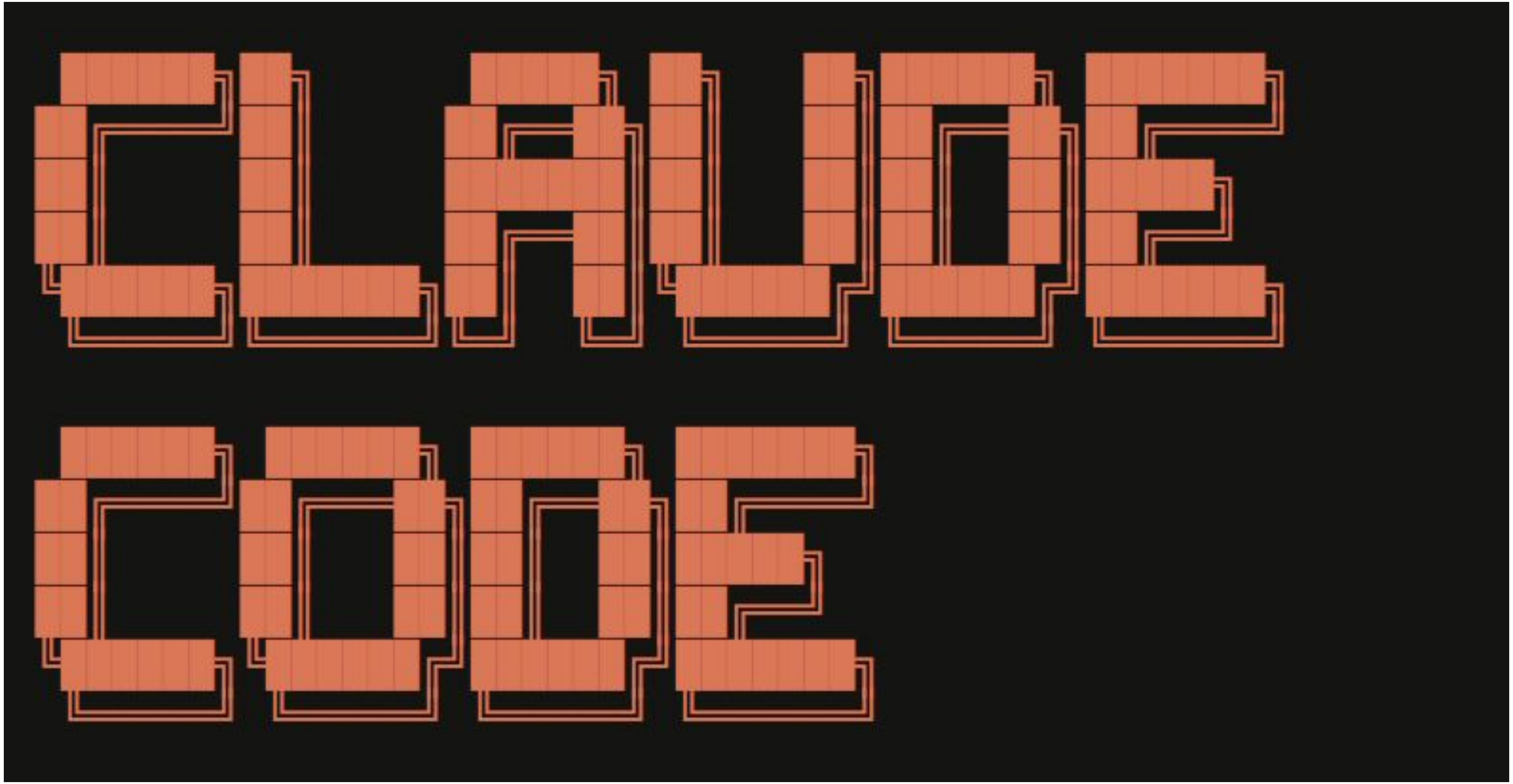
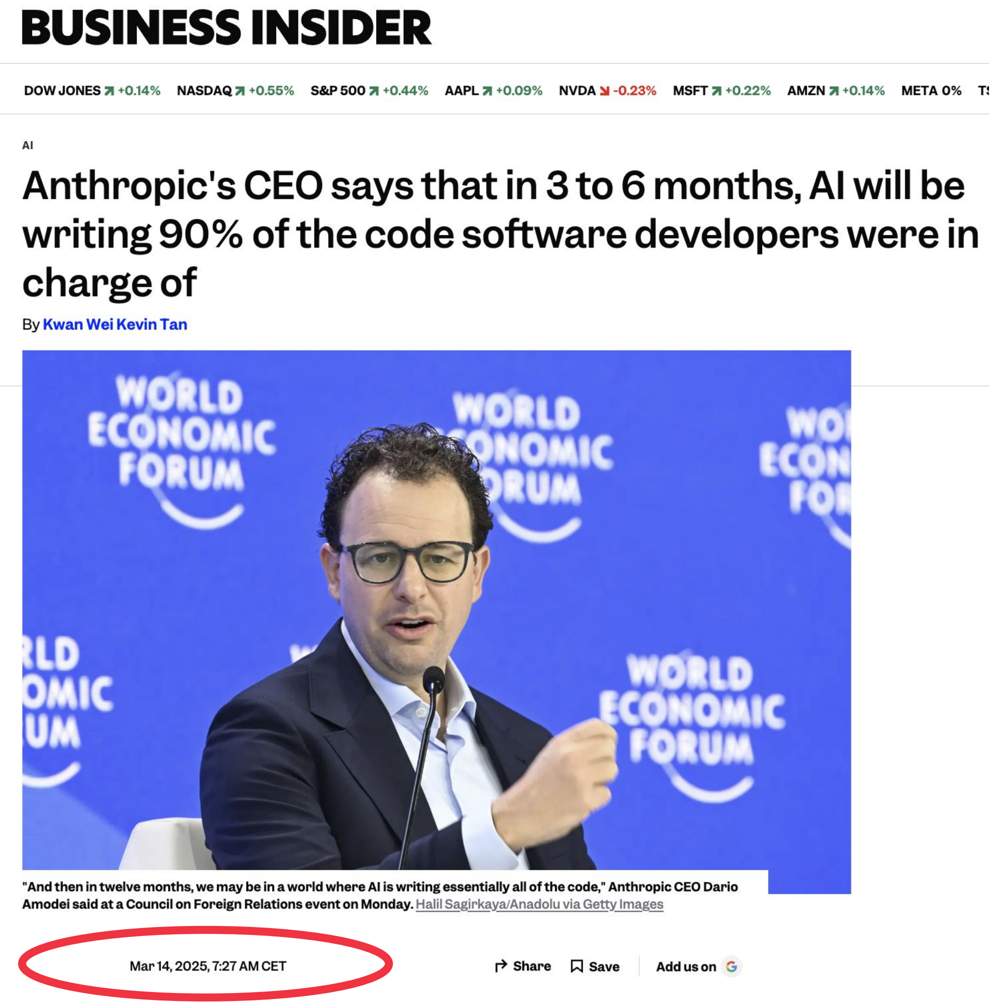
Programación con Agentes

Claude Code en acción

Diego Díaz
Chief Product Officer


CLAUDE
CODE



BUSINESS INSIDER

DOW JONES ↗ +0.14% NASDAQ ↗ +0.55% S&P 500 ↗ +0.44% AAPL ↗ +0.09% NVDA ↘ -0.23% MSFT ↗ +0.22% AMZN ↗ +0.14% META 0%

AI

Anthropic's CEO says that in 3 to 6 months, AI will be writing 90% of the code software developers were in charge of

