

Claude Code `</>`

DE LO BÁSICO AL DOMINIO

La guía completa para el desarrollo
de software agéntico **en 2026**



Instala y
Comienza



Prompt
Engineering



Workflows
Multiagente



Seguridad y
Mejores Prácticas



Proyectos
del mundo real

stripe

Casos de estudio
de Stripe
y Wiz

WIZ*



Automatización
CI/CD



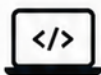
El futuro de la
programación
agéntica

Una guía completa y práctica del sistema de codificación agéntica
de Claude Code de Anthropic que está **transformando la forma**
en que se construye el software.

Desde la primera instalación hasta la implementación en empresas—domina
las herramientas, técnicas y flujos de trabajo en los que confían los mejores
equipos de ingeniería.



Ejemplos prácticos
en cada capítulo



Proyectos prácticos
que puedes construir



Lecciones de equipos
del mundo real



Basado en investigación
con más de 100K+ sesiones

S T E V E T.

Claude Code: De lo básico al dominio

La Guía Completa del Desarrollo de Software Agéntico en 2026

Steve T. Publications

Este libro está a la venta en <https://leanpub.com/claudecodees>

Esta versión se publicó en 2026-07-16



Éste es un libro de [Leanpub](#). Leanpub anima a los autores y publicadoras con el proceso de publicación. [Lean Publishing](#) es el acto de publicar un libro en progreso usando herramientas sencillas y muchas iteraciones para obtener retroalimentación del lector hasta conseguir el libro adecuado.

© 2026 Steve T. Publications

Índice general

La Guía Completa del Desarrollo de Software Agéntico en 2026	1
Un Manual Exhaustivo para Dominar Claude Code, Flujos de Trabajo Multiagente y Despliegue Empresarial	1
Introducción: El Ingeniero que Ya No Escribe	3
Lo Que Este Libro Te Enseñará	3
A Quién Va Dirigido Este Libro	4
Cómo Usar Este Libro	4
La Base de Evidencia	5
Una Nota Sobre Lo Que Este Libro No Cubre	5
La Tesis Central	5
El Panorama en 2026	6
Capítulo 1: La Revolución Agéntica en el Desarrollo de Software	7
Del Autocompletado a los Agentes: Una Evolución de Cinco Años	7
Qué Hace Diferente a Claude Code: Arquitectura Agéntica Terminal-First	8
El Flujo de Trabajo de Boris Cherny: Cómo el Creador Usa Realmente Claude Code	10
La División del Trabajo: Humano como Orquestador, Agente como Constructor	12
El Panorama de Agentes de Programación con IA en 2026	13
¿Por Qué Ahora? Adopción del Mercado y el Cambio hacia Flujos de Trabajo Agénticos	14
Análisis Crítico: Compromisos y Limitaciones de las Herramientas Agénticas	15
Capítulo 2: Anatomía de Claude Code	19
El Bucle Agéntico: Recopilar, Actuar, Verificar	19
Modelos Detrás de Escena: Sonnet, Opus, Fable y Selección de Modelo	19
Matemáticas de Gestión de Tokens: Entendiendo Tu Gasto	20
Herramientas Integradas: Operaciones de Archivos, Búsqueda, Ejecución, Web, Inteligencia de Código	20
El Directorio .claude: Configuración, CLAUDE.md y Memoria	20

Ventanas de Contexto: Cómo se Llenan, Auto-Compresión y Gestión de Tokens	20
Capítulo 3: Comenzando y Configurando	22
Instalación en Múltiples Plataformas: macOS, Linux, Windows, WSL, Homebrew, WinGet	22
Autenticación y Configuración de Cuenta	22
Tu Primera Sesión: Recorrido desde Cero hasta Código	22
Modos de Permiso: Default, Auto-accept, Plan y Auto Mode	23
Integraciones IDE: VS Code, JetBrains, Remote Control, Interfaz Web	23
Capítulo 4: Ingeniería de Prompts para Claude Code	24
El Arte del Prompt Claude: Especificidad, Contexto y Verificación	24
CLAUDE.md como Capa Persistente de Instrucciones	24
Ejemplos Few-Shot, Estructuración XML y Asignación de Roles	24
El Patrón “Entrevistame”: Dejar que Claude Haga Preguntas Antes de Programar	24
Encadenamiento de Prompts y Gestión de Estado Entre Sesiones	25
Patrones Comunes de Fallo y Cómo Evitarlos	25
Prueba Adversarial: Poniendo a Prueba la Salida de Claude Code	25
Capítulo 5: Flujos de Trabajo de Desarrollo del Mundo Real	26
Construyendo una Aplicación Full-Stack desde Cero (Proyecto Práctico)	26
Refactorizando Codebases Legados: Estudio de Caso de la Migración Scala-a-Java de Stripe	27
Depurando Bugs Complejos: Un Recorrido Paso a Paso	27
Migración de Código a Escala: La Migración Python-a-Go de Wiz	28
Incorporación a un Código Fuente Desconocido	28
Capítulo 6: Pruebas, Verificación y Aseguramiento de Calidad	29
Dándole a Claude Algo Contra Qué Verificar	29
Generación de Pruebas y Pruebas Auto-Curativas	29
Outcomes: Calificación Automatizada por Rúbricas	29
El Patrón Escritor/Revisor con Sesiones Paralelas	29
Habilidades de Revisión de Código y Estrategias de Revisión Adversarial	29
Integración CI/CD: GitHub Actions, GitLab CI y PRs Automatizados	30
Capítulo 7: Automatización y Escalado	31
Modo Headless: Claude Code No Interactivo para Scripts y Pipelines	31
Patrones de Abanico (Fan-out): Procesando Miles de Archivos en Paralelo	31
Rutinas Claude Code: Tareas Programadas y Flujos de Trabajo Impulsados por Eventos	32

Hooks Pre-commit, Linting y Puertas Automatizadas de Calidad de Código	32
Gestión de Costos y Presupuesto de Tokens a Escala	32
Capítulo 8: Patrones Agénticos Avanzados	33
Sub-Agentes: Delegando Investigación y Verificación	33
Habilidades: Creando Conocimiento de Dominio Reutilizable y Flujos de Trabajo	33
Servidores MCP: Conectando a Sistemas Externos (GitHub, Bases de Datos, Jira) .	33
Hooks: Aplicación Automatizada de Políticas Entre Sesiones	33
Orquestación Multiagente: Agentes Principales, Sub-Agentes Especialistas, Sistemas de Archivos Compartidos	34
Agentes Gestionados Claude y la Función “Dreaming”	34
Capítulo 9: Despliegue y Listado para Producción	36
Desplegando Aplicaciones Construidas con Claude Code a Plataformas Cloud . .	36
Configuración de Entorno y Gestión de Secretos	36
Monitoreo y Observabilidad: OpenTelemetry, Analíticas de Uso, Rastreo de Tokens	36
Optimización de Rendimiento: Caché de Prompts, Selección de Modelo, Eficiencia de Contexto	36
De Prototipo a Producción: Un Estudio de Caso con Rakuten	36
Capítulo 10: Colaboración de Equipo y Despliegue Empresarial	38
Claude Code a Escala: El Despliegue de 1.370 Ingenieros de Stripe	38
Configuraciones Gestionadas, Seguridad Empresarial y Cumplimiento	38
CLAUDE.md Compartido, Librerías de Habilidades y Distribución de Plugins . .	39
Integrando Claude Code en Flujos de Trabajo de Equipo Existentes	39
Capítulo 11: Análisis Profundo de Seguridad	40
Vulnerabilidades Documentadas: CVE-2025-59536 y CVE-2026-21852	40
Inyección de Prompt y Contenido Malicioso en Archivos	40
Exposición de Secretos en Carga de Contexto	41
La Bandera —dangerously-skip-permissions: Cuándo y Cómo Usarla Seguramente	41
Sandboxing: Docker, Dev Containers y Aislamiento VM	41
Arquitectura de Seguridad Empresarial: MDM, Gateways LLM, Monitoreo de Agentes	41
Capítulo 12: El Futuro de la Programación Agéntica	42
Hoja de Ruta de Anthropic: Dreaming, Outcomes, Orquestación Multiagente . . .	42
La Investigación sobre Experiencia: Por Qué el Conocimiento del Dominio Gana al Antecedente de Programación	42
Ocho Tendencias Moldeando la Programación Agéntica en 2026 y Más Allá	42

