E-book · 148 páginas

22 receitas testadas no Windows 11

148



13 capítulos

22 receitas

Claude Code

guia de campo em português

Edição de outubro de 2026

Material independente, sem vínculo com a Anthropic. Claude e Claude Code são marcas da Anthropic

Como ler este guia

Este é um livro de consulta. Os capítulos 1 a 6 levam do primeiro pedido a uma instalação bem configurada. Do 7 ao 11 está o que faz o Claude Code trabalhar mais tempo sozinho: skills, agentes, hooks, conectores e continuidade. O 12 ensina a acompanhar as mudanças, e o 13 traz 22 receitas prontas para tarefas de escritório. Os apêndices foram feitos para consultar com o terminal aberto.

`v2.1.296+` Assim aparece a versão mínima de um recurso, sempre na margem, ao lado do parágrafo que o usa.

Fonte de tudo: code.claude.com/docs, 228 páginas baixadas em 15/10/2026, e o changelog oficial até a 2.1.301.

A versão de cada fato

O Claude Code muda toda semana: entre 24/09 e 15/10/2026 saíram dezoito versões, da 2.1.282 à 2.1.301. Por isso, na margem, cada recurso que depende de versão traz a versão mínima em que funciona, e cada seção termina com a página da documentação consultada e a data. Se o seu `claude --version` mostra um número menor, o recurso pode não existir na sua máquina.

Windows em primeiro lugar

Quando a sintaxe muda entre sistemas, o exemplo aparece em PowerShell, com a variação em bash ao lado. `~/.claude/` é a pasta `.claude` dentro do seu diretório de usuário; no Windows, `C:\Users\<você>\.claude\`.

As três caixas

Na prática
O pedido ou o comando como você digita, pronto para copiar.

Cuidado
O erro que custa caro e não avisa: campo ignorado, gasto alto, dado perdido.

Atalho
O caminho mais curto para o mesmo resultado.

A gramática dos diagramas

Todo diagrama do livro usa as mesmas formas. Azul cheio é sempre a sua conversa; contorno azul é um agente; a área tracejada é um contexto que você não vê; amarelo é o que volta para você. Aprendida aqui, a legenda vale do capítulo 1 ao 13.



Figura 0 A legenda de todos os diagramas.

Material independente. Este guia não é afiliado, patrocinado nem endossado pela Anthropic. Claude e Claude Code são marcas da Anthropic, citadas só para descrever o conteúdo. Os exemplos são genéricos e foram escritos para este livro.

Sumário

Treze capítulos e quatro apêndices, 148 páginas. A barra mostra o tamanho de cada capítulo. As receitas do capítulo 13 ocupam uma página cada: o pedido para copiar, o que esperar e como conferir.

Como ler este guia p. 2

O que este guia cobre e o que deixa de fora p. 5

Por onde começar: quatro caminhos pelo livro p. 6

1 Primeiros passos

p. 7 8 páginas

1.1 Instalar no Windows, no Mac e no Linux
1.2 Entrar: planos, Console e provedores de nuvem
1.3 Onde rodar: terminal, Desktop, IDE, web e celular
1.4 A primeira sessão, do pedido ao diff
1.5 Conferir versão e ambiente: --version, /status e /doctor

2 O jeito certo de pedir

p. 15 8 páginas

2.1 O laço do agente: ler, agir, conferir
2.2 O pedido em cinco partes
2.3 Planejar antes de mexer: o modo plan
2.4 Conferir o resultado sem confiar na palavra
2.5 Dez erros que fazem a IA errar

3 Permissões e segurança

p. 23 10 páginas

3.1 Os seis modos, de Manual a bypassPermissions
3.2 Regras allow, ask e deny, com exemplos
3.3 Auto mode: o que o classificador olha
3.4 Sandbox, caminhos protegidos e prompt injection
3.5 O que sai da sua máquina: retenção e telemetria

4 Configuração: settings.json em camadas

p. 33 8 páginas

4.1 Os níveis e quem ganha
4.2 Um settings.json de projeto, comentado
4.3 Variáveis de ambiente que valem a pena
4.4 /config, /update-config e /permissions

5 CLAUDE.md, regras e memória

p. 41 8 páginas

5.1 De onde vem cada CLAUDE.md
5.2 Escrever pouco e bem: menos de 200 linhas
5.3 Regras modulares com paths:
5.4 Auto memory e AGENTS.md
5.5 Auditar instruções antigas com /doctor prompt-audit

6 Contexto e custo

p. 49 10 páginas

6.1 De onde vem cada parte do contexto
6.2 /context, /compact, /clear, /rewind e /resume
6.3 Compactação automática e ociosa
6.4 Modelos, effort e cache de prompt
6.5 Ler o /usage e gastar menos sem perder qualidade

7 Skills

p. 59 8 páginas

7.1 Skill, comando ou CLAUDE.md?
7.2 A primeira skill: SKILL.md e frontmatter
7.3 As nativas: /run, /verify, /doctor, /update-config
7.4 Rodar uma skill num subagente
7.5 Testar e iterar uma skill

8 Agentes: subagentes e workflows — p. 67 · 12 páginas

8.1 O que um subagente faz e o que devolve
8.2 Nativos, seus e de plugin
8.3 Escrever um subagente
8.4 Chamar, acompanhar e retomar
8.5 Workflows: quando o plano vira script
8.6 Quando não usar

9 Hooks — p. 79 · 10 páginas

9.1 Os eventos numa linha do tempo
9.2 O primeiro hook: formatar depois de editar
9.3 Bloquear: exit code 2 e onFailure: 'block'
9.4 Hooks de prompt, de agente, HTTP e MCP
9.5 Depurar um hook que não dispara

10 MCP, conectores, plugins e mods — p. 89 · 10 páginas

10.1 MCP em cinco minutos: claude mcp add e escopos
10.2 Conectores do claude.ai e OAuth
10.3 Plugins: instalar, criar e publicar
10.4 Mods: painéis, comandos e interceptação
10.5 Marketplace da Anthropic e de terceiros
Quando não usar

11 Continuidade e trabalho em paralelo — p. 99 · 8 páginas

11.1 /goal: até a condição ser cumprida
11.2 /loop, tarefas agendadas e routines
11.3 Worktrees: várias tarefas sem conflito
11.4 Sessões em segundo plano e agent view
Quando não usar

12 Manter atualizado — p. 107 · 6 páginas

12.1 Atualização automática: latest e stable
12.2 Homebrew e WinGet não se atualizam sozinhos
12.3 Changelog e /release-notes
12.4 O que mudou da 2.1.282 à 2.1.301
Armadilhas

