

SEU AGENTE NO CLAUDE CODE EM 5 CAMADAS

Do terminal vazio a um agente que sabe quem você é,
lembra do que aprendeu e executa seus processos.
Sem super prompt e sem saber programar.

01 Contexto

02 Memória

03 Skills

04 Sub-agentes

05 Rotinas

Por que camadas, e não um super prompt

Um agente bom não nasce de um prompt de 2 mil palavras. Ele nasce de cinco camadas que você monta uma vez e o Claude Code reaproveita em toda sessão.

05

Rotinas

tarefas que rodam sozinhas

04

Sub-agentes

especialistas para o trabalho pesado

03

Skills

processos que viram comando

02

Memória

o que ele aprende com você

01

Contexto

quem você é e como trabalha

Cada camada se apoia na de baixo. Comece sempre pelo contexto.

Super prompt

Você cola o mesmo textão toda vez, ele esquece no meio da conversa e cada pedido começa do zero.

Agente em camadas

Você monta uma vez. Ele já sabe quem você é, lembra das suas correções e repete seus processos com um comando.

PASSO ZERO

Instale o Claude Code

Você precisa de uma assinatura do Claude (Pro, Max, Team ou Enterprise) ou de uma conta no Claude Console, e de uma pasta onde o agente vai trabalhar. Não precisa saber programar.

- 1 **Instale.** Abra o terminal e cole o comando do seu sistema.

Sistema	Comando
macOS, Linux ou WSL	<code>curl -fsSL https://claude.ai/install.sh bash</code>
Windows (PowerShell)	<code>irm https://claude.ai/install.ps1 iex</code>
Windows (CMD)	<code>curl -fsSL https://claude.ai/install.cmd -o install.cmd && install.cmd && del install.cmd</code>

No Windows, se o terminal mostra `PS C:\` você está no PowerShell. Se mostra só `C:\`, está no CMD. O instalador não mostra barra de progresso, então espere ele terminar.

- 2 **Confira.** Abra um terminal novo e rode `claude --version`. Apareceu um número de versão, deu certo.
- 3 **Abra na sua pasta.** Entre na pasta e digite `claude`. Na primeira vez ele pede login pelo navegador.

COLE NO TERMINAL

```
cd minha-pasta
claude
```

01

CAMADA 1

Contexto: o CLAUDE.md

O `CLAUDE.md` é um arquivo de texto que o Claude lê no começo de toda sessão. É ali que fica tudo que ele precisa saber sempre: quem você é, o que você faz e como você quer as coisas.

Onde ele fica

Arquivo	Vale para
<code>~/.claude/CLAUDE.md</code>	Todos os seus projetos (preferências pessoais)
<code>CLAUDE.md</code> na raiz da pasta	Só aquele projeto

Atalho: dentro do Claude, digite `/init`. Ele analisa a pasta e cria um `CLAUDE.md` inicial, que você ajusta depois.

Ou monte com uma entrevista

COLE NO CLAUDE

Me ajuda a montar meu CLAUDE.md. Não escreve nada ainda.
Primeiro me entrevista, poucas perguntas por vez, sobre:

- o que eu faço e pra quem
- como eu falo e como quero que você responda
- as ferramentas que eu uso
- o que você nunca deve fazer

Quando tiver o suficiente, escreve um CLAUDE.md curto e direto, separando o que vale pra todos os projetos do que vale só pra este.

Um modelo simples para preencher

MODELO DE CLAUDE.MD

```
# Quem eu sou
[seu nome, o que você faz]

# O que eu vendo e pra quem
[produto, cliente ideal]

# Como falar
[tom, palavras que você usa, palavras proibidas]

# Como trabalhar comigo
[respostas curtas? perguntar antes de agir?]

# Nunca faça
[o que ele não pode fazer]
```

Regra de ouro: mantenha o arquivo curto. A documentação oficial recomenda menos de 200 linhas, porque arquivo longo ocupa espaço e piora a obediência. Processo grande e passo a passo não entram aqui: viram skill na camada 3.

02

Memória: o que ele aprende com você

A memória é o contrário do `CLAUDE.md`. No `CLAUDE.md` você escreve as regras. Na memória, o próprio Claude anota o que aprendeu trabalhando com você: uma correção que você repetiu, um comando útil, uma preferência que apareceu.

CLAUDE.md

Quem escreve: você.

Guarda: regras e instruções.

Carrega: em toda sessão.

Memória automática

Quem escreve: o Claude.

Guarda: aprendizados e padrões.

Carrega: em toda sessão.

Já vem ligada

A memória automática fica ativa por padrão quando você usa o Claude Code no seu computador. Cada projeto tem a sua pasta, em `~/.claude/projects/<projeto>/memory/`, com um índice chamado `MEMORY.md`.

Para ver e editar

Digite `/memory` dentro do Claude. Ali você abre os arquivos, liga ou desliga a memória automática e apaga o que não quer que ele guarde. Tudo é texto simples.

Para mandar ele lembrar de algo

COLE NO CLAUDE

Lembra disso: minhas legendas nunca levam travessão.

Se for uma regra que precisa valer sempre, peça para colocar no lugar certo:

COLE NO CLAUDE

Adiciona isso no `CLAUDE.md`.

Na prática: o que você decidiu vai no `CLAUDE.md`. O que você corrigiu no caminho, deixa a memória guardar.

03

Skills: processos que viram comando

Skill é um processo que você repete, escrito uma vez só. Em vez de explicar toda vez "pesquisa o tema, faz 3 ganchos, escreve o roteiro, revisa", você guarda isso numa skill e chama com `/nome-da-skill`.

