



Introdução aos agentes de IA na pesquisa acadêmica

Dia 1

Felipe Lamarca

IESP-UERJ · MAPE · DOXA · NECON

Feito com Claude Code 😊

2 de junho de 2026

- 01 Chats e agentes
- 02 Instalação: Mac e Windows
- 03 Integrações: VSCode, Positron e terminal
- 04 Configurando o ambiente: `.claude`, CLAUDE.md, skills e rules
- 05 Sites de projeto e sites pessoais

Chats e agentes

Você provavelmente já usa IA

- Interface de chat: você digita, o modelo responde
- Cada conversa começa do zero: sem memória de sessões anteriores, sem acesso ao seu ambiente de trabalho
- Interações **transacionais**: úteis para tarefas pontuais, mas sem continuidade

O chat tem limites para pesquisa

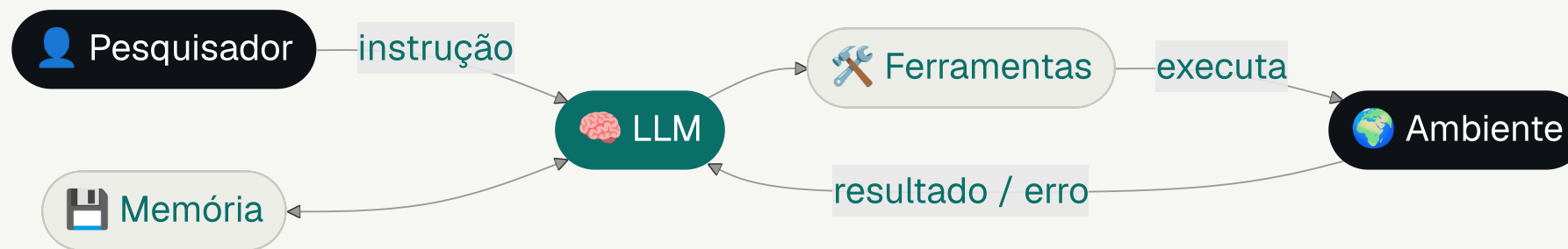
- Sem acesso ao seu ambiente de trabalho (arquivos, código, terminal)
- Sem memória entre sessões
- Cada etapa do ciclo de pesquisa — coleta, análise, documentação, comunicação — exige sua coordenação manual

O que é um agente de IA?

Em vez de responder a perguntas pontuais, o agente planeja ações, executa código e itera sobre tarefas complexas com supervisão mínima.

Chat	Agente
Responde	Age
Uma troca	Múltiplas etapas
Você executa	Ele executa

Anatomia de um agente



 contexto da sessão + CLAUDE.md  terminal · arquivos · APIs · web

Claude Code

- CLI que roda diretamente no terminal — ou integrado ao VSCode e ao Positron
- Lê e escreve arquivos, executa código, acessa o `git` e o shell
- Trata-se do caso mais desenvolvido de agente de IA para uso em pesquisa

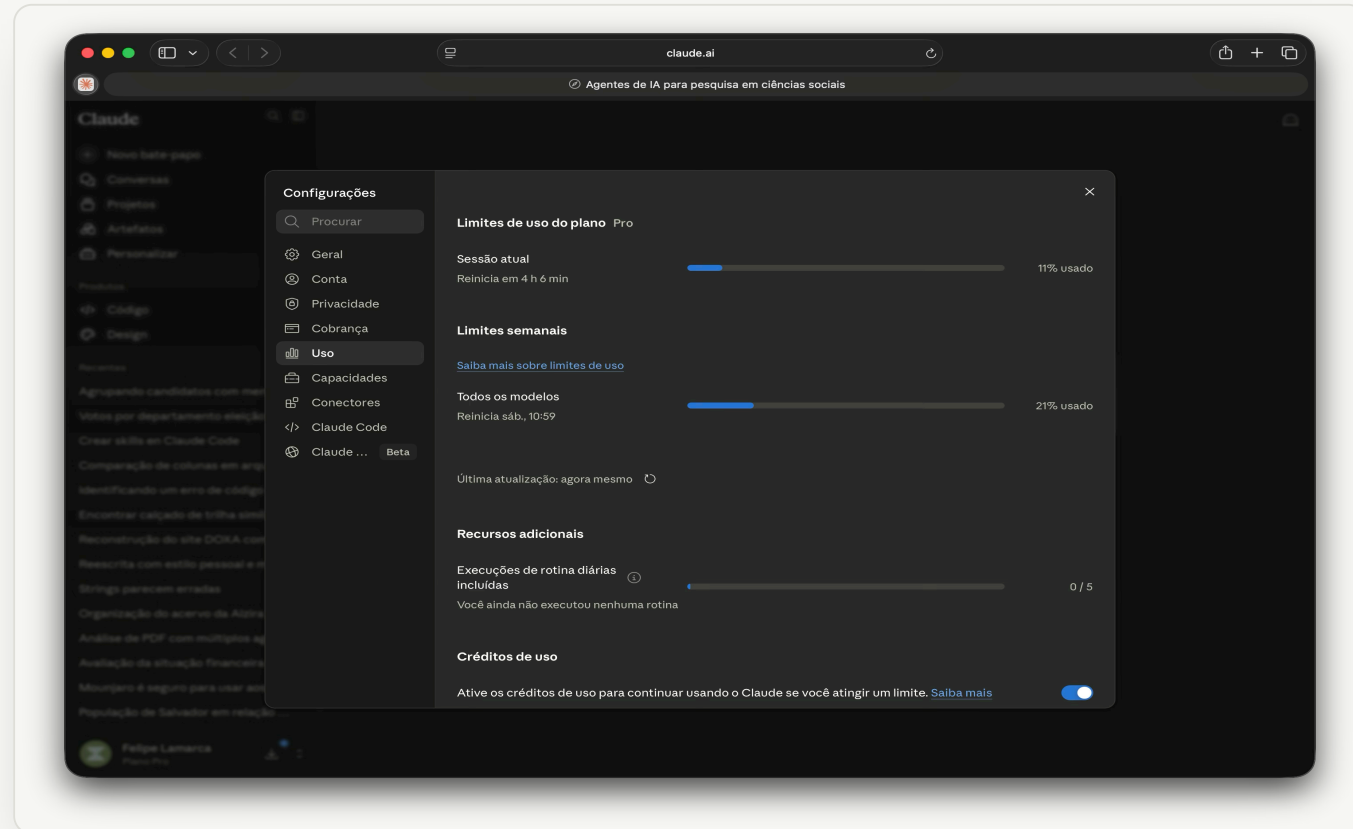
Roda o script, encontra o erro, corrige e re-executa — sem você abrir o terminal.

