

Skills do Claude Code: o que são, como criar e como usar para automatizar qualquer tarefa

Um comando. Um processo completo. Como skills transformam o Claude Code em um sistema de trabalho.

SKILLS CLAUDE CODE COMO USAR

RESPOSTA DIRETA

Skills no Claude Code são instruções reutilizáveis que ensinam o Claude a executar uma tarefa complexa com um comando simples. Em vez de descrever o processo toda vez que você precisa de um relatório semanal ou de uma revisão de copy, você cria uma skill uma vez e depois só roda `/nome-da-skill`. A IA segue o processo exato que você definiu.

A diferença entre um agente e uma skill no Claude Code é sutil mas importante: agentes definem um personagem ou especialista (quem o Claude é), skills definem um processo (o que o Claude faz, passo a passo). Na prática, você usa os dois juntos: um agente de copywriting com uma skill de revisão de copy produz resultados mais consistentes do que usar só um deles.

A diferença real entre skills, agentes e comandos no Claude Code

Agente: define quem o Claude é naquela interação. Persona, especialidade, tom, restrições. Exemplo: 'Você é um estrategista de marketing com 10 anos de experiência em SaaS B2B.'

Skill: define o que o Claude faz quando você chama o comando. Processo, formato de saída, etapas, verificações. Exemplo: quando você roda `/audit-copy`, o Claude executa 7 passos de análise e devolve um relatório no formato especificado.

Comando: a forma de acionar a skill. Você digita `/nome` e o processo começa. Sem precisar reescrever as instruções toda vez.

Na prática: você tem um agente de copywriter com 5 skills (audit-copy, rewrite-headline, adapt-for-email, generate-cta, check-tone). Cada skill é um comando que o agente sabe executar. O sistema é modular.

Como criar sua primeira skill em 15 minutos

Skills ficam em `~/claude/skills/` ou em `./claude/skills/` dentro do projeto.

Crie a pasta: `mkdir -p ~/claude/skills/minha-skill/`

Crie o arquivo da skill: `~/claude/skills/minha-skill/SKILL.md`

Escreva as instruções da skill:

```
``` # Skill: Revisão de Email
```

Esta skill revisa qualquer email profissional seguindo estes critérios:

### ## Processo

1. Leia o email completo 2. Identifique o objetivo central (informar, solicitar, confirmar, vender) 3. Avalie clareza do assunto: é específico? gera expectativa correta? 4. Analise o corpo: primeiro parágrafo captura atenção? CTA está claro? 5. Verifique tom: adequado para o destinatário e objetivo? 6. Identifique redundâncias e jargão desnecessário

### ## Formato de saída

- Score geral (1-10) - 3 pontos fortes - 3 pontos de melhoria com sugestão de correção - Versão revisada do assunto (se necessário) - Versão revisada do primeiro parágrafo (se necessário) ```

Configure o comando em `~/claude/settings.json` para que o `/audit-email` ative essa skill.

Pronto: da próxima vez que precisar revisar um email, você cola o texto e roda `/audit-email`.

## Skills que eu uso toda semana (exemplos reais)

Relatório semanal: skill que lê meu arquivo de notas da semana, identifica as decisões tomadas, os bloqueios e os próximos passos, e gera um relatório em formato específico para o time. Rodo toda sexta em 2 minutos.

Auditoria de copy: skill com 12 critérios de copy (clareza, especificidade, CTA, tom, tamanho, estrutura). Qualquer texto que vai para um canal externo passa por essa skill antes.

Briefing de conteúdo: dado um tema e um público, a skill gera um briefing completo com objetivo, ângulo, estrutura de tópicos, perguntas a responder e fontes sugeridas. Substitui 45 minutos de planejamento.

Checklist de reunião: antes de qualquer reunião importante, a skill lê o contexto (agenda, participantes, decisões anteriores) e gera um checklist de pontos a cobrir e perguntas a fazer.

Síntese de pesquisa: cola até 10 textos de fontes diferentes, a skill identifica convergências, divergências e lacunas. 5 minutos de resultado versus 2 horas de leitura manual.

## Como organizar skills para não virar bagunça

Skills globais (disponíveis em todos os projetos): `~/claude/skills/`

Skills de projeto (disponíveis só naquele projeto): `~/claude/skills/` dentro da pasta do projeto.

Convenção de nomenclatura recomendada: - `audit-[objeto]`: skills de revisão (audit-copy, audit-email, audit-code) - `generate-[objeto]`: skills de geração (generate-brief, generate-report) - `analyze-[objeto]`: skills de análise (analyze-data, analyze-feedback) - `adapt-[objeto]`: skills de adaptação (adapt-for-linkedin, adapt-for-email)

Documentação mínima em cada skill: uma linha de descrição no topo do arquivo dizendo o que a skill faz e quando usar. Você não vai lembrar de 20 skills em detalhes. A documentação economiza tempo quando você volta depois de 2 semanas.

Revisão periódica: skills que você não usa há 30 dias provavelmente podem ser removidas ou simplificadas. Menos skills bem calibradas é melhor do que muitas skills medíocres.

## Skills prontas vs criar do zero: o que faz sentido

Criar skills do zero tem valor quando você tem um processo muito específico para o seu contexto: terminologia do seu setor, formato que só você usa, critérios que vieram de anos de trabalho na sua área.

Para processos mais genéricos (revisão de copy, geração de briefing, análise de dados), começar com skills prontas e ajustar é mais eficiente do que construir do zero. O núcleo do processo já está validado. Você só personaliza para o seu contexto.

O Meu Time de IA inclui mais de 500 comandos prontos organizados por área, que funcionam como skills nativas do Claude Code. Instalação em 2 minutos, disponíveis imediatamente.

### DÚVIDAS FREQUENTES

## Perguntas e respostas

Quantas skills posso ter?	▼
Skills funcionam entre projetos diferentes?	▼
Como compartilhar skills com o time?	▼
Skills têm limite de tamanho?	▼
Dá para ter skills que usam outras skills?	▼