Onde ela fica

Pasta	Vale para
<code>~/ .claude/skills/nome-da-skill/SKILL.md</code>	Todos os seus projetos
<code>.claude/skills/nome-da-skill/SKILL.md</code>	Só aquele projeto

Como é por dentro

EXEMPLO DE SKILL.MD

```
---
name: roteiro-reel
description: Escreve roteiro de reel a partir de uma ideia.
  Use quando eu pedir um reel ou roteiro.
---

1. Me pergunte a ideia e o objetivo do vídeo.
2. Escreva 3 opções de gancho diferentes.
3. Escreva o roteiro no gancho que eu escolher, com frases curtas.
4. Revise tirando tudo que soa como texto de IA.
```

A linha mais importante é a description. É ela que o Claude lê para decidir sozinho quando usar a skill, então comece pelo "quando usar".

Deixe o Claude criar por você

COLE NO CLAUDE

Quero transformar um processo que eu repito numa skill do Claude Code.
Me entrevista primeiro: quando eu uso, o que eu te passo,
quais são as etapas, o que você precisa conferir e o que eu recebo no final.
Depois cria a skill com o nome [NOME] na pasta certa.

Troque `[NOME]` pelo nome da sua skill. Comece por um processo pequeno que você faz toda semana.

04

CAMADA 4

Sub-agentes: especialistas

Sub-agente é um assistente especializado que o Claude chama para uma tarefa específica. Ele trabalha com o próprio contexto, as próprias ferramentas e o próprio jeito, e devolve só o resultado para a conversa principal.

Por que isso importa: uma pesquisa grande ou uma revisão pesada enche a conversa de informação. Com o sub-agente, essa bagunça fica com ele e a conversa principal continua limpa e focada.

Onde ele fica

Pasta	Vale para
<code>~/ .claude/agents/</code>	Todos os seus projetos
<code>.claude/agents/</code>	Só aquele projeto

Como é por dentro

EXEMPLO: REVISOR-DE-COPY.MD

name: revisor-de-copy

description: Revisa legendas e roteiros antes de publicar.

Use sempre que eu terminar um texto.

tools: Read, Grep

Você é um revisor de copy para Instagram.

Aponte frases que soam como IA, promessas exageradas e erros de português.

Sugira a correção, não reescreva o texto inteiro.

Só `name` e `description` são obrigatórios. A `description` diz ao Claude quando delegar, e a linha `tools` limita o que o especialista pode fazer.

Deixe o Claude criar por você

COLE NO CLAUDE

```
Cria um sub-agente chamado revisor-de-copy em .claude/agents/.  
Ele revisa legendas e roteiros antes de eu publicar,  
aponta o que soa artificial e sugere correções.  
Ele só pode ler arquivos, não pode editar.
```

Para chamar, peça pelo nome ("usa o revisor-de-copy nesse texto") ou deixe o Claude decidir sozinho pela descrição. Comece com um especialista só: dois ou três bem definidos valem mais que dez genéricos.

05

CAMADA 5

Rotinas: ele trabalha sozinho

Até aqui, o agente só trabalha quando você abre o terminal. A última camada muda isso: você deixa tarefas marcadas para rodar sozinhas.

Tarefas agendadas

Dentro do Claude Code, digite `/schedule` e descreva o que quer, em português mesmo. O Claude faz as perguntas que faltam e salva a tarefa.

COLE NO CLAUDE

```
/schedule toda segunda às 8h, lê as anotações da semana  
e me monta o planejamento de conteúdo
```

Essas tarefas rodam na nuvem da Anthropic, então funcionam com o computador desligado. Você também cria e acompanha tudo em claude.ai/code/routines.

Atenção: a tarefa na nuvem só enxerga os repositórios do GitHub e os conectores (Google Drive, Notion, Slack e afins) que você liberar para ela. Se as anotações estão só no seu computador, use a opção do app de desktop.

Opção	Onde roda	Bom para
<code>/schedule</code>	Na nuvem	Tarefas fixas: diária, semanal ou uma vez numa data
Agendadas no app de desktop	No seu computador	Tarefas que precisam dos arquivos da sua máquina
<code>/loop</code>	Na sessão aberta	Repetir algo enquanto você trabalha

As tarefas agendadas na nuvem precisam de um plano Pro, Max, Team ou Enterprise, ainda estão em fase de testes e consomem o mesmo limite de uso das suas conversas. Escreva o pedido completo, porque ninguém vai estar ali para responder dúvida.

Bônus: hooks

Um hook é um gatilho fixo: toda vez que algo acontece, um comando roda. Por exemplo, avisar quando o Claude termina uma tarefa ou impedir que ele mexa numa pasta. É mais técnico, então deixe para quando as outras quatro camadas estiverem redondas. O passo a passo está no [guia oficial de hooks](#).

Checklist: monte na ordem

- ☐ Instalei o Claude Code e rodei `claude --version`
- ☐ Criei meu `CLAUDE.md` com o prompt da entrevista ou com `/init`
- ☐ Abri o `/memory` e conferi que a memória automática está ligada
- ☐ Transformei um processo que repito toda semana na minha primeira skill
- ☐ Criei um sub-agente para revisar ou pesquisar
- ☐ Agendei uma tarefa com `/schedule`

Não tente fazer as cinco no mesmo dia. Monte o contexto hoje, use por uma semana e só então passe para a próxima camada. Cada uma fica melhor quando a anterior já está rodando.

Me conta até onde você chegou

Manda um print do seu primeiro CLAUDE.md ou da sua primeira skill na DM. Toda semana tem conteúdo novo de IA no perfil.

@luiggiwebdev

FONTES · DOCUMENTAÇÃO OFICIAL DO CLAUDE CODE

- [Quickstart: instalação](#)
- [CLAUDE.md e memória](#)
- [Skills](#) · [Sub-agentes](#) · [Tarefas agendadas](#)