El Rol Cambiante de los Ingenieros de Software	42
Qué Significa Esto para el Futuro del Trabajo y la Economía del Conocimiento . .	43
Capítulo 13: Conclusión: Convertirse en un Ingeniero Agéntico	44
Sintetizando el Viaje de Principiante a Maestro	44
Los Nuevos Modelos Mentales del Desarrollo de Software	44
Construyendo Tu Práctica de Claude Code: Un Plan de Acción Personal	44
Mirando Hacia Adelante: Hacia Dónde Nos Llevan los Próximos Cinco Años . . .	44
Referencias	45

La Guía Completa del Desarrollo de Software Agéntico en 2026

**Un Manual Exhaustivo para Dominar Claude Code,
Flujos de Trabajo Multiagente y Despliegue
Empresarial**

Acerca de este libro: Esta es una guía exhaustiva de Claude Code, el sistema de programación agéntico de Anthropic que ha cambiado fundamentalmente la forma en que se construye el software. Ya sea que seas un desarrollador nuevo en la programación asistida por IA o un ingeniero experimentado que busca dominar flujos de trabajo multiagente, este libro te lleva desde la primera instalación hasta el despliegue empresarial. Cubre la arquitectura de Claude Code, técnicas de ingeniería de prompts, estudios de caso reales de Stripe y Wiz, automatización de CI/CD, mejores prácticas de seguridad y el futuro de la programación agéntica. Cada capítulo incluye ejemplos prácticos, proyectos hands-on y orientación accionable fundamentada en investigación sobre cientos de miles de sesiones reales de Claude Code.

Introducción: El Ingeniero que Ya No Escribe

A finales de 2025, Boris Cherny, el creador y jefe de Claude Code en Anthropic, hizo una observación que habría sonado como ciencia ficción hace apenas tres años. No había escrito manualmente ni una sola línea de código desde el noviembre anterior. En su lugar, describía lo que quería, Claude Code descubría cómo construirlo, ejecutaba las pruebas, corregía los fallos y enviaba pull requests. Estaba ejecutando cinco sesiones locales de terminal y de cinco a diez instancias remotas de forma concurrente, enviando de diez a treinta pull requests por día [1].

El creador de la herramienta está describiendo cómo la empresa que la construyó opera realmente cada día, y esto es solo un punto de datos en una historia mucho más grande.

A mayo de 2026, más del 80 por ciento del código fusionado en el propio código fuente de producción de Anthropic fue escrito por Claude, aumentando desde unos pocos dígitos individuales antes de que Claude Code se lanzara en vista previa de investigación en febrero de 2025 [7]. Los ingenieros en Anthropic no escriben implementaciones desde cero. Describen lo que quieren, Claude Code descubre cómo construirlo, ejecuta las pruebas, corrige los fallos y envía los pull requests. Los ingenieros humanos revisan la salida, toman decisiones arquitectónicas y pasan al siguiente problema [8].

Lo Que Este Libro Te Enseñará

Este libro es una guía exhaustiva de Claude Code, el sistema de programación agéntico de Anthropic que ha cambiado fundamentalmente la forma en que se construye el software. Ya sea que seas un desarrollador nuevo en la programación asistida por IA o un ingeniero experimentado que busca dominar flujos de trabajo multiagente, este libro te lleva desde la primera instalación hasta el despliegue empresarial. Cubre la arquitectura de Claude Code, técnicas de ingeniería de prompts, estudios de caso reales de Stripe, Wiz y Rakuten, automatización de CI/CD, mejores prácticas de seguridad y el futuro de la programación agéntica. Cada capítulo incluye ejemplos prácticos, proyectos hands-on y orientación accionable fundamentada en investigación sobre cientos de miles de sesiones reales de Claude Code.

Al finalizar este libro, serás capaz de:

- Instalar y configurar Claude Code en múltiples plataformas e IDEs
- Escribir prompts que produzcan de forma fiable código de calidad de producción en el primer intento
- Construir aplicaciones full-stack desde cero usando el patrón de entrevista y el desarrollo guiado por especificaciones
- Configurar bucles de prueba y verificación para que Claude se autocorrija sin tu intervención
- Automatizar tareas repetitivas usando el modo headless, patrones de abanico (fan-out) y rutinas programadas
- Diseñar arquitecturas multiagente con agentes principales, sub-agentes especializados y sistemas de archivos compartidos
- Desplegar Claude Code a escala empresarial con configuraciones gestionadas, SSO y políticas de gobernanza
- Comprender y mitigar los riesgos de seguridad inherentes en las herramientas de programación agéntica, incluyendo inyección de prompts, exposición de secretos y ataques a la cadena de suministro

A Quién Va Dirigido Este Libro

Este libro asume que tienes una familiaridad básica con el desarrollo de software: sabes qué es una terminal, has escrito código en al menos un lenguaje y entiendes conceptos como control de versiones, pruebas y despliegue. No necesitas ser experto en ningún lenguaje de programación en particular. Claude Code funciona a través de lenguajes, y los modelos mentales que este libro enseña son transferibles independientemente de tu stack tecnológico.

Si eres un desarrollador junior que ha usado herramientas de autocompletado y las ha encontrado útiles pero limitadas, este libro te mostrará qué sucede cuando vas más allá del autocompletado. Si eres un ingeniero senior o líder de equipo evaluando Claude Code para tu organización, este libro te dará la profundidad técnica y el contexto estratégico necesarios para tomar decisiones informadas. Si eres un gerente de producto, diseñador, fundador o profesional no técnico que quiere construir software sin escribir código línea por línea, este libro te mostrará cómo Claude Code abre esa puerta.

Cómo Usar Este Libro

El libro está organizado en tres partes, cada una construyendo sobre la anterior:

Parte I (Capítulos 1-3): Fundamentos. Los capítulos 1 y 2 establecen el marco conceptual. Entenderás dónde encaja Claude Code en la evolución de las herramientas de programación con IA, qué hace que su arquitectura agéntica sea diferente del autocompletado y los asistentes inline, y cómo funciona su bucle interno bajo el capó. El capítulo 3 te guía a través de la instalación y tu primera sesión.

