Prompt Engineering

Categoría	
Estatus	Terminado
Finalizado el:	@5 de julio de 2025 19:53
Modificado	@5 de julio de 2025 19:53
Creado	@5 de julio de 2025 16:06
Tipo	📕 Libro

Cuándo usar la ingeniería de prompts

No todos los criterios de éxito o evaluaciones fallidas se resuelven mejor con la ingeniería de prompts. Por ejemplo, la latencia y el costo a veces se pueden mejorar más fácilmente seleccionando un modelo diferente.

La ingeniería de prompts es mucho más rápida que otros métodos de control del comportamiento del modelo, como el finetuning, y a menudo puede producir saltos en el rendimiento en mucho menos tiempo. Aquí hay algunas razones para considerar la ingeniería de prompts sobre el finetuning:

- **Eficiencia de recursos**: El fine-tuning requiere GPUs de alta gama y gran memoria, mientras que la ingeniería de prompts solo necesita entrada de texto, haciéndola mucho más amigable con los recursos.
- **Rentabilidad**: Para servicios de IA basados en la nube, el fine-tuning incurre en costos significativos. La ingeniería de prompts usa el modelo base, que típicamente es más barato.
- **Mantenimiento de actualizaciones del modelo**: Cuando los proveedores actualizan los modelos, las versiones con fine-tuning podrían necesitar reentrenamiento. Los prompts generalmente funcionan entre versiones sin cambios.

- **Ahorro de tiempo**: El fine-tuning puede llevar horas o incluso días. En contraste, la ingeniería de prompts proporciona resultados casi instantáneos, permitiendo una rápida resolución de problemas.
- **Necesidades mínimas de datos**: El fine-tuning necesita datos etiquetados sustanciales específicos de la tarea, que pueden ser escasos o costosos. La ingeniería de prompts funciona con aprendizaje de pocos ejemplos o incluso sin ejemplos.
- **Flexibilidad e iteración rápida**: Prueba rápidamente varios enfoques, ajusta prompts y ve resultados inmediatos. Esta experimentación rápida es difícil con el fine-tuning.
- **Adaptación al dominio**: Adapta fácilmente los modelos a nuevos dominios proporcionando contexto específico del dominio en los prompts, sin reentrenamiento.
- **Mejoras en la comprensión**: La ingeniería de prompts es mucho más efectiva que el finetuning para ayudar a los modelos a entender y utilizar mejor el contenido externo como documentos recuperados.
- **Preserva el conocimiento general**: El fine-tuning corre el riesgo de olvido catastrófico, donde el modelo pierde conocimiento general. La ingeniería de prompts mantiene las capacidades amplias del modelo.
- **Transparencia**: Los prompts son legibles por humanos, mostrando exactamente qué información recibe el modelo. Esta transparencia ayuda en la comprensión y depuración.

Cómo hacer ingeniería de prompts

Las páginas de ingeniería de prompts en esta sección se han organizado desde las técnicas más efectivas en general hasta las técnicas más especializadas. Cuando soluciones problemas de rendimiento, sugerimos que pruebes estas técnicas en orden, aunque el impacto real de cada técnica dependerá de tu caso de uso.

1. **Generador de prompts**
2. **Sé claro y directo**
3. **Usa ejemplos (multishot)**
4. **Deja que Claude piense (cadena de pensamiento)**

Mejores prácticas de ingeniería de prompts para Claude 4

Esta guía proporciona técnicas específicas de ingeniería de prompts para los modelos Claude 4 (Opus 4 y Sonnet 4) para ayudarte a lograr resultados óptimos en tus aplicaciones.

Principios generales

Sé explícito con tus instrucciones

Los modelos Claude 4 responden bien a instrucciones claras y explícitas. Ser específico sobre el resultado deseado puede ayudar a mejorar los resultados. Los clientes que deseen el comportamiento “por encima y más allá” de los modelos anteriores de Claude podrían necesitar solicitar estos comportamientos de manera más explícita con Claude 4.

Ejemplo: Creación de un panel de análisis.

Menos efectivo: Crea un panel de análisis

Más efectivo: Crea un panel de análisis. Incluye tantas características e interacciones relevantes como sea posible. Ve más allá de lo básico para crear una implementación completa con todas las funciones.

Añade contexto para mejorar el rendimiento

Proporcionar contexto o motivación detrás de tus instrucciones, como explicarle a Claude por qué tal comportamiento es importante, puede ayudar a Claude 4 a entender mejor tus objetivos y ofrecer respuestas más específicas.

Ejemplo: Preferencias de formato

Menos efectivo: NUNCA uses puntos suspensivos.

Más efectivo: Tu respuesta será leída en voz alta por un motor de texto a voz, así que nunca uses puntos suspensivos ya que el motor de texto a voz no sabrá cómo pronunciarlos.

Claude es lo suficientemente inteligente para generalizar a partir de la explicación.

Sé vigilante con ejemplos y detalles

Los modelos Claude 4 prestan atención a los detalles y ejemplos como parte del seguimiento de instrucciones. Asegúrate de que tus ejemplos se alineen con los comportamientos que quieres fomentar y minimicen los comportamientos que quieres evitar.

Orientación para situaciones específicas

Controla el formato de las respuestas

Hay algunas formas que hemos encontrado particularmente efectivas para dirigir el formato de salida en los modelos Claude 4:

1. **Dile a Claude qué hacer en lugar de qué no hacer**
    - En lugar de: “No uses markdown en tu respuesta”
    - Intenta: “Tu respuesta debe estar compuesta de párrafos de prosa que fluyan suavemente.”
2. **Usa indicadores de formato XML**
    - Intenta: “Escribe las secciones de prosa de tu respuesta en etiquetas <parrafos_de_prosa_que_fluyen_suavemente>.”
3. **Haz coincidir el estilo de tu prompt con la salida deseada**

El estilo de formato utilizado en tu prompt puede influir en el estilo de respuesta de Claude. Si sigues experimentando problemas de direccionamiento con el formato de salida, recomendamos que, en la medida de lo posible, hagas coincidir el estilo de tu prompt con el estilo de salida deseado. Por ejemplo, eliminar el markdown de tu prompt puede reducir la cantidad de markdown en la salida.

Aprovecha las capacidades de pensamiento y pensamiento intercalado

Claude 4 ofrece capacidades de pensamiento que pueden ser especialmente útiles para tareas que involucran reflexión después del uso de herramientas o razonamiento complejo de múltiples pasos. Puedes guiar su pensamiento inicial o intercalado para obtener mejores resultados.

