

# How to edit videos with AI

The same system I use for my videos, set up on your computer. You record, answer a few quick questions, and the AI transcribes, cuts, animates, adds captions, fixes the sound and delivers the finished video.



## IN THIS GUIDE

- 02 How it works
- 03 Before you start
- 04 What each tool does
- 05 Step 1: what you install by hand
- 07 Step 2: the master prompt
- 08 After the setup
- 09 If something goes wrong
- 10 Appendix: the full master prompt

02

# How it works

The core idea: the animations aren't made by dragging things around a timeline. They're written in code, like a web page, and AI writes that code really well. And the system adapts to what you say: Claude Code reads your speech and creates animations that follow each word.

## THE PRODUCTION LINE FOR EACH VIDEO

- 1 Record yourself** talking to the camera, vertically. A phone works. Then put the video in the project's raw/ folder.
- 2 Ask Claude Code to edit it.** Before starting, it runs a quick interview: captions only, captions with zooms and cards, or full animation? What needs to appear on screen?
- 3 It transcribes the video** with Whisper, with the exact timing of each word, and cuts the silences.
- 4 It reads what you said and proposes the animations:** every number, list, comparison or step you mention becomes something on screen, coming in on the exact word. It shows you this in a storyboard, the plan for each scene with its timing.
- 5 You approve or ask for changes.** It animates, adds captions and sound effects, sets the loudness and renders with HyperFrames.
- 6 You watch and ask for tweaks.** It usually takes two or three rounds.

What gives every video the same look are three files the master prompt creates with you: the editing rules (CLAUDE.md), the visual identity and the channel positioning. Claude reads these files before every edit.

03

# Before you start

Six things to check before the first command.

## CLAUDE PLAN

### Pro or higher

Claude Code isn't included in the free plan. It's the only paid part of the system.

## COMPUTER

### Mac with macOS 13 or newer, or Windows 10 (version 1809) or newer

Claude Code requirement.

## MEMORY

### 8 GB or more

Recommended. Rendering video uses a lot of memory.

## FREE SPACE

### About 15 GB

Programs, the transcription model and the videos.

## PROJECT FOLDER

### Outside iCloud, OneDrive or Google Drive

Large videos in a synced folder freeze and disappear from the disk.

## TIME

### About 1 hour

On the first day: installation and the visual identity interview.

## Mac or Windows?

The system was built and tested on a Mac. It should work on Windows, since every tool has a Windows version, but that's where more adjustments may come up. If rendering doesn't work on Windows, the fix is WSL (Linux inside Windows), and Claude Code itself will guide you.

04

# What each tool does

You'll hardly use any of them directly: Claude Code does. But it's good to know what's going on your computer.

| TOOL                                                                                                                                                       | WHAT IT IS                                                                                                                     | WHAT IT DOES IN THE SYSTEM                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  <b>Terminal (Mac) / PowerShell (Windows)</b><br>Already on your computer | The window where you type commands to the computer instead of clicking                                                         | It's where you install things and open Claude Code                                           |
|  <b>Claude Code</b><br>Included in Claude Pro                             | Claude working directly on your computer: it opens files, runs programs and creates folders, always asking for your permission | It's the editor. It does the rest of the process                                             |
|  <b>Homebrew (Mac only)</b><br>Free                                      | An "app store" for the Terminal: it installs programs with one command                                                         | Installs Node.js, FFmpeg, Git, Python and Whisper                                            |
|  <b>winget (Windows only)</b><br>Free                                   | The same idea as Homebrew, already in Windows 10 and 11                                                                        | Same job on Windows                                                                          |
|  <b>Node.js</b><br>Free                                                 | The engine that runs programs written in JavaScript outside the browser                                                        | Makes HyperFrames work                                                                       |
|  <b>FFmpeg</b><br>Free                                                  | The "Swiss Army knife" of video and audio                                                                                      | Cuts the silences, joins the parts, sets the loudness and creates the final file             |
|  <b>Git</b><br>Free                                                     | A program that keeps versions of files                                                                                         | Claude Code uses it to download tools and, on Windows, to run commands                       |
|  <b>Python</b><br>Free                                                  | A programming language                                                                                                         | Runs the system's small scripts: sound effects, word timings                                 |
|  <b>Whisper</b><br>Free, runs on your computer                          | An AI that turns speech into text, with the exact timing of each word                                                          | Makes the caption light up on the right word and the animation come in the second you say it |
|  <b>HyperFrames</b><br>Free (open source)                               | Turns an animated web page (HTML) into video, frame by frame                                                                   | It's where the animations are created and rendered                                           |
|  <b>Google Fonts</b><br>Free                                            | A font library                                                                                                                 | The fonts of your visual identity                                                            |

## 05 STEP 1 · WHAT YOU INSTALL BY HAND

# On Mac

Just these commands. Copy each one, paste it into the Terminal window (Cmd+V) and press Enter. Claude Code installs the rest with you in step 2.

