

# Mastering Claude

La guía definitiva  
para programadores:  
agentes, automatización  
y arquitecturas de IA



Francisco Montes de Oca Molina

2026

# Mastering Claude

---

La guía definitiva para programadores: agentes, automatización y arquitecturas de IA

Francisco Montes de Oca Molina

2026

# Índice

---

## Índice

### Mastering Claude

---

La guía definitiva para programadores: agentes, automatización y arquitecturas de IA

Prólogo

---

### PARTE I – FUNDAMENTOS Y MENTALIDAD

---

Capítulo 1: ¿Por qué Claude cambia el trabajo del programador?

Capítulo 2: Cómo piensa Claude – modelos, tokens y contexto

Capítulo 3: Configurar tu entorno de trabajo con Claude

Convenciones importantes

Estructura del proyecto

Qué NO hacer

Capítulo 4: El arte del prompting para programadores

Capítulo 5: Técnicas avanzadas de prompting

---

### PARTE II – CLAUDE COMO COPILOTO DE CÓDIGO

---

Capítulo 6: Generación de código con criterio

Capítulo 7: Explicación y comprensión de código

Capítulo 8: Debugging asistido por IA

Capítulo 9: Refactorización y mejora de código

---

Capítulo 10: Testing asistido por IA

---

Capítulo 11: Documentación y revisión de código

---

## PARTE III – ARQUITECTURAS Y SISTEMAS

---

Capítulo 12: Tool Use – Dar manos al modelo

---

Capítulo 13: RAG – Claude con conocimiento externo

---

Capítulo 14: Memoria en sistemas de IA

---

Capítulo 15: Flujos de trabajo y automatización

---

Capítulo 16: Agentes – De conversación a sistema

---

## PARTE IV – CLAUDE CODE Y EL STACK MODERNO

---

Capítulo 17: Claude Code – El entorno de desarrollo con IA

---

Stack técnico

---

Convenciones críticas

---

Estructura de directorios

---

Decisiones de arquitectura

---

Trampas conocidas

---

Capítulo 18: Skills y comandos personalizados

---

Criterio de severidad

---

Capítulo 19: Claude Code en profundidad – comandos, contexto y productividad diaria

---

Capítulo 20: settings.json, permisos y hooks

---

Capítulo 21: Claude Code y Git – commits, PRs y trabajo en equipo

---

Capítulo 22: Un día real con Claude Code

---

Capítulo 23: Model Context Protocol (MCP)

---

Capítulo 24: Extended Thinking y capacidades avanzadas

---

Capítulo 25: Integraciones y herramientas de desarrollo

---

## PARTE V – PROYECTO REAL DE PRINCIPIO A FIN

---

Capítulo 26: Arquitectura del proyecto – Agente revisor de código

---

Capítulo 27: Construyendo el agente paso a paso

---

Capítulo 28: Orquestación – Planner, Coder, Reviewer

---

Capítulo 29: Producción – Seguridad, costos y observabilidad

---

## APÉNDICES

---

Apéndice A: Biblioteca de prompts listos para usar

---

Apéndice B: Blueprint de proyecto real

---

Stack

---

Convenciones críticas

---

Qué NO hacer

---

Apéndice C: Referencias oficiales

---

# Mastering Claude

---

# La guía definitiva para programadores: agentes, automatización y arquitecturas de IA

Francisco Montes de Oca Molina

---

## Prólogo

Llevo dieciocho años construyendo software. He trabajado en startups que desaparecieron en seis meses y en empresas que crecieron hasta millones de usuarios. He escrito código que funcionó en producción por años y código que rompió producción en minutos. He liderado equipos y he sido el único desarrollador en el cuarto. Y en todo ese tiempo, la lección más valiosa no fue ningún patrón de diseño ni ningún framework: fue aprender a tener criterio sobre cuándo usar qué.

Ese criterio tarda tiempo en construirse. Se forma a base de errores costosos, de bugs que te quitan el sueño, de decisiones de arquitectura que seis meses después te hacen arrepentirte. No hay atajo para eso. Pero sí hay herramientas que pueden reducir la fricción del camino.

La inteligencia artificial es una de ellas. Y Claude, en particular, es la que más he usado en los últimos años para trabajo técnico real.

Cuando empecé a usarla en serio, mi primera reacción fue resistencia. La narrativa dominante en ese momento era “la IA va a reemplazar a los programadores”, y yo —como probablemente tú— no quería creerlo ni quería explorarlo demasiado por miedo a validar algo que me parecía absurdo. Pero la curiosidad ganó. Empecé a usar Claude para tareas pequeñas: explicar un error raro, proponer una estructura para un módulo nuevo, generar el esqueleto de unos tests. Y lo que descubrí fue algo completamente diferente a lo que prometían los titulares.

La IA no reemplazó al programador. Lo multiplicó.

No es exageración. Hay tareas que antes me tomaban cuarenta y cinco minutos y ahora me toman doce. No porque Claude las haga por mí de forma mágica, sino porque elimina la fricción de arrancar, de buscar el patrón correcto, de recordar la sintaxis exacta de una librería que uso una vez al mes. Esa fricción pequeña, repetida decenas de veces al día, se acumula. Y cuando la eliminas, recuperas algo muy valioso: energía mental.

Esa energía mental extra es lo que más me ha cambiado el trabajo. No es solo velocidad. Es claridad. Cuando no estás gastando atención en escribir boilerplate o en buscar la quinta vez cómo se hace un reduce en TypeScript, tienes más espacio para pensar en lo que realmente importa: el diseño, la arquitectura, los edge cases, las implicaciones del negocio. Las decisiones que sí requieren criterio humano.

Y eso tiene un efecto secundario que no esperaba: mejor balance de vida. Cuando terminas tu trabajo más rápido y con menos fricción, el margen de tiempo que recuperas no tiene que ir a trabajar más. Puede ir a vivir mejor. A estar con tu familia. A descansar. A aprender algo que no tiene nada que ver con código. Eso, para mí, es el beneficio más honesto y menos hypeado de usar IA bien.