13 Na prática: 22 receitas de escritório — p. 113 · 24 páginas

Antes da primeira receita e armadilhas
13.1 Planilhas, receitas 1 a 5
13.2 Documentos e relatórios, 6 a 9
13.3 PDFs e arquivos, 10 a 13
13.4 Pastas, 14 a 16
13.5 Fotos e recibos, 17 a 19
13.6 E-mail e texto, 20 a 22

A Apêndices — p. 137 · 12 páginas

A Atalhos de teclado
B Cola de comandos e flags
C Solução de problemas
D Índice remissivo

Receitas do capítulo 13, uma por página: orçamento em Excel a partir de uma lista; conferir uma planilha e apontar os erros; tabela de preços com margem e reajuste; controle de estoque com alerta de mínimo; conciliar o extrato com os recebimentos; e-mails e anotações viram relatório em Word; relatório de visita a partir de anotações e fotos; contrato a partir do seu modelo; resumo do mês em uma página; renomear PDFs pelo conteúdo; juntar PDFs na ordem certa; separar um PDF escaneado por documento; tirar a tabela de um PDF para o Excel; organizar a pasta Downloads; achar duplicados sem apagar nada; arquivar por cliente e por mês; fotos de recibos viram planilha de despesas; fotos renomeadas pela data em que foram tiradas; reduzir fotos para e-mail; mensagens de cobrança a partir dos vencidos; responder pedido de orçamento com a sua tabela; ata de reunião a partir de anotações soltas.

O que este guia cobre e o que deixa de fora

O assunto é o Claude Code como ferramenta de trabalho: a CLI no terminal e as telas que usam o mesmo motor, o app Desktop, as extensões de editor, a web e o celular. O livro cobre o que muda o resultado do dia a dia: como pedir, o que o Claude pode fazer sem perguntar, onde mora cada configuração, quanto contexto e cota cada escolha consome, e como fazer o trabalho seguir sozinho com skills, agentes, hooks, conectores e /goal.

Para quem é. Para quem usa ou vai usar o Claude Code no trabalho e prefere ler em português. Não é preciso saber programar para as receitas do capítulo 13; os capítulos 9 e 10 pedem alguma intimidade com JSON e com a linha de comando.

O que está dentro

- Instalação no Windows, com as diferenças do Mac e do Linux nas notas (capítulo 1).
- Permissões, configuração em camadas, CLAUDE.md e memória (capítulos 3 a 5).
- Contexto e custo: o que enche a janela, o que gasta a cota e como ler o `/usage` (capítulo 6).
- Skills, subagentes, workflows, hooks, MCP, plugins e mods, cada um com exemplo que se copia (capítulos 7 a 10).
- Continuidade e paralelo: `/goal`, `/loop`, tarefas agendadas e worktrees (capítulo 11).
- Como acompanhar as versões, e o que mudou da 2.1.282 à 2.1.301 (capítulo 12).
- 22 receitas de escritório, uma por página, com o pedido, o que esperar e como conferir (capítulo 13).

O que fica de fora

- **Programar com o Agent SDK ou chamar a API diretamente.** O SDK aparece só onde muda o comportamento do Claude Code; construir aplicações com ele é outro livro.
- **O chat do claude.ai.** Ele entra apenas quando um recurso é compartilhado, como os conectores.
- **Preço de plano em reais.** Os valores mudam e dependem de câmbio e imposto; o livro ensina a ler o próprio uso, não a escolher plano pelo preço.
- **Ranking de modelos.** O capítulo 6 diz o que o modelo e o effort mudam no custo e no cache, sem eleger um vencedor.

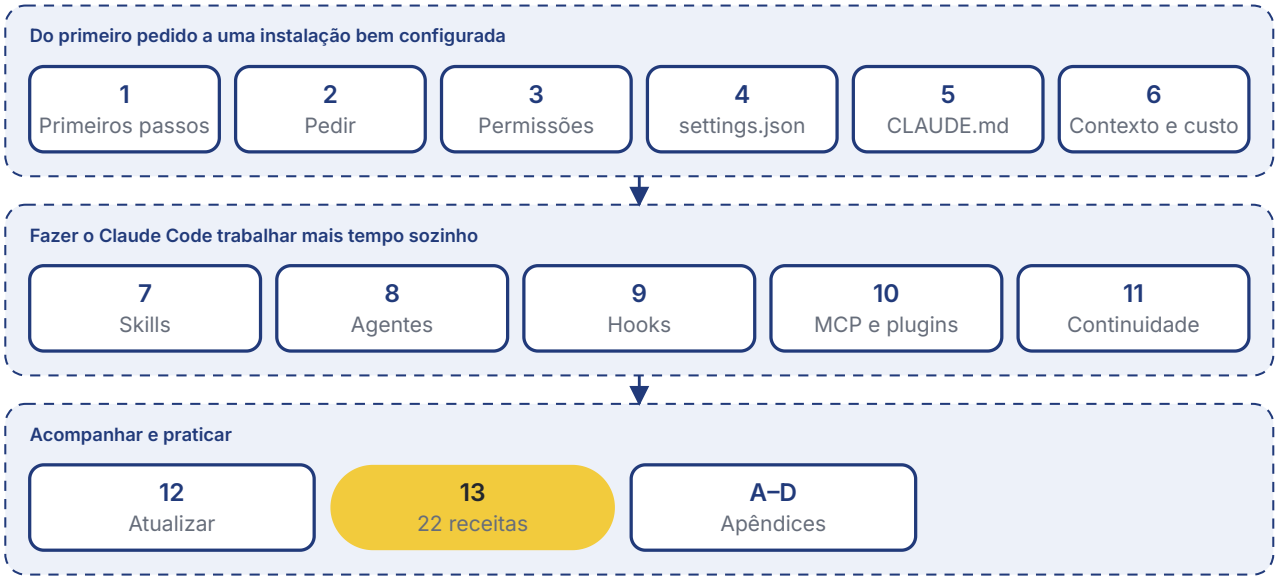
Edição de outubro de 2026. Tudo foi conferido na documentação oficial baixada em 15/10/2026, com o Claude Code 2.1.301. Recurso lançado depois disso não está aqui; o capítulo 12 mostra como descobrir o que mudou.

> ⚠ **Cuidado**
>
> Comando de exemplo não substitui ler o que ele faz na sua máquina. Toda receita que renomeia ou move arquivo mostra antes a tabela do que vai fazer e para: confira antes de liberar. Nenhuma apaga arquivo nem envia nada.

Por onde começar: quatro caminhos pelo livro

Ninguém precisa ler as 148 páginas na ordem. Escolha a linha que descreve você hoje e siga os capítulos na ordem indicada; o resto do livro fica para consulta. Cada capítulo abre com uma página que diz a versão mínima, a documentação conferida e o que convém ler antes.