**1** **Open the Terminal.** Press Cmd+Space, type Terminal and press Enter.

**2** 🍺 **Install Homebrew.** It will ask for your Mac password. Nothing shows on screen while you type: that's normal, type it and press Enter.

```
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

At the end, Homebrew shows a part called **Next steps** with two or three command lines. Copy and run those lines too: they let your Mac find Homebrew. To check, run `brew --version`: a version number should appear.

**3** ✨ **Install Claude Code.**

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

Close the Terminal, open it again and run `claude --version` to check.

**4** **Create the project folder and go into it.** It lives in your user folder, outside iCloud.

```
mkdir -p ~/edicao-ia && cd ~/edicao-ia
```

**5** ✨ **Open Claude Code.**

```
claude
```

The first time, it opens the browser so you can sign in to your Claude account. Then go to step 2.

## 06 STEP 1 · WHAT YOU INSTALL BY HAND

# On Windows

Just these commands. Copy each one, paste it into the PowerShell window (Ctrl+V or right-click) and press Enter. Claude Code installs the rest with you in step 2.

**1 Open PowerShell.** Click Start, type PowerShell and press Enter. No need to open it as administrator.

**2  Install Git.** If it asks whether you accept the terms, type Y and press Enter. If Windows asks whether the app can make changes, click Yes.

```
winget install Git.Git
```

**3  Install Claude Code.**

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

Close PowerShell, open it again and run `claude --version` to check.

**4 Create the project folder and go into it.** It lives in your user folder, outside OneDrive.

```
mkdir $HOME\edicao-ia; cd $HOME\edicao-ia
```

**5  Open Claude Code.**

```
claude
```

The first time, it opens the browser so you can sign in to your Claude account. Then go to step 2.

## Every time you use the system later

Open the Terminal or PowerShell, go into the folder and type `claude`.

```
Mac cd ~/edicao-ia
```

```
Windows cd $HOME\edicao-ia
```

07

## Step 2: the master prompt

With Claude Code open in the project folder, copy the whole prompt, paste it and press Enter. It works in five phases and stops at the end of each one. Whenever it asks permission to run a command, read what it explained and approve. If Windows or your Mac asks for a password or confirmation, you're the one who types it.

### Copy the prompt from the website, not from the PDF

Copying long text from a PDF breaks the lines and picks up the page numbers. The full text is in the appendix (pages 10 to 14) for you to read. To copy it, use the button on the website: it has the full version and the just-one-video version.

Copy the prompt at [orodmoraais.pages.dev/en](https://orodmoraais.pages.dev/en)

### THE FIVE PHASES

|    |                     |                                                                                             |
|----|---------------------|---------------------------------------------------------------------------------------------|
| 01 | <b>Diagnosis</b>    | Checks your computer, what's already installed and whether the folder is outside the cloud. |
| 02 | <b>Installation</b> | Installs what's missing, explaining each tool, and checks that rendering works.             |
| 03 | <b>Interview</b>    | Builds your visual identity: colors, fonts, captions, editing style and loudness.           |
| 04 | <b>Setup</b>        | Creates the editing rules, the identity, the quality standards and the video template.      |
| 05 | <b>Delivery</b>     | Explains your day to day and writes the user guide in the folder.                           |

### Just want to edit one video, without building a channel?

The prompt builds a permanent system, with a visual identity and rules for every video. For a single video (a promotion, an ad, a birthday video), choose "Just one video" on the website. It already comes with these changes:

- 1 In **PHASE 3**, items 1, 2 and 8 are removed. Items 3 to 7 become quick questions about this video.
- 2 In **PHASE 4**, the file list is replaced by: "Create only the raw/, videos/ and out/ folders and a short CLAUDE.md with the rules below." The **QUALITY STANDARDS**, **ANIMATION THAT FOLLOWS THE SPEECH** and **CLAUDE.md RULES** blocks stay.
- 3 **PHASE 5 becomes**: "Edit the video I put in raw/v1.mov, starting with the quick interview from rule 1."

Phases 1 and 2 stay the same: without them, nothing works.

08

# After the setup: each new video

- 1 Record** vertically, with good light in front of you and the mic close to your mouth. Audio is what most gives away an amateur video.
- 2 Put the video in the** raw/ folder with the video number: v2.mov, v3.mov...
- 3 Put the material in the** raw/ref/ folder: logo, product photos, screenshots, whatever needs to appear.
- 4 In Claude Code, write:**