Este libro parte de esa experiencia. Está escrito para programadores que conocen su oficio — que saben TypeScript, que entienden cómo funciona una API REST, que han sufrido bugs de producción — pero que todavía no han encontrado la manera de integrar Claude de forma efectiva en su trabajo diario.

No está escrito para admirar la tecnología desde lejos. Está escrito para usarla con los pies en la tierra.

La promesa de este libro es simple: al terminar de leerlo, sabrás usar Claude como copiloto, no como muleta. La diferencia es importante. Un copiloto te ayuda a hacer mejor tu trabajo mientras tú sigues siendo quien decide la dirección. Una muleta te quita responsabilidad y te hace dependiente. El primero te hace mejor programador. El segundo te hace peor.

Para lograrlo, hay tres patrones que van a repetirse a lo largo de estos capítulos. Primero: la diferencia entre un prompt débil y un prompt fuerte no es magia, es comunicación con contexto. Aprenderás a construir prompts que funcionan porque tienen estructura, no porque tienen palabras clave mágicas. Segundo: la iteración es parte del proceso. El primer resultado de Claude es casi siempre un borrador, y eso está bien. Los mejores resultados

llegan después de dos o tres intercambios bien dirigidos. Tercero: siempre verificas. Claude puede equivocarse con mucha fluidez verbal. La responsabilidad de revisar, probar y validar siempre es tuya.

Esa última responsabilidad no es una carga. Es lo que hace que tu trabajo siga siendo valioso. Claude puede generar el código; tú decides si es correcto, si es seguro, si encaja en la arquitectura, si resuelve el problema real. Esa decisión final es irremplazable.

Puedes leer este libro de principio a fin o usarlo como referencia. Los capítulos están contruidos para que tengan sentido en orden, especialmente los de las primeras dos partes. Pero si ya tienes experiencia básica con la API y quieres ir directo a agentes o a la parte de producción, puedes saltar a la Parte III o IV sin perderte el hilo.

Lo que no debes hacer es leerlo sin aplicarlo. Cada capítulo tiene código que funciona. Cópialo, rómpelo, mejóralo. Esa es la única forma en que este libro va a cambiar cómo programas.

---

# PARTE I – FUNDAMENTOS Y MENTALIDAD

---

# Capítulo 1: ¿Por qué Claude cambia el trabajo del programador?

Existe una narrativa que lleva años circulando en foros técnicos, conferencias y titulares de tecnología: la inteligencia artificial va a reemplazar a los programadores. Es una afirmación que genera reacciones fuertes. Algunos la celebran como si fuera inevitable. Otros la descartan como hype puro. Y la mayoría de los desarrolladores la escuchan con una mezcla incómoda de escepticismo y ansiedad.

La narrativa es falsa. Pero entender por qué es falsa importa casi tanto como entender qué sí cambia.

## Por qué persiste el mito del reemplazo

El argumento suena lógico a primera vista: si una herramienta puede escribir código, y los programadores escriben código, entonces la herramienta puede hacer el trabajo del programador. Pero esa lógica tiene un error fundamental de categoría. Confunde escribir código con hacer software.

Hacer software implica entender un problema de negocio que muchas veces está mal definido. Implica tomar decisiones de arquitectura bajo restricciones reales de tiempo, equipo y presupuesto. Implica anticipar cómo va a fallar algo antes de que falle. Implica negociar con un cliente que quiere todo para ayer. Implica mantener un sistema que fue escrito por cuatro personas distintas a lo largo de tres años, con estándares inconsistentes y sin tests.

Claude puede escribir una función. No puede reemplazar el criterio que se necesita para saber qué función escribir, cuándo no escribirla, cómo integrarla con el sistema existente sin romper nada, o cómo explicarle al equipo por qué esta aproximación es mejor que aquella.

El mito persiste porque los demos son impresionantes. Ver a un modelo generar un componente React completo en diez segundos se ve espectacular. Pero los demos siempre muestran el caso feliz en un contexto

limpio. No muestran el legacy code con veinte años de decisiones acumuladas. No muestran los requisitos que cambian tres veces en una semana. No muestran la conversación con el usuario de negocio que describe el problema de una manera y quiere decir otra.

## Qué sí cambia: los puntos de fricción

Lo que sí cambia, y cambia de forma significativa, son los puntos de fricción que drenan energía durante el día de un programador.

Piensa en tu última semana de trabajo. ¿Cuánto tiempo pasaste escribiendo boilerplate que ya sabías cómo hacer, pero que igual tomaba tiempo porque había que hacerlo? ¿Cuánto tiempo buscaste en la documentación la sintaxis exacta de algo que usas una vez al mes? ¿Cuánto tiempo explicaste un bug a alguien, o a ti mismo en voz alta, antes de encontrar la causa? ¿Cuánto tiempo tardaste en escribir los tests para una función cuyo comportamiento ya conocías perfectamente?

Esas actividades no son el núcleo del oficio del programador. Son la fricción que rodea al núcleo. Y Claude es excepcionalmente bueno en reducir esa fricción.

Una tarea que antes tomaba cuarenta y cinco minutos — escribir un endpoint con validación, manejo de errores y tests — ahora puede tomar doce. No porque Claude lo haga todo, sino porque genera el ochenta por ciento del código estructural mientras tú revisas, ajustas y agregas lo que requiere criterio real: las reglas de negocio específicas, los edge cases que vienen del conocimiento del dominio, la forma correcta de manejar los errores según el contrato de la API.

Esos doce minutos incluyen tu revisión. No es que generes y aceptes ciegamente. Lees el código, lo entiendes, lo pruebas. Pero el punto de partida es mucho más avanzado.

## El modelo del copiloto

La metáfora más útil para entender la relación correcta con Claude es la del copiloto de aviación. El copiloto no pilota el avión. Asiste al piloto, monitorea sistemas, ejecuta procedimientos de checklist, comunica con control de tráfico aéreo. Pero cuando hay que tomar una decisión crítica en condiciones difíciles, el piloto manda.