Ejemplo de prompt

Después de recibir los resultados de la herramienta, reflexiona cuidadosamente sobre su calidad y determina los próximos pasos óptimos antes de proceder. Utiliza tu pensamiento para planificar e iterar basándote en esta nueva información, y luego toma la mejor acción siguiente.

Optimiza la llamada de herramientas en paralelo

Los modelos Claude 4 sobresalen en la ejecución de herramientas en paralelo. Tienen una alta tasa de éxito en el uso de llamadas de herramientas en paralelo sin ningún tipo de indicación para hacerlo, pero algunas indicaciones menores pueden aumentar este comportamiento a una tasa de éxito de uso de herramientas en paralelo de ~100%. Hemos encontrado que este prompt es el más efectivo:

Prompt de ejemplo para agentes: Para máxima eficiencia, siempre que necesites realizar múltiples operaciones independientes, invoca todas las herramientas relevantes simultáneamente en lugar de secuencialmente.

Reduce la creación de archivos en codificación agéntica

Los modelos Claude 4 a veces pueden crear nuevos archivos para propósitos de prueba e iteración, particularmente cuando trabajan con código. Este enfoque permite a Claude usar archivos, especialmente scripts de Python, como un "borrador temporal" antes de guardar su salida final. El uso de archivos temporales puede mejorar los resultados, particularmente para casos de uso de codificación agéntica.

Prompt de ejemplo: Si creas archivos temporales nuevos, scripts o archivos auxiliares para iteración, limpia estos archivos eliminándolos al final de la tarea.

Mejora la generación de código visual y frontend

Para la generación de código frontend, puedes dirigir los modelos Claude 4 para crear diseños complejos, detallados e interactivos proporcionando un estímulo explícito:

Prompt de ejemplo: No te contengas. Da lo mejor de ti.

También puedes mejorar el rendimiento de frontend de Claude en áreas específicas proporcionando modificadores adicionales y detalles sobre en qué enfocarse:

- "Incluye tantas características e interacciones relevantes como sea posible"
- "Añade detalles cuidadosos como estados de hover, transiciones y microinteracciones"
- "Crea una demostración impresionante que muestre capacidades de desarrollo web"
- "Aplica principios de diseño: jerarquía, contraste, equilibrio y movimiento"

Consideraciones de migración

Al migrar de Sonnet 3.7 a Claude 4:

1. **Sé específico sobre el comportamiento deseado**: Considera describir exactamente lo que te gustaría ver en la salida.

2. **Enmarca tus instrucciones con modificadores**: Añadir modificadores que animen a Claude a aumentar la calidad y el detalle de su salida puede ayudar a dar mejor forma al rendimiento de Claude. Por ejemplo, en lugar de “Crea un panel de análisis”, usa “Crea un panel de análisis. Incluye tantas características e interacciones relevantes como sea posible. Ve más allá de lo básico para crear una implementación completa con todas las funciones.”
3. **Solicita características específicas explícitamente**: Las animaciones y elementos interactivos deben solicitarse explícitamente cuando se deseen.

Part. 3

Generar automáticamente plantillas de prompts iniciales

El generador de prompts es particularmente útil como herramienta para resolver el “problema de la página en blanco” y darle un punto de partida para pruebas e iteraciones posteriores.

A veces, la parte más difícil de usar un modelo de IA es descubrir cómo hacerle prompts de manera efectiva. Para ayudar con esto, hemos creado una herramienta de generación de prompts que guía a Claude para generar plantillas de prompts de alta calidad adaptadas a sus tareas específicas. Estas plantillas siguen muchas de nuestras mejores prácticas de ingeniería de prompts.

Pruebe el generador de prompts ahora directamente en la **Consola**.

Part. 4

Usar plantillas y variables de prompt

Al implementar una aplicación basada en LLM con Claude, tus llamadas a la API típicamente consistirán en dos tipos de contenido:

- **Contenido fijo:** Instrucciones estáticas o contexto que permanece constante a través de múltiples interacciones
- **Contenido variable:** Elementos dinámicos que cambian con cada solicitud o conversación, como:
    - Entradas del usuario

- Contenido recuperado para Generación Aumentada por Recuperación (RAG)
- Contexto de la conversación como el historial de la cuenta del usuario
- Datos generados por el sistema como resultados del uso de herramientas alimentados desde otras llamadas independientes a Claude

Una **plantilla de prompt** combina estas partes fijas y variables, usando marcadores de posición para el contenido dinámico. En la **Anthropic Console**, estos marcadores de posición se denotan con **{{dobles llaves}}**, haciéndolos fácilmente identificables y permitiendo pruebas rápidas de diferentes valores.

Cuándo usar plantillas y variables de prompt

Siempre deberías usar plantillas y variables de prompt cuando esperes que cualquier parte de tu prompt se repita en otra llamada a Claude (solo a través de la API o la **Anthropic Console**. **claude.ai** actualmente no soporta plantillas o variables de prompt).

Las plantillas de prompt ofrecen varios beneficios:

- **Consistencia:** Aseguran una estructura consistente para tus prompts a través de múltiples interacciones
- **Eficiencia:** Intercambian fácilmente el contenido variable sin reescribir el prompt completo
- **Capacidad de prueba:** Prueban rápidamente diferentes entradas y casos límite cambiando solo la porción variable
- **Escalabilidad:** Simplifican la gestión de prompts a medida que tu aplicación crece en complejidad
- **Control de versiones:** Rastrean fácilmente los cambios en la estructura de tu prompt a lo largo del tiempo manteniendo el seguimiento solo en la parte central de tu prompt, separada de las entradas dinámicas

La **Anthropic Console** utiliza intensivamente plantillas y variables de prompt para soportar características y herramientas para todo lo anterior, como con el:

- **Generador de prompts:** Decide qué variables necesita tu prompt y las incluye en la plantilla que genera
- **Mejorador de prompts:** Toma tu plantilla existente, incluyendo todas las variables, y las mantiene en la plantilla mejorada que genera
- **Herramienta de evaluación:** Permite probar, escalar y rastrear versiones de tus prompts fácilmente separando las porciones variables y fijas de tu plantilla de prompt

Ejemplo de plantilla de prompt

Consideremos una aplicación simple que traduce texto del inglés al español. El texto traducido sería variable ya que esperarías que este texto cambie entre usuarios o llamadas a Claude. Este texto traducido podría ser recuperado dinámicamente de bases de datos o de la entrada del usuario.