Se você...	Leia	Para quê
Nunca usei o Claude Code	**1 → 2 → 3**	Instalar, fazer o primeiro pedido e entender o que ele pode fazer sem perguntar.
Quero resolver tarefas de escritório	**1 → 13 → C**	Instalar, ir direto às 22 receitas e usar o apêndice C quando algo travar.
Já uso e quero gastar menos e errar menos	**4 → 5 → 6**	Configuração em camadas, CLAUDE.md enxuto e o que enche a janela e a cota.
Quero que ele trabalhe mais tempo sozinho	**7 → 8 → 9 → 11**	Skills, agentes, hooks e continuidade com /goal e worktrees.



Mapa do livro Os capítulos em três blocos. As 22 receitas, em amarelo, são o que volta para você: cada uma vira um arquivo pronto.

Como achar um assunto

- **Pelo nome do comando ou da opção:** o índice remissivo, no apêndice D (p. 148).
- **Pelo sintoma** (não instala, não entra, hook que não dispara, campo ignorado): a solução de problemas, no apêndice C (p. 144).
- **Pela tecla:** os atalhos, no apêndice A (p. 138); comandos e flags de uma linha, no apêndice B (p. 141).

Os capítulos conversam entre si

Quando um assunto depende de outro, o texto aponta o capítulo, como em "(cap. 6)". No PDF, os títulos do sumário e os marcadores na lateral do leitor levam direto à página.

Antes de pedir algo que apaga ou envia, leia o capítulo 3. Ele é curto e é onde se decide o que o Claude faz sem perguntar.

Capítulo 1

Primeiros passos

Antes do primeiro pedido há três escolhas: como instalar, com que conta entrar e em que tela trabalhar. Este capítulo instala o Claude Code no Windows, com as diferenças do Mac e do Linux nas notas, faz o login, compara o terminal, o app Desktop, as extensões de editor, a web e o celular, conduz a primeira sessão do pedido ao diff e termina com os comandos que dizem se a instalação está em ordem.

Neste capítulo

1.1	Instalar no Windows, no Mac e no Linux	8
1.2	Entrar: planos, Console e provedores de nuvem	10
1.3	Onde rodar: terminal, Desktop, IDE, web e celular	11
1.4	A primeira sessão, do pedido ao diff	12
1.5	Conferir versão e ambiente: --version, /status e /doctor	14

Versão mínima. A instalação vale para qualquer versão atual. O modo auto como modo inicial em todos os planos chegou na 2.1.284 (Pro, Max e Team já o tinham), e o painel de diff ao lado da conversa pede a 2.1.287.

Conferido em code.claude.com/docs/en/setup, /en/authentication, /en/platforms, /en/quickstart e /en/troubleshoot-install, páginas baixadas em 15/10/2026. Os comandos de verificação foram rodados no Windows 11, PowerShell 5.1, com o Claude Code 2.1.301.

Antes, leia nada: este é o ponto de partida. As permissões que aparecem na primeira sessão são o assunto do capítulo 3.

1.1 Instalar no Windows, no Mac e no Linux

O Claude Code é um programa de linha de comando distribuído como um único executável, sem depender de Node.js nem de outro runtime. Roda no Windows 10 versão 1809 ou mais novo e no Windows Server 2019, no macOS 13 ou mais novo, no Ubuntu 20.04+, no Debian 10+ e no Alpine 3.19+, com 4 GB de memória e processador x64 ou ARM64. Precisa de internet, porque o modelo roda nos servidores da Anthropic, e de estar num dos países atendidos pela empresa.

No CMD o comando é outro: `curl -fsSL https://claude.ai/install.cmd -o install.cmd && install.cmd && del install.cmd`.

Mac, Linux e WSL: `curl -fsSL https://claude.ai/install.sh | bash` no terminal. O executável vai para `~/.local/bin/`.

Prefere não usar terminal? O app Desktop já traz o Claude Code (seção 1.3).

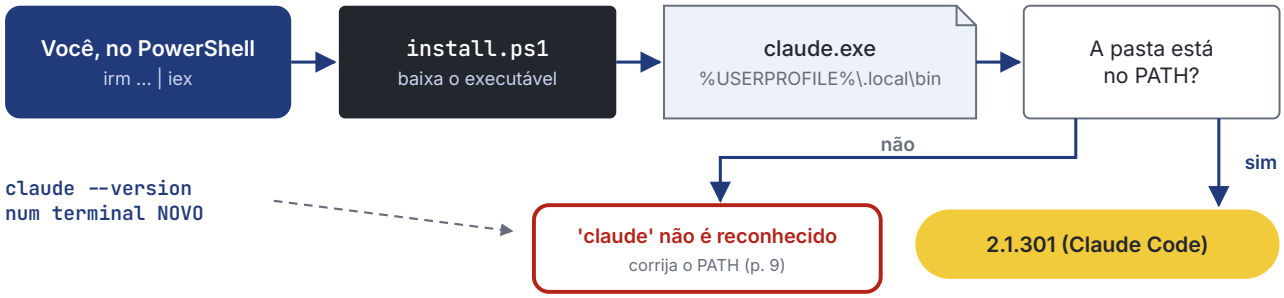
Windows: o instalador nativo

Abra o PowerShell pelo menu Iniciar (o prompt começa com `PS C:\`; sem o `PS`, é o CMD) e cole:

PowerShell

```
irm https://claude.ai/install.ps1 | iex
```

Não precisa ser administrador. O comando baixa o executável sem mostrar progresso e o deixa em `%USERPROFILE%\.local\bin\claude.exe`. Ao terminar, abra uma janela nova do terminal e digite `claude --version`: a resposta é o número da versão seguido de `(Claude Code)`.



A janela onde você rodou o instalador guarda o PATH antigo: teste sempre numa janela aberta depois.
O instalador avisa em "Setup notes:" quando a pasta não está no PATH, mas não muda o PATH por você.

Figura 1.1 O caminho do instalador nativo. Quase todo "não funcionou" depois de instalar é o último degrau: a pasta fora do PATH.

Git for Windows: opcional, mas recomendado

Sem ele, o Claude Code roda os comandos do sistema pela ferramenta PowerShell. Com ele, ganha também a ferramenta Bash, que usa o Git Bash, e o Monitor, que acompanha processos em segundo plano. Instale-o (git-scm.com) com **Add to PATH** marcado. Se ele estiver fora da pasta padrão, aponte o caminho no `settings.json` (capítulo 4):

Onde ele procura. Em `C:\Program Files\Git` e `C:\Program Files (x86)\Git`, depois no `git` do PATH. O arquivo apontado tem de se chamar `bash.exe` ou `sh.exe`; `git-bash.exe` é ignorado.

