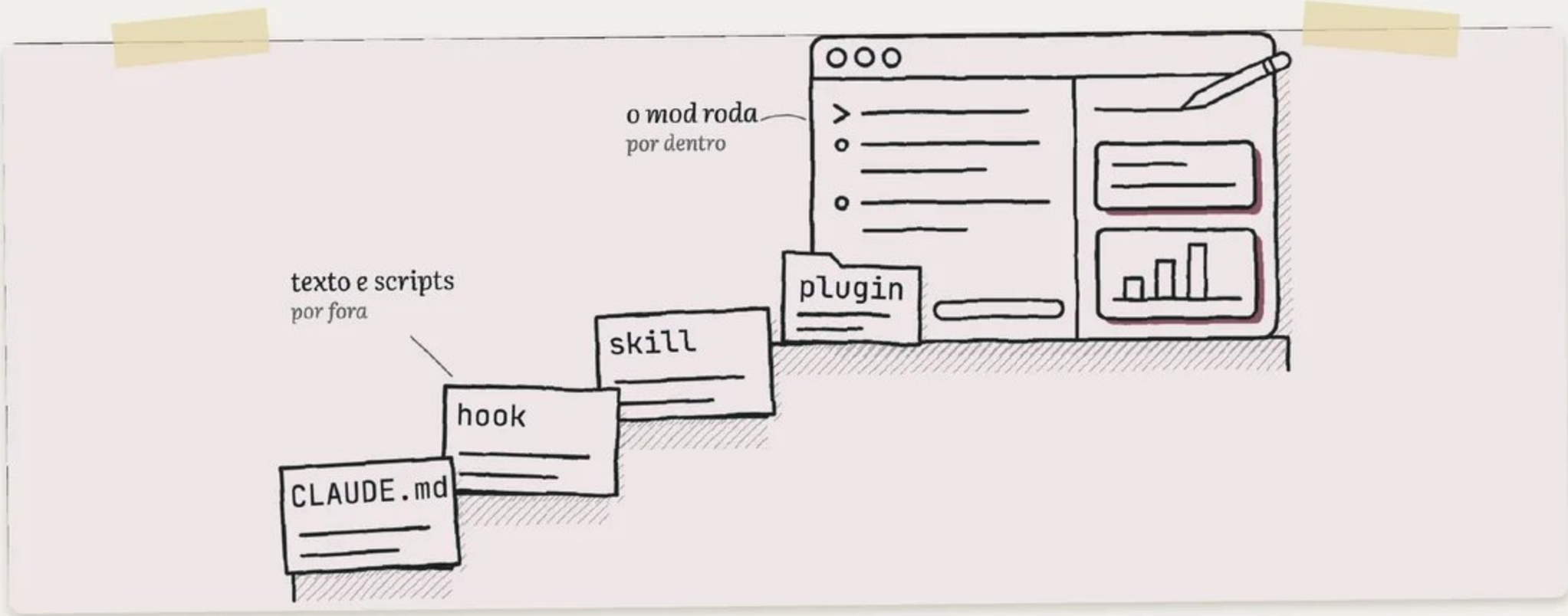
Vol. 04 · ficha 1 · nº 029

02/10/2026 · IA

Mods do Claude Code

O que são e as peças que vieram antes · parte 1 de 2



Código que roda dentro do Claude Code

Um **mod** é um plugin do Claude Code com código que roda dentro do programa: reage aos eventos da sessão, guarda estado, registra comandos e desenha na interface.

01/10/2026

o anúncio dos mods, que chegam ligados por padrão

2.1.287

a versão do Claude Code em que os mods ficaram oficiais

19 meses

desde o lançamento do Claude Code, em 24/02/2025

até ali, tudo entregava texto ou ferramentas ao modelo, ou rodava um script por fora

Um painel só pode vir do programa.

o modelo — a IA, o Claude

Lê texto e responde.

Ele pede as ferramentas, mas quem as executa é o programa.

o Claude Code — o programa no terminal

Manda, executa e desenha.

Manda o texto ao modelo, executa as ferramentas que ele pede (ler um arquivo, rodar um comando) e desenha a tela.

Contexto é tudo o que o modelo tem na frente para responder: as instruções, a conversa, os arquivos lidos. Ele tem tamanho máximo e é medido em **tokens**, os pedaços de palavra em que o modelo conta o texto e pelos quais se cobra.

Do lançamento ao mod: 19 meses

Em cada degrau, a mesma pergunta: essa peça consegue pôr algo novo na interface?



O que cada peça alcança

texto para o modelo — 5 peças

CLAUDE.md

As instruções permanentes, entregues ao modelo em toda sessão.