Tú eres el piloto. Claude es el copiloto.

Esta distinción importa porque determina cómo usas la herramienta. Un copiloto bien usado reduce tu carga cognitiva en las tareas mecánicas para que puedas concentrar tu atención en las que requieren juicio. Un copiloto mal usado — cuando le delegas decisiones que debería tomar el piloto— crea riesgo real.

La pregunta correcta no es “¿puede Claude hacer esto?” sino “¿debería ser Claude quien decida esto?”. Para escribir el esqueleto de un módulo: sí. Para decidir cómo manejar la autenticación en tu sistema de pagos: no, o al menos no sin revisión cuidadosa.

## Lo que Claude no hace bien

Ser honesto sobre las limitaciones no es pesimismo. Es lo que te permite usar la herramienta sin sorpresas desagradables.

Claude no tiene juicio arquitectónico genuino. Puede proponer opciones, comparar trade-offs y generar documentación de decisiones, pero no conoce tu sistema, tu equipo, tu deuda técnica ni las restricciones implícitas que vienen de años de contexto organizacional. Una propuesta de arquitectura de Claude es un punto de partida para pensar, no una recomendación final.

Claude no entiende las reglas de negocio implícitas. Sabe lo que le dices. Lo que no le dices —las excepciones que todos en el equipo conocen pero nunca están escritas, el cliente especial que tiene reglas distintas, la razón histórica por la que ese módulo hace algo que parece raro— no lo conoce y no puede adivinarlo bien.

Claude no valida seguridad de forma confiable. Puede identificar patrones comunes de vulnerabilidades, pero no es un auditor de seguridad. Código que Claude genera y que no tiene errores obvios puede tener vulnerabilidades sutiles que solo se detectan con revisión experta.

Claude no siempre tiene razón aunque suene seguro. Esta es la característica más peligrosa: el modelo genera texto fluido y confiado incluso cuando está equivocado. Un programador que no revisa código de Claude por parecer correcto está tomando un riesgo real.

## El tiempo recuperado

Cuando eliminas dos horas de trabajo repetitivo por semana, la pregunta que aparece es: ¿qué haces con ese tiempo?

La respuesta más obvia —y menos interesante— es trabajar más. Hacer más features, cerrar más tickets, asumir más proyectos. Eso puede tener valor, pero no es el beneficio más importante.

El beneficio más importante es recuperar energía mental para lo que sí requiere tu mejor atención. Pensar mejor el diseño. Revisar el código de tu equipo con más cuidado. Dedicar tiempo a entender el sistema completo en vez de solo el módulo que tienes que entregar esta semana. Aprender algo nuevo sin sentir que estás robando tiempo de trabajo urgente.

Y fuera del trabajo: estar presente cuando no estás trabajando. El programador que llega a casa con la mente saturada de cosas por hacer no descansa bien. La IA no elimina ese problema, pero reduce la acumulación de tareas pequeñas que contribuyen a él.

## La identidad del programador

Hay algo más que quiero nombrar directamente, porque lo escucho con frecuencia en conversaciones con otros desarrolladores.

Usar IA para generar código no te hace menos programador. El valor de un desarrollador experimentado no está en su velocidad de tipeo ni en la cantidad de APIs de librerías que tiene memorizadas. Está en su criterio. En su capacidad de diseñar sistemas que escalen, de anticipar fallos, de tomar decisiones técnicas con información incompleta, de comunicar esas decisiones a personas que no son técnicas.

Todo eso sigue siendo tuyo cuando usas Claude. De hecho, cuando la fricción baja, esas capacidades se expresan más. Tienes más tiempo para diseñar bien porque tardas menos en escribir lo que ya sabes que necesitas escribir.

---

### ✗ Prompt débil:

```
Make a function for users.
```

### ✓ Prompt fuerte:

Escribe una función TypeScript llamada `groupUsersByRole` con esta firma:

```
function groupUsersByRole(users: Array<{ id: string; name: string; role: 'admin' | 'editor' | 'viewer' }>): Record<string, string[]>
```

Requisitos:

- Sin `any`
- Maneja arreglo vacío retornando `{}`
- El valor de cada llave es un arreglo de nombres (no ids)
- Incluye un ejemplo de uso en comentario
- Explica la complejidad temporal en una línea

La diferencia no está en la longitud. Está en la especificidad: tipo de entrada, tipo de salida, restricciones explícitas, comportamiento en edge cases. Claude no necesita adivinar nada. El resultado es directamente usable.

---

**Mini resumen:** Claude cambia el trabajo del programador reduciendo la fricción en tareas mecánicas, no reemplazando el criterio técnico. El modelo del copiloto —tú decides, Claude asiste— es la forma correcta de pensar la relación. El tiempo que se recupera bien usado se traduce en mejor diseño, mayor calidad y mejor balance de vida.

---

## Capítulo 2: Cómo piensa Claude — modelos, tokens y contexto

Para usar Claude bien, no necesitas entender los detalles matemáticos de cómo funciona un transformador. Pero sí necesitas entender algunos conceptos que afectan directamente cómo diseñas tus interacciones, cuánto cuestan y qué puedes esperar de los resultados.

### Qué es un modelo de lenguaje, en términos prácticos

Un modelo de lenguaje grande es, en esencia, un sistema entrenado para predecir texto. Durante el entrenamiento, procesó cantidades masivas de texto — código, documentación, libros, conversaciones — y aprendió los patrones estadísticos de cómo las palabras y los conceptos se relacionan entre sí.

Cuando le haces una pregunta, el modelo no busca la respuesta en una base de datos. Genera texto token por token, eligiendo en cada paso la continuación más probable dadas tus instrucciones, el historial de la conversación y los patrones aprendidos durante el entrenamiento. Es predicción de texto a escala masiva, con una capacidad emergente de razonamiento que nadie diseñó explícitamente pero que aparece cuando el modelo es suficientemente grande y el entrenamiento suficientemente extenso.