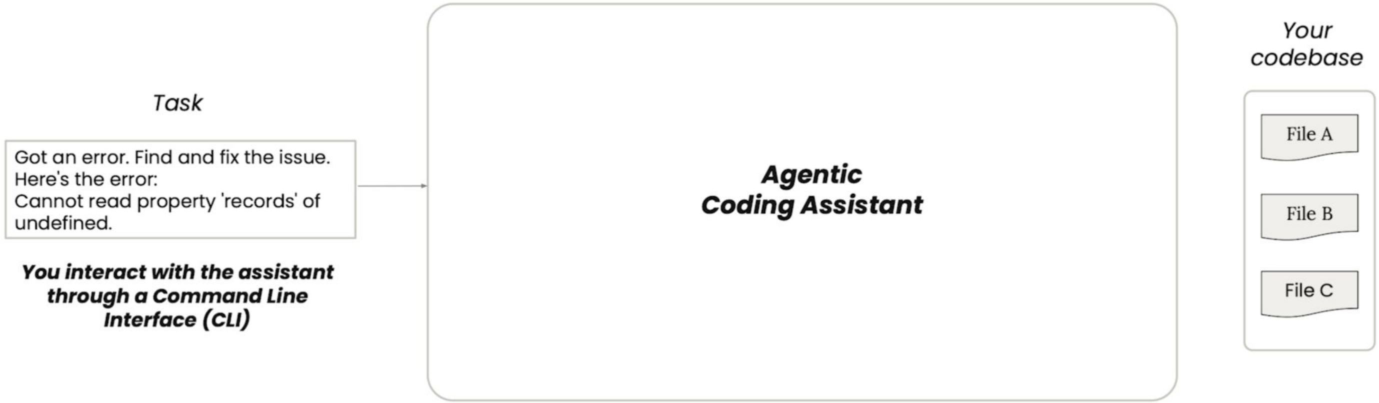
By Kwan Wei Kevin Tan

"And then in twelve months, we may be in a world where AI is writing essentially all of the code," Anthropic CEO Dario Amodei said at a Council on Foreign Relations event on Monday. Halil Sagirkaya/Anadolu via Getty Images

Mar 14, 2025, 7:27 AM CET

¿Que es Claude code?

CLI Agentic Coding Assistant



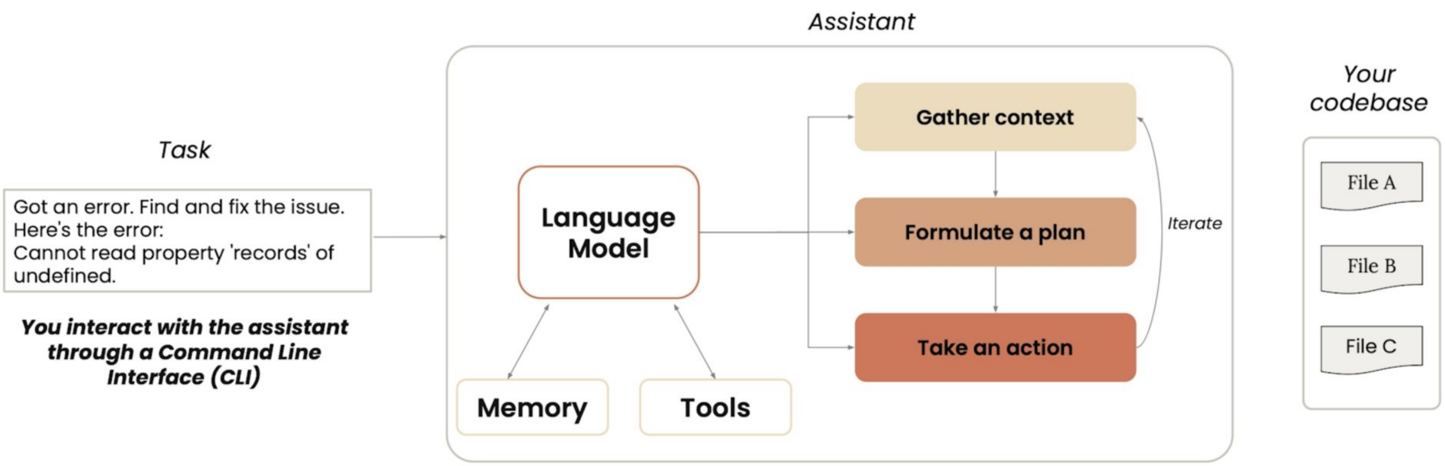
Task

Got an error. Find and fix the issue.
Here's the error:
Cannot read property 'records' of undefined.

You interact with the assistant through a Command Line Interface (CLI)

¿Que es Claude code?

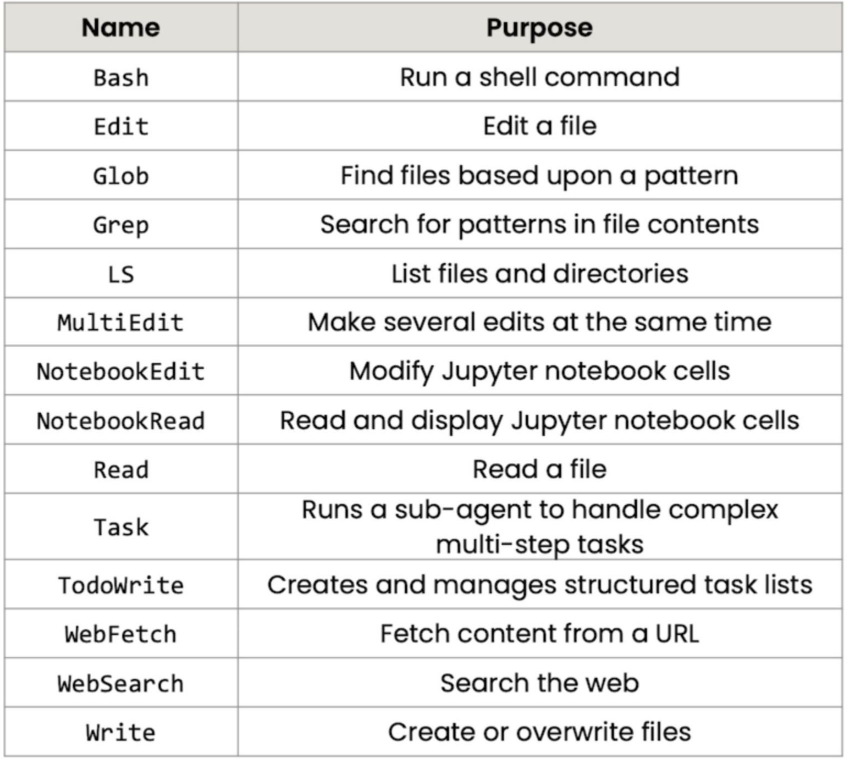
CLI Agentic Coding Assistant



Got an error. Find and fix the issue.
Here's the error:
Cannot read property 'records' of undefined.