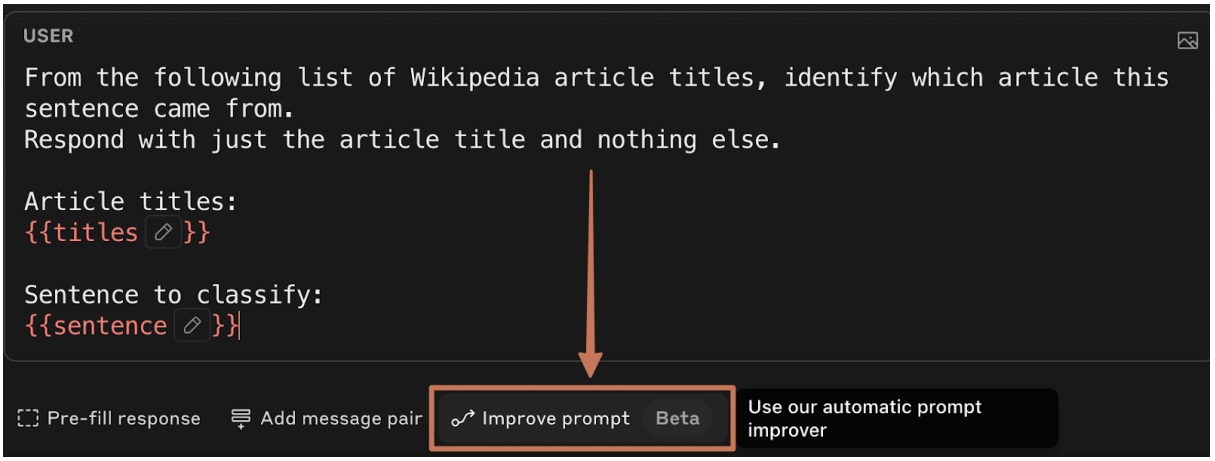
Por lo tanto, para tu aplicación de traducción, podrías usar esta simple plantilla de prompt:

Translate this text from English to Spanish: {{text}}

Part. 5

Usa nuestro mejorador de prompts para optimizar tus prompts

El mejorador de prompts te ayuda a iterar y mejorar rápidamente tus prompts mediante análisis y mejoras automatizadas. Se destaca en hacer que los prompts sean más robustos para tareas complejas que requieren alta precisión.



Antes de comenzar

Necesitarás:

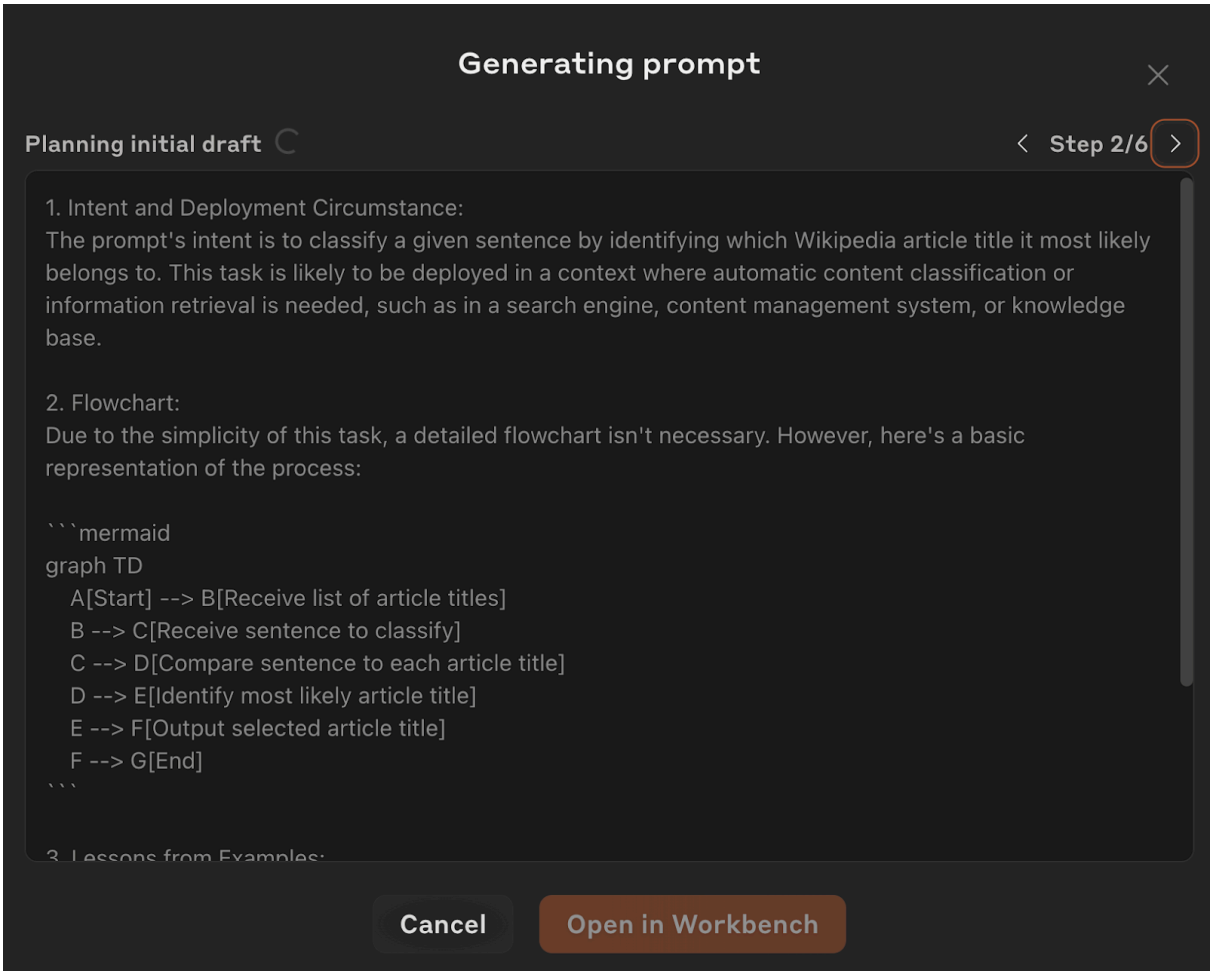
- Una **plantilla de prompt** para mejorar
- Retroalimentación sobre problemas actuales con las salidas de Claude (opcional pero recomendado)
- Ejemplos de entradas y salidas ideales (opcional pero recomendado)

Cómo funciona el mejorador de prompts

El mejorador de prompts mejora tus prompts en 4 pasos:

1. **Identificación de ejemplos**: Localiza y extrae ejemplos de tu plantilla de prompt
2. **Borrador inicial**: Crea una plantilla estructurada con secciones claras y etiquetas XML
3. **Refinamiento de cadena de pensamiento**: Agrega y refina instrucciones detalladas de razonamiento
4. **Mejora de ejemplos**: Actualiza los ejemplos para demostrar el nuevo proceso de razonamiento

Puedes ver estos pasos en tiempo real en el modal de mejora.