Esto explica algo importante: Claude puede dar respuestas plausibles pero incorrectas. El modelo no verifica sus afirmaciones contra una fuente de verdad. Genera lo que es estadísticamente coherente con el contexto. Cuando el contexto que le das es claro y completo, las respuestas son buenas. Cuando el contexto es ambiguo, el modelo rellena huecos con lo que parece más probable, que puede no coincidir con la realidad de tu proyecto.

La implicación práctica: siempre verificas. No porque Claude sea malo, sino porque así funciona.

## Tokens: la moneda del modelo

Antes de entender el contexto, necesitas entender los tokens, porque son la unidad básica con la que trabaja el modelo.

Un token no es exactamente una palabra. Es un fragmento de texto que puede ser una palabra completa, parte de una palabra, un signo de puntuación o un espacio. En inglés y en código, la proporción aproximada es de 3-4 caracteres por token. En español y otros idiomas con más morfología, tiende a ser un poco más.

Como referencia práctica: - Una función TypeScript de 20 líneas: aproximadamente 120-150 tokens - Un archivo TypeScript de 200 líneas: aproximadamente 1,200-1,500 tokens - Este capítulo completo: aproximadamente 2,500-3,000 tokens - Un repositorio mediano con 50 archivos: puede superar fácilmente los 100,000 tokens

Los tokens importan por dos razones. Primero, el costo: los modelos cobran por tokens de entrada y de salida. Segundo, el límite de contexto: cada modelo tiene un número máximo de tokens que puede procesar en una sola llamada.

```

import Anthropic from '@anthropic-ai/sdk';

const client = new Anthropic({ apiKey: process.env.ANTHROPIC_API_KEY
});

const response = await client.messages.create({
  model: 'claude-sonnet-4-5',
  max_tokens: 1024,
  messages: [{ role: 'user', content: "Explica este error: Cannot
read property of undefined" }],
});

console.log(response.content[0].text);
console.log(`Tokens usados: ${response.usage.input_tokens} entrada,
${response.usage.output_tokens} salida`);

```

El campo `response.usage` es tu mejor aliado para entender el costo real de cada llamada. Guarda esos números cuando estés diseñando sistemas que van a hacer muchas llamadas.

## El contexto y sus límites

El contexto es todo lo que el modelo puede "ver" en una llamada: el system prompt, el historial de mensajes, el texto que adjuntas. Claude 3.5 Sonnet y versiones posteriores tienen ventanas de contexto de 200,000 tokens, lo que equivale a varios libros o a un repositorio mediano completo.

Pero el tamaño del contexto disponible no significa que debas llenarlo todo. Hay dos problemas con el contexto excesivo.

Primero, el costo. Pagas por cada token de entrada, incluyendo el contexto. Un sistema prompt de 10,000 tokens que envías en cada llamada puede representar una fracción significativa del costo total si haces muchas llamadas.

Segundo, la calidad. Esto es menos obvio: el contexto irrelevante no solo ocupa espacio, activamente daña la calidad de las respuestas. El modelo le da atención a todo lo que está en el contexto. Si llenas el contexto con información que no es relevante para la tarea actual, el modelo dedica atención a esa información y puede distraerse de lo que importa.

La regla práctica: incluye todo lo que es relevante, nada de lo que no lo es.

Cuando un archivo o repositorio es demasiado grande para el contexto, tienes tres estrategias: - **Resumir**: extrae las partes relevantes del código o documentación y pásalas en lugar del archivo completo - **Fragmentar**: divide el trabajo en partes más pequeñas, cada una con el contexto específico que necesita - **Recuperar**: usa RAG (Capítulo 13) para encontrar y pasar solo los fragmentos relevantes



Figura 1. Todo lo que entra en la ventana de contexto en una llamada al modelo.

## Las familias de modelos

Anthropic ofrece diferentes modelos con distintas capacidades y costos. Entender cuándo usar cada uno es parte importante de trabajar con Claude de manera efectiva.

**Claude Haiku** es el modelo más rápido y económico. Responde en fracciones de segundo y su costo es significativamente menor que los modelos más grandes. Es ideal para tareas de clasificación, enrutamiento, resúmenes cortos, reescrituras simples y cualquier cosa donde no necesitas razonamiento complejo. Si estás construyendo un sistema que hace muchas llamadas simples —clasificar si un error es crítico o no, decidir a qué agente enviar una tarea, generar un mensaje de error amigable a partir de una excepción técnica— Haiku es la elección correcta.

**Claude Sonnet** es el modelo de trabajo general. Equilibra bien capacidad de razonamiento, velocidad y costo. Para la mayoría de las tareas de programación — generar código, debuggear, refactorizar, escribir tests, explicar sistemas — Sonnet es el punto de partida correcto. Cuando en este libro decimos “Claude” sin especificar modelo, nos referimos a Sonnet.

**Claude Opus** es el modelo más capaz. Tiene la mayor profundidad de razonamiento y es el mejor para problemas complejos: decisiones de arquitectura con muchas variables, debugging de sistemas distribuidos con comportamiento emergente, análisis de trade-offs en contextos con restricciones múltiples. También es el más caro y el más lento. Úsalo cuando el problema genuinamente lo necesite, no por defecto.

La regla de selección es simple: empieza con Sonnet. Si los resultados son consistentemente buenos, quédate ahí. Si necesitas más profundidad en razonamiento para tareas específicas, prueba Opus para esas tareas. Si tienes tareas de alta frecuencia y baja complejidad, muévelas a Haiku.

## No determinismo y sus implicaciones

El mismo prompt enviado dos veces no produce el mismo resultado. Esto no es un bug, es una característica fundamental del diseño. El modelo usa muestreo probabilístico, no búsqueda determinista.

Para código, esto tiene implicaciones prácticas. No puedes asumir que porque Claude generó una solución específica ayer, generará la misma hoy. Los tests que validan el output de Claude tienen que evaluar comportamiento, no texto exacto.