You interact with the assistant through a Command Line Interface (CLI)

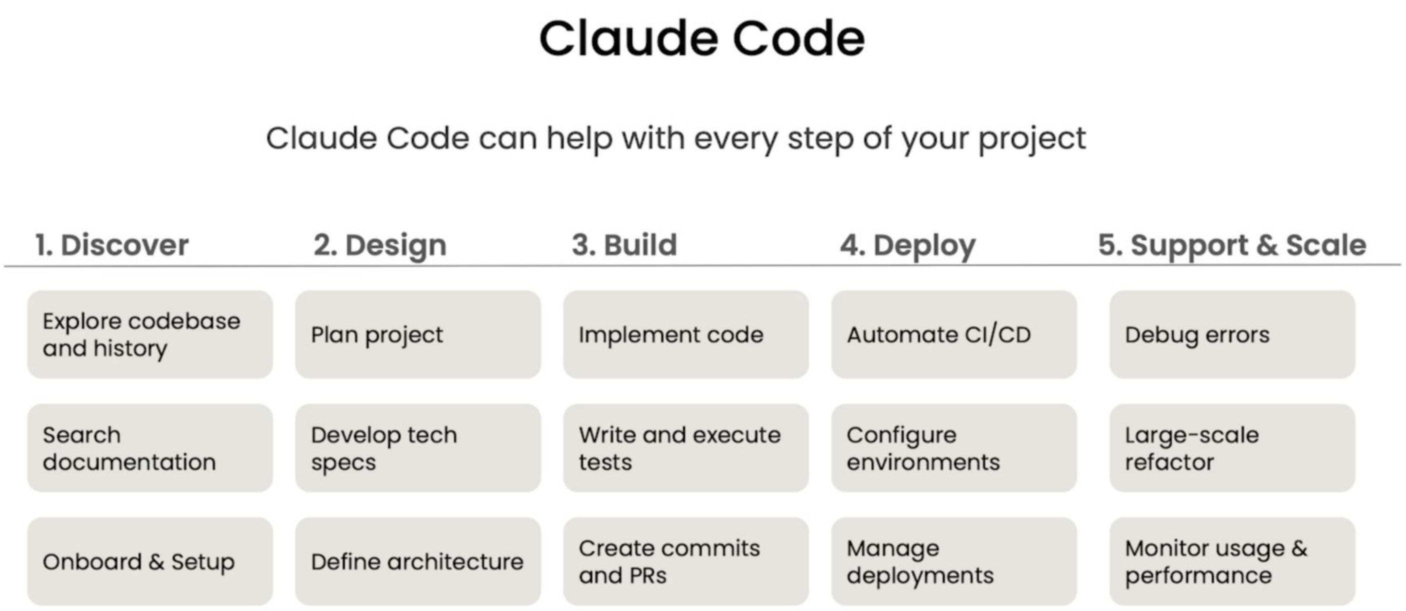
¿Qué herramientas tiene?



Name	Purpose
`Bash`	Run a shell command
`Edit`	Edit a file
`Glob`	Find files based upon a pattern
`Grep`	Search for patterns in file contents
`LS`	List files and directories
`MultiEdit`	Make several edits at the same time
`NotebookEdit`	Modify Jupyter notebook cells
`NotebookRead`	Read and display Jupyter notebook cells
`Read`	Read a file
`Task`	Runs a sub-agent to handle complex multi-step tasks
`TodoWrite`	Creates and manages structured task lists
`WebFetch`	Fetch content from a URL
`WebSearch`	Search the web
`Write`	Create or overwrite files

+ MCPs

¿Qué puede hacer Claude Code?



Claude Code

Claude Code can help with every step of your project

1. Discover	2. Design	3. Build	4. Deploy	5. Support & Scale
Explore codebase and history	Plan project	Implement code	Automate CI/CD	Debug errors
Search documentation	Develop tech specs	Write and execute tests	Configure environments	Large-scale refactor
Onboard & Setup	Define architecture	Create commits and PRs	Manage deployments	Monitor usage & performance

Casos de uso

Crear una app a partir de un diseño en Figma y probarlo

- Figma MCP, Playwright

Trabajar con notebooks

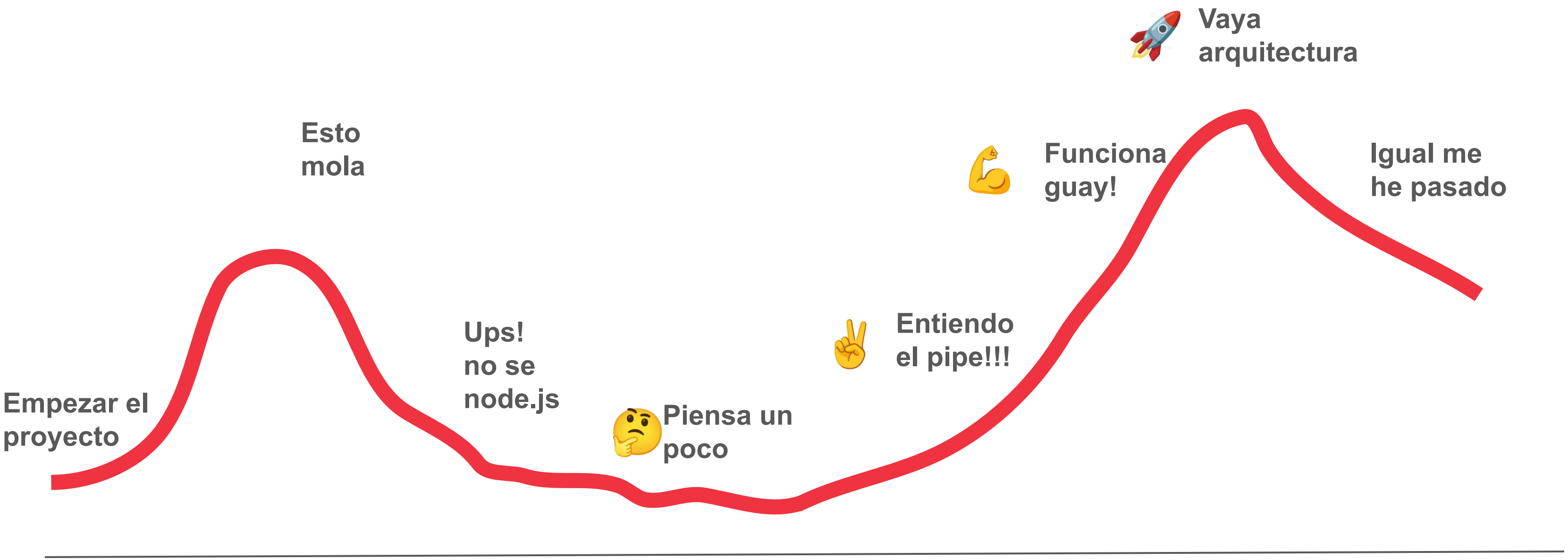
Integrar con Github para hacer Pull Request y revisarlas.

