

Un prompt no puede hacerlo todo

Hasta acá mejoraste **lo que le das** al modelo. Este bloque mejora **cómo lo procesa**, y las tres técnicas comparten una intuición: un problema grande resuelto de una sola vez es imposible de auditar. Todo el bloque cabe en una palabra: descomponer.

QUÉ VAS A PODER HACER

- 1 Hacer visible el razonamiento del modelo para poder auditarlo.
- 2 Detectar un error de razonamiento sin ser experto en el tema.
- 3 Partir una tarea grande en eslabones y saber dónde cortar.
- 4 Convertir el mejor resultado en una plantilla que use todo tu equipo.

LAS 3 TÉCNICAS

TÉCNICA	QUÉ HACE	ÚSALA CUANDO	LO QUE TE DA DE VERDAD
Cadena de pensamiento	Muestra el paso a paso antes de responder	Cálculos, análisis, cualquier cosa con supuestos	Ver en qué paso se equivocó
Self-consistency	Resuelve lo mismo por varios caminos y compara	Plata, contratos, decisiones con trade-offs	Detectar que tu enunciado era ambiguo
Prompt chaining	Parte la tarea en prompts encadenados	Cuando un solo prompt hace 3 o más cosas	Iterar un paso sin rehacer el resto

LAS TRES, ESCRITAS

CADENA DE PENSAMIENTO que muestre el trabajo

"Antes de darme la respuesta final, muestra tu razonamiento paso a paso. Numera cada paso y explica qué supuesto usas en cada uno."

No es para que piense mejor: es para que TÚ audites dónde falló. El error casi nunca está en el resultado, está en el paso 3.

SELF-CONSISTENCY que se contradiga a propósito

"Resuelve esto por tres métodos independientes, sin mirar los anteriores. Al final compara los tres resultados: si coinciden dímelos; si difieren, explica exactamente de dónde viene la diferencia."

Si divergen, contrasta el punto débil, que suele ser TU enunciado.

ANTES Y DESPUÉS: PARTIR EL MONSTRUO

✗ ANTES

Lee este log, encuentra el problema, dime cómo arreglarlo, escribe el mensaje para el cliente y prepara el resumen para la reunión.

Cinco tareas en un prompt. Si algo sale mal no sabes cuál de las cinco falló, y el contexto se llena antes de llegar al final.

✓ DESPUÉS

1. "Lee este log y lista las anomalías, sin interpretarlas todavía."
2. "De estas anomalías, ordena las 3 causas raíz más probables y di qué evidencia apoya cada una."
3. "Para la causa #1, escribe los pasos de verificación concretos en nuestro sistema."
4. "Escribe el update para el cliente: qué pasó, impacto, qué sigue. 100 palabras, sin jerga."

Cada eslabón en un chat nuevo, pegando solo la salida anterior. Si el mensaje al cliente no te gusta, repites solo el paso 4.

DÓNDE CORTAR LA CADENA

Corta donde cambia el TIPO de trabajo:

extraer ≠ interpretar ≠ redactar ≠ adaptar de tono

Regla práctica: si en tu prompt hay un "y después", ahí va la tijera.

Y cada eslabón en un CHAT NUEVO, pegando solo la salida anterior.
Encadenar dentro del mismo chat te devuelve al prompt gigante.

La ventaja real del encadenado no es la calidad de cada paso: es que **cada eslabón se puede iterar por separado**. Si el post no te gustó, no vuelves a extraer los puntos del reporte.

EL MODO DEL MODELO Y EL TRUCO DE LA PLANTILLA

EL MODO CAMBIA CUÁNTO PIENSA, NO CUÁNTO SABE

rápido	redactar, reformatear, traducir, resumir
razonamiento	cálculos, código, debugging, análisis multipaso
pro / máximo	decisión estratégica con muchas variables

Más razonamiento NO es mejor en tareas simples: pagas y esperas de más.

CONVIERTE EL BUEN RESULTADO EN PLANTILLA el truco más subutilizado

"Este enfoque es exactamente el que necesito. Conviértelo en una plantilla reutilizable: mantén la estructura y el razonamiento, y reemplaza los datos específicos por variables entre corchetes."

Y la próxima vez: "Usando este template, resuelve el caso de [X]"

Ganancia triple: acota la creatividad · estandariza al equipo · acelera.

Un modelo rápido CON plantilla suele ganarle a uno de razonamiento SIN ella.

1. ¿Para qué sirve realmente la cadena de pensamiento?

No tanto para que el modelo piense mejor, sino para que tú puedas auditarlo. El error casi nunca está en el resultado final: está en un paso intermedio, y sin verlo no lo encuentras.

2. Si los tres métodos dan resultados distintos, ¿qué aprendiste?

Que hay una ambigüedad, y casi siempre está en tu enunciado, no en el modelo. Es un hallazgo valioso, no un fracaso.

3. ¿Por qué un prompt gigante rinde peor que varios chicos?

Porque llena la ventana de contexto: aparecen olvidos, contradicciones y alucinaciones. Y si algo sale mal, no sabes cuál de las cinco tareas falló.

4. ¿Cuál es la ventaja práctica más importante del encadenado?

Que puedes iterar un eslabón veinte veces sin tocar los demás.

5. ¿Qué error se comete al encadenar dentro del mismo chat?

Arrastras todo el contexto anterior y terminas con el mismo problema del prompt gigante. Cada paso va en un chat limpio con solo la salida previa.

6. ¿Por qué una plantilla mejora el resultado además de ahorrar tiempo?

Porque acota la creatividad del modelo, que ya no reinventa el enfoque cada vez, y estandariza el resultado entre distintas personas del equipo.

LOS 4 ERRORES MÁS COMUNES

❗ Pedir cadena de pensamiento para todo, incluso para "reescribe esto más corto".

❗ Leer el paso a paso por encima sin revisar los supuestos: ahí está el error.

❗ Encadenar arrastrando toda la conversación previa.

❗ Llegar al resultado perfecto y cerrar el chat sin pedir la plantilla.