La temperatura es el parámetro que controla cuánta aleatoriedad tiene el muestreo. Una temperatura de 0 hace al modelo más determinista (elige siempre la opción más probable). Una temperatura alta hace las respuestas más variadas y creativas. Para código, la temperatura por defecto es generalmente la correcta. No necesitas ajustarla salvo en casos específicos donde quieres explorar variaciones o donde necesitas más predictibilidad.

## La calidad del contexto

Un principio que vale más que cualquier técnica específica: el contexto de calidad produce resultados de calidad.

El contexto de calidad tiene tres características. Es relevante: incluye lo que el modelo necesita para hacer la tarea, no más. Es completo: no omite restricciones, convenciones, comportamientos esperados que el modelo necesita conocer para dar una respuesta útil. Es preciso: describe correctamente lo que existe, no lo que crees que existe o lo que quisieras que existiera.

El contexto pobre tiene el efecto contrario. Si le das a Claude un stack trace sin el código relevante, tiene que adivinar la causa. Si le das el código sin describir el comportamiento esperado, tiene que asumir qué quieres que

haga. Si le das restricciones vagas ("hazlo escalable"), no puede cumplirlas porque no tienen significado operacional.

La inversión en construir buen contexto al inicio de una sesión siempre se recupera en la calidad de los resultados.

**Mini resumen:** Claude es un sistema de predicción de texto entrenado en datos masivos, no un motor de búsqueda ni un oráculo. Los tokens determinan costo y límites. El contexto de calidad produce resultados de calidad — irrelevante no es neutro, activamente daña. Sonnet es el modelo de trabajo por defecto; Haiku para volumen y simplicidad, Opus para razonamiento complejo.

---

## Capítulo 3: Configurar tu entorno de trabajo con Claude

Hay tres formas distintas de trabajar con Claude, y cada una tiene su lugar. Confundirlas —o usar siempre la misma para todo— es uno de los errores más comunes de quienes empiezan.

### Las tres vías de acceso

**Claude.ai (interfaz web)** es la manera más rápida de empezar. Abres el navegador, escribes, recibes respuesta. Es excelente para exploración, para hacer preguntas que no requieren contexto del proyecto, para experimentar con prompts, para conversaciones largas donde quieres ver el hilo completo. Su limitación es que no tiene acceso a tu codebase y cada sesión empieza desde cero.

Cuándo usar `claude.ai`: preguntas conceptuales ("¿cuál es la diferencia entre `Promise.all` y `Promise.allSettled?`"), exploración de enfoques antes de comprometerse con uno, aprendizaje de nuevas tecnologías, conversaciones de diseño de alto nivel.

**La API directamente** es para cuando construyes algo. Si quieres integrar Claude en una aplicación, automatización, script o agente, usas la API. Tienes control total sobre el modelo, el contexto, los parámetros y la forma en que procesas los resultados. El costo es por uso, sin suscripción.

Cuándo usar la API: construcción de herramientas internas, automatizaciones en CI/CD, scripts que procesan código o documentación, sistemas de agentes.

**Claude Code (CLI)** es para trabajo diario dentro de tu codebase. Es un cliente de línea de comandos que entiende la estructura de tu repositorio, puede leer y editar archivos, ejecutar comandos y mantener contexto entre operaciones. No es un chat — es un colaborador que vive en tu terminal.

Cuándo usar Claude Code: refactorizaciones que afectan múltiples archivos, debugging en contexto real del proyecto, escritura de tests para módulos existentes, revisión de código con acceso al repositorio completo.

## Configuración de la API

El proceso es directo:

```
# Instalar el SDK
npm add @anthropic-ai/sdk

# O con npm
npm install @anthropic-ai/sdk
```

```
// src/lib/claude.ts
import Anthropic from '@anthropic-ai/sdk';

// La API key viene de una variable de entorno, nunca hardcodeada
const client = new Anthropic({ apiKey: process.env.ANTHROPIC_API_KEY
});

export default client;
```

```

// Primera llamada completa
import client from './lib/claude';

async function primerLlamada(): Promise<void> {
  const response = await client.messages.create({
    model: 'claude-sonnet-4-5',
    max_tokens: 1024,
    messages: [
      { role: 'user', content: '¿Cuál es la diferencia entre
interface y type en TypeScript?' },
    ],
  });

  const textBlock = response.content.find(b => b.type === 'text');
  if (textBlock && textBlock.type === 'text') {
    console.log(textBlock.text);
  }
}

primerLlamada().catch(console.error);

```

Para las variables de entorno, el flujo estándar es:

```

# .env (en .gitignore, nunca en el repositorio)
ANTHROPIC_API_KEY=sk-ant-...

# Cargarlo con dotenv en desarrollo
pnpm add -D dotenv

```

```
// Al inicio del proceso, antes de cualquier import del SDK
import './dotenv/config';
```

La API key no va en el código. Nunca. Si la commiteas accidentalmente, rótaela de inmediato en [console.anthropic.com](https://console.anthropic.com).

## Instalación de Claude Code

```
# Instalación global
npm install -g @anthropic-ai/claude-code

# Autenticación
claude auth login

# Verificar que funciona
claude --version
```

Una vez instalado, navegas al directorio de tu proyecto y simplemente escribes `claude`. Se abre una sesión interactiva donde puedes hacer peticiones en lenguaje natural y Claude Code tiene acceso a los archivos de tu repositorio.

```
cd mi-proyecto
claude

# Dentro de la sesión:
# > Explicame la estructura de este proyecto
# > Hay un test fallando en src/users/users.test.ts, ayúdame a diagnosticarlo
# > Refactoriza el módulo de autenticación para usar zod en la validación de inputs
```

## El archivo CLAUDE.md

Este es probablemente el concepto más subestimado para quienes empiezan con Claude Code. Un archivo `CLAUDE.md` en la raíz del proyecto le da a Claude contexto persistente sobre tu proyecto — contexto que no tienes que repetir en cada sesión.

