EcomLabs

GUÍA PRÁCTICA

Claude Code: cuatro hábitos para que tu límite de uso dure, y chuleta de comandos

Por qué cada mensaje gasta más que el anterior, los cuatro hábitos de la documentación para estirar tu límite de uso (/clear, /compact, sesiones cortas y el modelo según la tarea) y una chuleta con los comandos para ver cuánto contexto llevas.

PARA	Quien usa Claude Code con un plan de suscripción y se queda sin límite a mitad del día
PROBADO CON	Solo el cálculo del ejemplo, con Node 22; los comandos salen de la documentación y no se ejecutaron en una sesión nueva
ACTUALIZADA	12 de noviembre de 2026
DOCUMENTACIÓN	code.claude.com/docs/en/costs
ARTÍCULO	ecomlabs.dev/g/tokens
AUTOR	Luis Morales · @luismoralesdev

Índice

Antes de empezar	Por qué se acaba tu límite de uso
Paso 01	Los cuatro hábitos
Paso 02	Chuleta de comandos
Al terminar	Checklist

ANTES DE EMPEZAR

Por qué se acaba tu límite de uso

Claude Code hace una petición nueva con cada mensaje y el modelo no recuerda nada entre peticiones, así que reenvía el prompt del sistema, el contexto del proyecto y cada mensaje y resultado anteriores. La caché abarata ese reenvío, pero según la página de costos de la documentación, una pregunta de una línea en una sesión abierta todo el día sigue consumiendo uso por toda la conversación.

Ejemplo ilustrativo, un modelo aritmético calculado con Node 22 (15 000 tokens de contexto base más 3000 por turno, una petición por turno, sin caché): una sesión de 20 turnos envía 930 000 tokens y dos sesiones de 10 turnos envían 630 000, un 32 % menos para el mismo trabajo. No mide tu plan: la documentación no publica cuotas en tokens, solo un límite de sesión de cinco horas y uno semanal en Pro y Max.

PASO 01

Los cuatro hábitos

HÁBITO	CUÁNDO	QUÉ HACE O CUESTA
`/clear`	Al pasar a una tarea sin relación	Contexto vacío; no cuesta nada. Vuelves con `/resume`
`/compact [instrucciones]`	Pausa natural en la misma tarea	Resume el historial; es una petición grande. Con caché caliente cuesta una fracción; tras una pausa larga reprocesa todo
Sesiones cortas	Una tarea por sesión	Menos contexto por petición. La caché dura una hora en suscripción dentro del plan, cinco minutos con créditos, API o nube
`/model`	Al inicio de la sesión	Cada modelo tiene su caché: cambiar a media sesión lee todo sin caché. Sonnet para casi todo, Opus para lo complejo

- **Compactar con foco**: en el `CLAUDE.md` de la raíz, una sección `# Compact instructions` con `When you are using compact, please focus on test output and code changes`, el texto de la documentación.
- **Al retomar**: `/resume` envía la conversación completa otra vez; si la caché venció, la pagas sin descuento.

> **CUIDADO**
> `/model nombre` lo guarda como predeterminado para sesiones nuevas. Para cambiar solo esta sesión, abre el selector con `/model` y pulsa `s` sobre la fila.

PASO 02

Chuleta de comandos

COMANDO	CUÁNDO	QUÉ HACE
`/context`	Antes de una tarea larga	Cuadrícula del uso del contexto, con sugerencias
`/usage`	A mitad del día	Barras del plan y desglose por skills, subagentes, plugins y MCP; alias `/cost`, `/stats`
`/rename [nombre]`	Antes de `/clear`	Nombra la sesión para retomarla después
`/clear`	Al cambiar de tarea	Conversación nueva (alias `/reset`, `/new`)
`/compact [instrucciones]`	Pausa en la misma tarea	Resume; acepta instrucciones de foco
`/model`	Al inicio	Cambia de modelo; sin argumento abre el selector

Una rutina para el día, tal cual la escribirías en la sesión:

```
/context
/usage
/rename carrito-descuentos
/clear
/model
/compact conserva los cambios en components/carrito/ y los comandos de prueba
```

AL TERMINAR

Checklist

- [ ] Una tarea por sesión; `/rename` y `/clear` al terminar.
- [ ] `/compact` solo en pausas naturales, con instrucciones de foco.
- [ ] Modelo y esfuerzo elegidos al inicio, no a media tarea.
- [ ] `/context` antes de una tarea larga y `/usage` a mitad del día.
- [ ] No retomas una sesión vieja si puedes empezar limpio.
- [ ] Tareas programadas detenidas al acabar: se disparan aunque la sesión esté inactiva.