Lo que obtienes

El mejorador de prompts genera plantillas con:

- Instrucciones detalladas de cadena de pensamiento que guían el proceso de razonamiento de Claude y típicamente mejoran su rendimiento
- Organización clara usando etiquetas XML para separar diferentes componentes
- Formato estandarizado de ejemplos que demuestra el razonamiento paso a paso desde la entrada hasta la salida
- Prellenados estratégicos que guían las respuestas iniciales de Claude

Cómo usar el mejorador de prompts

1. Envía tu plantilla de prompt
2. Agrega cualquier retroalimentación sobre problemas con las salidas actuales de Claude (por ejemplo, “los resúmenes son demasiado básicos para audiencias expertas”)
3. Incluye ejemplos de entradas y salidas ideales
4. Revisa el prompt mejorado

Generar ejemplos de prueba

¿Aún no tienes ejemplos? Usa nuestro **Generador de Casos de Prueba** para:

1. Generar entradas de muestra
2. Obtener respuestas de Claude
3. Editar las respuestas para que coincidan con tus salidas ideales
4. Agregar los ejemplos pulidos a tu prompt

Cuándo usar el mejorador de prompts

El mejorador de prompts funciona mejor para:

- Tareas complejas que requieren razonamiento detallado
- Situaciones donde la precisión es más importante que la velocidad
- Problemas donde las salidas actuales de Claude necesitan mejoras significativas

Ejemplo de mejora

Así es como el mejorador de prompts mejora un prompt básico de clasificación:

Prompt original

De la siguiente lista de títulos de artículos de Wikipedia, identifica de qué artículo provino esta oración.
Responde solo con el título del artículo y nada más.

Títulos de artículos:
{{titles}}

Oración a clasificar:
{{sentence}}

Prompt mejorado

Eres un sistema inteligente de clasificación de texto especializado en emparejar oraciones con títulos de artículos de Wikipedia. Tu tarea es identificar a qué artículo de Wikipedia pertenece más probablemente una oración dada, basándote en una lista proporcionada de títulos de artículos.

Primero, revisa la siguiente lista de títulos de artículos de Wikipedia:
<article_titles>
{{titles}}
</article_titles>

Ahora, considera esta oración que necesita ser clasificada:
<sentence_to_classify>
{{sentence}}
</sentence_to_classify>

Tu objetivo es determinar qué título de artículo de la lista proporcionada coincide mejor con la oración dada. Sigue estos pasos:

1. Enumera los conceptos clave de la oración
2. Compara cada concepto clave con los títulos de los artículos
3. Clasifica los 3 títulos más relevantes y explica por qué son relevantes
4. Selecciona el título de artículo más apropiado que mejor abarque o se relacione con el contenido de la oración

Envuelve tu análisis en etiquetas <analysis>. Incluye lo siguiente:

- Lista de conceptos clave de la oración
- Comparación de cada concepto clave con los títulos de los artículos

- Clasificación de los 3 títulos más relevantes con explicaciones
- Tu elección final y razonamiento

Después de tu análisis, proporciona tu respuesta final: el único título de artículo de Wikipedia más apropiado de la lista.

Muestra solo el título del artículo elegido, sin ningún texto o explicación adicional.

⬆️

Observa cómo el prompt mejorado:

- Agrega instrucciones claras de razonamiento paso a paso
- Usa etiquetas XML para organizar el contenido
- Proporciona requisitos explícitos de formato de salida
- Guía a Claude a través del proceso de análisis

Solución de problemas

Problemas comunes y soluciones:

- **Los ejemplos no aparecen en la salida**: Verifica que los ejemplos estén correctamente formateados con etiquetas XML y aparezcan al inicio del primer mensaje del usuario
- **Cadena de pensamiento demasiado verbosa**: Agrega instrucciones específicas sobre la longitud deseada de la salida y el nivel de detalle
- **Los pasos de razonamiento no coinciden con tus necesidades**: Modifica la sección de pasos para que coincida con tu caso de uso específico

---

Part. 6

Sé claro, directo y detallado

Cuando interactúes con Claude, piensa en él como un empleado brillante pero muy nuevo (con amnesia) que necesita instrucciones explícitas. Como

cualquier empleado nuevo, Claude no tiene contexto sobre tus normas, estilos, pautas o formas preferidas de trabajo. Cuanto más precisamente expliques lo que quieres, mejor será la respuesta de Claude.

> **La regla de oro para prompts claros**
>
> Muestra tu prompt a un colega, idealmente alguien que tenga un contexto mínimo sobre la tarea, y pídele que siga las instrucciones. Si está confundido, Claude probablemente también lo estará.

Cómo ser claro, contextual y específico

- **Proporciona información contextual a Claude:** Al igual que podrías desempeñarte mejor en una tarea si tuvieras más contexto, Claude se desempeñará mejor si tiene más información contextual. Algunos ejemplos de información contextual:
  - Para qué se utilizarán los resultados de la tarea
  - A qué audiencia está destinada la salida
  - De qué flujo de trabajo forma parte la tarea y dónde se ubica esta tarea en ese flujo de trabajo
  - El objetivo final de la tarea o cómo es una tarea completada con éxito
- **Sé específico sobre lo que quieres que Claude haga:** Por ejemplo, si quieres que Claude genere solo código y nada más, dilo.
- **Proporciona instrucciones como pasos secuenciales:** Usa listas numeradas o viñetas para asegurarte mejor de que Claude realice la tarea exactamente como quieres.

Ejemplos

Part. 7

Usar ejemplos (prompting multishot) para guiar el comportamiento de Claude

Los ejemplos son tu arma secreta para conseguir que Claude genere exactamente lo que necesitas. Al proporcionar algunos ejemplos bien elaborados en tu prompt, puedes mejorar dramáticamente la precisión, consistencia y calidad de las salidas de Claude. Esta técnica, conocida como prompting few-shot o multishot, es particularmente efectiva para tareas que requieren salidas estructuradas o adherencia a formatos específicos.

> **Potencia tus prompts**: Incluye de 3 a 5 ejemplos diversos y relevantes para mostrarle a Claude exactamente lo que quieres. Más ejemplos = mejor rendimiento, especialmente para tareas complejas.

¿Por qué usar ejemplos?

- **Precisión**: Los ejemplos reducen la mala interpretación de las instrucciones.
- **Consistencia**: Los ejemplos imponen una estructura y estilo uniformes.
- **Rendimiento**: Los ejemplos bien elegidos mejoran la capacidad de Claude para manejar tareas complejas.