How Anthropic teams use Claude Code

24 jul 2025 • 6 min read

Mi primer proyecto con CC

Mi proyecto de verano – timeline



Empezar el proyecto

Esto mola

Ups! no se node.js

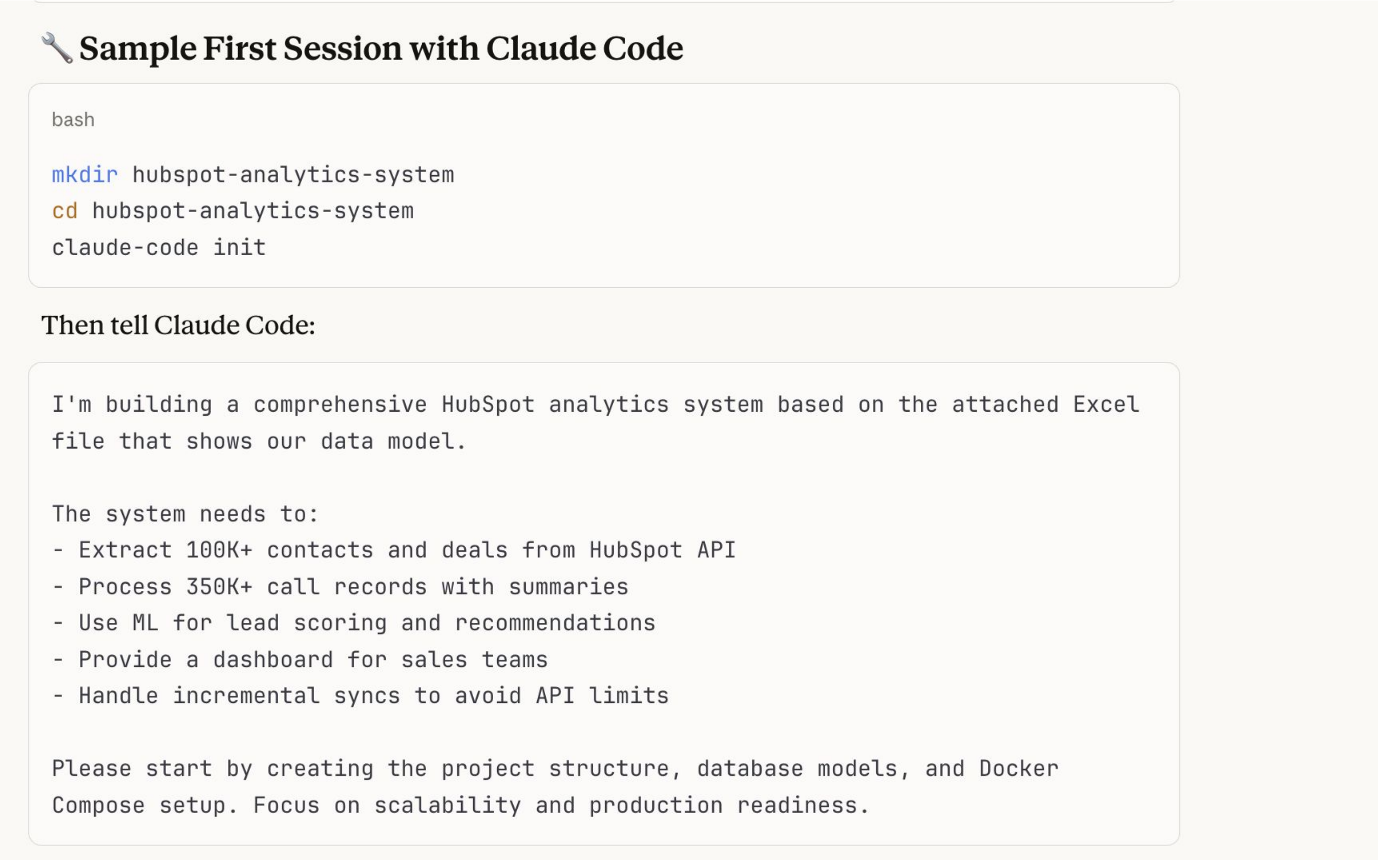
🤔 Piensa un poco

✌️ Entiendo el pipe!!!

💪 Funciona guay!

🚀 Vaya arquitectura

Igual me he pasado



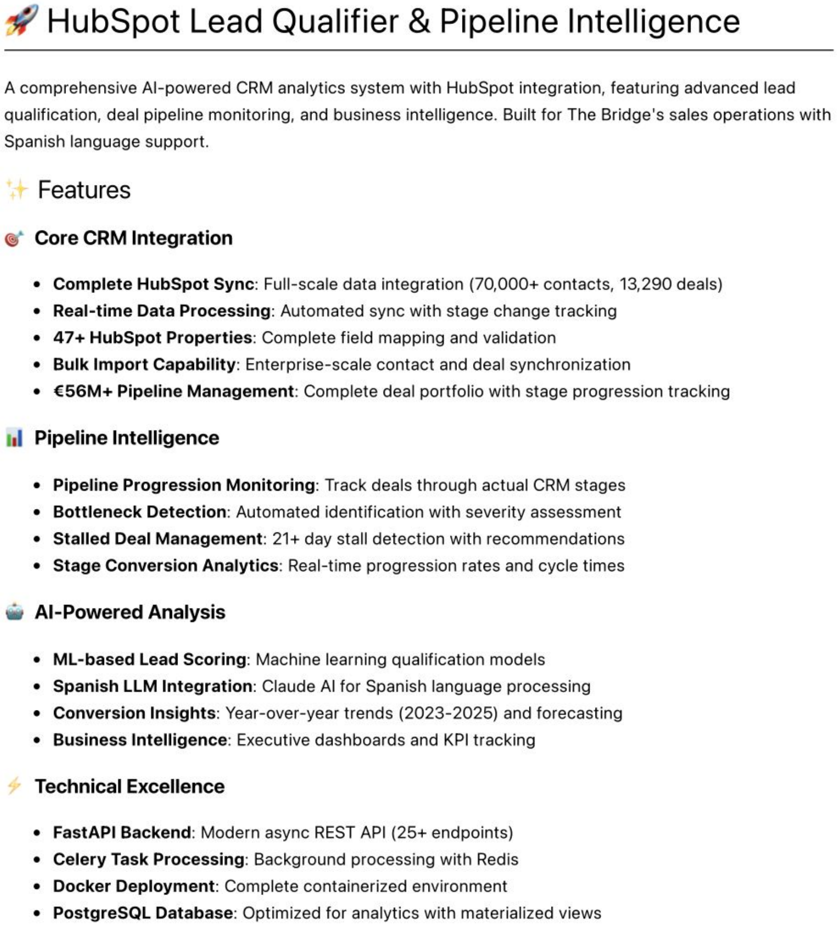
🔧 Sample First Session with Claude Code

bash

```bash
mkdir hubspot-analytics-system
cd hubspot-analytics-system
claude-code init
```

Then tell Claude Code:

```
I'm building a comprehensive HubSpot analytics system based on the attached Excel
file that shows our data model.

The system needs to:
- Extract 100K+ contacts and deals from HubSpot API
- Process 350K+ call records with summaries
- Use ML for lead scoring and recommendations
- Provide a dashboard for sales teams
- Handle incremental syncs to avoid API limits

Please start by creating the project structure, database models, and Docker
Compose setup. Focus on scalability and production readiness.
```



🚀 HubSpot Lead Qualifier & Pipeline Intelligence

A comprehensive AI-powered CRM analytics system with HubSpot integration, featuring advanced lead qualification, deal pipeline monitoring, and business intelligence. Built for The Bridge's sales operations with Spanish language support.

✨ Features

🎯 Core CRM Integration

- **Complete HubSpot Sync**: Full-scale data integration (70,000+ contacts, 13,290 deals)
- **Real-time Data Processing**: Automated sync with stage change tracking
- **47+ HubSpot Properties**: Complete field mapping and validation
- **Bulk Import Capability**: Enterprise-scale contact and deal synchronization
- **€56M+ Pipeline Management**: Complete deal portfolio with stage progression tracking