Parte II (Capítulos 4-8): Habilidades Centrales. Los capítulos 4 al 6 te enseñan las habilidades prácticas de trabajar con Claude Code: ingeniería de prompts, flujos de trabajo de desarrollo del mundo real y estrategias de prueba. Los capítulos 7 y 8 cubren automatización, escalado y patrones agénticos avanzados. Aquí es donde desarrollas la memoria muscular que separa las sesiones productivas de las frustrantes.

Parte III (Capítulos 9-13): Producción y Más Allá. Los capítulos 9 y 10 abordan el despliegue, la observabilidad y la colaboración empresarial. El capítulo 11 es un análisis profundo de seguridad. El capítulo 12 mira el futuro de la programación agéntica. El capítulo 13 sintetiza todo en orientación accionable para convertirte en un ingeniero agéntico efectivo.

Puedes leer este libro de portada a portada, o saltar a los capítulos más relevantes para tus necesidades actuales. Cada capítulo está diseñado para ser un análisis profundo autónomo, al tiempo que construye sobre conceptos introducidos anteriormente.

La Base de Evidencia

La base de evidencia para este libro incluye el análisis exhaustivo de la documentación oficial de Anthropic y las publicaciones de su blog de ingeniería, la investigación de la compañía sobre aproximadamente 400.000 sesiones de Claude Code de unos 235.000 usuarios entre octubre de 2025 y abril de 2026 [6], estudios de caso de clientes publicados por Anthropic en claude.com/customers/, análisis técnicos de terceros de Check Point Research y otras firmas de seguridad, y flujos de trabajo contribuidos por la comunidad. Donde las afirmaciones no pueden verificarse de forma independiente, se atribuyen a su fuente. Donde la evidencia es inconclusa o controvertida, el texto lo dice.

Una Nota Sobre Lo Que Este Libro No Cubre

Este libro se centra en Claude Code como herramienta de desarrollo para trabajar con tus propios codebases. No cubre la API de Claude ni la construcción de aplicaciones que usen la API de Claude, aunque los conceptos de este libro te darán una base sólida para entender cómo funcionan los modelos subyacentes de Claude. Tampoco cubre Claude Cwork (el espacio de trabajo colaborativo), Claude Design o Claude Security, que son productos separados en el portafolio de Anthropic.

La Tesis Central

La tesis central de este libro es directa: Claude Code representa un cambio de paradigma del autocompletado de código a la programación agéntica. El humano se convierte en un orquestador que define objetivos y revisa resultados, mientras Claude se encarga de la ejecución. Pero este cambio de paradigma viene con riesgos reales: vulnerabilidades de seguridad en archivos de configuración, desafíos de gestión de contexto, incidentes de degradación de calidad y la necesidad de gobernanza de grado empresarial. Dominar esta herramienta requiere no solo aprender comandos, sino desarrollar nuevos modelos mentales para la delegación, la verificación y la colaboración con un agente autónomo.

El Panorama en 2026

Claude Code no es la única herramienta de programación agéntica disponible. El panorama competitivo incluye GitHub Copilot, Cursor, OpenAI Codex, Google Antigravity y otros, cada uno con estrategias y fortalezas distintas. Entender sus diferencias es esencial para elegir la herramienta correcta para tu flujo de trabajo. Pero Claude Code ocupa una posición única: es terminal-first pero funciona en múltiples interfaces, opera a nivel de proyecto en lugar de nivel de archivo, y ha demostrado la adopción más agresiva entre equipos empresariales.

La tecnología avanza rápido. Claude Code ya ha pasado por cambios significativos desde su lanzamiento, con lanzamientos importantes de funciones a mediados de 2026 que añadieron Dreaming (curación de memoria entre sesiones), Outcomes (calificación automatizada de calidad) y orquestación multiagente en la conferencia “Code with Claude” del 6 de mayo de 2026 [3]. El lineup de modelos incluye Claude Fable 5, que obtiene un 95,0 por ciento en SWE-bench Verified, Claude Opus 4.8 con un 88,6 por ciento y Claude Sonnet 4.6 como el valor predeterminado de producción [2]. Los principios fundamentales de uso efectivo, los modelos mentales de delegación y verificación, y la comprensión arquitectónica de cómo opera Claude Code permanecen constantes independientemente de las actualizaciones del modelo.

Los capítulos que siguen proporcionan una guía completa para dominar Claude Code.

Capítulo 1: La Revolución Agéntica en el Desarrollo de Software

Del Autocompletado a los Agentes: Una Evolución de Cinco Años

Para entender dónde encaja Claude Code en el panorama, ayuda trazar la evolución de la programación asistida por IA durante aproximadamente cinco años. La progresión ha pasado por cuatro fases distintas, cada una construyendo sobre la anterior y expandiendo lo posible.

La primera fase, abarcando aproximadamente de 2019 a 2021, fue el autocompletado de código basado en reglas. Herramientas como TabNine usaban modelos estadísticos entrenados en código de código abierto para sugerir la siguiente línea o función mientras un desarrollador escribía. Estas herramientas eran genuinamente útiles para código boilerplate y patrones comunes, pero no podían razonar sobre la intención, la arquitectura o los cambios multi-archivo. Completaban líneas; no entendían proyectos.

La segunda fase, desde 2021 hasta aproximadamente mediados de 2024, fue el autocompletado con modelos de lenguaje grandes (LLM). GitHub Copilot, lanzado en 2021 con el modelo Codex de OpenAI, representó un salto cuántico. En lugar de predicción estadística, usaba un modelo de lenguaje neuronal para generar sugerencias de código. Los desarrolladores podían resaltar una función y pedir un conjunto de pruebas, o describir lo que querían en comentarios y recibir implementaciones funcionales. La experiencia fue transformadora para muchos desarrolladores, pero permaneció fundamentalmente reactiva. El desarrollador escribía, el modelo respondía, y el desarrollador aceptaba o rechazaba cada sugerencia.

La tercera fase, emergiendo a finales de 2024 y acelerándose a lo largo de 2025, introdujo asistentes inline e IDEs nativos de IA. Herramientas como Cursor (un fork de VS Code) y las propias integraciones de Claude fueron más allá de sugerencias de un solo archivo. Podían leer todo el contexto del proyecto, entender cómo se relacionaban los archivos entre sí, y hacer cambios coordinados a través de múltiples archivos. El desarrollador podía hacer una pregunta sobre el código fuente y recibir una respuesta fundamentada en la estructura real del código.

Claude Code, lanzado como herramienta de disponibilidad general el 22 de mayo de 2025 junto con Claude 4, representa la cuarta fase: programación agéntica. A diferencia de sus predecesores, Claude Code no espera que se acepte o rechace una sugerencia. Recibe un objetivo, planifica una secuencia de acciones, las ejecuta usando herramientas reales de desarrollo (leyendo archivos, editando código, ejecutando pruebas, usando git), evalúa los resultados y ajusta su enfoque. El desarrollador define el objetivo y revisa el resultado en lugar de guiar cada paso [18].

El cambio del autocompletado a la agencia cambia la relación fundamental entre humano y máquina. Con el autocompletado, tú haces el trabajo y la herramienta asiste con sugerencias. Con un sistema agéntico, delegas trabajo a un colaborador capaz que planifica, ejecuta e itera de forma independiente.