WSL 2. Instale dentro da distribuição, com o comando do Linux, e rode `claude` no terminal do WSL. É o caminho para ter o sandbox de comandos, que não existe no Windows nativo (capítulo 3).

Arquivo `~\.claude\settings.json`

```
{
  "env": {
    "CLAUDE_CODE_GIT_BASH_PATH": "C:\\Program Files\\Git\\bin\\bash.exe"
  }
}
```

Fonte code.claude.com/docs/en/setup, "System requirements", "Install Claude Code" e "Set up on Windows". Página consultada em 15/10/2026, Claude Code 2.1.301.

Quando o terminal diz que `claude` não existe

A mensagem `'claude' is not recognized` quase sempre quer dizer que a pasta do executável não está no PATH, a lista de pastas onde o Windows procura programas. Confira em duas linhas:

PowerShell

```
Test-Path "$env:USERPROFILE\.local\bin\claude.exe"
$env:PATH -split ';' | Select-String '\.local\\bin'
```

A primeira deve responder `True`; a segunda, imprimir `C:\Users\<você>\.local\bin`. Se a segunda não imprimir nada, acrescente a pasta ao PATH do seu usuário, o que vale para todas as janelas futuras, e abra um terminal novo:

PowerShell

```
$atual = [Environment]::GetEnvironmentVariable('PATH', 'User')
[Environment]::SetEnvironmentVariable('PATH',
  "$atual;$env:USERPROFILE\.local\bin", 'User')
```

Testado em 15/10/2026 no PowerShell 5.1: as duas linhas de conferência respondem como descrito.

Terminal de editor (VS Code, Cursor) herda o PATH do editor: feche e reabra o editor inteiro, não só o terminal.

Mac e Linux: `echo 'export PATH="$HOME/.local/bin:$PATH"' >> ~/.zshrc` e depois `source ~/.zshrc`; no bash do Linux, `~/.bashrc`; no do Mac, `~/.bash_profile`.

Outros jeitos de instalar

Método	Comando	Atualiza sozinho?
Nativo	`irm https://claude.ai/install.ps1 | iex`	Sim, em segundo plano
WinGet	`winget install Anthropic.ClaudeCode`	Não, por padrão: `winget upgrade Anthropic.ClaudeCode`
Homebrew	`brew install --cask claude-code`	Não, por padrão: `brew upgrade claude-code`
npm	`npm install -g @anthropic-ai/claude-code`	Sim, se a pasta global do npm aceitar gravação
apt, dnf, apk	Repositórios assinados da Anthropic	Não: pelo gerenciador do sistema

O nativo é o recomendado. Para o canal stable, cerca de uma semana atrás e sem as versões com regressão grave: `& ([scriptblock]::Create((irm https://claude.ai/install.ps1))) stable`.

npm pede Node.js 22 ou mais novo, mas instala o mesmo executável. Nunca com `sudo`.

Atualizações e canais: capítulo 12.

⚠ Cinco erros de instalação no Windows

- **`'irm' is not recognized`:** você está no CMD. Abra o PowerShell ou use o comando do CMD da página anterior.
- **`The token '&&' is not a valid statement separator`:** o comando do CMD foi colado no PowerShell.
- **`A parameter cannot be found that matches parameter name 'fsSL'`:** o comando do Mac foi colado no PowerShell, onde `curl` é apelido de outro comando.
- **"does not support 32-bit Windows" num PC de 64 bits:** você abriu o "Windows PowerShell (x86)". Abra o que não tem x86 no nome.
- **`claude` abre o app Desktop:** uma versão antiga do Desktop registrou um `Claude.exe` que passa na frente da CLI. Atualize o Desktop.

Desinstalar o nativo: apague `%USERPROFILE%\.local\bin\claude.exe` e a pasta `%USERPROFILE%\.local\share\claude`. Configurações e histórico ficam em `~\.claude\` e `~\.claude.json`; só apague se quiser perder tudo.

Fonte code.claude.com/docs/en/troubleshoot-install e code.claude.com/docs/en/setup, "Verify your PATH", "Wrong install command on Windows", "Install with npm" e "Uninstall Claude Code". Página consultada em 15/10/2026, Claude Code 2.1.301.

1.2 Entrar: planos, Console e provedores de nuvem

O Claude Code não tem conta própria: ele usa uma destas. O plano gratuito do claude.ai não inclui o Claude Code.

Conta	Quem paga	Como entra
Pro ou Max	Sua assinatura do claude.ai, dentro dos limites de uso do plano	Login no navegador
Team ou Enterprise	A organização; o administrador convida você	Login com a conta convidada
Console	A conta da API, cobrada por token	Login, opção Console
Bedrock, Agent Platform, Foundry	A conta de nuvem da empresa (Amazon, Google Cloud, Microsoft)	Variáveis de ambiente, sem navegador

Na primeira vez que você roda `claude`, o navegador abre para o login. Concluído, o terminal mostra `Login successful` e pede Enter. A credencial fica guardada e o login não se repete; para trocar de conta, digite `/login` dentro da sessão, e para sair, `/logout`.

O navegador não abriu? Tecle **c** para copiar o endereço e cole-o no navegador.

WSL, SSH ou contêiner: se o navegador mostrar um código em vez de voltar ao terminal, cole-o em `Paste code here if prompted`.

Onde fica: no Windows, em `%USERPROFILE%\.claude\.credentials.json`, protegido pelas permissões da sua pasta de usuário; no Mac, no Keychain.

`v2.1.242+` Console sem criar chave de API: "Sign in with your Console account".

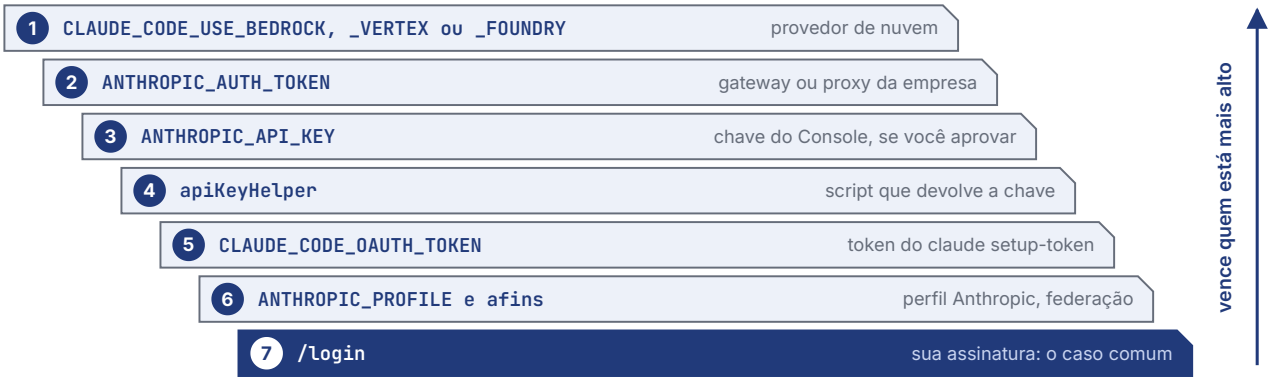


Figura 1.2 Com várias credenciais na máquina, vale a mais alta. A assinatura do `/login` é o último degrau: qualquer credencial acima dela, presente (e, no caso da chave, aprovada), ganha.