📊 Pipeline Intelligence

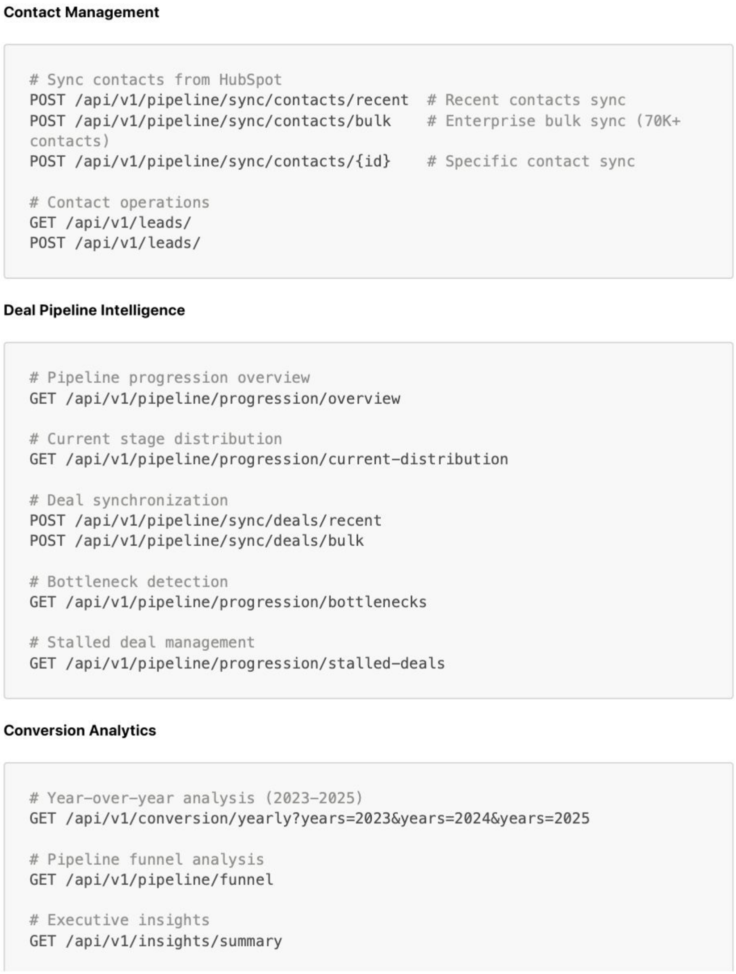
- **Pipeline Progression Monitoring**: Track deals through actual CRM stages
- **Bottleneck Detection**: Automated identification with severity assessment
- **Stalled Deal Management**: 21+ day stall detection with recommendations
- **Stage Conversion Analytics**: Real-time progression rates and cycle times

🤖 AI-Powered Analysis

- **ML-based Lead Scoring**: Machine learning qualification models
- **Spanish LLM Integration**: Claude AI for Spanish language processing
- **Conversion Insights**: Year-over-year trends (2023-2025) and forecasting
- **Business Intelligence**: Executive dashboards and KPI tracking

⚡ Technical Excellence

- **FastAPI Backend**: Modern async REST API (25+ endpoints)
- **Celery Task Processing**: Background processing with Redis
- **Docker Deployment**: Complete containerized environment
- **PostgreSQL Database**: Optimized for analytics with materialized views



Contact Management

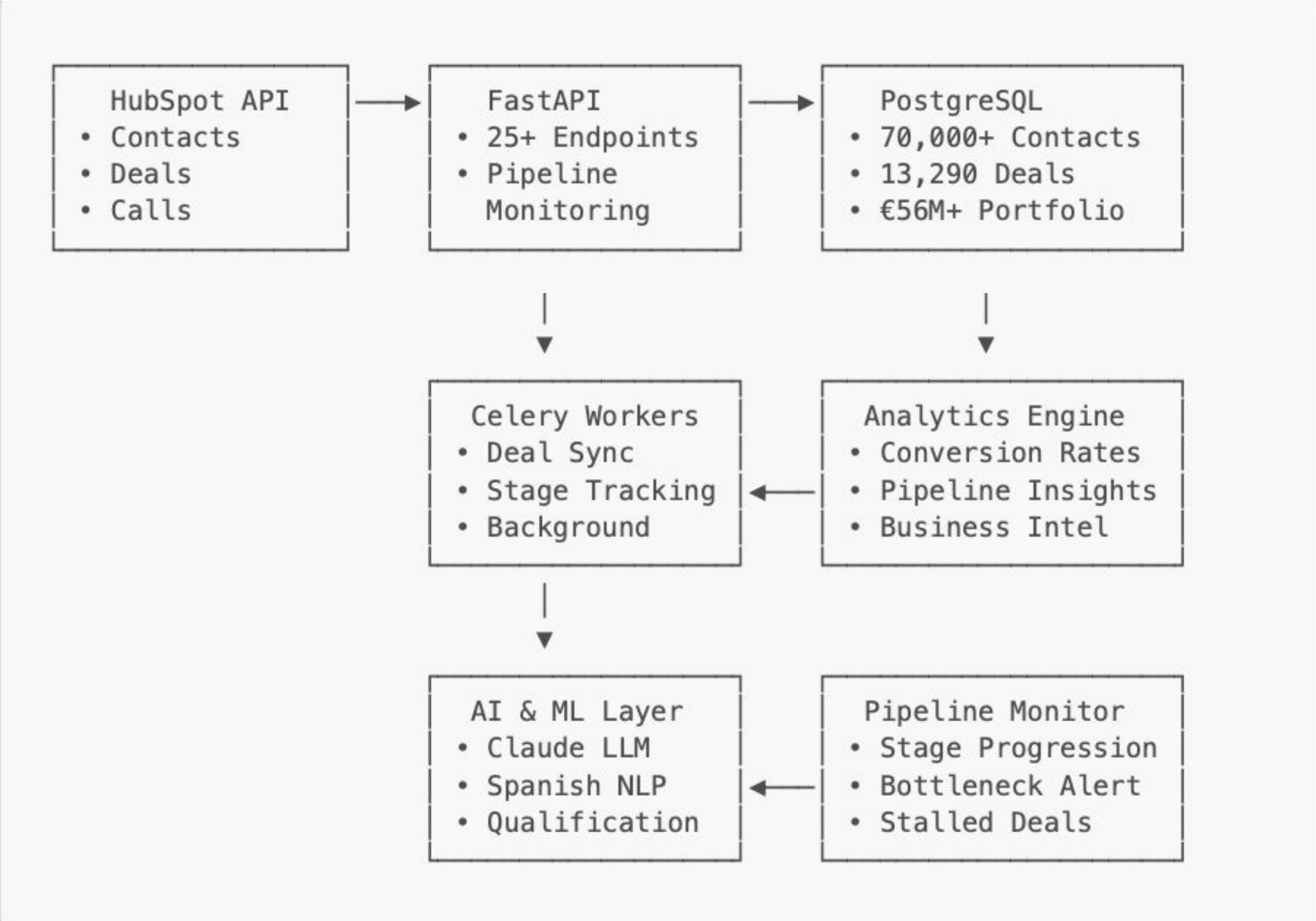
```
# Sync contacts from HubSpot
POST /api/v1/pipeline/sync/contacts/recent  # Recent contacts sync
POST /api/v1/pipeline/sync/contacts/bulk    # Enterprise bulk sync (70K+
contacts)
POST /api/v1/pipeline/sync/contacts/{id}    # Specific contact sync

# Contact operations
GET /api/v1/leads/
POST /api/v1/leads/
```

Deal Pipeline Intelligence

```
# Pipeline progression overview
GET /api/v1/pipeline/progression/overview

# Current stage distribution
GET /api/v1/pipeline/progression/current-distribution

# Deal synchronization
POST /api/v1/pipeline/sync/deals/recent
POST /api/v1/pipeline/sync/deals/bulk

# Bottleneck detection
GET /api/v1/pipeline/progression/bottlenecks

# Stalled deal management
GET /api/v1/pipeline/progression/stalled-deals
```

Conversion Analytics

```
# Year-over-year analysis (2023-2025)
GET /api/v1/conversion/yearly?years=2023&years=2024&years=2025

# Pipeline funnel analysis
GET /api/v1/pipeline/funnel

# Executive insights
GET /api/v1/insights/summary
```



HubSpot API
- Contacts
- Deals
- Calls

FastAPI
- 25+ Endpoints
- Pipeline Monitoring

PostgreSQL
- 70,000+ Contacts
- 13,290 Deals
- €56M+ Portfolio

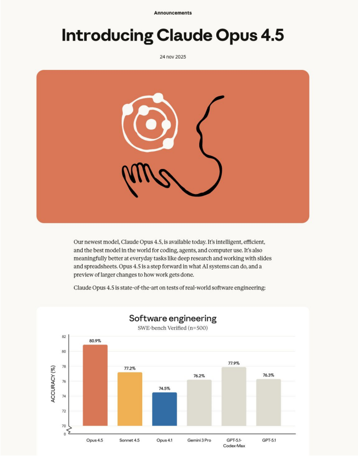
Celery Workers
- Deal Sync
- Stage Tracking
- Background

Analytics Engine
- Conversion Rates
- Pipeline Insights
- Business Intel

AI & ML Layer
- Claude LLM
- Spanish NLP
- Qualification

Pipeline Monitor
- Stage Progression
- Bottleneck Alert
- Stalled Deals

¿Cómo ha evolucionado CC?