Los propios datos de Anthropic reflejan esta evolución. Un análisis que preserva la privacidad de aproximadamente 400.000 sesiones de Claude Code desde octubre de 2025 hasta abril de 2026 reveló que la composición del trabajo ha cambiado dramáticamente durante esos siete meses [6]. La proporción de sesiones dedicadas a corregir código roto cayó del treinta y tres por ciento al diecinueve por ciento, mientras que el software operativo creció del catorce por ciento al veintiuno por ciento. Escribir código nuevo y analizar datos se duplicaron aproximadamente, pasando de alrededor del diez por ciento al veinte por ciento. Las tareas en sí también se volvieron más valiosas: el valor económico estimado de la sesión promedio aumentó un veintisiete por ciento.

A medida que los agentes se vuelven más capaces, los desarrolladores dejan de usarlos para pequeñas correcciones y comienzan a usarlos para desarrollo de extremo a extremo, despliegue y operaciones. La herramienta ya no solo escribe código; está ejecutando pipelines, desplegando aplicaciones, analizando datos y generando documentación.

Qué Hace Diferente a Claude Code: Arquitectura Agéntica Terminal-First

El panorama de herramientas de programación con IA en 2026 se ha consolidado alrededor de varias plataformas dominantes, cada una con una estrategia distinta y un conjunto de fortalezas. Entender sus diferencias es esencial para elegir la herramienta correcta para tu flujo de trabajo.

GitHub Copilot, mantenido por Microsoft y GitHub, es la opción más ampliamente compatible. Se integra con diez o más IDEs incluyendo VS Code, JetBrains, Neovim, Xcode, Eclipse, Zed y SQL Server Management Studio. Su fortaleza reside en la distribución y la integración del ecosistema. Copilot soporta flexibilidad de modelos a través de OpenAI

(GPT-5.1 hasta 5.4), Anthropic (Haiku 4.5, Sonnet 4.6, Opus 4.6), Google (Gemini 3 Pro y Flash) y xAI Grok Code. El nivel empresarial añade bases de conocimiento personalizadas, resúmenes de pull request, controles de administrador finos y inicio de sesión único SAML. Con 10 dólares al mes para desarrolladores individuales y 19 dólares por asiento al mes para equipos del nivel Business, permanece como la opción de mejor valor para asistencia básica de programación con IA. Sus debilidades son un acoplamiento más estrecho al ecosistema de GitHub y una capacidad limitada de ejecución autónoma multi-paso.

Cursor, desarrollado por Anysphere, es un IDE completamente nativo de IA construido sobre un fork de VS Code. Su fortaleza reside en la profundidad de integración. Cada capa de edición es consciente de IA: completaciones de pestaña, Composer 2 para ediciones agénticas multi-archivo, agentes de nube en segundo plano y Bugbot para revisión automatizada de pull request. Cursor ofrece la experiencia más profunda de IDE nativo de IA con 20 dólares al mes para desarrolladores individuales y 40 dólares por asiento al mes para equipos. Está certificado SOC 2 Tipo 2 con pruebas de penetración anuales, gestión de asientos SCIM e inicio de sesión único SAML/OIDC. Su limitación principal es que requiere la migración completa del equipo al IDE Cursor, y su nivel Hobby gratuito excluye agentes de nube y funciones avanzadas.

Claude Code, desarrollado por Anthropic, toma un enfoque fundamentalmente diferente. Es terminal-first pero funciona a través de múltiples interfaces: la CLI, extensión de VS Code, plugin de JetBrains, aplicación de escritorio, interfaz web en claude.ai/code, Slack y pipelines CI/CD. Su característica definitoria es la capacidad agéntica. Claude Code opera a nivel de proyecto, leyendo todo el código fuente, planificando a través de múltiples archivos, ejecutando cambios, ejecutando pruebas e iterando sobre los fallos. El desarrollador define el objetivo y revisa el resultado [18].

Los datos del panorama competitivo de principios de 2026 muestran un cambio notable en la preferencia de los desarrolladores. Claude Code logró una calificación del 46 por ciento de “más querido” entre los desarrolladores en la encuesta de febrero de 2026 del Pragmatic Engineer a 15.000 ingenieros de software, comparado con Cursor al 19 por ciento y GitHub Copilot al 9 por ciento [50]. Esto no se trata solo de funciones; refleja un reconocimiento creciente de que la agencia importa más que el autocompletado.

La tabla de comparación a continuación resume las diferencias clave:

Dimensión	Claude Code	GitHub Copilot	Cursor
Interfaz principal	Terminal + extensiones IDE	Plugins IDE (10+ editores)	IDE nativo de IA (fork de VS Code)
Capacidad agéntica	Autonomía completa multi-paso	Sugerencias inline + espacio de trabajo	Ediciones multi-archivo, agentes de nube
Opciones de modelo	Solo modelos Anthropic	OpenAI, Anthropic, Google, xAI	Múltiples proveedores de modelos
Integración de herramientas externas	Servidores MCP (estándar abierto)	Extensiones del ecosistema GitHub	Extensiones específicas de Cursor
Integración de Slack	Flujo de trabajo asíncrono nativo	Limitado	Ninguno
Seguridad empresarial	Listo para HIPAA, SSO, RBAC	SSO SAML, indemnización de IP	SOC 2 Tipo 2, retención cero de datos
Precio individual	20\$/mes (Pro)	10\$/mes (Pro)	20\$/mes (Pro)

La elección entre estas herramientas no es necesariamente de uno u otro. Muchos equipos grandes de ingeniería ejecutan Copilot para amplitud multi-IDE mientras despliegan Claude Code o Cursor para tareas complejas de alto impacto que requieren razonamiento profundo del código fuente y ejecución autónoma. El factor crítico es adaptar la herramienta a la tarea: el autocompletado simple se beneficia de la amplia compatibilidad IDE, mientras que la refactorización compleja multi-archivo y el desarrollo de funciones de extremo a extremo se benefician de la arquitectura agéntica de Claude Code.

El Flujo de Trabajo de Boris Cherny: Cómo el Creador Usa Realmente Claude Code

La mejor evidencia sobre cómo debería usarse Claude Code proviene de su creador. Boris Cherny, que anteriormente trabajó en Meta donde fue responsable de la calidad del código en toda la compañía, lidera el equipo de Claude Code en Anthropic. Su trayectoria profesional desde la ingeniería de calidad de código hasta construir herramientas de programación con IA le da una perspectiva única sobre lo que los desarrolladores necesitan [49]. Desde finales de 2025, el cien por ciento de su propio código ha sido escrito por Claude Code. Envía de

diez a treinta pull requests por día y ejecuta cinco sesiones locales de terminal junto con de cinco a diez instancias remotas de forma concurrente [1].

El flujo de trabajo de Cherny es sistemático, no casual. Evita la personalización, confiando en Claude Code tal como viene. Cada sesión local usa un git checkout distinto en lugar de ramas o worktrees, previniendo conflictos entre sesiones paralelas. Las sesiones remotas se ejecutan en segundo plano mediante el operador & y se desplazan con --teleport, aunque nota que del diez al veinte por ciento son abandonadas debido a escenarios imprevistos [2,29].

Para programación, Cherny prefiere Opus 4.5 (ahora sucedido por Opus 4.8) con thinking habilitado, citando calidad, fiabilidad y capacidades de uso de herramientas superiores en comparación con Sonnet. Nota que aunque Sonnet puede ser más rápido por paso, el rendimiento general es más rápido con Opus porque el primer intento tiene éxito con más frecuencia [2,29,47].