Creando ejemplos efectivos

Para máxima efectividad, asegúrate de que tus ejemplos sean:

- **Relevantes**: Tus ejemplos reflejan tu caso de uso real.
- **Diversos**: Tus ejemplos cubren casos extremos y desafíos potenciales, y varían lo suficiente para que Claude no capte inadvertidamente patrones no intencionados.
- **Claros**: Tus ejemplos están envueltos en etiquetas `<example>` (si son múltiples, anidados dentro de etiquetas `<examples>`) para estructura.

> Pídele a Claude que evalúe tus ejemplos en cuanto a relevancia, diversidad o claridad. O haz que Claude genere más ejemplos basados en tu conjunto inicial.

Ejemplos

Part. 8

Deja que Claude piense (encadenamiento de pensamiento) para mejorar el rendimiento

Cuando se enfrenta a tareas complejas como investigación, análisis o resolución de problemas, dar espacio a Claude para pensar puede mejorar dramáticamente su rendimiento. Esta técnica, conocida como **encadenamiento de pensamiento (CoT)**, **anima a Claude a desglosar los problemas paso a paso, lo que lleva a resultados más precisos y matizados.**

Antes de implementar CoT

¿Por qué dejar pensar a Claude?

- **Precisión:** Resolver problemas paso a paso reduce errores, especialmente en matemáticas, lógica, análisis o tareas generalmente complejas.
- **Coherencia:** El pensamiento estructurado conduce a respuestas más cohesivas y bien organizadas.
- **Depuración:** Ver el proceso de pensamiento de Claude te ayuda a identificar dónde los prompts pueden no ser claros.

¿Por qué no dejar pensar a Claude?

- La mayor longitud de salida puede afectar la latencia.
- No todas las tareas requieren un pensamiento profundo. Usa **CoT** juiciosamente para asegurar el equilibrio correcto entre rendimiento y latencia.

> Usa CoT para tareas que un humano necesitaría pensar detenidamente, como matemáticas complejas, análisis de múltiples pasos, escritura de documentos complejos o decisiones con muchos factores.

Cómo hacer prompts para el pensamiento

Las técnicas de encadenamiento de pensamiento a continuación están **ordenadas de menos a más complejas**. Los métodos menos complejos ocupan menos espacio en la ventana de contexto, pero generalmente también son menos potentes.

> **Consejo CoT**: Siempre haz que Claude muestre su pensamiento. ¡Sin mostrar su proceso de pensamiento, no ocurre ningún pensamiento!

- **Prompt básico**: Incluye "Piensa paso a paso" en tu prompt.
  - Carece de guía sobre *cómo* pensar (lo cual es especialmente inadecuado si una tarea es muy específica para tu aplicación, caso de uso u organización)
- **Prompt guiado**: Describe pasos específicos para que Claude siga en su proceso de pensamiento.
  - Carece de estructuración para facilitar la extracción y separación de la respuesta del pensamiento.
- **Prompt estructurado**: Usa etiquetas XML como `<thinking>` y `<answer>` para separar el razonamiento de la respuesta final.

Ejemplos robustos

attachment:a3cf3125-b675-4012-ad2d-8b562889f64c:Deja_que_Claude_piense_(encadenamiento_de_pensamiento)_para_mejorar_el_rendimiento_-_Anthropic.pdf

---

Part. 9

Usa etiquetas XML para estructurar tus prompts

Cuando tus prompts involucran múltiples componentes como contexto, instrucciones y ejemplos, las etiquetas XML pueden ser un cambio radical. Ayudan a Claude a analizar tus prompts con mayor precisión, lo que lleva a resultados de mayor calidad.

> **Consejo XML**: Usa etiquetas como `<instructions>`, `<example>`, y `<formatting>` para separar claramente diferentes partes de tu prompt. Esto evita que Claude confunda las instrucciones con ejemplos o contexto.

¿Por qué usar etiquetas XML?

- **Claridad:** Separa claramente diferentes partes de tu prompt y asegura que esté bien estructurado.
- **Precisión:** Reduce errores causados por Claude al malinterpretar partes de tu prompt.
- **Flexibilidad:** Encuentra, agrega, elimina o modifica fácilmente partes de tu prompt sin reescribir todo.
- **Capacidad de análisis:** Hacer que Claude use etiquetas XML en su salida facilita la extracción de partes específicas de su respuesta mediante post-procesamiento.

> No existen etiquetas XML “mejores” canónicas con las que Claude haya sido entrenado en particular, aunque recomendamos que los nombres de tus etiquetas tengan sentido con la información que rodean.

Mejores prácticas de etiquetado

1. **Sé consistente**: Usa los mismos nombres de etiquetas en todos tus prompts y refiérete a esos nombres de etiquetas cuando hables sobre el contenido (por ejemplo, `Usando el contrato en las etiquetas <contract>...`).
2. **Anida etiquetas**: Debes anidar etiquetas `<outer><inner></inner></outer>` para contenido jerárquico.

> **Consejo para usuarios avanzados**: Combina etiquetas XML con otras técnicas como prompting multishot (`<examples>`) o cadena de pensamiento (`<thinking>`, `<answer>`). Esto crea prompts súper estructurados y de alto rendimiento.

attachment:c4da741b-fd31-4cd5-9709-fb81cf466fc5:Usa_etiquetas_XML_para_estructurar_tus_prompts_-_Anthropic.pdf

---

Part. 10

Asignando un rol a Claude con un prompt de sistema

Cuando uses Claude, puedes mejorar dramáticamente su rendimiento utilizando el parámetro `system` para asignarle un rol. Esta técnica, conocida como prompting de rol, es la forma más poderosa de usar prompts de sistema con Claude.

¡El rol adecuado puede transformar a Claude de un asistente general a tu experto virtual en un dominio específico!

> **Consejos para prompts de sistema**: Usa el parámetro `system` para establecer el rol de Claude. Pon todo lo demás, como instrucciones específicas de la tarea, en el turno del `user`

¿Por qué usar prompting de rol?

- **Mayor precisión:** En escenarios complejos como análisis legal o modelado financiero, el prompting de rol puede aumentar significativamente el rendimiento de Claude.

- **Tono adaptado:** Ya sea que necesites la brevedad de un CFO o el estilo de un redactor publicitario, el prompting de rol ajusta el estilo de comunicación de Claude.
- **Mejor enfoque:** Al establecer el contexto del rol, Claude se mantiene más dentro de los límites de los requisitos específicos de tu tarea.