Announcements

Introducing Claude Opus 4.5

24 nov 2025

Our newest model, Claude Opus 4.5, is available today. It's intelligent, efficient, and the best model in the world for coding, agents, and computer use. It's also meaningfully better at everyday tasks like deep research and working with slides and spreadsheets. Opus 4.5 is a step forward in what AI systems can do, and a preview of larger changes to how work gets done.

Claude Opus 4.5 is state-of-the-art on tests of real-world software engineering:

Software engineering

SWE-bench Verified (n=500)

Subagents
Hooks
Skills

Opus Plan
Ultrathink
Background Tasks

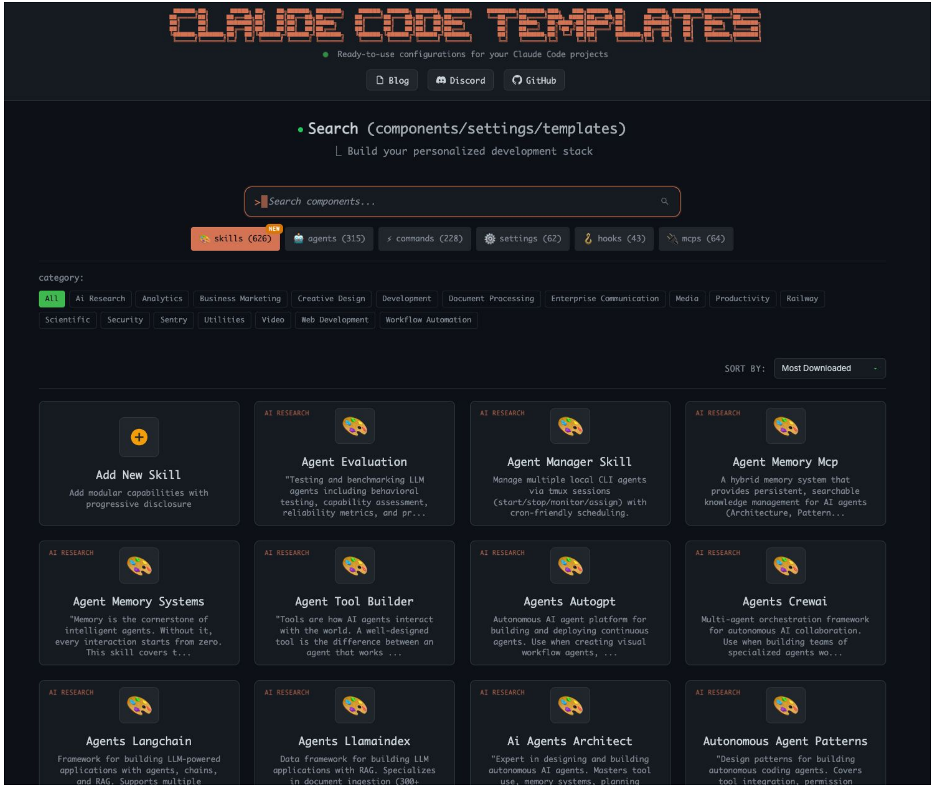
Plugin system
LSP (Language Server protocol)
Explore SubAgent
Contex Improvementes

Septiembre

Noviembre

Diciembre

Extensiones



CLAUDE CODE TEMPLATES

● Ready-to-use configurations for your Claude Code projects

● Search (components/settings/templates)

└ Build your personalized development stack

category:

SORT BY: Most Downloaded

Add New Skill
Add modular capabilities with progressive disclosure

AI RESEARCH
Agent Evaluation
"Testing and benchmarking LLM agents including behavioral testing, capability assessment, reliability metrics, and pr...

AI RESEARCH
Agent Manager Skill
Manage multiple local CLI agents via tmux sessions (start/stop/monitor/assign) with cron-friendly scheduling.

AI RESEARCH
Agent Memory Mcp
A hybrid memory system that provides persistent, searchable knowledge management for AI agents (Architecture, Pattern...

AI RESEARCH
Agent Memory Systems
"Memory is the cornerstone of intelligent agents. Without it, every interaction starts from zero. This skill covers t...

AI RESEARCH
Agent Tool Builder
"Tools are how AI agents interact with the world. A well-designed tool is the difference between an agent that works ...

AI RESEARCH
Agents Autogpt
Autonomous AI agent platform for building and deploying continuous agents. Use when creating visual workflow agents, ...

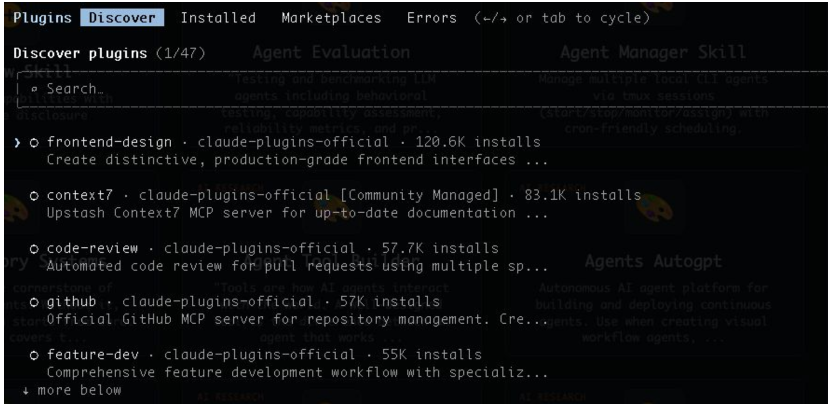
AI RESEARCH
Agents Crewai
Multi-agent orchestration framework for autonomous AI collaboration. Use when building teams of specialized agents wo...

AI RESEARCH
Agents Langchain
Framework for building LLM-powered applications with agents, chains, and RAG. Supports multiple

AI RESEARCH
Agents Llamaindex
Data framework for building LLM applications with RAG. Specializes in document ingestion (300+