La retención de conocimiento está institucionalizada a través de un archivo CLAUDE.md mantenido por el equipo en git. Este documento registra errores, convenciones de estilo, directrices de diseño y plantillas de PR para mejorar iterativamente la IA. Cherny etiqueta los pull requests de colegas con @.claude para inyectar nuevos aprendizajes, manteniendo el archivo en aproximadamente 2,5k tokens [2]. Este es un insight clave: CLAUDE.md no es un archivo de configuración estático, sino un documento vivo que mejora con el tiempo a través de la experiencia colectiva del equipo.

Su proceso comienza con una fase de planificación iterativa usando el modo Plan antes de cambiar a auto-aceptar ediciones, lo que típicamente tiene éxito en el primer intento. Operaciones diarias como commits y PRs son automatizadas mediante comandos slash almacenados en .claude/commands/, minimizando la prompting explícita. Un comando /commit-push-pr se ejecuta docenas de veces al día pre-computando el estado de git inline para reducir la latencia del modelo. Para evitar fallos de CI por inconsistencias de formato, configura un hook PostToolUse que ejecuta automáticamente bun run format || true en coincidencias de WriteEdit [2,46].

La seguridad se maneja otorgando comandos bash seguros específicos (por ejemplo, bun run build:*, cc:*) a través de /permissions, eliminando en gran medida la necesidad de --dangerously-skip-permissions. La bandera peligrosa está reservada exclusivamente para tareas de larga ejecución en sandbox para prevenir interrupciones repetidas. Cherny enfatiza que proporcionar un bucle de retroalimentación de verificación (como ejecutar conjuntos de pruebas o automatización de navegador) es crítico, aumentando la calidad final de salida de dos a tres veces. Nota que Claude prueba cada cambio implementado en claude.ai/code usando la extensión Claude para Chrome [2,48].

Este enfoque estructurado asegura que el código llegue a revisión ya pulido, permitiendo a los ingenieros enfocarse en la dirección y la supervisión arquitectónica. El patrón es claro: planificar primero, luego automatizar los pasos rutinarios, verificar rigurosamente y dejar que Claude maneje la ejecución.

La División del Trabajo: Humano como Orquestador, Agente como Constructor

Uno de los hallazgos más importantes de la investigación de Anthropic sobre 400.000 sesiones de Claude Code es la clara división del trabajo que emerge entre humano y agente [6]. En una sesión típica, las personas toman aproximadamente el setenta por ciento de las decisiones de planificación (qué hacer, qué enfoque tomar, qué cuenta como terminado) mientras Claude toma aproximadamente el ochenta por ciento de las decisiones de ejecución (qué archivos cambiar, qué código escribir, qué comandos ejecutar).

Esta división no es accidental. Refleja tanto la preferencia humana como la capacidad del agente. Los humanos son naturalmente atraídos a la planificación y la toma de decisiones porque ahí es donde la experiencia del dominio importa más. Un contador que dirige a Claude para construir un script de conciliación aporta conocimiento de las reglas de negocio, los casos extremos al cierre de mes y los criterios de verificación que definen el éxito. Claude aporta conocimiento de estructura de código, APIs de librerías, frameworks de prueba y técnicas de depuración.

Los datos muestran esta división consistentemente a través de ocupaciones. Las Ocupaciones Informáticas y Matemáticas (el grupo de usuarios más grande) alcanzan éxito verificado en aproximadamente el treinta por ciento de sus sesiones en general [6]. Los usuarios de otras profesiones alcanzan éxito verificado aproximadamente veintiséis por ciento de las veces. Entre las sesiones que producen código, esos números son del treinta y cuatro por ciento y veintinueve por ciento respectivamente. Cada uno de los diez grupos ocupacionales no-informáticos más grandes cae dentro de siete puntos porcentuales de los ingenieros de software en términos de tasa de éxito. Las ocupaciones de gestión en realidad se clasifican más altas en éxito verificado, ligeramente por encima de las ocupaciones de ingeniería de software.

Cuanta mayor experiencia del dominio aporta una persona a una sesión, más trabajo hace Claude por instrucción. En sesiones típicamente principiantes, cada prompt activa aproximadamente cinco acciones de Claude y alrededor de 600 palabras de salida. Las sesiones de expertos activan cadenas de acción más del doble de largas (12 acciones) llevando cinco veces la salida (3.200 palabras) [6]. Esta brecha aparece dentro de cada tipo de trabajo y cada banda de valor de tarea.

La forma de la colaboración importa para cómo piensas sobre Claude Code. No es un reemplazo para un desarrollador. Es un multiplicador de fuerza para la experiencia del dominio. Un ingeniero senior haciendo su primera pregunta de Rust es principiante en Rust. Un contador que nunca ha usado Python pero le dice a Claude exactamente qué reglas de conciliación debe aplicar un script en Python y atrapa el caso extremo que maneja mal al cierre de mes es experto en esa tarea [6]. La herramienta recompensa la comprensión del dominio del problema, no la competencia en un lenguaje de programación.

La capacitación debería enfocarse en desarrollar claridad en la definición del problema, criterios de verificación y conocimiento del dominio, en lugar de aprender nueva sintaxis o referencias de API. Los usuarios más efectivos de Claude Code no son necesariamente los mejores programadores; son aquellos que pueden articular qué necesita ser construido, por qué importa y cómo verificar que funciona.

El Panorama de Agentes de Programación con IA en 2026

Claude Code no es la única herramienta de programación agéntica disponible. El panorama competitivo incluye varios participantes notables:

OpenAI Codex, el sucesor del modelo Codex original que impulsaba las primeras funciones de GitHub Copilot, ha evolucionado hacia un sistema agéntico independiente. Opera de forma similar a Claude Code con edición multi-archivo y ejecución autónoma. La fortaleza de OpenAI reside en su masiva base de desarrolladores y su integración con ChatGPT Workspace. Su debilidad es un modelo de precios más opaco para uso empresarial.

Google Antigravity 2.0, anunciado a principios de 2026, aprovecha los modelos Gemini de Google para tareas de programación. Se integra profundamente con Google Cloud y Vertex AI, haciéndolo una elección natural para equipos ya invertidos en el ecosistema Google. Su rendimiento en benchmarks de SWE-bench va detrás de Claude Fable 5 y OpenAI Codex [8].

Devin, el agente de programación autónomo original de Cognition Labs (adquirido por Amazon), pionerizó el concepto de un agente que puede planificar y ejecutar tareas completas de software. Aunque estableció la categoría, sus capacidades han sido superadas por Claude Code en la mayoría de los benchmarks a mediados de 2026.

Amazon Q Developer, integrado en el ecosistema AWS, proporciona capacidades de programación agéntica con integración profunda a servicios AWS. Es una elección sólida para equipos cuya infraestructura depende fuertemente de AWS.

Las dinámicas competitivas en este espacio están cambiando rápidamente. Lo que una vez fue una carrera de modelos (qué LLM es más inteligente programando) se ha convertido en una carrera de arneses (qué plataforma orquesta mejor las capacidades del modelo, herramientas externas y supervisión humana). La estrategia de Anthropic en el evento “Code with Claude” de mayo de 2026 reforzó esto: no se anunciaron nuevos modelos. En su lugar, se lanzaron cinco funciones de infraestructura: Dreaming, Outcomes, orquestación multiagente, Claude Finance con diez agentes pre-construidos y Add-ins [3,38]. El mensaje fue claro. El cuello de botella para los sistemas de agentes de producción no es la capacidad del modelo. Es el andamiaje alrededor del modelo.