Cómo asignarle un rol a Claude

Usa el parámetro `system` en la **API de Messages** para establecer el rol de Claude:

```
import anthropic

client = anthropic.Anthropic()

response = client.messages.create(
    model="claude-3-7-sonnet-20250219",
    max_tokens=2048,
    system="You are a seasoned data scientist at a Fortune 500 company.",
# ← prompt de rol
    messages=[
        {"role": "user", "content": "Analyze this dataset for anomalies: <datase
t>{{DATASET}}</dataset>"}
    ]
)

print(response.content)
```

> **Consejo para prompting de rol**: ¡Experimenta con los roles! Un `científico de datos` podría ver diferentes perspectivas que un `estratega de marketing` para los mismos datos. Un `científico de datos especializado en análisis de insights de clientes para empresas Fortune 500` podría producir resultados aún diferentes!

Ejemplos

attachment:96bbc0c1-5638-492f-914c-44a44d6ce539:Asignando_un_rol_a_Claude_con_un_prompt_de_sistema_-_Anthropic.pdf

---

Part. 11

Prellenar la respuesta de Claude para un mayor control de la salida

> El prellenado solo está disponible para modos que no son de pensamiento extendido. Actualmente no es compatible con el pensamiento extendido.

Al usar Claude, tienes la capacidad única de guiar sus respuestas prellenando el mensaje del `Assistant`. Esta poderosa técnica te permite dirigir las acciones de Claude, omitir preámbulos, imponer formatos específicos como JSON o XML, e incluso ayudar a Claude a mantener la consistencia del personaje en escenarios de juego de rol.

En algunos casos donde Claude no está funcionando como se esperaba, unas pocas frases prellenadas pueden mejorar enormemente el rendimiento de Claude. ¡Un poco de prellenado tiene un gran impacto!

Cómo prellenar la respuesta de Claude

Para prellenar, incluye el texto inicial deseado en el mensaje del `Assistant` (la respuesta de Claude continuará desde donde termina el mensaje del `Assistant`):

```
import anthropic

client = anthropic.Anthropic()
response = client.messages.create(
    model="claude-opus-4-20250514",
    max_tokens=1024,
    messages=[
        {"role": "user", "content": "What is your favorite color?"},
        {"role": "assistant", "content": "As an AI assistant, I don't have a favorit
e color, But if I had to pick, it would be green because"}  # Prefill here
```

```
    ]
)
```

> ⚠️El contenido prellenado no puede terminar con espacios en blanco al final. Un prellenado como `"As an AI assistant, I "` (con un espacio al final) resultará en un error.

attachment:b1d8782a-98e5-476b-b64b-9905cab0017d:Prellenar_la_respuesta_de_Claude_para_un_mayor_control_de_la_salida_-_Anthropic.pdf

---

Part. 12

Encadena prompts complejos para un rendimiento más sólido

Cuando trabajas con tareas complejas, Claude a veces puede fallar si intentas manejar todo en un solo prompt. El prompt de cadena de pensamiento (CoT) es excelente, pero ¿qué pasa si tu tarea tiene múltiples pasos distintos que requieren cada uno un pensamiento profundo?

Aquí entra el encadenamiento de prompts: dividir tareas complejas en subtareas más pequeñas y manejables.

¿Por qué encadenar prompts?

1. **Precisión**: Cada subtarea recibe la atención completa de Claude, reduciendo errores.
2. **Claridad**: Las subtareas más simples significan instrucciones y resultados más claros.
3. **Trazabilidad**: Identifica y corrige fácilmente problemas en tu cadena de prompts.

Cuándo encadenar prompts

Usa el encadenamiento de prompts para tareas de múltiples pasos como síntesis de investigación, análisis de documentos o creación iterativa de contenido. Cuando una tarea involucra múltiples transformaciones, citas o instrucciones, el encadenamiento evita que Claude omita o maneje mal los pasos.

Recuerda: ¡Cada eslabón en la cadena recibe la atención completa de Claude!

> **Consejo de depuración**: Si Claude omite un paso o tiene un mal desempeño, aísla ese paso en su propio prompt. Esto te permite ajustar los pasos problemáticos sin rehacer toda la tarea.

Cómo encadenar prompts

1. **Identifica subtareas**: Divide tu tarea en pasos distintos y secuenciales.
2. **Estructura con XML para transferencias claras**: Usa etiquetas XML para pasar resultados entre prompts.
3. **Ten un objetivo de tarea única**: Cada subtarea debe tener un objetivo único y claro.
4. **Itera**: Refina las subtareas según el rendimiento de Claude.

Ejemplos de flujos de trabajo encadenados:

- **Análisis de múltiples pasos**: Ver los ejemplos legales y de negocios a continuación.
- **Tuberías de creación de contenido**: Investigación → Esquema → Borrador → Edición → Formato.
- **Procesamiento de datos**: Extraer → Transformar → Analizar → Visualizar.
- **Toma de decisiones**: Recopilar información → Listar opciones → Analizar cada una → Recomendar.
- **Bucles de verificación**: Generar contenido → Revisar → Refinar → Re-revisar.

> **Consejo de optimización**: Para tareas con subtareas independientes (como analizar múltiples documentos), crea prompts separados y ejecútalos en paralelo para mayor velocidad.

Ejemplos

attachment:1ae96f36-6917-475e-a3db-9c153cc12e30:Encadena_prompts_complejos_para_un_rendimiento_ms_slido_-_Anthropic.pdf

---

Part. 13

Consejos para el uso de contexto largo

La ventana de contexto extendida de Claude (200K tokens para los modelos Claude 3) permite manejar tareas complejas y ricas en datos. Esta guía te ayudará a aprovechar este poder de manera efectiva.

Consejos esenciales para prompts con contexto largo

- **Coloca los datos extensos en la parte superior**: Ubica tus documentos largos y entradas (~20K+ tokens) cerca de la parte superior de tu prompt, por encima de tu consulta, instrucciones y ejemplos. Esto puede mejorar significativamente el rendimiento de Claude en todos los modelos.

> Las consultas al final pueden mejorar la calidad de la respuesta hasta en un 30% en las pruebas, especialmente con entradas complejas y multi-documento.