```
> Edit the video raw/v2.mov
```

For very complex animations, type /model opus first.
- 5 Answer the quick interview:** captions only, captions with zooms and cards, or full animation, and what needs to appear on screen.
- 6 Approve the storyboard** or ask for changes. Only then does it render.
- 7 Watch** and ask for tweaks with the exact time ("at 0:42, the caption covers the price"). The finished video goes to out/.

## TIPS THAT MAKE A DIFFERENCE

The more you say in the interview, the better. "I want the price big when I say 29.90" becomes exactly what you pictured.

Captions only uses much less of your Claude limit than full animation. Pick the level for each video.

A new Claude Code conversation for each video. Long conversations use a lot more.

### The system learns from you

It starts with the same tools and the same quality standards I use. What it picks up over time is your style: through rule 15 of the prompt, every correction you make becomes a rule in CLAUDE.md, and the next video comes out closer to what you want. Liked an animation? Ask for it to become a rule or part of the template.

09

# If something goes wrong

The most common errors and what to do about each one.

| WHAT YOU SEE                                            | WHAT TO DO                                                                                                                                                                    |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| command not found: claude or "claude is not recognized" | Close the terminal and open it again. If it continues, run <code>claude doctor</code> and show the result to Claude in the app.                                               |
| command not found: brew (Mac)                           | You skipped Homebrew's <b>Next steps</b> lines. Run the install command again and copy those lines from the end.                                                              |
| 'irm' is not recognized (Windows)                       | You're in Command Prompt, not PowerShell. Open PowerShell.                                                                                                                    |
| Claude Code asks you to sign in and nothing happens     | Check that your plan is Pro or higher. The free plan doesn't include Claude Code.                                                                                             |
| "Resource deadlock avoided" or a file that disappears   | The folder is in the cloud (iCloud, OneDrive). Move the project to <code>~/edicao-ia</code> .                                                                                 |
| Rendering takes a long time                             | That's normal: it takes a picture of every frame. A 1-minute video can take several minutes.                                                                                  |
| Rendering fails on Windows                              | Ask Claude Code: "the HyperFrames render failed, investigate and, if needed, set up WSL for me".                                                                              |
| The caption has a wrong word                            | Tell Claude Code the right word and the time. It fixes the transcription and the caption.                                                                                     |
| You hit your Claude usage limit                         | Use a new conversation for each video and keep <code>opusplan</code> as the default. Simple tasks run on the smaller models. At the captions-only level, usage is much lower. |

## 10 APPENDIX · PART 1 OF 5

# The master prompt, in full

Here it is for you to read. To paste it into Claude Code, copy it with the button.

[Copy the prompt](#)

You are going to build with me, from scratch, a system to edit short vertical videos (Reels, TikTok, Shorts) with AI, just like @orodmoraais's. The system adapts to what I say: you read what I said in the video and create animations that follow each word. I don't know how to code. Explain each step in simple English, with one sentence on what each tool does, and ask my permission before installing anything or running any command that changes my computer. Never use em dashes in the texts you write for me.

The system uses: Claude Code (you), Node.js 22 or newer, FFmpeg, Git, Python 3, Whisper (transcription with the timing of each word), HyperFrames (animations written in HTML and GSAP and rendered to video) and fonts from Google Fonts.

Work in 5 phases, in this order. At the end of each phase, tell me in 2 lines what is done and ask if you can continue.

## PHASE 1 · DIAGNOSIS

1. Find out my system (macOS or Windows), its version, the processor, the memory and the free space. If I have less than 8 GB of memory or less than 15 GB free, warn me before continuing.
2. Check what is already installed (Homebrew or winget, node, ffmpeg, git, python, whisper) and show me a table: tool, what it does in one sentence, installed or not, version.