Sin `CLAUDE.md`, cada sesión empieza desde cero y tienes que explicar tu stack, tus convenciones y tus restricciones. Con `CLAUDE.md`, Claude llega a cada sesión ya sabiendo cómo trabaja tu proyecto.

```
# CLAUDE.md - Mi Proyecto API

## Stack técnico
- Node.js 20 + TypeScript 5.x en modo strict
- Express para HTTP
- Prisma con PostgreSQL
- Vitest para tests
- Zod para validación de inputs

## Comandos principales
```bash
pnpm dev          # servidor en localhost:3000 con hot reload
pnpm test         # vitest run (sin watch)
pnpm test:watch  # vitest en modo watch
pnpm lint         # eslint . --ext .ts
pnpm build        # tsc --noEmit && esbuild src/index.ts
```

## Convenciones importantes

- TypeScript strict: sin any, sin as en lugares que no sean necesarios
- Validación con zod en TODOS los endpoints antes de procesar
- Tests en archivos \*.test.ts junto al archivo que prueban
- Errores HTTP con formato: { error: string, code: string }
- Nunca modificar el schema de Prisma sin discutir primero

## Estructura del proyecto

```
src/  
  routes/      - handlers HTTP (Express Router)  
  services/    - lógica de negocio pura  
  lib/         - utilidades, clientes externos  
  types/       - tipos compartidos  
  middleware/  - Express middleware  
prisma/  
  schema.prisma
```

## Qué NO hacer

- No usar callbacks donde hay equivalente `async/await`
- No dejar `console.log` en código que no sea de debugging temporal
- No crear archivos en la raíz del proyecto sin preguntar

Este archivo vive en el repositorio y se comparte con el equipo. Cuando alguien nuevo llega al proyecto y usa Claude Code, ya tiene el contexto del proyecto sin tener que configurar nada.

## El ritual de inicio de sesión

Una sesión productiva con Claude tiene una estructura. No es obligatoria, pero produce resultados más consistentes:

**Primero, contexto del proyecto** — si no tienes `CLAUDE.md` o necesitas agregar contexto específico para la sesión:

```
Contexto: API REST en TypeScript con Express y Prisma. Hoy voy a trabajar en el módulo de autenticación. El sistema usa JWT con refresh tokens, implementado en src/auth/.
```

**Segundo, el objetivo concreto de la sesión:**

```
Objetivo: el endpoint POST /auth/refresh está fallando cuando el refresh token está expirado. Devuelve 500 en lugar de 401. Necesito diagnosticar y corregir.
```

**Tercero, las restricciones que aplican:**

```
Restricciones: no cambiar el formato de respuesta de los errores. No
modificar la firma de los endpoints existentes. Mantener TypeScript
strict.
```

Con ese inicio, la primera respuesta de Claude ya va en la dirección correcta.

**Mini resumen:** Hay tres formas de usar Claude con propósitos distintos. La API es para construir, Claude Code es para trabajar dentro del repositorio, `claude.ai` es para exploración. El archivo `CLAUDE.md` es la inversión de tiempo más rentable para trabajo diario con Claude Code — configúralo desde el primer día.

---

## Capítulo 4: El arte del prompting para programadores

El prompting no es un arte místico. No hay palabras mágicas ni trucos ocultos. Es comunicación con restricciones específicas: un emisor (tú) que tiene contexto que el receptor (Claude) no tiene, y que necesita transferir ese contexto de manera eficiente para obtener un resultado útil.

Si lo piensas así, los principios de un buen prompt son los mismos que los de una buena especificación técnica. Claridad, completitud, restricciones explícitas, criterio de éxito definido.

### La estructura de cuatro partes

Casi todo prompt efectivo para programación tiene cuatro componentes:

- 1. Contexto:** qué existe, qué sistema, qué stack, qué es lo relevante del estado actual. El contexto le dice a Claude en qué mundo está operando.
- 2. Objetivo:** qué quieres que produzca. Concreto, medible, sin ambigüedad. No "hazlo mejor" sino "refactoriza esta función para eliminar la duplicación de lógica de validación, manteniendo el comportamiento externo idéntico".
- 3. Restricciones:** qué no puede hacer, qué no puede cambiar, qué convenciones debe seguir, qué herramientas puede usar. Las restricciones son tan importantes como el objetivo porque eliminan el espacio de soluciones incorrectas.
- 4. Formato de salida:** cómo quieres que responda. ¿Solo el código? ¿Código con explicación? ¿Una tabla de opciones? ¿Solo los cambios, no el archivo completo? Cuando no especificas, Claude elige, y no siempre elige lo más útil para ti.

Veamos cómo se construye esto incrementalmente:

```
# Solo el objetivo (dóbil)
Escribe una función para paginar resultados.

# Objetivo + contexto
Estoy construyendo una API REST con Express y TypeScript.
Escribe una función para paginar resultados de consultas a la base de
datos.

# Objetivo + contexto + restricciones
Estoy construyendo una API REST con Express y TypeScript strict.
Escribe una función 'paginate<T>' que tome un array de items y
parámetros de paginación,
y devuelva { data: T[], total: number, page: number, pageSize:
number, hasNext: boolean }.
Sin any. Maneja casos de page <= 0 o pageSize <= 0 lanzando un error
con mensaje descriptivo.

# Completo: objetivo + contexto + restricciones + formato de salida
Estoy construyendo una API REST con Express y TypeScript strict.
Escribe una función 'paginate<T>' que tome un array de items y
parámetros de paginación,
y devuelva { data: T[], total: number, page: number, pageSize:
number, hasNext: boolean }.
Sin any. Maneja page <= 0 o pageSize <= 0 lanzando un error
descriptivo.
Dato: 1) la función con tipos completos, 2) un ejemplo de uso, 3) dos
casos de prueba con Vitest.
```

Cada adición elimina ambigüedad y hace más probable que el resultado sea directamente útil.

## System prompts vs mensajes de usuario

Cuando usas la API directamente, tienes la opción de usar un `system prompt`. La distinción importa:

El `system prompt` establece el contexto global y las instrucciones que aplican a toda la conversación. Aquí van: el rol de Claude en este sistema, las convenciones del proyecto, las restricciones permanentes, el formato de respuesta esperado.

Los `mensajes de usuario` contienen las tareas específicas dentro de ese contexto.

