

Como instalar o Claude Code no Windows, Mac e Linux

Como instalar o Claude Code no Windows, Mac e Linux pelo terminal, responder aos pedidos de permissão, criar o CLAUDE.md e usar subagentes.

1. Em 1 minuto
2. Faça agora
3. O que ele faz no seu computador
4. Permissões: o que responder
5. CLAUDE.md: as regras do projeto
6. Subagentes: vários trabalhando ao mesmo tempo
7. Deu problema?
8. Para quem quer ir além
9. Fontes

01 Em 1 minuto

O QUE É

O Claude Code é a ferramenta de programação da Anthropic. Ele trabalha dentro da pasta do seu projeto: lê arquivos, edita código e roda comandos.

O QUE VOCÊ GANHA

Um Claude que age no seu computador, e não só responde com texto.

QUANTO CUSTA

O plano gratuito do claude.ai não inclui. Precisa de assinatura Pro, Max, Team ou Enterprise, ou de uma conta do Claude Console com créditos pré-pagos.

02 Faça agora

Se você nunca usou, comece pelo Passo 1 e siga na ordem. Em 5 passos ele está instalado e respondendo.

PASSO 1

Abra o terminal

O terminal é onde você cola os comandos deste guia.

- **No Mac:** abra o app **Terminal**.
- **No Windows:** abra o **PowerShell** ou o **Prompt de Comando** (CMD).

Palavra difícil: Terminal é uma janela onde você dá ordens ao computador escrevendo, em vez de clicar.

Atenção: Antes, confira se você tem um plano que inclui o Claude Code. Os preços estão em claude.com/pricing.

PASSO 2

Instale com um comando

Copie o comando do seu sistema, cole no terminal e aperte Enter.

Mac, Linux ou WSL:

```
curl -fsSL https://claude.ai/install.sh | bash
```

Windows, no PowerShell (a linha começa com `PS C:\`):

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

Windows, no CMD (a linha começa com `C:\`, sem o `PS`):

```
curl -fsSL https://claude.ai/install.cmd -o install.cmd && install.cmd && del install.cmd
```

Dica: O comando não mostra progresso enquanto baixa. Espere terminar.

PASSO 3

Confira se instalou

Feche o terminal, abra uma janela nova e rode:

```
claude --version
```

Tem que aparecer um número de versão seguido de `(Claude Code)`.

PASSO 4

Abra o Claude Code na pasta do projeto

Entre na pasta do projeto. Troque o caminho pelo da sua pasta:

```
cd /caminho/da/sua/pasta
```

Depois, abra o Claude Code:

```
claude
```

No primeiro uso, ele pede para entrar na sua conta. O login termina no navegador e fica salvo.

Palavra difícil: Pasta do projeto é a pasta onde ficam os arquivos em que ele vai trabalhar.

PASSO 5

Faça o primeiro pedido

Escreva em português mesmo, como numa conversa. Por exemplo:

```
o que esse projeto faz?
```

Depois, peça uma mudança pequena. Ele acha o arquivo certo e mostra a alteração. Se pedir confirmação, escolha **Yes**.

03 O que ele faz no seu computador

Age, em vez de só responder

No chat comum, o Claude só responde com texto. No Claude Code, ele tem ferramentas para agir no seu computador.

Cada resultado volta para ele decidir o próximo passo. Por exemplo, um erro ou a saída de um teste.

Tipo	O que ele consegue fazer
Arquivos	Ler arquivos, editar código, criar arquivos novos, renomear e reorganizar
Busca	Achar arquivos por nome e procurar texto dentro deles
Execução	Rodar comandos no terminal, ligar servidores, rodar testes, usar o git
Web	Pesquisar na internet e abrir documentação

Ele escolhe sozinho qual ferramenta usar. Num pedido como "conserte os testes que estão falhando", a documentação descreve a sequência: ele roda os testes, lê o erro, procura os arquivos, lê o código, edita e roda os testes de novo para conferir.

> **Palavra difícil:** Git é o programa que guarda o histórico de mudanças do código.

O que ele enxerga

- **O projeto:** os arquivos da pasta e das subpastas. Arquivos de fora, só com a sua permissão.
- **O terminal:** qualquer comando que você mesmo poderia rodar ali.
- **O git:** o branch atual, as mudanças ainda não salvas e os commits recentes.
- **O CLAUDE.md:** as regras do projeto (mais abaixo).

Parar e desfazer

- **Parar:** aperte `Esc` a qualquer momento.
- **Desfazer:** aperte `Esc` duas vezes e volte a um ponto anterior.

> **Atenção:** Desfazer vale só para arquivos. O que vai para fora do computador, como um banco de dados ou um deploy, não volta.

04 Permissões: o que responder

Quando precisa da sua aprovação, ele mostra o que vai fazer (o comando ou o arquivo) e pergunta se pode seguir. As

opções aparecem em inglês:

- **Yes:** aprova só esta vez.
- **Yes, and don't ask again:** aprova e para de perguntar sobre aquilo.
- **Yes, and switch to auto mode:** aprova e muda para o modo automático. Não aparece em todo pedido.
- **No:** recusa. Aperte `Tab` e escreva o motivo, que ele usa para mudar de rumo.