3. Check whether this folder is inside a cloud-synced folder (iCloud, OneDrive, Google Drive, Dropbox). If it is, help me move the project out: large videos in a synced folder freeze and disappear from the disk.

## 11 APPENDIX · PART 2 OF 5

**PHASE 2 · INSTALLATION**

Install only what is missing, one tool at a time, explaining first what it is.

- macOS: use Homebrew. If it isn't installed, stop and give me the official command to run in the Terminal (it asks for my password) and remind me to run the "Next steps" lines it shows at the end. Then: `brew install node ffmpeg git python whisper-cpp`.
- Windows: use winget: `Git.Git`, `OpenJS.NodeJS.LTS`, `Gyan.FFmpeg` and `Python.Python.3.12`. For Whisper, use `whisper.cpp` (the Windows build from the official GitHub repository) or, if that fails, `pip install openai-whisper`. If a command isn't recognized after installing, ask me to close and reopen the terminal, and check again.
- Download Whisper's multilingual "medium" model. If the computer is weak, use "small" and tell me.
- Install HyperFrames as a Claude Code plugin (claude plugin marketplace add `heygen-com/hyperframes` and then `claude plugin install hyperframes@hyperframes`) and run its diagnostics (`npx hyperframes doctor`).
- Check that rendering works with a 3-second animated text and delete the test file afterwards.

**PHASE 3 · INTERVIEW**

Now interview me to build my visual identity. At most 4 questions at a time, in rounds. For each question, give 2 to 4 concrete options, say which one you recommend and why, and let me decide. Cover:

1. Channel: name and @, topic, who I'm talking to, language, platforms, tone (serious, light, fun), what I never want in my videos.
  2. References: ask for 2 or 3 profiles or videos I like, screenshots of them and what I like about each one.
  3. Recording: camera or phone, framing, setting, whether I talk to the camera the whole video.
  4. Visual identity: card background color, text color, one accent color (or one per topic), two Google Fonts (one for titles and one for text, with 3 combinations for me to choose from) and my logo, if I have one. Generate a sample image of each option for me to see before deciding.
  5. Captions: position, size, how many words at a time, color of the word being spoken, whether keywords change color.
  6. Editing style: face zoom, info cards, PIP (me in a small window when the screen has a lot of information), sound effects (yes or no, which ones), background music (yes or no).
  7. Audio: final loudness. Explain the difference between -14 LUFS (the platforms' standard) and -10 LUFS (louder, what @orodmoraais uses) and recommend one.
  8. Data: if my videos show numbers or facts, how to cite the source on screen.
- At the end, show me everything in a table and ask for my OK before writing the files.

## 12 APPENDIX · PART 3 OF 5

**PHASE 4 · SETUP**

Create in this folder:

- CLAUDE.md: the rules that apply to every edit (list below, adapted to my identity), written for you to read at the start of every session.
- brand/identidade-visual.md: colors with hex codes, fonts and sizes, safe zones, layouts (normal and PIP), captions, motion, sound and loudness, with the quality and animation standards below.
- brand/posicionamento.md: channel, audience, tone and what never goes into my videos.
- brand/safe-zones-1080x1920.png: a transparent image with the areas where the Instagram and TikTok interface covers the video painted red, to check every frame. Look up each platform's current measurements. As a starting point, @orodmoraais uses, in a 1080x1920 frame: nothing essential with y below 270 or above 1540, x below 120 or above 960, x above 900 below y 840, and essential text above y 1420.
- brand/briefs/: where you save the brief you write yourself for each video, from the transcription and the quick interview, in these sections: 1 interview answers, 2 material, 3 transcribed speech, 4 scene by scene (with the exact word that triggers each event), 5 captions, 6 error checklist.
- Folders raw/ (the raw videos, named v1.mov, v2.mov...), raw/ref/ (logos, photos, screenshots and references), videos/ (one project per video), out/ (the finished videos), assets/fonts/ and assets/sfx/.
- tools/: scripts to transcribe with Whisper (with the timing of each word), cut silences, measure the face in the video, synthesize soft sound effects (click, swoosh, ding, key) and master the audio.
- videos/\_molde/: a HyperFrames project already with my identity, the captions, the normal layout and the PIP layout. Every new video starts by copying the template.
- .claude/settings.json with "model": "opusplan" (Opus plans, Sonnet executes, to spend less) and two subagents in .claude/agents/, "transcricao" and "conferencia", both with model: haiku.
- GUIA.md: how I use the system day to day, in simple language.

## 13 APPENDIX · PART 4 OF 5

**QUALITY STANDARDS** (they go in identidade-visual.md; these are the values @orodmoraais refined over many videos):

- Final frame 1080x1920. Measure the face in the raw video (top of the hair and chin) and calculate the allowed zoom so the face never gets closer than 60 px to the captions or the cards. In normal parts, zoom between 110% and 115%; punch-ins only at turning points.