```
const response = await client.messages.create({
  model: 'claude-sonnet-4-0',
  max_tokens: 2048,
  system: 'Eres un asistente técnico para un proyecto TypeScript con Express y Prisma.

  Convenciones del proyecto:
  - TypeScript strict, sin any
  - Validación de inputs con zod
  - Tests con Vitest
  - Errores HTTP: { error: string, code: string }

  Responde siempre con código comentado y explicaciones concisas.',
  messages: [
    {
      role: 'user',
      content: 'Escribe un middleware de autenticación JWT que valide el header Authorization.',
    },
  ],
});
```

Si siempre vas a trabajar en el mismo proyecto, poner el contexto del proyecto en el system prompt es más eficiente que repetirlo en cada mensaje.

## Cuatro pares de prompts débiles vs fuertes

### Par 1: Solicitud de función

❌ Débil:

```
Escribe una función para validar emails.
```

✅ Fuerte:

```
Escribe una función TypeScript `validateEmail(email: unknown): email  
is string` que:  
- Retorne true solo si el argumento es un string con formato de email  
válido (RFC 5322 básicos)  
- Use una regex razonable, no una librería externa  
- Maneje correctamente: strings vacíos, null, undefined, números,  
objetos  
Incluye 5 casos de prueba que cubran casos edge.
```

### Par 2: Reporte de bug

❌ Débil:

Mi función no funciona, ¿qué está mal?

```
function processOrder(order) {  
  return order.items.reduce((sum, item) => sum + item.price, 0);  
}
```

✓ Fuerte:

Esta función falla con `TypeError` cuando `order.items` está vacío o cuando un ítem no tiene `price`.

```
function processOrder(order: { items: Array<{ name: string; price?: number }> }): number {  
  return order.items.reduce((sum, item) => sum + item.price, 0);  
}
```

Error: `TypeError: Cannot read properties of undefined (reading 'price')`

Comportamiento esperado: retornar 0 si items está vacío, ignorar items sin price.

Restricción: mantener la misma firma, no usar `any`.

Identifica la causa raíz y propón el fix mínimo.

### Par 3: Solicitud de tests

✗ Débil:

Escribe tests para esta función.

✔ Fuerte:

```
Escribe tests con Vitest para esta función 'groupUsersByRole'.  
Quiero cubrir: caso normal con múltiples roles, arreglo vacío, un  
solo usuario por rol,  
usuario con rol no estándar (TypeScript no lo permitiría pero podrías  
recibirlo en runtime).  
Formato: describe/it, sin mocks (la función es pura), nombres de test  
descriptivos en español.
```

#### Par 4: Explicación de código

✘ Débil:

```
Explica este código.
```

✔ Fuerte:

```
Explica esta función a alguien que conoce TypeScript bien pero acaba  
de entrar al proyecto.  
Enfócate en: qué problema resuelve, por qué usa este enfoque en lugar  
de uno más simple,  
cuáles son los efectos secundarios que tengo que considerar si quiero  
modificarla,  
y qué invariantes asumo que son verdad en tiempo de ejecución.
```

### Few-shot: ejemplos en el prompt

Cuando quieres que Claude siga un patrón específico que es difícil de describir con palabras, la manera más eficiente es mostrar ejemplos:

Convierte cada uno de estos mensajes de error técnico a un mensaje amigable para el usuario.

Sigue exactamente este formato: [Error técnico] + [Mensaje para usuario]

Ejemplos:

`CONNECTION_ERROR` + No pudimos conectar con el servidor. Intenta de nuevo en un momento.

`ValidationError: email is required` + Por favor ingresa tu correo electrónico.

`JWT expired` + Tu sesión ha expirado. Ingresá de nuevo para continuar.

Ahora convierte estos:

- `ETIMEDOUT`
- `Error: User not found`
- `Cannot read property 'id' of null`

Los ejemplos eliminan la ambigüedad sobre formato y tono mejor que cualquier descripción verbal.

## Pedir razonamiento

Para tareas complejas, pedir que Claude explique su razonamiento antes de dar el resultado produce mejores resultados. No es magia — es que el modelo también genera mejor código cuando “piensa en voz alta” sobre el problema antes de escribir la solución.

Antes de escribir el código, explica brevemente:

- 1) cuáles son las partes complejas de este problema
- 2) qué enfoque vas a usar y por qué
- 3) qué edge cases necesitan manejo especial

Luego escribe la implementación.

## La iteración es el proceso

Un error común: creer que un buen prompt debería producir el resultado perfecto en el primer intento. No es así. La iteración es parte del flujo normal, no una señal de que algo salió mal.

El primer prompt establece la dirección. El segundo refina. El tercero ajusta detalles. Ese proceso de tres pasos es más rápido y produce mejores resultados que intentar escribir el prompt perfecto de entrada.

# Iteración 1: dirección

Escribe un hook de React para manejar el estado de un formulario de login.

# Iteración 2: refinamiento

Agrega validación con zod. El esquema es: { email: email string, password: min 8 caracteres }.

Maneja los errores de validación por campo.

# Iteración 3: ajuste

Cambia el estado de error para que sea string | null en lugar de string | undefined.

Y agrega un estado 'isSubmitting' que sea true mientras se procesa el formulario.

Cada paso es corto y claro. No reescribes desde cero — refinas desde donde estás.

**Mini resumen:** Un prompt efectivo tiene cuatro partes: contexto, objetivo, restricciones y formato de salida. Las restricciones son tan importantes como el objetivo. La iteración de tres pasos es más eficiente que buscar el prompt perfecto. El razonamiento explícito mejora la calidad del código en tareas complejas.

---

## Capítulo 5: Técnicas avanzadas de prompting

Una vez que tienes la estructura básica de un buen prompt, hay un conjunto de técnicas que producen mejoras significativas en situaciones específicas. No todas aplican en todo momento — parte del trabajo es saber cuándo usar cada una.

### Chain of thought: pensar antes de escribir

Chain of thought (cadena de pensamiento) es pedirle a Claude que razona sobre el problema paso a paso antes de llegar a la respuesta. Para tareas simples no hace diferencia. Para tareas complejas — algoritmos no triviales, debugging de sistemas con múltiples piezas, decisiones de arquitectura — la diferencia puede ser considerable.

La intuición es similar a lo que pasa con un humano: un programador que se detiene a pensar el problema antes de escribir código casi siempre produce una solución mejor que uno que empieza a teclear de inmediato.

✘ Sin chain of thought:

```
Diseña una función que procese pedidos concurrentes sin condiciones de carrera.
```

✔ Con chain of thought:

```
Voy a pedirte que diseñes una función que procese pedidos concurrentes.
```

```
Antes de escribir el código:
```

1. Identifica las condiciones de carrera posibles en un escenario de múltiples workers
2. Evalúa tres enfoques de sincronización y sus trade-offs para este caso
3. Elige el más apropiado y explica por qué

```
Después escribe la implementación en TypeScript.
```

El resultado con chain of thought tiende a ser más fundamentado y con menos errores sutiles porque el modelo ha "considerado" los problemas antes de generar la solución.

## Role prompting: contexto de perspectiva

Definir un rol específico para Claude no es un truco de magia verbal.

Funciona porque el rol actúa como contexto que guía el tipo de respuesta.

Un "senior TypeScript engineer haciendo code review" va a enfocarse en cosas diferentes a un "desarrollador TypeScript" genérico.

```
Actúa como senior engineer de seguridad revisando un PR de autenticación.
```

```
Tu foco: vulnerabilidades, manejo de secretos, validación de inputs, surface de ataque, problemas de autorización.
```

```
No te preocupes por style o legibilidad.
```

```
[código del PR]
```

Los roles más útiles para trabajo de programación: - Senior engineer de [tecnología específica] para code review enfocado - Arquitecto de sistemas para decisiones de diseño - Revisor de seguridad para análisis de vulnerabilidades - Autor de la función para defender decisiones de diseño

El rol debe ser específico y tener sentido para la tarea. "Actúa como experto" sin más detalle no aporta nada útil.

## Salida estructurada: forzar formato JSON

Cuando Claude es parte de un sistema — no solo una herramienta de conversación — necesitas que su salida sea parseable de manera confiable. La manera más directa es pedirle JSON explícitamente.

```

import Anthropic from '@anthropic-ai/sdk';

const client = new Anthropic({ apiKey: process.env.ANTHROPIC_API_KEY
});

interface CodeAnalysis {
  bugs: string[];
  improvements: string[];
  riskLevel: 'low' | 'medium' | 'high';
  testCoverage: { missing: string[]; recommended: string[] };
}

async function analyseCode(code: string): Promise<CodeAnalysis> {
  const response = await client.messages.create({
    model: 'claude-sonnet-4-5',
    max_tokens: 1024,
    messages: [{
      role: 'user',
      content: 'Analiza este código TypeScript y responde ÚNICAMENTE
con JSON válido, sin texto adicional.
El JSON debe seguir exactamente esta estructura:
{
  "bugs": ["descripción de bug 1", "..."],
  "improvements": ["mejora sugerida 1", "..."],
  "riskLevel": "low" | "medium" | "high",
  "testCoverage": {
    "missing": ["caso no cubierto 1", "..."],
    "recommended": ["test recomendado 1", "..."]
  }
}
}

```

Código a analizar:

Para sistemas en producción, agrega validación con zod después del parse:

```
import { z } from "zod";

const CodeAnalysisSchema = z.object({
  bugs: z.array(z.string()),
  improvements: z.array(z.string()),
  riskLevel: z.enum(['low', 'medium', 'high']),
  testCoverage: z.object({
    missing: z.array(z.string()),
    recommended: z.array(z.string()),
  }),
});

// Después del JSON.parse:
const validated = CodeAnalysisSchema.parse(JSON.parse(cleaned));
```

## Conversaciones multi-turno: el arreglo de mensajes

Las conversaciones con contexto acumulado se construyen añadiendo mensajes al arreglo `messages`. El modelo ve toda la historia y responde en ese contexto.

```
const messages: Anthropic.MessageParam[] = [];

async function chat(userMessage: string): Promise<string> {
  messages.push({ role: 'user', content: userMessage });

  const response = await client.messages.create({
    model: 'claude-sonnet-4-5',
    max_tokens: 2048,
    messages,
  });

  const assistantMessage = response.content.find(b => b.type ===
'text');
  if (!assistantMessage || assistantMessage.type !== 'text') {
    throw new Error('Respuesta inesperada');
  }

  // Agregar la respuesta al historial para la siguiente vuelta
  messages.push({ role: 'assistant', content: assistantMessage.text
});

  return assistantMessage.text;
}

// Uso:
await chat('Tengo una función que agrupa usuarios por rol. ¿Cómo la
haría eficiente?');
await chat('Ahora agrega manejo de usuarios con roles múltiples.');
```

```
await chat('¿Cuál sería el tipo de retorno correcto en ese caso?');
```

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudiobook.com](http://masterclaudiobook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudioebook.com](http://masterclaudioebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudiobook.com](http://masterclaudiobook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudiobook.com](http://masterclaudiobook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudiobook.com](http://masterclaudiobook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudefbook.com](http://masterclaudefbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudefbook.com](http://masterclaudefbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudefbook.com](http://masterclaudefbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudiobook.com](http://masterclaudiobook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudiobook.com](http://masterclaudiobook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudiobook.com](http://masterclaudiobook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudefbook.com](http://masterclaudefbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudefbook.com](http://masterclaudefbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudefbook.com](http://masterclaudefbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudefbook.com](http://masterclaudefbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudefbook.com](http://masterclaudefbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudefbook.com](http://masterclaudefbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudiobook.com](http://masterclaudiobook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

# ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudebook.com](http://masterclaudebook.com)

## ? Contenido bloqueado

Obtén el libro completo en  
[masterclaudbook.com](http://masterclaudbook.com)