Ele pode começar sem perguntar quase nada

Nas versões atuais, o terminal começa no **modo automático**. Um segundo modelo revisa cada ação e bloqueia as arriscadas, em vez de perguntar a você.

Por padrão, ele bloqueia:

- baixar e rodar código da internet;
- mandar dados sensíveis para fora;
- deploy em produção;
- apagar de vez arquivos que já existiam.

> **Atenção:** A própria documentação avisa: o modo automático reduz as perguntas, mas não garante segurança.

Para ele perguntar sempre

Aperte `Shift+Tab` durante a conversa até aparecer `⏸ manual mode on` embaixo da caixa de texto. No modo Manual, ele pergunta antes de editar arquivo e de rodar comando.

05 CLAUDE.md: as regras do projeto

Toda conversa nova começa do zero, sem lembrar das anteriores. O CLAUDE.md resolve isso.

É um arquivo de texto com as regras do projeto. O Claude Code lê esse arquivo no começo de toda conversa.

Crie com um comando

Dentro do Claude Code, na pasta do projeto, digite:

```
/init
```

Ele analisa o projeto e cria o arquivo com os comandos de build e de teste e os padrões que encontrar. Se o CLAUDE.md já existe, ele sugere melhorias em vez de apagar.

O que escrever nele

O que você explicaria de novo em toda conversa:

- comandos de build e de teste;
- padrões de código;
- onde fica cada coisa;
- regras de "sempre faça X".

Dica: Seja concreto. "Rode `npm test` antes do commit" funciona melhor que "teste suas mudanças".

Atenção: O CLAUDE.md é contexto, não trava. Ele lê e tenta seguir, mas instrução vaga ou contraditória pode ser ignorada.

06 Subagentes: vários trabalhando ao mesmo tempo

Um subagente é um assistente separado que o Claude Code chama para uma parte da tarefa. Quando termina, ele devolve só um resumo para a conversa principal.

> **Palavra difícil:** Janela de contexto é a memória da conversa. Cada subagente tem a própria, então a conversa principal não enche com tudo o que ele leu.

Os que já vêm prontos

Ele chama sozinho quando a tarefa combina:

- **Explore:** só lê. Procura e analisa o código sem mudar nada.
- **Plan:** pesquisa o projeto para montar um plano, no modo Plan.
- **General-purpose:** para tarefas de vários passos que pedem pesquisar e também alterar arquivos.

Peça vários de uma vez

Eles rodam em segundo plano, enquanto você continua. Para pesquisas independentes, peça assim:

```
pesquise os módulos de login, banco de dados e
API em paralelo, com subagentes separados
```

Cada um investiga a sua parte e o Claude junta o resultado. Quando um subagente precisa de permissão, o pedido aparece na sua tela com o nome dele.

Atenção: Cada subagente gasta da mesma cota de uso da conversa principal. Muitos ao mesmo tempo gastam mais.

07 Deu problema?

O que aconteceu	O que fazer
Apareceu `command not found: claude`	Feche e abra o terminal. Se continuar, a pasta da instalação não está no PATH: veja "Fix your PATH" na página de problemas de instalação (Fontes).
Deu erro com `&&` ou `irm` no Windows	Você usou o comando do terminal errado. `irm` é do PowerShell, `curl ... install.cmd` é do CMD.
Apareceu `syntax error` ou `403` ao instalar	Veio uma página em vez do instalador. Tente de novo em alguns minutos ou instale pelo Homebrew ou WinGet (em "Para quem quer ir além").
Apareceu `App unavailable in region`	O Claude Code não está disponível no seu país.
Não consigo entrar com a conta grátis	O plano gratuito do claude.ai não inclui o Claude Code.
Ele não pergunta antes de agir	Está no modo automático. Aperte `Shift+Tab` até aparecer `manual mode on`.
Ele não segue o CLAUDE.md	Rode `/context` e veja se o arquivo está em **Memory files**. Deixe as regras mais concretas.

O que aconteceu	O que fazer
Outro problema	Rode `/doctor` dentro do Claude Code, ou `claude doctor` no terminal se ele nem abrir.

Pronto, o principal acabou aqui. Daqui pra baixo é a parte técnica, pra quem quer entender os detalhes.

08 Para quem quer ir além

Onde mais ele roda

Ele roda no terminal, no VS Code, no JetBrains, num app de computador e no navegador. Este guia cobre o terminal, que é o que o vídeo mostra.