O degrau que mais engana é o terceiro. Com uma `ANTHROPIC_API_KEY` no ambiente, o Claude Code pergunta uma vez se deve usá-la; aprovada, ela passa na frente da assinatura, e o uso vai para a conta da API. Em `claude -p` não há pergunta: a chave presente é usada.

> ⚠ **Cuidado**
>
> Assinou o Max e apareceu cobrança no Console, ou erro de organização desativada? Procure uma chave esquecida:
>
> ```
> Get-ChildItem Env:ANTHROPIC_*      # lista as variáveis do terminal
> Remove-Item Env:ANTHROPIC_API_KEY  # tira só desta janela
> ```
>
> Se ela vier das variáveis do sistema, tire-a lá também. Depois, `/status` mostra qual credencial está em uso e marca a que ficou de fora.

Para scripts e CI, `claude setup-token` gera um token de um ano da assinatura, para pôr em `CLAUDE_CODE_OAUTH_TOKEN`. Ele só faz pedidos ao modelo.

`v2.1.210+` `/status` mostra a linha Login como vencida antes de o primeiro pedido falhar.

Fonte code.claude.com/docs/en/authentication, "Log in to Claude Code", "Credential management" e "Authentication precedence". Página consultada em 15/10/2026, Claude Code 2.1.301.

Duas contas na mesma máquina

Para manter a conta do trabalho e a pessoal logadas ao mesmo tempo, dê a cada uma a sua pasta de configuração pela variável `CLAUDE_CONFIG_DIR`, com settings, histórico e login próprios. Esta função cria o comando `claude-trabalho`:

PowerShell, no perfil (`notepad $PROFILE`)

```
function claude-trabalho {
  $antes = $env:CLAUDE_CONFIG_DIR
  $env:CLAUDE_CONFIG_DIR = "$HOME\.claude-trabalho"
  try { claude @args } finally { $env:CLAUDE_CONFIG_DIR = $antes }
}
```

Testado em 15/10/2026: numa pasta nova, `claude-trabalho auth status` responde "não logado" e aponta a pasta certa.

Perfil não carrega? O PowerShell bloqueia scripts por padrão: `Set-ExecutionPolicy -Scope CurrentUser RemoteSigned`.

Mac e Linux: `alias claude-work='CLAUDE_CONFIG_DIR=~/.claude-work claude'`.

Fonte code.claude.com/docs/en/authentication, "Log in with multiple accounts" e "Generate a long-lived token". Página consultada em 15/10/2026, Claude Code 2.1.301.

1.3 Onde rodar: terminal, Desktop, IDE, web e celular

O motor é o mesmo em toda parte; muda onde o trabalho acontece e o que ele enxerga.

Onde	Para quê	O que precisa
Terminal	Tudo; o único com `claude -p` para scripts e com o Agent SDK	A instalação da seção 1.1
Desktop	Revisar diffs na tela, várias sessões lado a lado	O app Claude, aba **Code**; o Claude Code vem dentro
VS Code	Trabalhar sem sair do editor, com diff no próprio arquivo	Extensão "Claude Code", VS Code 1.94 ou mais novo
JetBrains	IntelliJ, PyCharm, WebStorm e afins	Plugin do JetBrains Marketplace e a CLI instalada
Web	Tarefa longa que continua com o computador desligado	claude.ai/code e um repositório no GitHub (ou um local, enviado com `claude --cloud`)
Celular	Lançar e acompanhar tarefas de longe	App Claude (iOS, Android), conta claude.ai

Desktop no Windows: instaladores x64 e ARM64. No Linux, versão beta por apt.

A extensão do VS Code traz uma cópia própria da CLI, que não entra no PATH. Para usar `claude` no terminal do editor, faça também a instalação da seção 1.1.

Cursor e outros derivados do VS Code instalam a mesma extensão.

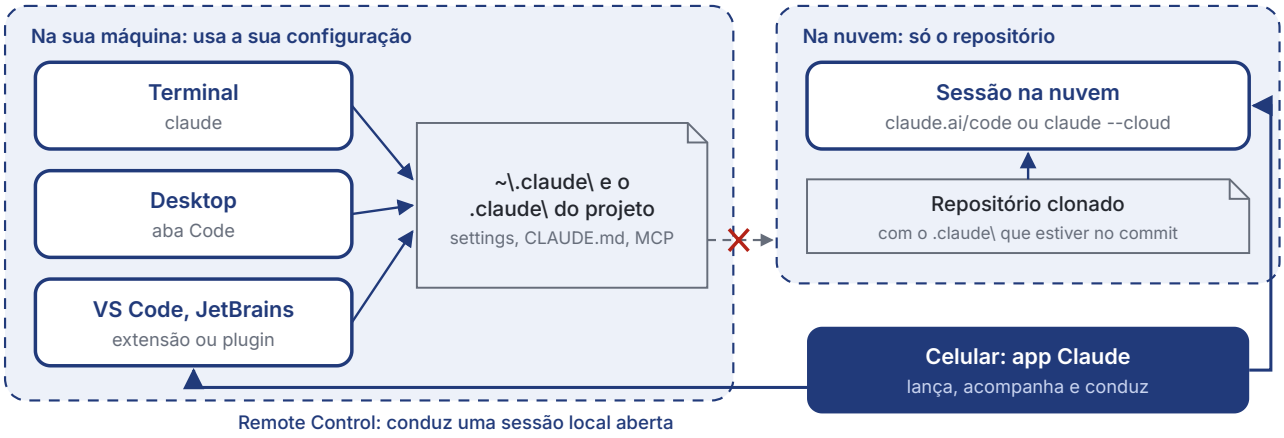


Figura 1.3 As telas locais dividem a mesma configuração: o que você ajusta no terminal vale no Desktop e no editor. A sessão na nuvem recebe só o repositório: o que está apenas na sua máquina não vai. O celular é controle remoto, não roda código.