¿Por Qué Ahora? Adopción del Mercado y el Cambio hacia Flujos de Trabajo Agénticos

La aceleración de la adopción de programación agéntica en 2026 es impulsada por varios factores convergentes.

Primero, las capacidades del modelo han alcanzado un umbral donde los agentes autónomos pueden manejar de forma fiable tareas multi-paso. Claude Fable 5 obtiene un 95 por ciento en SWE-bench Verified [22], y Claude Opus 4.8 (lanzado el 28 de mayo de 2026) entrega un juicio agéntico sustancialmente mejor con aproximadamente cuatro veces menos defectos de código pasados por alto que su predecesor [28]. Estos representan un cambio cualitativo en fiabilidad.

Segundo, la infraestructura para ejecutar agentes a escala ha madurado. La caché de prompts logra una tasa de reutilización de prefijo del 92 por ciento a través de las sesiones de Claude Code, reduciendo los costos aproximadamente un 81 por ciento y haciendo el uso sostenido de agentes económicamente viable [26]. El Protocolo de Contexto de Modelo (MCP), ahora gestionado por la Agentic AI Foundation, es implementado por Anthropic, OpenAI Agents SDK, Cursor, Cloudflare Agents y Microsoft [17], creando un estándar abierto para la integración agente-herramienta.

Tercero, la adopción del mercado ha cruzado un punto de inflexión. La proporción de proyectos de GitHub con actividad de agentes de programación se ha más que duplicado desde finales de 2025 [6]. Claude Code superó los 2.500 millones de dólares en ingresos anualizados para febrero de 2026 [8]. En la propia Anthropic, el ochenta por ciento del nuevo código de producción es ahora escrito por Claude Code [7]. El impulso comunitario va de la mano con la adopción empresarial: el repositorio de Claude Code en GitHub superó las 127.000 estrellas a mediados de 2026 [25].

Cuarto, el incentivo económico es convincente. Los estudios de caso de Stripe, Wiz,

Ramp y Rakuten demuestran ganancias de productividad difíciles de ignorar. Un equipo que puede migrar una librería de Python de 50.000 líneas a Go en veinte horas en lugar de dos o tres meses ha cambiado fundamentalmente sus restricciones de capacidad. Una compañía que reduce la entrega de funciones de veinticuatro días a cinco ha comprimido su ciclo de desarrollo de producto aproximadamente un ochenta por ciento.

El cambio hacia la programación agéntica no es una tendencia temporal ni un ciclo de hype. Representa un cambio estructural en cómo se construye el software, comparable a la transición de tarjetas perforadas a teclados o de lenguaje ensamblador a lenguajes de alto nivel. Las herramientas están cambiando, los flujos de trabajo están cambiando y las habilidades que más importan están cambiando.

La pregunta ya no es si la programación agéntica se volverá mainstream, sino cómo los equipos pueden adaptarse a ella de forma efectiva. Este libro busca proporcionar la profundidad técnica y el contexto estratégico necesarios para responder esa pregunta.

Análisis Crítico: Compromisos y Limitaciones de las Herramientas Agénticas

Antes de sumergirse en cómo usar Claude Code de forma efectiva, vale la pena examinar qué no pueden hacer las herramientas de programación agéntica y dónde introducen nuevos riesgos más allá de los flujos de trabajo de desarrollo tradicionales. Entender estos límites previene errores costosos.

La rotación de contexto es real. Incluso con una ventana de 1M tokens, las sesiones que acumulan extensas salidas de herramientas, registros de error y diffs multi-archivo eventualmente activarán la auto-compresión. Cuando ocurre la compresión, Claude puede perder el rastro de instrucciones anteriores o contextos sutiles establecidos en la primera mitad de una sesión. El incidente de degradación de calidad de abril de 2026 de Anthropic lo demostró vívidamente: un bug de caché que limpiaba el historial de razonamiento en cada turno causó olvido, repetición y elecciones extrañas de herramientas a través de miles de sesiones [12]. El postmortem del 23 de abril de 2026 reveló que tres cambios separados a nivel de producto causaron la degradación entre el 4 de marzo y el 20 de abril de 2026: un cambio de esfuerzo de razonamiento predeterminado de alto a medio (4 de marzo), un bug de optimización de caché que descartaba el razonamiento en cada turno en lugar de una vez por hora inactiva (26 de marzo) y una reducción de verbosidad del prompt del sistema que limitaba el texto de llamadas de herramienta a menos de veinticinco palabras (16 de abril) [12,33,34]. Los tres fueron corregidos para el 20 de abril en la versión 2.1.116. La lección no es que Claude Code sea poco fiable; es que incluso un solo cambio sutil en el arnés alrededor

de un LLM puede casarse en degradación de calidad medible a través de una gran base de usuarios.

Claude Code no puede manejar de forma fiable tareas que no han sido probadas. SWE-bench mide la capacidad de un modelo para corregir bugs conocidos en repositorios de código abierto. No mide la capacidad de un modelo para construir una aplicación desconocida desde cero. Claude Code puede hacer ambas cosas, pero los niveles de confianza difieren significativamente. La puntuación del 95 por ciento en SWE-bench Verified para Fable 5 refleja el rendimiento en un benchmark específico con casos de prueba conocidos y una señal de fallo bien definida. Construir una aplicación novedosa a partir de una descripción de alto nivel implica requisitos ambiguos, decisiones arquitectónicas y compensaciones que ningún benchmark captura.

Entender la brecha entre el rendimiento del benchmark y la capacidad del mundo real requiere entender cómo funciona SWE-bench. Cada tarea de SWE-bench es un issue de GitHub de un proyecto real de código abierto, emparejado con el estado del repositorio en el momento en que se presentó el issue y un conjunto de pruebas existentes que definen si una corrección es correcta. El modelo recibe la descripción del issue, explora el repositorio e intenta escribir un cambio de código que haga pasar las pruebas existentes. El benchmark tiene tres propiedades que lo hacen más fácil que el desarrollo del mundo real: el problema está bien definido (el issue describe un bug específico), el criterio de éxito es determinista (las pruebas pasan o fallan) y el código fuente es familiar (el modelo fue entrenado en código de código abierto que incluye estos repositorios).

El desarrollo del mundo real carece de las tres propiedades. Cuando le pides a Claude que “construya una API de gestión de tareas”, los requisitos son ambiguos (¿qué campos debería tener una tarea? ¿qué modelo de autenticación?), el criterio de éxito es subjetivo (¿la API se siente correcta? ¿el manejo de errores es adecuado?) y el código fuente puede ser completamente novedoso (Claude nunca ha visto tu proyecto antes). La puntuación del 95 por ciento en SWE-bench no significa que Claude tendrá éxito el 95 por ciento de las veces en tareas de desarrollo reales. Significa que Claude tiene éxito el 95 por ciento de las veces en una clase específica de tareas bien definidas de corrección de bugs con suites de prueba deterministas.