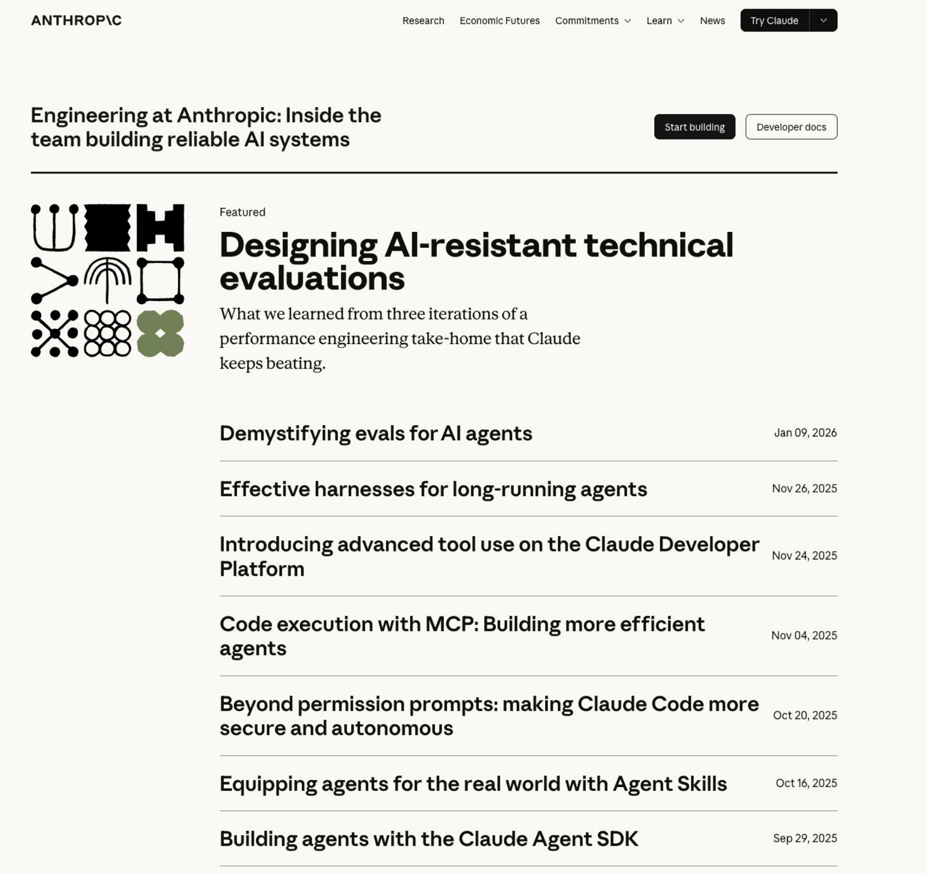
AI RESEARCH
Ai Agents Architect
"Expert in designing and building autonomous AI agents. Masters tool use, memory systems, planning

AI RESEARCH
Autonomous Agent Patterns
"Design patterns for building autonomous coding agents. Covers tool integration, permission



```
Plugins  Discover   Installed   Marketplaces   Errors  (←/→ or tab to cycle)

Discover plugins (1/47)
┌─────────────────────────────────────────────
│ ⌕ Search…
└─────────────────────────────────────────────

❯ ○ frontend-design · claude-plugins-official · 120.6K installs
    Create distinctive, production-grade frontend interfaces ...

  ○ context7 · claude-plugins-official [Community Managed] · 83.1K installs
    Upstash Context7 MCP server for up-to-date documentation ...

  ○ code-review · claude-plugins-official · 57.7K installs
    Automated code review for pull requests using multiple sp...

  ○ github · claude-plugins-official · 57K installs
    Official GitHub MCP server for repository management. Cre...

  ○ feature-dev · claude-plugins-official · 55K installs
    Comprehensive feature development workflow with specializ...
 ↓ more below
```

El futuro



Engineering at Anthropic: Inside the team building reliable AI systems

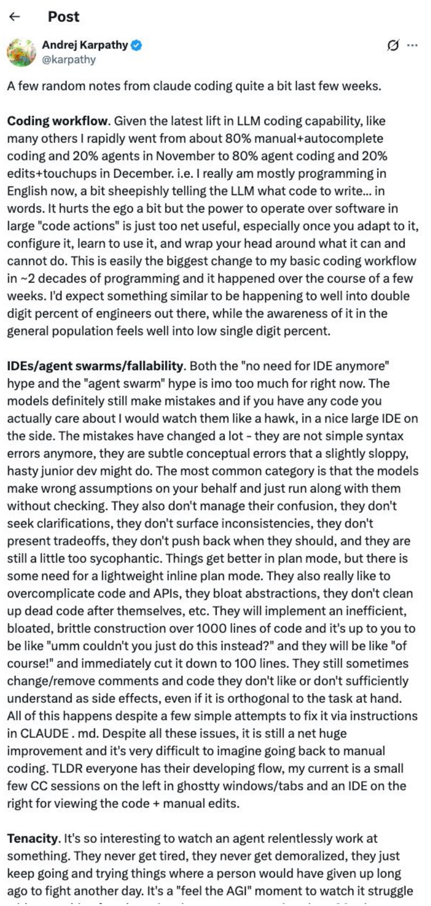
Featured

Designing AI-resistant technical evaluations

What we learned from three iterations of a performance engineering take-home that Claude keeps beating.

Demystifying evals for AI agents	Jan 09, 2026
Effective harnesses for long-running agents	Nov 26, 2025
Introducing advanced tool use on the Claude Developer Platform	Nov 24, 2025
Code execution with MCP: Building more efficient agents	Nov 04, 2025
Beyond permission prompts: making Claude Code more secure and autonomous	Oct 20, 2025
Equipping agents for the real world with Agent Skills	Oct 16, 2025
Building agents with the Claude Agent SDK	Sep 29, 2025

https://code.claude.com/docs/en/best-practices



Post

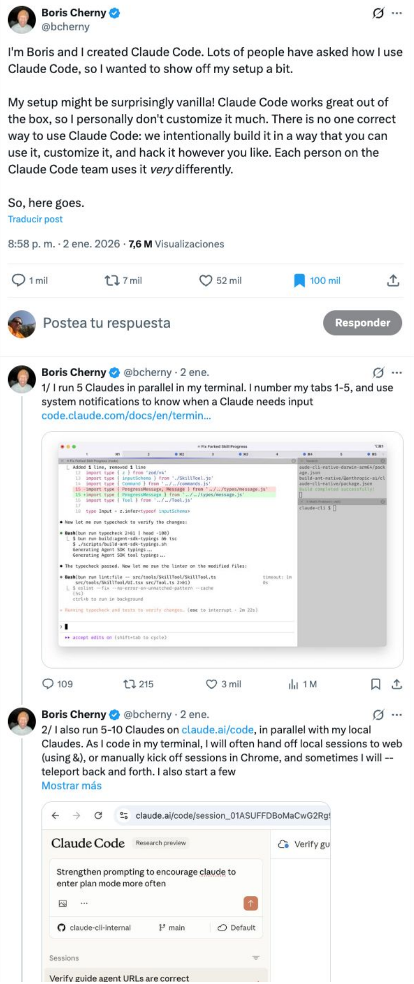
Andrej Karpathy @karpathy

A few random notes from claude coding quite a bit last few weeks.

Coding workflow. Given the latest lift in LLM coding capability, like many others I rapidly went from about 80% manual+autocomplete coding and 20% agents in November to 80% agent coding and 20% edits+touchups in December. i.e. I really am mostly programming in English now, a bit sheepishly telling the LLM what code to write... in words. It hurts the ego a bit but the power to operate over software in large "code actions" is just too net useful, especially once you adapt to it, configure it, learn to use it, and wrap your head around what it can and cannot do. This is easily the biggest change to my basic coding workflow in ~2 decades of programming and it happened over the course of a few weeks. I'd expect something similar to be happening to well into double digit percent of engineers out there, while the awareness of it in the general population feels well into low single digit percent.