Fonte code.claude.com/docs/en/platforms, "Where to run Claude Code"; /en/desktop-quickstart, /en/vs-code e /en/jetbrains, "Install". Página consultada em 15/10/2026, Claude Code 2.1.301.

Dá para misturar telas no mesmo projeto. Uma conversa começada no terminal continua no Desktop com `/desktop`; uma sessão local pode ser conduzida do celular ou do navegador com `/remote-control`, enquanto o computador fica ligado; e `claude --cloud "tarefa"` cria uma sessão nova na nuvem, que segue mesmo que você desligue a máquina; a conversa do terminal não vai junto.

Nuvem e celular pedem conta claude.ai: chave do Console e provedores de nuvem não chegam lá. A web está nos planos Pro, Max e Team, e no Enterprise com o assento certo.

`/desktop` pede Mac ou Windows x64 e assinatura.

Fonte code.claude.com/docs/en/platforms e /en/web-quickstart, "Where to run Claude Code", "Work when you are away from your terminal" e "Compare ways to run Claude Code". Página consultada em 15/10/2026, Claude Code 2.1.301.

1.4 A primeira sessão, do pedido ao diff

Abra o PowerShell, entre na pasta do projeto e inicie:

PowerShell

```
cd C:\projetos\meu-app
claude
```

A pasta certa importa. Abrir o Claude Code na sua pasta de usuário põe todos os seus arquivos ao alcance dele. Abra sempre na raiz do projeto.

Sem projeto à mão? Uma pasta com um único script já serve para o primeiro teste.

Na primeira vez numa pasta, o Claude Code pergunta se você confia nos arquivos dela, porque as configurações do projeto podem liberar ferramentas. Aceito, aparecem a versão, o modelo e a pasta de trabalho acima do prompt. O Claude trabalha nessa pasta e nas subpastas e lê os arquivos à medida que precisa: você não anexa nada. Comece perguntando, sem pedir mudança:

```
o que este projeto faz? onde fica o ponto de entrada e como eu rodo
os testes?
```

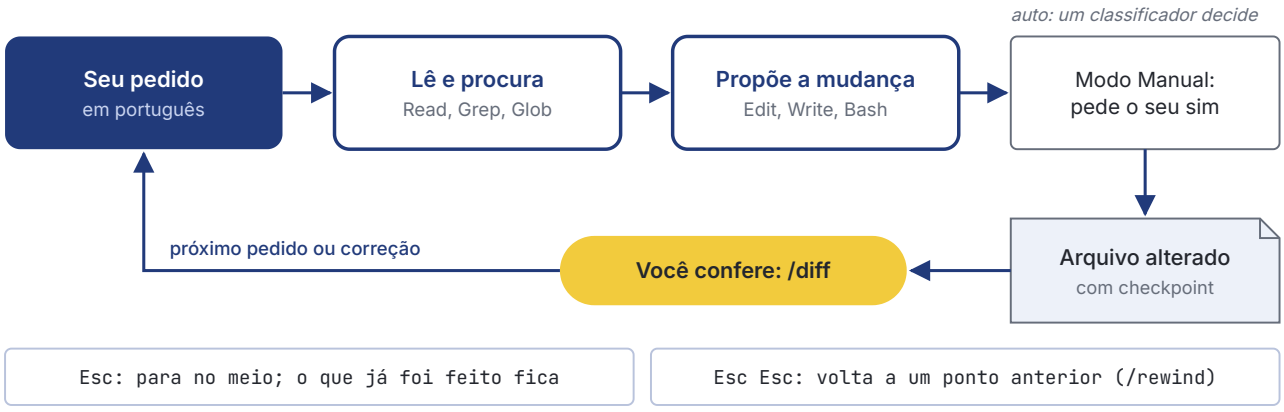


Figura 1.4 O laço de cada pedido. Você entra com o pedido e sai com o diff; no meio, o Claude lê, propõe e, conforme o modo, pede licença.

Fonte code.claude.com/docs/en/quickstart, "Start your first session" e "Ask your first question"; /en/permissions, "Project allow rules and workspace trust". Página consultada em 15/10/2026, Claude Code 2.1.301.

O modo em que a sessão começa

Salvo configuração em contrário, uma sessão de terminal começa em modo **auto**: o Claude edita arquivos e roda comandos sem pedir licença a cada passo, e um classificador em segundo plano barra o que parece arriscado. Se o modelo não aceitar o auto ou a organização o tiver desligado, a sessão começa em Manual. A barra de status mostra `⏵⏵ auto mode on`. Para acompanhar cada passo nas primeiras sessões, aperte **Shift+Tab** uma vez: a barra passa a `⏸ manual mode on`, e toda edição e todo comando que não seja só leitura esperam o seu sim.

`v2.1.284+` auto como modo inicial em todo plano, no terminal e no VS Code (a doc diz 2.1.283; o changelog fecha a mudança na 2.1.284). Pro, Max e Team já começavam em auto antes.

O ciclo do Shift+Tab: auto, Manual, acceptEdits, plan e de volta. No Windows, **Alt+M** faz o mesmo quando o Shift+Tab não responde.

Modo padrão fixo no `settings.json`: capítulo 3.

`v2.1.287+` com a renderização em tela cheia, num repositório git e num terminal de 110 colunas ou mais, o `/diff` abre um painel ao lado da conversa.

Na prática

Para já começar em Manual, abra a sessão assim no PowerShell e então faça o primeiro pedido de mudança:

PowerShell

```
claude --permission-mode manual
```

```
acrescente uma validação que recuse e-mail vazio no formulário
de cadastro
```

O Claude acha o arquivo, mostra a mudança e pergunta. **Yes** aprova esta; a opção seguinte libera as edições até o fim da sessão; **No**, ou Esc, recusa e deixa você dizer o que quer diferente. Depois, `/diff` mostra o que mudou no repositório git, com uma vista por pedido do que o Claude editou.

Tecla ou comando	O que faz
`Esc`	Para a resposta no meio; o que já foi feito fica, e você corrige o rumo
`Esc Esc`	Com o prompt vazio, abre o rewind: volta a conversa, o código ou os dois a um ponto anterior
`! comando`	Roda um comando do sistema e põe a saída na conversa
`/exit`	Sai; Ctrl+D duas vezes faz o mesmo
`claude -c`	Volta à última conversa desta pasta; `claude -r` deixa escolher qual

Pergunta de uma linha, sem abrir sessão: `claude -p "O que este projeto faz? Uma frase."` responde e sai. Testado em 15/10/2026.