Esta distinción importa para decisiones arquitectónicas. Si estás usando Claude Code para corregir bugs en un código fuente existente con una suite de prueba exhaustiva, puedes tener alta confianza en la salida. Si estás usando Claude Code para diseñar e implementar un sistema nuevo desde cero, necesitas significativamente más supervisión humana, verificación y refinamiento iterativo. La puntuación del benchmark es una señal útil de capacidad bruta de programación, pero no es un predictor fiable de éxito en el mundo real en tareas novedosas.

Claude Code puede alucinar en cadenas multi-paso. Un solo paso puede producir una salida correcta, pero los errores se compunden a través de una cadena de diez o veinte pasos. Un archivo que debería haberse leído en el paso tres pudo haber sido editado en el paso cinco, y la comprensión de Claude del estado del código fuente se vuelve obsoleta. Por eso es esencial la verificación en cada paso. La investigación de Huang et al. (2026) probó siete clientes MCP ampliamente usados, incluyendo Claude Code, y encontró que los ataques de envenenamiento de herramientas pueden inyectar instrucciones maliciosas a través de datos devueltos por sistemas externos conectados [14]. Cuando Claude lee un README de GitHub, un resultado de consulta de base de datos o un mensaje de Slack que contiene instrucciones incrustadas, esas instrucciones se convierten en parte de su contexto y pueden manipular su comportamiento.

Claude Code no puede tomar decisiones sobre requisitos de negocio. Puede implementar lo que le pidas, pero no puede determinar si una función debería existir, qué diseño de API es óptimo o qué compensaciones son aceptables. Estos son juicios humanos. La investigación sobre 400.000 sesiones confirma esto: los humanos toman aproximadamente el setenta por ciento de las decisiones de planificación mientras Claude maneja aproximadamente el ochenta por ciento de las decisiones de ejecución [6].

Claude Code introduce riesgos reales de seguridad en tu entorno de desarrollo. Como documentó Check Point Research, CVE-2025-59536 permitía la ejecución remota de código a través de archivos de configuración de proyecto maliciosos antes de que apareciera cualquier diálogo de confianza, y CVE-2026-21852 permitía el robo de claves API mediante manipulación de ANTHROPIC_BASE_URL [11]. Estas son vulnerabilidades documentadas que afectaron despliegues reales y demostraron cómo los archivos de configuración, tradicionalmente tratados como metadatos pasivos, pueden convertirse en vectores de ataque ejecutables.

Más allá de Claude Code, el ecosistema MCP más amplio enfrenta desafíos sistémicos de seguridad. Un aviso de vulnerabilidad de Ox Security (abril de 2026) identificó que IDEs y asistentes de código incluyendo Windsurf, Claude Code, Cursor, Gemini CLI y GitHub Copilot son vulnerables a inyección de comandos a través de sus archivos de configuración JSON de MCP [17]. El único CVE emitido en ese momento fue para Windsurf, pero el patrón de ataque se aplica a través del ecosistema. NIST caracteriza la inyección de prompts como “el mayor defecto de seguridad de la IA generativa” y OWASP lo clasifica como la vulnerabilidad número uno en su Top 10 de Aplicaciones LLM [35].

Claude Code tiene límites de uso y restricciones de cómputo. A principios de 2026, muchos usuarios del plan Pro reportaron alcanzar los límites de uso en diez a quince minutos de usar Sonnet, con Opus volviéndose indisponible poco después de comenzar [27]. Anthropic duplicó los límites de uso en mayo de 2026 en respuesta a cuellos de botella de

cómputo, pero la restricción fundamental permanece: Claude Code es un recurso compartido, y el uso intensivo eventualmente alcanzará los límites de tasa [39]. El plan Pro incluye 20 dólares al mes de créditos de agente, Max 5x incluye 100 dólares y Max 20x incluye 200 dólares [2]. Para trabajo de desarrollo sostenido, la mayoría de los equipos encuentran que es necesario actualizar a Max 5x o superior para evitar interrupciones.

Claude Code no puede reemplazar la revisión de código humana. Cada cambio generado por IA debería pasar por el mismo proceso de revisión que el código escrito por desarrolladores. Cada cambio generado por IA que toque archivos críticos de seguridad debería requerir dos revisores. El agente es un colaborador, no un emisor autónomo [20]. Este principio fue reforzado por la experiencia de Stripe: Scott MacVicar, quien lidera el equipo de infraestructura de desarrolladores de Stripe, enfatizó que tratar la IA como reemplazo en lugar de colaborador es el mayor desafío en la adopción [8].

Bloqueo de proveedor y dependencia del ecosistema. Claude Code está diseñado para funcionar mejor con modelos Anthropic y se integra estrechamente con la plataforma Claude. Aunque usa estándares abiertos como MCP, migrarse desde Claude Code a otro agente o volver al desarrollo tradicional conlleva costos reales: flujos de trabajo aprendidos, archivos CLAUDE.md, habilidades personalizadas, hooks y configuraciones de servidores MCP son todos específicos de Claude Code. Los equipos deberían evaluar si las ganancias de productividad superan el riesgo de bloqueo, especialmente para proyectos con ciclos de vida largos.

Estas limitaciones no son debilidades de Claude Code per se. Son características de cualquier sistema que delega trabajo a un agente basado en LLM. Entenderlas es el primer paso hacia usar Claude Code de forma efectiva y segura.

Capítulo 2: Anatomía de Claude Code

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

El Bucle Agéntico: Recopilar, Actuar, Verificar

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Rastreando el Bucle Agéntico: Un Recorrido Completo de Terminal

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Qué Hace Diferente al Bucle Agéntico Frente al Autocompletado

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Interrumpir y Redirigir el Bucle

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Modelos Detrás de Escena: Sonnet, Opus, Fable y Selección de Modelo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Elegiendo Modelos: Un Marco de Decisión Práctico

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Sobrecarga de Tokens por Definiciones de Herramienta

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Matemáticas de Gestión de Tokens: Entendiendo Tu Gasto

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Herramientas Integradas: Operaciones de Archivos, Búsqueda, Ejecución, Web, Inteligencia de Código

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

El Directorio .claude: Configuración, CLAUDE.md y Memoria

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Ventanas de Contexto: Cómo se Llenan, Auto-Compresión y Gestión de Tokens

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Cómo se Llena el Contexto: Un Recorrido del Presupuesto de Tokens

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Cómo Funciona la Auto-Compresión Bajo el Capó

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Controlando Qué Sobrevive a la Compresión

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Estrategias para Gestionar el Contexto Efectivamente

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Capítulo 3: Comenzando y Configurando

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Instalación en Múltiples Plataformas: macOS, Linux, Windows, WSL, Homebrew, WinGet

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

La Aplicación de Escritorio de Claude Code

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Solución de Problemas Comunes de Instalación

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Autenticación y Configuración de Cuenta

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Tu Primera Sesión: Recorrido desde Cero hasta Código

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Rastreo de Sesión de Terminal: Tu Primera Interacción

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Configuración Paso a Paso: .claudeignore, settings.json y Modos de Permiso

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Modos de Permiso: Default, Auto-accept, Plan y Auto Mode

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Integraciones IDE: VS Code, JetBrains, Remote Control, Interfaz Web

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Capítulo 4: Ingeniería de Prompts para Claude Code

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

El Arte del Prompt Claude: Especificidad, Contexto y Verificación

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

CLAUDE.md como Capa Persistente de Instrucciones

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Ejemplos Few-Shot, Estructuración XML y Asignación de Roles