- **Estructura el contenido del documento y los metadatos con etiquetas XML**: Cuando uses múltiples documentos, envuelve cada documento en etiquetas `<document>` con subtags `<document_content>` y `<source>` (y otros metadatos) para mayor claridad.
    - **Ejemplo de estructura multi-documento:**

```
<documents>
 <document index="1">
  <source>annual_report_2023.pdf</source>
  <document_content>
   {{ANNUAL_REPORT}}
  </document_content>
 </document>
 <document index="2">
  <source>competitor_analysis_q2.xlsx</source>
  <document_content>
   {{COMPETITOR_ANALYSIS}}
  </document_content>
 </document>
</documents>

Analiza el informe anual y el análisis de la competencia. Identifica las v
entajas estratégicas y recomienda áreas de enfoque para el Q3.
```

- **Fundamenta las respuestas en citas**: Para tareas con documentos largos, pide a Claude que cite primero las partes relevantes de los documentos antes de realizar su tarea. Esto ayuda a Claude a filtrar el “ruido” del resto del contenido del documento.
  - **Ejemplo de extracción de citas**

```
Eres un asistente médico AI. Tu tarea es ayudar a los médicos a diagno
sticar posibles enfermedades de los pacientes.

<documents>
 <document index="1">
  <source>patient_symptoms.txt</source>
  <document_content>
   {{PATIENT_SYMPTOMS}}
  </document_content>
 </document>
 <document index="2">
  <source>patient_records.txt</source>
```

```
  <document_content>
   {{PATIENT_RECORDS}}
  </document_content>
 </document>
 <document index="3">
  <source>patient01_appt_history.txt</source>
  <document_content>
   {{PATIENT01_APPOINTMENT_HISTORY}}
  </document_content>
 </document>
</documents>

Encuentra citas de los registros del paciente y el historial de citas que s
ean relevantes para diagnosticar los síntomas reportados del paciente.
Colócalas en etiquetas <quotes>. Luego, basándote en estas citas, enu
mera toda la información que ayudaría al médico a diagnosticar los sínt
omas del paciente. Coloca tu información diagnóstica en etiquetas <inf
o>.
```

attachment:9660d85d-0b00-40b4-a826-a5622cba31c6:Consejos_para_el_uso_de_contexto_largo_-_Anthropic.pdf

---

Part. 14

Consejos para el pensamiento extendido

Esta guía proporciona estrategias y técnicas avanzadas para aprovechar al máximo las funciones de pensamiento extendido de Claude. El pensamiento extendido permite a Claude resolver problemas complejos paso a paso, mejorando el rendimiento en tareas difíciles.

Consulta **Modelos de pensamiento extendido** para obtener orientación sobre cuándo usar el pensamiento extendido.

Antes de comenzar

Esta guía presupone que ya has decidido utilizar el modo de pensamiento extendido y has revisado nuestros pasos básicos sobre **cómo empezar con el pensamiento extendido**, así como nuestra **guía de implementación del pensamiento extendido**.

Consideraciones técnicas para el pensamiento extendido

- Los tokens de pensamiento tienen un presupuesto mínimo de 1024 tokens. Recomendamos que comiences con el presupuesto mínimo de pensamiento y lo incrementes gradualmente según tus necesidades y la complejidad de la tarea.
- Para cargas de trabajo donde el presupuesto óptimo de pensamiento supera los 32K, recomendamos que utilices **procesamiento por lotes** para evitar problemas de red. Las solicitudes que empujan al modelo a pensar por encima de 32K tokens causan solicitudes de larga duración que podrían chocar contra los tiempos de espera del sistema y los límites de conexiones abiertas.
- El pensamiento extendido funciona mejor en inglés, aunque los resultados finales pueden estar en **cualquier idioma que Claude soporte**.
- Si necesitas un pensamiento por debajo del presupuesto mínimo, recomendamos usar el modo estándar, con el pensamiento desactivado, con el tradicional encadenamiento de pensamiento mediante etiquetas XML (como `<thinking>`). Consulta **encadenamiento de pensamiento**.

Técnicas de prompt para el pensamiento extendido

Usa primero instrucciones generales, luego soluciona problemas con instrucciones más detalladas paso a paso

Claude a menudo funciona mejor con instrucciones de alto nivel para pensar profundamente sobre una tarea en lugar de una guía prescriptiva paso a paso. La creatividad del modelo para abordar problemas puede superar la capacidad humana para prescribir el proceso de pensamiento óptimo.

Por ejemplo, en lugar de:

> Piensa en este problema matemático paso a paso:
> 1. Primero, identifica las variables
> 2. Luego, establece la ecuación
> 3. A continuación, resuelve para x
> ...

Considera:

> Por favor, piensa en este problema matemático de manera exhaustiva y co n gran detalle.
> Considera múltiples enfoques y muestra tu razonamiento completo.
> Prueba diferentes métodos si tu primer enfoque no funciona.

Dicho esto, Claude todavía puede seguir eficazmente pasos de ejecución estructurados complejos cuando sea necesario. El modelo puede manejar listas aún más largas con instrucciones más complejas que las versiones anteriores. Recomendamos que comiences con instrucciones más generalizadas, luego leas el resultado del pensamiento de Claude e iteres para proporcionar instrucciones más específicas para dirigir su pensamiento a partir de ahí.

Prompting multishot con pensamiento extendido

El **prompting multishot** funciona bien con el pensamiento extendido. Cuando proporcionas a Claude ejemplos de cómo pensar a través de problemas, seguirá patrones de razonamiento similares dentro de sus bloques de pensamiento extendido.

Puedes incluir ejemplos de pocos disparos (few-shot) en tu prompt en escenarios de pensamiento extendido utilizando etiquetas XML como `<thinking>` o `<scratchpad>` para indicar patrones canónicos de pensamiento extendido en esos ejemplos.

Claude generalizará el patrón al proceso formal de pensamiento extendido. Sin embargo, es posible que obtengas mejores resultados dando a Claude libertad para pensar de la manera que considere mejor.

Ejemplo:

Voy a mostrarte cómo resolver un problema matemático, luego quiero que r esuelvas uno similar.

Problema 1: ¿Cuánto es el 15% de 80?

<thinking>
Para encontrar el 15% de 80:
1. Convierte 15% a decimal: 15% = 0.15
2. Multiplica: 0.15 × 80 = 12
</thinking>

La respuesta es 12.

Ahora resuelve este:
Problema 2: ¿Cuánto es el 35% de 240?

Maximizando el seguimiento de instrucciones con pensamiento extendido

Claude muestra una mejora significativa en el seguimiento de instrucciones cuando el pensamiento extendido está habilitado. El modelo típicamente:

1. Razona sobre las instrucciones dentro del bloque de pensamiento extendido
2. Ejecuta esas instrucciones en la respuesta

Para maximizar el seguimiento de instrucciones:

- Sé claro y específico sobre lo que quieres
- Para instrucciones complejas, considera dividirlas en pasos numerados que Claude debe seguir metódicamente
- Permite a Claude suficiente presupuesto para procesar completamente las instrucciones en su pensamiento extendido

Usando el pensamiento extendido para depurar y dirigir el comportamiento de Claude

Puedes usar el resultado del pensamiento de Claude para depurar la lógica de Claude, aunque este método no siempre es perfectamente confiable.