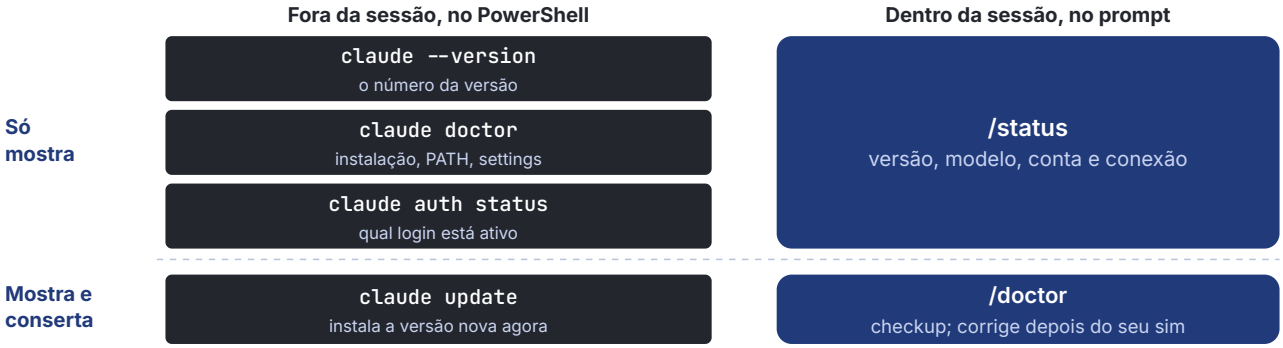
Cuidado

O rewind desfaz o que o Claude editou com as ferramentas de edição, mas não o que um comando apagou ou moveu (`rm`, `mv`, um script) nem o que mudou fora do Claude Code. Antes de uma mudança grande, faça um commit: "faça commit das minhas mudanças com uma mensagem descritiva" já resolve.

Fonte code.claude.com/docs/en/permission-modes e code.claude.com/docs/en/interactive-mode, "Available modes", "Which mode a session starts in", "Keyboard shortcuts" e "Review changes with /diff"; /en/checkpointing, "Limitations". Página consultada em 15/10/2026, Claude Code 2.1.301.

1.5 Conferir versão e ambiente: --version, /status e /doctor

Seis comandos dizem o que está instalado, com que conta e o que está fora do lugar.



Os da esquerda funcionam mesmo quando a sessão não abre. Comece por eles.

Figura 1.5 Os comandos de diagnóstico. `claude doctor` só lê; `/doctor` propõe correções e só aplica depois que você confirma.

Saída típica, resumida

```
PS> claude doctor
Running: native (2.1.301)
Path: C:\Users\voce\.local\bin\claude.exe
Config install method: native
Search: OK (bundled)
Auto-updates: enabled
No installation issues found.
```

Testado em 15/10/2026: `claude doctor` e `claude auth status --text` no PowerShell 5.1, na 2.1.301.

Sessão que nem abre? `claude --safe-mode` inicia sem CLAUDE.md, skills, plugins, hooks e MCP: se abrir assim, o defeito está numa personalização.

`/doctor` também audita instruções antigas: capítulo 5.

`Path` diz qual executável respondeu; `Search`, se a busca embutida funciona. `claude auth status --text` mostra login e e-mail e sai com código 1 se ninguém estiver logado. O `/doctor` vai além do `/status`: olha PATH, settings, extensões e CLAUDE.md.

Próximo capítulo. O jeito certo de pedir: o laço do agente e o pedido em cinco partes.

⚠ **Armadilhas**

- **WinGet e Homebrew, por padrão, não se atualizam.** O `claude doctor` mostra a última tentativa.
- **Duas instalações.** Nativa e npm ao mesmo tempo: roda a que vem primeiro no PATH. `where.exe claude` lista todas.

Para levar deste capítulo

- No Windows: `irm https://claude.ai/install.ps1 | iex` e um terminal novo. "Não reconhecido" é PATH.
- Uma `ANTHROPIC_API_KEY` no ambiente passa na frente da assinatura. `/status` mostra qual vale.
- A sessão de terminal começa em auto. Para ver cada passo, Shift+Tab até Manual.
- Fora da sessão, `claude doctor`; dentro, `/status` e `/doctor`.

Fonte code.claude.com/docs/en/cli-reference e code.claude.com/docs/en/commands, "CLI commands" (claude doctor, claude auth status, --safe-mode) e "All commands" (/status, /doctor); /en/troubleshoot-install, "Check for conflicting installations". Página consultada em 15/10/2026, Claude Code 2.1.301.

13.5 Fotos e recibos · receita 17 de 22 TESTADA

17 Fotos de recibos viram planilha de despesas

Você termina com um `despesas.xlsx` com data, fornecedor, valor, categoria e o nome da foto de cada recibo, uma coluna que marca as leituras duvidosas e o total em fórmula.

Você precisa de Excel instalado e as fotos em `.jpg` ou `.png`, uma por recibo, com o papel inteiro e em boa luz.

Testada em 15/10/2026: este pedido, com este texto, rodou duas vezes no Claude Code 2.1.300, Windows 11, Office 16, numa pasta de exemplo com nomes acentuados. A conferência da página é a que pegou os erros.

Foto boa vale mais que pedido bom. O Claude recebe a imagem reduzida quando ela é grande; papel amassado, sombra e foto torta viram linha em Conferir.

Sem Excel (não testado): `despesas.csv` com as mesmas colunas; o total vira número fixo.

Cole o pedido

```
Na pasta atual há fotos de recibos e notas (jpg ou png).
Monte a planilha despesas.xlsx com as colunas Data, Fornecedor, Valor, Categoria, Arquivo e Conferir.
Regras:
- Leia cada foto. Valor em reais, como número, com duas casas.
- Categoria: Alimentação, Transporte, Material, Serviço ou Outros.
- Se algum dado estiver ilegível ou duvidoso, não adivinhe: preencha o que deu e, na coluna Conferir, escreva o que precisa ser conferido.
- Uma linha Total no fim, com fórmula de soma dos valores.
- Use o Excel deste computador pelo PowerShell. Não instale nada. Se o Excel não estiver instalado, pare e me diga.
No fim, me diga o total e quais fotos ficaram para conferir.
```

O que esperar

1. Ele abre as fotos uma a uma (ler não pede licença) e diz o que leu.
2. Pede licença para o PowerShell que monta a planilha.
3. Diz o total e quais fotos ficaram para conferir.

Como conferir

- Para cada linha marcada em Conferir, abra a foto e corrija à mão.
- Em três linhas sem marca, compare data e valor com a foto. Vírgula e ponto trocados mudam o valor: 1.250,00 vira 1,25.
- Clique no total: `=SOMA(...)`. Conte as linhas: uma por foto.