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Por Qué Funcionan Estas Técnicas: Las Mecánicas Detrás de la Ingeniería de Prompts

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

El Patrón “Entrevistame”: Dejar que Claude Haga Preguntas Antes de Programar

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Encadenamiento de Prompts y Gestión de Estado Entre Sesiones

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Patrones Comunes de Fallo y Cómo Evitarlos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Prueba Adversarial: Poniendo a Prueba la Salida de Claude Code

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Capítulo 5: Flujos de Trabajo de Desarrollo del Mundo Real

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Construyendo una Aplicación Full-Stack desde Cero (Proyecto Práctico)

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Fase 1: Exploración y Especificación

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Fase 2: Implementación

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Fase 3: Integración del Frontend

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Fase 4: Depurando un Bug Real

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Fase 5: Prueba Adversarial

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Manejando la Compresión de Contexto a Mitad de Implementación

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Qué Te Enseña Este Flujo de Trabajo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Refactorizando Codebases Legados: Estudio de Caso de la Migración Scala-a-Java de Stripe

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

El Estudio de Caso de Stripe: Análisis Profundo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

El Flujo de Trabajo de Migración a Escala

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

El Cuello de Botella de Revisión

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Depurando Bugs Complejos: Un Recorrido Paso a Paso

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Migración de Código a Escala: La Migración Python-a-Go de Wiz

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Incorporación a un Código Fuente Desconocido

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Capítulo 6: Pruebas, Verificación y Aseguramiento de Calidad

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Dándole a Claude Algo Contra Qué Verificar

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Generación de Pruebas y Pruebas Auto-Curativas

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Outcomes: Calificación Automatizada por Rúbricas

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Escribiendo una Rúbrica Outcomes: Un Recorrido Paso a Paso

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

El Patrón Escritor/Revisor con Sesiones Paralelas

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Habilidades de Revisión de Código y Estrategias de Revisión Adversarial

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Integración CI/CD: GitHub Actions, GitLab CI y PRs Automatizados

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Configuración CI/CD de Grado Producción

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Revisiones Automatizadas de PR con Ejecución Condicional

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Creación Automatizada de Pull Requests

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Consideraciones de Seguridad en CI/CD

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Requisitos de Cumplimiento y Auditoría

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Capítulo 7: Automatización y Escalado

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Modo Headless: Claude Code No Interactivo para Scripts y Pipelines

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Patrones de Abanico (Fan-out): Procesando Miles de Archivos en Paralelo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Fan-out Secuencial (Simple)

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Fan-out Paralelo (Alto Rendimiento)

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Fan-out con Agregación de Resultados

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Fan-out con Control de Concurrencia y Backoff

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Análisis de Costo para Patrones Fan-out

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Rutinas Claude Code: Tareas Programadas y Flujos de Trabajo Impulsados por Eventos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Hooks Pre-commit, Linting y Puertas Automatizadas de Calidad de Código

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Gestión de Costos y Presupuesto de Tokens a Escala

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Capítulo 8: Patrones Agénticos Avanzados

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Sub-Agentes: Delegando Investigación y Verificación

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Economía de Tokens de Sub-Agentes

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Habilidades: Creando Conocimiento de Dominio Reutilizable y Flujos de Trabajo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Servidores MCP: Conectando a Sistemas Externos (GitHub, Bases de Datos, Jira)

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Hooks: Aplicación Automatizada de Políticas Entre Sesiones

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Orquestación Multiagente: Agentes Principales, Sub-Agentes Especialistas, Sistemas de Archivos Compartidos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Configurando Orquestación Multiagente: Paso a Paso

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Resolviendo Conflictos de Fusión en Flujos de Trabajo Multiagente

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Parseando Datos de Observabilidad de Ejecuciones Multiagente

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Agentes Gestionados Claude y la Función “Dreaming”

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Dreaming: Agentes Auto-Mejorables

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Outcomes: Calificación Automatizada de Calidad

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Precios y Despliegue de Agentes Gestionados Claude

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Orquestación Multiagente: Patrones y Modos de Fallo

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Claude Finance y Add-ins

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Capítulo 9: Despliegue y Listado para Producción

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Desplegando Aplicaciones Construidas con Claude Code a Plataformas Cloud

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Configuración de Entorno y Gestión de Secretos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Monitoreo y Observabilidad: OpenTelemetry, Analíticas de Uso, Rastreo de Tokens

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Optimización de Rendimiento: Caché de Prompts, Selección de Modelo, Eficiencia de Contexto

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

De Prototipo a Producción: Un Estudio de Caso con Rakuten

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

La Estrategia “AI-nization” de Rakuten

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Implementación Técnica

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Desarrollo Paralelo a Escala

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Trayectoria Futura: Agentes Ambientales

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Capítulo 10: Colaboración de Equipo y Despliegue Empresarial

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Claude Code a Escala: El Despliegue de 1.370 Ingenieros de Stripe

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Configuraciones Gestionadas, Seguridad Empresarial y Cumplimiento

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Gobernanza de Identidad y Claves API

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Enrutamiento Centralizado de Tráfico

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Gobernanza de Herramientas MCP

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Sandbox Local y Aplicación de Permisos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Recorrido Completo de Despliegue Empresarial End-to-End

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Marcos de Cumplimiento

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Control de Acceso Basado en Roles

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

CLAUDE.md Compartido, Librerías de Habilidades y Distribución de Plugins

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Integrando Claude Code en Flujos de Trabajo de Equipo Existentes

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Capítulo 11: Análisis Profundo de Seguridad

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Vulnerabilidades Documentadas: CVE-2025-59536 y CVE-2026-21852

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Inyección de Prompt y Contenido Malicioso en Archivos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Inyección de Prompt vía Archivos

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Inyección de Prompt en CI/CD

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Salvaguardas de Privacidad

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Salvaguardas Adicionales

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Clasificaciones NIST y OWASP

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Exposición de Secretos en Carga de Contexto

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

La Bandera —dangerously-skip-permissions: Cuándo y Cómo Usarla Seguramente

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Sandboxing: Docker, Dev Containers y Aislamiento VM

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Arquitectura de Seguridad Empresarial: MDM, Gateways LLM, Monitoreo de Agentes

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Capítulo 12: El Futuro de la Programación Agéntica

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Hoja de Ruta de Anthropic: Dreaming, Outcomes, Orquestación Multiagente

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

La Investigación sobre Experiencia: Por Qué el Conocimiento del Dominio Gana al Antecedente de Programación

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Ocho Tendencias Moldeando la Programación Agéntica en 2026 y Más Allá

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Bloqueo de Proveedor y Dependencia del Ecosistema

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

El Rol Cambiante de los Ingenieros de Software

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

El Panorama Legal y Ético

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Qué Significa Esto para el Futuro del Trabajo y la Economía del Conocimiento

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Capítulo 13: Conclusión: Convertirse en un Ingeniero Agéntico

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Sintetizando el Viaje de Principiante a Maestro

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Los Nuevos Modelos Mentales del Desarrollo de Software

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Construyendo Tu Práctica de Claude Code: Un Plan de Acción Personal

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Mirando Hacia Adelante: Hacia Dónde Nos Llevan los Próximos Cinco Años

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.

Referencias

Este contenido no está disponible en el libro de muestra. El libro se puede comprar en Leanpub en <https://leanpub.com/claudecodees>.