- Normal layout: me in full screen, short information at the top (y 280 to 450) and captions right below (y 470 to 560).
- PIP layout: me in a 420x560 window in the top right corner (x 540 to 960, y 280 to 840), rounded corners; the content takes x 120 to 900, y 870 to 1410; captions at y 1420 to 1510. PIP enters and exits animated in 0.6 s, without cutting the video.
- Captions: groups of 1 to 3 words, one line, centered. Words not yet spoken stay at 35% opacity and each one lights up the instant it's spoken. Keywords in the accent color. The caption disappears when the screen shows exactly the sentence I'm saying.
- Audio: high-pass filter at 80 Hz, de-esser, +2 dB at 3 kHz, compressor (threshold -24 dB, 4:1, attack 5 ms, release 100 ms), limiter at -1 dBTP and two-pass linear normalization (chosen loudness, peak -1, LRA 7).
- Sound effects synthesized by script, with a gain of 0.4 relative to the voice.
- Check screenshots: at 0 s, every 2 s and on the last frame, at phone size, with the safe-zone overlay.

**ANIMATION THAT FOLLOWS THE SPEECH** (goes in identidade-visual.md):

After transcribing, read the speech and mark the moments that call for something on screen. The video adapts to what I said, never the other way around.

- Number or price: the number comes in big on the spoken word; if it's a comparison, it becomes bars or a count.
- List or steps (first, second...): items come in one by one and stay on screen like a scoreboard.
- Comparison (before and after, this versus that): two columns side by side.
- Cause and effect: arrows that draw themselves from one idea to the other.
- Brand, app, place or person name: official logo, icon or photo from raw/ref/, never an invented image.
- Demonstration (an app, a website, a screen): recreate the interface in HTML and animate it as if it were being used live.
- Keyword or punchline: highlight in the caption or a short card at the top.
- Question or turning point: punch-in on the face.

Don't animate everything: choose the moments that help understanding. After a few seconds of a screen full of information, go back to the face.

## 14 APPENDIX · PART 5 OF 5

**CLAUDE.md RULES** (keep all of them):

1. Before each edit, do a quick interview, with options: (a) editing level: captions and cuts only, captions with zooms and simple cards, or full animation that follows the speech; (b) where it will be posted; (c) what needs to appear on screen (names, prices, dates, logo, address, @); (d) material I have (logo, photos, screenshots, data with a source), which goes in raw/ref/; (e) anything to cut or avoid. The depth of the work follows the chosen level: at the captions-only level, no animation and no long storyboard.
2. Before rendering, show me the storyboard: a table with time, speech, layout, what comes on screen, the exact word it comes in on and the sound. Wait for my OK. At the captions-only level, a sample screenshot is enough.
3. The caption is what I said. Transcribe the raw video with Whisper in three passes (whole file with a list of names, whole file without the list, segment by segment), keep what at least two passes confirm and fix only names and terms. Write down doubts in NOTAS.md and ask me.
4. Fine cut: silences over 0.3 s become 0.22 s. After an important sentence, leave 0.5 s of breathing room.
5. Each element comes in on the exact word where I talk about it and stays while the idea is being spoken. List items arrive 0.08 s before the word.
6. Lots of information on screen = PIP. No captions repeating what the chart already shows.
7. Nothing essential outside the safe zones. Check the screenshots with the overlay.
8. Lines, arrows and outlines draw themselves live, never appear ready.
9. Smooth motion: entrances of 0.3 to 0.5 s, exits of 0.25 s. Nothing blinks, spins or bounces without a reason.
10. Sound effects low and never on top of an important word.
11. Never change brand/ or the projects of previous videos. The notes for each video go in that project's NOTAS.md.
12. Any number, data, screenshot or AI answer that appears on screen must be real and have a source. Only official logos.
13. Use the HyperFrames skills when they help, but these rules come first.
14. Before showing me anything, run the "conferencia" subagent.
15. At the end of each video, write here in CLAUDE.md what I corrected and what worked, as a new rule. This is how the system gets better with every video.

**PHASE 5 · DELIVERY**

Explain to me in up to 10 lines what my day to day will look like and write it in GUIA.md. To edit a video, I'll just put the file in raw/ and write "Edit the video raw/vN.mov"; you start with the quick interview from rule 1. Also explain that, for very complex animations, it's worth typing /model opus first.



**Copy the whole prompt in one click**

Opens the website with the copy button: full version or just-one-video version.