Para hacer el mejor uso de esta metodología, recomendamos los siguientes consejos:

- **No recomendamos pasar el pensamiento extendido de Claude de vuelta en el bloque de texto del usuario, ya que esto no mejora el rendimiento y puede degradar los resultados.**
- **El prellenado del pensamiento extendido está explícitamente prohibido, y cambiar manualmente el texto de salida del modelo que sigue a su bloque de pensamiento probablemente degradará los resultados debido a la confusión del modelo.**

Cuando el pensamiento extendido está desactivado, el **prellenado** estándar del texto de respuesta del `assistant` sigue estando permitido.

> A veces Claude puede repetir su pensamiento extendido en el texto de salida del asistente. Si quieres una respuesta limpia, instruye a Claude para que no repita su pensamiento extendido y que solo muestre la respuesta.

Aprovechando al máximo las salidas largas y el pensamiento de forma extensa

Para casos de uso de generación de conjuntos de datos, prueba prompts como “Por favor, crea una tabla extremadamente detallada de...” para generar conjuntos de datos completos.

Para casos de uso como la generación de contenido detallado donde puedes querer generar bloques de pensamiento extendido más largos y respuestas más detalladas, prueba estos consejos:

- Aumenta tanto la longitud máxima del pensamiento extendido COMO pide explícitamente salidas más largas
- Para salidas muy largas (más de 20,000 palabras), solicita un esquema detallado con recuentos de palabras hasta el nivel de párrafo. Luego pide a Claude que indexe sus párrafos según el esquema y mantenga los recuentos de palabras especificados

> ⚠️ **No recomendamos que empujes a Claude a generar más tokens por el simple hecho de generar tokens. Más bien, te animamos a comenzar con un pequeño presupuesto de pensamiento y aumentarlo según sea necesario para encontrar la configuración óptima para tu caso de uso.**

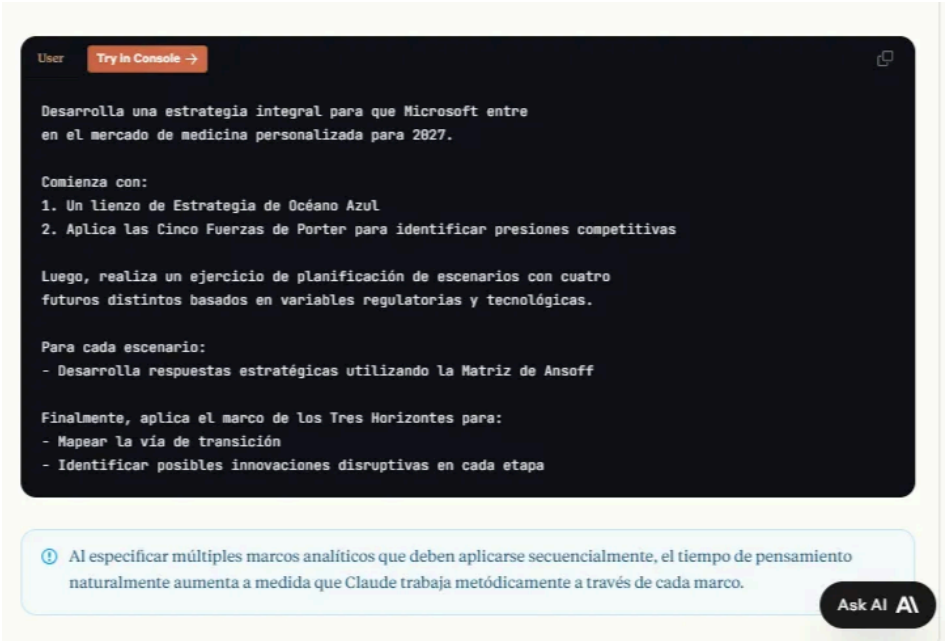
Prompts mejorados

User Try in Console →

```
Planifica un viaje de 7 días a Japón con las siguientes restricciones:
- Presupuesto de $2,500
- Debe incluir Tokio y Kioto
- Necesidad de acomodar una dieta vegetariana
- Preferencia por experiencias culturales sobre compras
- Debe incluir un día de senderismo
- No más de 2 horas de viaje entre ubicaciones por día
- Necesidad de tiempo libre cada tarde para llamadas a casa
- Debe evitar multitudes cuando sea posible
```

Con múltiples restricciones para equilibrar, Claude naturalmente funcionará mejor cuando se le dé más espacio para pensar cómo satisfacer todos los requisitos de manera óptima.

Ask AI



User Try in Console →

```
Desarrolla una estrategia integral para que Microsoft entre
en el mercado de medicina personalizada para 2027.

Comienza con:
1. Un lienzo de Estrategia de Océano Azul
2. Aplica las Cinco Fuerzas de Porter para identificar presiones competitivas

Luego, realiza un ejercicio de planificación de escenarios con cuatro
futuros distintos basados en variables regulatorias y tecnológicas.

Para cada escenario:
- Desarrolla respuestas estratégicas utilizando la Matriz de Ansoff

Finalmente, aplica el marco de los Tres Horizontes para:
- Mapear la vía de transición
- Identificar posibles innovaciones disruptivas en cada etapa
```

Al especificar múltiples marcos analíticos que deben aplicarse secuencialmente, el tiempo de pensamiento naturalmente aumenta a medida que Claude trabaja metódicamente a través de cada marco.

Ask AI

Haz que Claude reflexione y verifique su trabajo para mejorar la consistencia y el manejo de errores

Puedes usar prompting en lenguaje natural simple para mejorar la consistencia y reducir errores:

1. Pide a Claude que verifique su trabajo con una prueba simple antes de declarar una tarea completa
2. Instruye al modelo para que analice si su paso anterior logró el resultado esperado
3. Para tareas de programación, pide a Claude que ejecute casos de prueba en su pensamiento extendido

Ejemplo:

User Try in Console →

```
Escribe una función para calcular el factorial de un número.
Antes de terminar, por favor verifica tu solución con casos de prueba para:
- n=0
- n=1
- n=5
- n=10
Y corrige cualquier problema que encuentres.
```

attachment:74b48f59-5bda-415e-a318-40499b7ae31c:Consejos_para_el_pensamiento_extendido_-_Anthropic.pdf

CONTENIDO DENSO DE PENSAMIENTO EXTENDIDO CON EJEMPLOS