IDEs/agent swarms/fallability. Both the "no need for IDE anymore" hype and the "agent swarm" hype is imo too much for right now. The models definitely still make mistakes and if you have any code you actually care about I would watch them like a hawk, in a nice large IDE on the side. The mistakes have changed a lot - they are not simple syntax errors anymore, they are subtle conceptual errors that a slightly sloppy, hasty junior dev might do. The most common category is that the models make wrong assumptions on your behalf and just run along with them without checking. They also don't manage their confusion, they don't seek clarifications, they don't surface inconsistencies, they don't present tradeoffs, they don't push back when they should, and they are still a little too sycophantic. Things get better in plan mode, but there is some need for a lightweight inline plan mode. They also really like to overcomplicate code and APIs, they bloat abstractions, they don't clean up dead code after themselves, etc. They will implement an inefficient, bloated, brittle construction over 1000 lines of code and it's up to you to be like "umm couldn't you just do this instead?" and they will be like "of course!" and immediately cut it down to 100 lines. They still sometimes change/remove comments and code they don't like or don't sufficiently understand as side effects, even if it is orthogonal to the task at hand. All of this happens despite a few simple attempts to fix it via instructions in CLAUDE . md. Despite all these issues, it is still a net huge improvement and it's very difficult to imagine going back to manual coding. TLDR everyone has their developing flow, my current is a small few CC sessions on the left in ghostty windows/tabs and an IDE on the right for viewing the code + manual edits.

Tenacity. It's so interesting to watch an agent relentlessly work at something. They never get tired, they never get demoralized, they just keep going and trying things where a person would have given up long ago to fight another day. It's a "feel the AGI" moment to watch it struggle



Boris Cherny @bcherny

I'm Boris and I created Claude Code. Lots of people have asked how I use Claude Code, so I wanted to show off my setup a bit.

My setup might be surprisingly vanilla! Claude Code works great out of the box, so I personally don't customize it much. There is no one correct way to use Claude Code: we intentionally build it in a way that you can use it, customize it, and hack it however you like. Each person on the Claude Code team uses it *very* differently.

So, here goes.

Traducir post

8:58 p. m. · 2 ene. 2026 · **7,6 M** Visualizaciones

1 mil · 7 mil · 52 mil · 100 mil

Postea tu respuesta — Responder

Boris Cherny @bcherny · 2 ene.

1/ I run 5 Claudes in parallel in my terminal. I number my tabs 1-5, and use system notifications to know when a Claude needs input code.claude.com/docs/en/termin...

```
Added 1 line, removed 1 line
  import type { z } from 'zod/v4'
  import type { inputSchema } from './SkillTool.js'
  import type { Command } from '../../commands.js'
 -import type { ProgressMessage, Message } from '../../types/message.js'
 +import type { ProgressMessage } from '../../types/message.js'
  import type { Tool } from '../../Tool.js'

  type Input = z.infer<typeof inputSchema>

● Now let me run typecheck to verify the changes:

● Bash(bun run typecheck 2>&1 | head -100)
  ⎿ $ bun run build:agent-sdk-typings && tsc
    $ ./scripts/build-ant-sdk-typings.sh
    Generating Agent SDK typings...
    Generating Agent SDK tool typings...

● The typecheck passed. Now let me run the linter on the modified files:

● Bash(bun run lint:file -- src/tools/SkillTool/SkillTool.ts        timeout: 1m
      src/tools/SkillTool/UI.tsx src/Tool.ts 2>&1)                  0s
  ⎿ $ eslint --fix --no-error-on-unmatched-pattern --cache
    (5s)
    ctrl+b to run in background

· Running typecheck and tests to verify changes… (esc to interrupt · 2m 22s)

>

▸▸ accept edits on (shift+tab to cycle)
```

```
aude-cli-native-darwin-arm64/package.json
build-ant-native/@anthropic-ai/claude-cli-native/package.json
Build completed successfully!

claude-cli $
```

109 · 215 · 3 mil · 1 M

Boris Cherny @bcherny · 2 ene.

2/ I also run 5-10 Claudes on claude.ai/code, in parallel with my local Claudes. As I code in my terminal, I will often hand off local sessions to web (using &), or manually kick off sessions in Chrome, and sometimes I will -- teleport back and forth. I also start a few

Mostrar más

claude.ai/code/session_01ASUFFDBoMaCwG2Rg

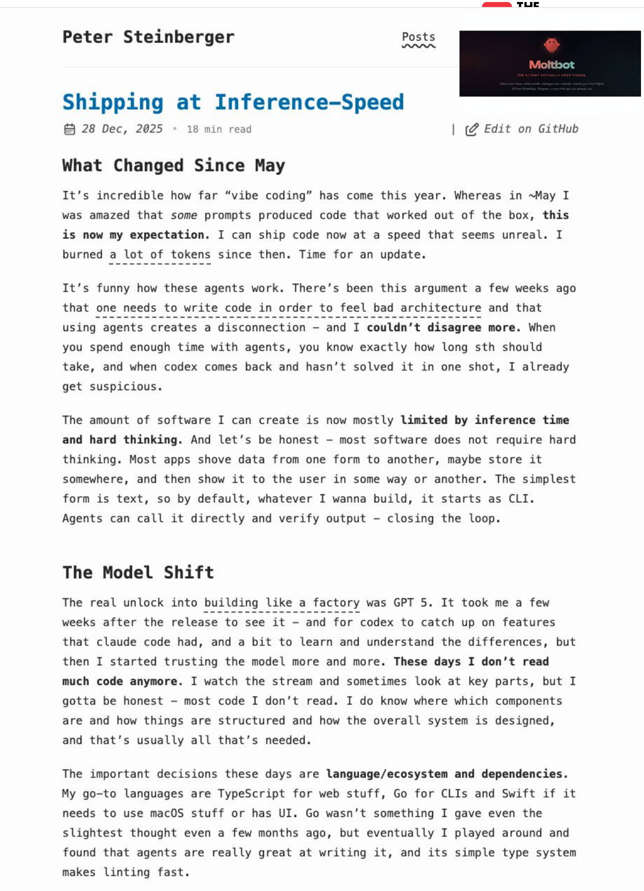
Claude Code Research preview

Strengthen prompting to encourage claude to enter plan mode more often

claude-cli-internal · main · Default

Sessions

Verify guide agent URLs are correct



Peter Steinberger — Posts

Shipping at Inference-Speed

28 Dec, 2025 • 18 min read | *Edit on GitHub*

What Changed Since May

It's incredible how far "vibe coding" has come this year. Whereas in ~May I was amazed that *some* prompts produced code that worked out of the box, **this is now my expectation.** I can ship code now at a speed that seems unreal. I burned a lot of tokens since then. Time for an update.

It's funny how these agents work. There's been this argument a few weeks ago that one needs to write code in order to feel bad architecture and that using agents creates a disconnection – and I **couldn't disagree more.** When you spend enough time with agents, you know exactly how long sth should take, and when codex comes back and hasn't solved it in one shot, I already get suspicious.

The amount of software I can create is now mostly **limited by inference time and hard thinking.** And let's be honest – most software does not require hard thinking. Most apps shove data from one form to another, maybe store it somewhere, and then show it to the user in some way or another. The simplest form is text, so by default, whatever I wanna build, it starts as CLI. Agents can call it directly and verify output – closing the loop.

The Model Shift

The real unlock into building like a factory was GPT 5. It took me a few weeks after the release to see it – and for codex to catch up on features that claude code had, and a bit to learn and understand the differences, but then I started trusting the model more and more. **These days I don't read much code anymore.** I watch the stream and sometimes look at key parts, but I gotta be honest – most code I don't read. I do know where which components are and how things are structured and how the overall system is designed, and that's usually all that's needed.

The important decisions these days are **language/ecosystem and dependencies.** My go-to languages are TypeScript for web stuff, Go for CLIs and Swift if it needs to use macOS stuff or has UI. Go wasn't something I gave even the slightest thought even a few months ago, but eventually I played around and found that agents are really great at writing it, and its simple type system makes linting fast.

https://x.com/bcherny/status/2007179832300581177?s=20

¡GRACIAS!

THE BRIDGE

Impulsa tu carrera digital

Good Bye Mate