Comando

Um prompt salvo, chamado por `/nome`. Poupa a digitação, e só.

Subagente

Um ajudante com contexto só dele. O que volta é texto.

Estilo de saída

O papel, o tom e o formato das respostas.

Skill

Instruções que o modelo carrega quando o pedido combina.

ferramentas — MCP

MCP

Liga o modelo a sistemas de fora: um banco de dados, um navegador. A interface continua a mesma.

script de fora — 2 peças

Hook

Um comando do sistema que roda sozinho num evento da sessão.

Linha de status

Uma linha no rodapé, sem clique, escrita por um script.

embalagem e cores — 2 peças

Plugin

A pasta que se instala de uma vez. Empacota o que já existia.

Tema

As cores da interface.

por dentro

faltava a peça que roda por dentro

Nenhuma dessas peças põe um painel na interface.

> O `CLAUDE.md`, o comando, o subagente, o estilo e a skill são texto para o modelo.
>
> O MCP dá ferramentas a ele.
>
> O hook e a linha de status rodam um script do lado de fora.
>
> O plugin embala, e o tema troca cores.

Faltava a peça que roda por dentro.

O mod é um plugin com uma diferença

Num plugin comum, o `hooks/hooks.json` lista os hooks: que comando do sistema rodar em cada evento. No mod, o mesmo arquivo aponta um módulo, um arquivo de código.

O módulo exporta `register(on)`. Cada `on(...)` liga uma função a um evento: uma chamada de ferramenta, o início de um turno, um pedaço da interface sendo desenhado.

A função recebe `$`, a API dos mods; `e`, o evento; e `next`, que passa o evento adiante, como num middleware.

quem não chama next responde no lugar

plugins/csr-cockpit/hooks/hooks.json

```
{ "modules": ["./register.tsx"] }
```

o código do mod

plugins/csr-cockpit/hooks/register.tsx

```
export const register: Register = on => {
  on('agent.spawn', async ($, e, next) => {
    const iniciado = await next(e)
    // guarda o agente e o pedido dele no estado da sessão
    return iniciado
  })
}
```

segue adiante como chegou

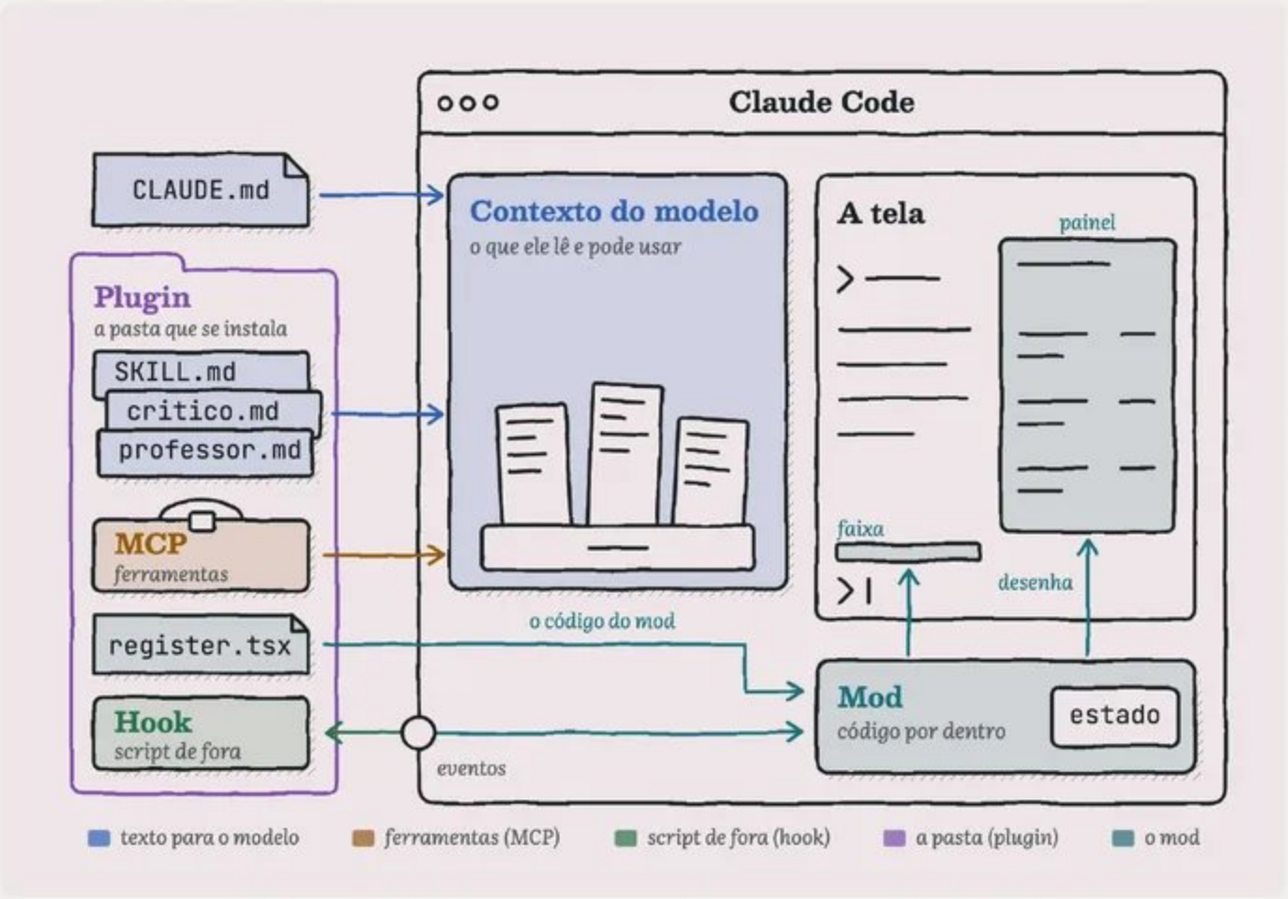
Um hook do csr-cockpit, o mod da parte 2: quando o Claude Code cria um subagente (`agent.spawn`), ele deixa o evento seguir e anota o agente e o pedido.

