección esté terminada.
```

05 / LO QUE DEBES SABER

Errores comunes y como evitarlos.

El evaluador que decide si tu condición se cumplió solo lee lo que Claude ya mostró en pantalla. No corre comandos por su cuenta ni lee archivos directamente. Por eso las condiciones deben describir algo que Claude pueda demostrar en la conversación: un test que corrió y pasó, un output específico, un número de archivos procesados. Si la condición es invisible, el evaluador no puede confirmarla. Tres reglas de operación: primero, si Claude no para nunca, tu condición es ambigua. Agrega un límite de seguridad: 'o para después de 20 turnos'. Segundo, la app de Claude Desktop debe estar abierta durante todo el loop. Si se cierra o el equipo entra en suspensión, el proceso termina. Tercero, /goal solo funciona en workspaces donde aceptaste el diálogo de confianza.

LÍMITES A TENER EN CUENTA

Condición máxima: 4,000 caracteres
Workspace trust debe estar aceptado
Claude Desktop debe estar abierto
El evaluador solo ve el output en pantalla
Agrega 'o para después de X turnos' si dudas

SIGUIENTE PASO

Sígueme en

Publico guías como esta cada semana. Todo lo que comparto lo pruebo en mis propios proyectos y con mis clientes. Si esto te fue útil, compártelo con alguien que lo necesite.

MÁS GUÍAS GRATIS EN 377CREATIVE.COM/RECURSOS