Se der errado	O que dizer
Valor com casas trocadas	“Na foto X o valor é 1.250,00; corrija.”
Preencheu campo ilegível	“Na foto X a data não é legível; deixe vazia e marque em Conferir.”
Duas fotos do mesmo recibo	“X e Y são o mesmo recibo; uma linha só.”
Categoria errada	Corrija na planilha: categoria é decisão sua.

Fonte code.claude.com/docs/en/tools-reference e code.claude.com/docs/en/permission-modes, “Read tool behavior” (imagens) e tabela de modos. Página consultada em 15/10/2026, Claude Code 2.1.301.

13.5 Fotos e recibos · receita 18 de 22 **TESTADO 16/10 · WINDOWS**

18 Fotos renomeadas pela data em que foram tiradas

Você termina com cada foto renomeada como `AAAA-MM-DD_HH-MM-SS.jpg`, pela data gravada pela câmera dentro do arquivo, com `renomeados.csv` e `desfazer.ps1`.

Você precisa de só o Windows. Fotos de câmera ou celular em `.jpg`.

Testada em 16/10/2026: este pedido, com este texto, rodou no Claude Code 2.1.301, Windows 11, numa pasta de exemplo, sem tela (ler, gravar e PowerShell liberados de antemão, no lugar da aprovação passo a passo). O resultado foi conferido pela lista desta página.

Foto que passou por mensageiro costuma chegar sem a data da câmera: o aplicativo apaga esses dados no envio. Essas ficam com o nome atual, como manda o pedido.

Cole o pedido

```
Na pasta atual há fotos .jpg. Renomeie cada uma pela data e hora em
que foi tirada, no padrão AAAA-MM-DD_HH-MM-SS.jpg.
Regras:
- Use a data gravada pela câmera dentro do arquivo (EXIF, a que o
Windows mostra como "Tirada em"), lida com o PowerShell e o .NET do
Windows. Não instale nada.
- A data de modificação do arquivo NÃO serve: ela muda quando a foto
é copiada. Foto sem a data da câmera fica com o nome atual; me liste
quais.
- Duas fotos no mesmo segundo: acrescente _2 na segunda.
- Antes de renomear, me mostre a tabela nome atual -> nome novo e
pare até eu responder.
- Grave renomeados.csv e desfazer.ps1 (uma linha Rename-Item -Lite-
ralPath por arquivo, sem laço, em UTF-8 com BOM).
```

O que esperar

1. Ele lê a data de cada foto e mostra a tabela, com as que não têm data. Responda.
2. Pede licença para gravar os dois arquivos e renomear.
3. Diz quantas renomeou e quais ficaram como estavam.

Como conferir

- Clique com o botão direito numa foto > Propriedades > Detalhes: “Tirada em” tem de bater com o nome.
- Conte as fotos antes e depois: `(Get-ChildItem *.jpg).Count`.
- Rode o desfazer no ensaio.

Se der errado	O que dizer
Usou a data de modificação	“Só a data da câmera; sem ela, mantenha o nome.”
Hora três horas adiantada	“Use a hora como está gravada, sem converter fuso.”
Fotos .heic do iPhone ficaram de fora	Esperado: o Windows não lê HEIC sem extensão. Converta antes, ou deixe-as.

Fonte code.claude.com/docs/en/permission-modes, tabela de modos e “Which mode a session starts in”. Página consultada em 15/10/2026, Claude Code 2.1.301.

13.5 Fotos e recibos · receita 19 de 22 **TESTADO 16/10 · WINDOWS**

19 Reduzir fotos para mandar por e-mail

Você termina com uma pasta `reduzidas` com cópias das fotos de no máximo 1600 pixels no lado maior, e os originais intactos. Uma foto de celular de 4 MB costuma cair para algumas centenas de KB.

Você precisa de só o Windows. Fotos `.jpg` ou `.png`.

Testada em 16/10/2026: este pedido, com este texto, rodou no Claude Code 2.1.301, Windows 11, numa pasta de exemplo, sem tela (ler, gravar e PowerShell liberados de antemão, no lugar da aprovação passo a passo). O resultado foi conferido pela lista desta página.

1600 pixels enchem a tela de um notebook. Para anexar a um relatório em Word (receita 7), 1280 basta.

Rotação. Celular grava a foto deitada e anota no arquivo que ela deve ser girada. A cópia reduzida pode perder essa anotação; daí a linha da tabela.

Cole o pedido

```
Na pasta atual há fotos. Faça cópias reduzidas na subpasta reduzidas.
Regras:
- Lado maior com no máximo 1600 pixels, mantendo a proporção. Foto já
menor que isso é só copiada.
- Salve em JPG com qualidade 80, com o mesmo nome do original e a ex-
tensão .jpg. Se dois originais derem o mesmo nome, acrescente _2 no
segundo.
- Não altere nem apague os originais.
- Script ou arquivo de apoio que você precisar criar fica só dentro
desta pasta, nunca fora dela.
- Use o PowerShell e o .NET do Windows (System.Drawing). Não instale
nada.
- Arquivo que não abrir (por exemplo, .heic), deixe de fora e me diga
qual.
No fim, me mostre o tamanho total da pasta antes e da pasta reduzi-
das, em MB.
```

O que esperar

1. Pede licença para o PowerShell que cria a pasta e grava as cópias.
2. Informa o tamanho antes e depois e as fotos que não abriu.
3. Os originais ficam onde estavam.

Como conferir

- Tamanho das duas pastas: `'{0:N1} MB' -f ((Get-ChildItem -File | Measure-Object Length -Sum).Sum / 1MB)`, dentro de cada uma.
- Abra uma reduzida em tela cheia: o texto que importa (placa, etiqueta) ainda se lê.
- Conte os arquivos das duas pastas, menos os que ele listou.

Se der errado	O que dizer
Reduziu os originais	Não deveria: recupere da cópia de ensaio e repita a regra.
Foto deitada saiu de pé (ou o contrário)	“Respeite a orientação gravada na foto (EXIF).”
Ainda grande demais	“Use 1280 pixels e qualidade 70.”

Fonte code.claude.com/docs/en/permission-modes, tabela de modos e “Which mode a session starts in”. Página consultada em 15/10/2026, Claude Code 2.1.301.

Gostou da amostra?

O livro completo tem 148 páginas: 13 capítulos (permissões, settings.json, CLAUDE.md, contexto e custo, skills, subagentes, hooks, MCP, workflows, continuidade) e as 22 receitas de escritório.

PDF na hora por R$ 39,90, com 7 dias de garantia:
https://pay.kiwify.com.br/LClZD5P

Sumário e páginas: https://guia-claude-code-8r2.pages.dev/

Material independente, sem vínculo com a Anthropic. Claude e Claude Code são marcas da Anthropic.