Requisitos completos

- **Conta:** assinatura Pro, Max, Team ou Enterprise, ou uma conta do Claude Console (acesso pela API com créditos pré-pagos).
- **Sistema:** macOS 13.0 ou mais novo, Windows 10 1809 ou mais novo, Ubuntu 20.04 ou mais novo, Debian 10 ou mais novo, ou Alpine Linux 3.19 ou mais novo.
- **Máquina:** 4 GB de RAM ou mais e internet ligada.
- **Windows:** o Git for Windows é recomendado, para o Claude Code usar o Bash. Sem ele, o Claude Code usa o PowerShell no lugar. Quem usa WSL não precisa.
- **Projeto:** uma pasta de projeto para ele trabalhar.

Instalar pelo Homebrew ou pelo WinGet

No Mac, pelo Homebrew:

```
brew install --cask claude-code
```

No Windows, pelo WinGet:

```
winget install Anthropic.ClaudeCode
```

A instalação pelo comando oficial se atualiza sozinha. Pelo Homebrew e pelo WinGet, não. Para atualizar:

```
brew upgrade claude-code
```

```
winget upgrade Anthropic.ClaudeCode
```

Comandos que ajudam desde o começo

Comando	O que faz
`/help`	Mostra os comandos disponíveis
`/clear`	Limpa a conversa
`/exit`	Fecha o Claude Code
`claude -c`	No terminal, continua a última conversa daquela pasta

Permissões em detalhe

- **Yes, and don't ask again:** para um comando, a regra fica salva em `.claude/settings.local.json` e vale nas próximas conversas daquele projeto. Para edição de arquivo, vale até o fim da conversa.
- **Modo Manual:** ler arquivos dentro da pasta do projeto não pede aprovação. Editar arquivos e rodar comandos pede, menos uma lista de comandos que só leem.
- **Versão:** o modo automático é o inicial no terminal da versão 2.1.283 em diante.

Os modos, trocados com `Shift+Tab` durante a conversa:

Modo	O que ele faz sem perguntar
Auto	Quase tudo, com a revisão automática bloqueando o que for arriscado
Manual	Só ler. Pergunta antes de editar arquivo e de rodar comando
Accept edits	Ler, editar arquivos e comandos simples de pasta, como `mkdir` e `mv`
Plan	Só pesquisa e propõe um plano, sem editar seus arquivos

Para começar uma conversa no modo Manual:

```
claude --permission-mode default
```

Para todas as conversas no seu computador começarem no Manual, salve isto em `~/.claude/settings.json`:

```
{
  "permissions": {
    "defaultMode": "default"
  }
}
```

Liberar ou bloquear comandos

Dentro do Claude Code, o comando `/permissions` lista as regras e em qual arquivo cada uma está. Existem três tipos:

- **Allow:** roda sem perguntar.
- **Ask:** pergunta sempre.
- **Deny:** nunca roda.

O bloqueio vence sempre. Se um comando cai numa regra Deny, nenhuma regra Allow libera.

> **Atenção:** Regra escrita no CLAUDE.md não muda permissão. Quem decide o que pode rodar são os modos e as regras do `/permissions`.

Onde fica o CLAUDE.md

Arquivo	Vale para
`CLAUDE.md` **na raiz do projeto (ou** `.claude/CLAUDE.md` **)**	O projeto, e quem usa o mesmo repositório
`~/.claude/CLAUDE.md`	Todos os seus projetos, só no seu computador
`CLAUDE.local.md` **na raiz do projeto**	Só você, só naquele projeto

A documentação recomenda ficar abaixo de 200 linhas.

Exemplo de CLAUDE.md

```
# Projeto: site da loja
- Rodar local: npm run dev
- Antes de commit: npm test
- Páginas ficam em src/pages/
- Indentação de 2 espaços
- Textos do site em português
- Nunca editar arquivos da pasta public/vendor/
```

Conferir e editar o CLAUDE.md

- `/memory` lista os arquivos CLAUDE.md e abre o que você escolher para editar.
- `/context` mostra se o arquivo carregou, na lista **Memory files**.

Subagentes em detalhe

Cada subagente trabalha na **própria janela de contexto**, com instruções, ferramentas e permissões próprias. Os que ele chama a partir da conversa principal rodam em segundo plano.

Por padrão, o limite é de 20 subagentes rodando juntos numa sessão.

Criar o seu subagente

Um subagente é um arquivo de texto. O jeito mais simples é pedir para o próprio Claude Code criar, dizendo o que ele faz e onde salvar:

- `.claude/agents/` dentro do projeto: vale só naquele projeto.
- `~/.claude/agents/`: vale em todos os seus projetos.

Para usar, peça pelo nome ("use o subagente revisor nas minhas mudanças") ou digite `@` e escolha na lista. O comando `/agents`, nas versões atuais, mostra esse lembrete: pedir ao Claude ou editar essas pastas.

09 Fontes

Conferido em 30/10/2026, na documentação oficial do Claude Code (Anthropic):

- Visão geral
- Quickstart
- Instalação e requisitos (Advanced setup)
- Como o Claude Code funciona
- Permissões
- Modos de permissão
- CLAUDE.md e memória
- Subagentes
- Problemas de instalação

Tem guia novo a cada vídeo.

Todos em marraads.online · siga @marraads