Por dentro e por fora

Com isso, um mod:

1. roda dentro do Claude Code, no processo dele, e não como um script à parte;
2. guarda estado (lembra o que viu) entre um evento e outro, pelo tempo da sessão;
3. registra um comando `/nome` que executa código na hora, sem passar pelo modelo;
4. desenha: um painel ao lado da conversa e uma faixa acima do prompt.

Das peças da tabela, é a primeira que roda ali dentro e a primeira que desenha um painel.



Onde ele roda, e o cuidado antes de instalar

onde roda — terminal e desktop

Os mods funcionam no terminal e na aba Code do aplicativo de desktop.

O aplicativo traz a própria cópia do Claude Code, que pode estar numa versão anterior: `claude --version` mostra a versão do terminal, não a do aplicativo.

desde quando — vale em out/2026

Em setembro de 2026, um acesso antecipado, só para quem ligava uma variável de ambiente.

A partir da 2.1.287, oficiais e ligados por padrão.

Atenção

Um mod é código que roda com as suas permissões, sem isolamento: pode ler e gravar arquivos, iniciar processos e usar a rede. Instale só mods de autores em quem você confia e, antes, leia o código ou rode `claude plugin validate <pasta>`, que lista os eventos que ele escuta e as chamadas que faz, sem executar nada.

As confusões mais comuns

skill · agente · plugin 1/5

Não são alternativas.

A skill é uma instrução. O agente executa uma tarefa numa conversa à parte. O plugin é a pasta que entrega os dois.

mod 2/5

Mod é plugin, não skill.

A skill é um texto que o modelo lê. O mod é código que o Claude Code executa. Os dois chegam dentro de um plugin.

build 3/5

O mod não tem build.

Nada de `npm install` nem de empacotador: o Claude Code lê o código direto da pasta.

instalação 4/5

Não se instala uma skill solta de dentro de um plugin.

A unidade de instalação é o plugin, e tudo o que está na pasta dele vem junto.

marketplace 5/5

Marketplace não é uma loja.

É o catálogo de onde os plugins se instalam: um repositório git com um arquivo que lista os itens. Um repositório público no GitHub basta.

Na parte 2, um mod pronto

✓ A **parte 2** mostra um mod pronto, o csr-cockpit. Para ir além do básico: a página **Extend Claude Code** e os **mods de exemplo** da Anthropic.

blog.cesarschutz.com.br/posts/claude-code-csr-cockpit

code.claude.com/docs/en/features-overview

github.com/anthropics/claude-code-playground › claude-code/mods

As fontes do post

Anthropic: Claude 3.7 Sonnet and Claude Code (o lançamento, 24/02/2025)

Claude: Claude Code mods (01/10/2026), os plugins (09/10/2025) e as Agent Skills (16/10/2025)

Claude Code: o changelog, o CHANGELOG.md no GitHub e as versões no npm

Documentação: memória, MCP, hooks, subagentes, estilos, skills, plugins e mods

Cesar Schutz: o claude-code-kit

Do livro



Volume 04 · 2 artigos

IA

Ver o livro →