Quanto custa? 💰

Plano	Preço	O que inclui
Pro	US\$ 20/mês	Acesso ao Claude Code
Max 5×	US\$ 100/mês	5× mais uso que o Pro
Max 20×	US\$ 200/mês	20× mais uso que o Pro
API	Por token	Para automação e uso intenso

Meu plano é o Pro — mais ou menos suficiente para uso intensivo. Quando tenho mais pressa, deixo um valor para uso via API

Acompanhando o gasto



No dashboard da Anthropic: uso de tokens, gastos por período e alertas de limite — tudo em tempo real

Outros agentes de IA

Ferramenta	Empresa	Diferencial
Claude Code	Anthropic	CLI + IDE; forte em raciocínio e texto
Codex	OpenAI	Equivalente da OpenAI para terminal
Cursor	Cursor AI	IDE própria centrada em IA
Cline	Open-source	Extensão VSCode; suporta vários modelos

A escolha depende do modelo que você prefere e de como você trabalha — **o raciocínio que vamos aprender hoje vale para todos**

Chat vs. agente: vamos ver na prática

Antes de instalar, uma demonstração concreta da diferença

Tarefa: gerar código que plota uma distribuição normal em Python

- **No chat:** você recebe o código, copia, cola no script, roda, corrige erros manualmente...
- **No Claude Code:** o agente cria o arquivo, roda, detecta erros, corrige — tudo na mesma sessão

Vamos voltar a essa pasta depois de instalar e configurar o VSCode

Instalação

Instalação

Baixar o app em **claude.com/download** — disponível para Mac e Windows

Depois, abrir o Claude e pedir:

“Me ajude a instalar o Claude Code no meu computador”

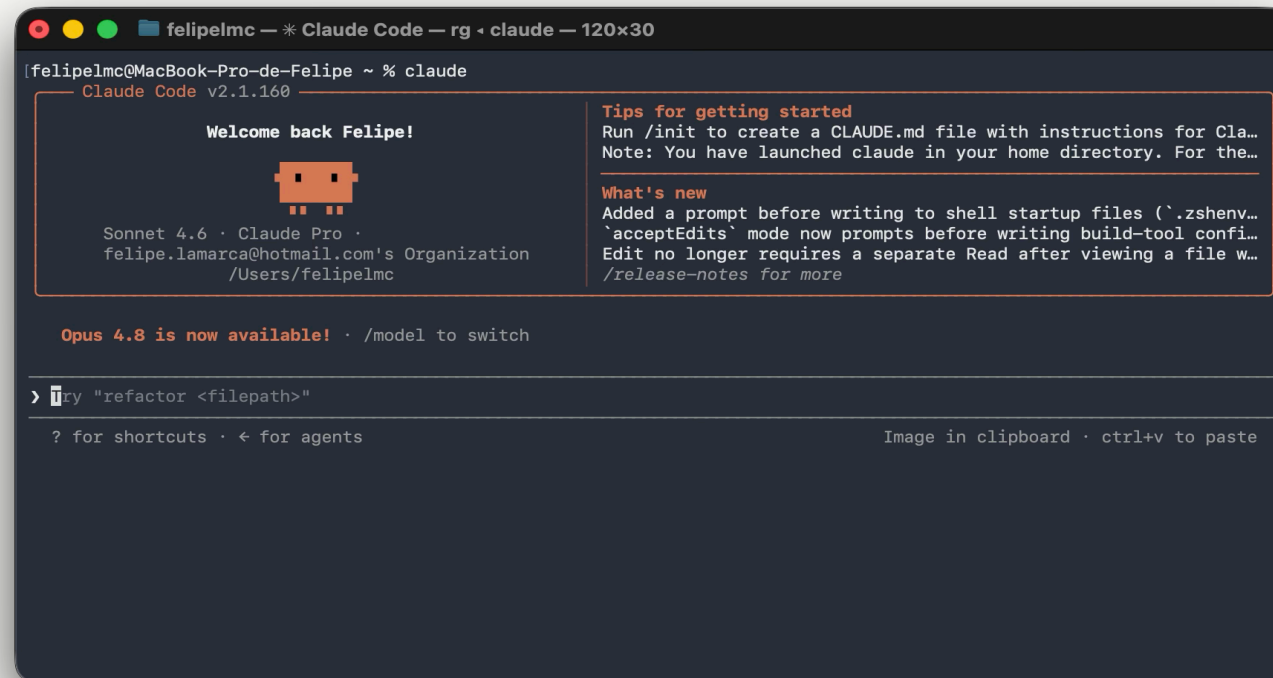
O próprio Claude guia a instalação — detecta o sistema operacional, instala as dependências e verifica que funcionou

Integrações com IDEs

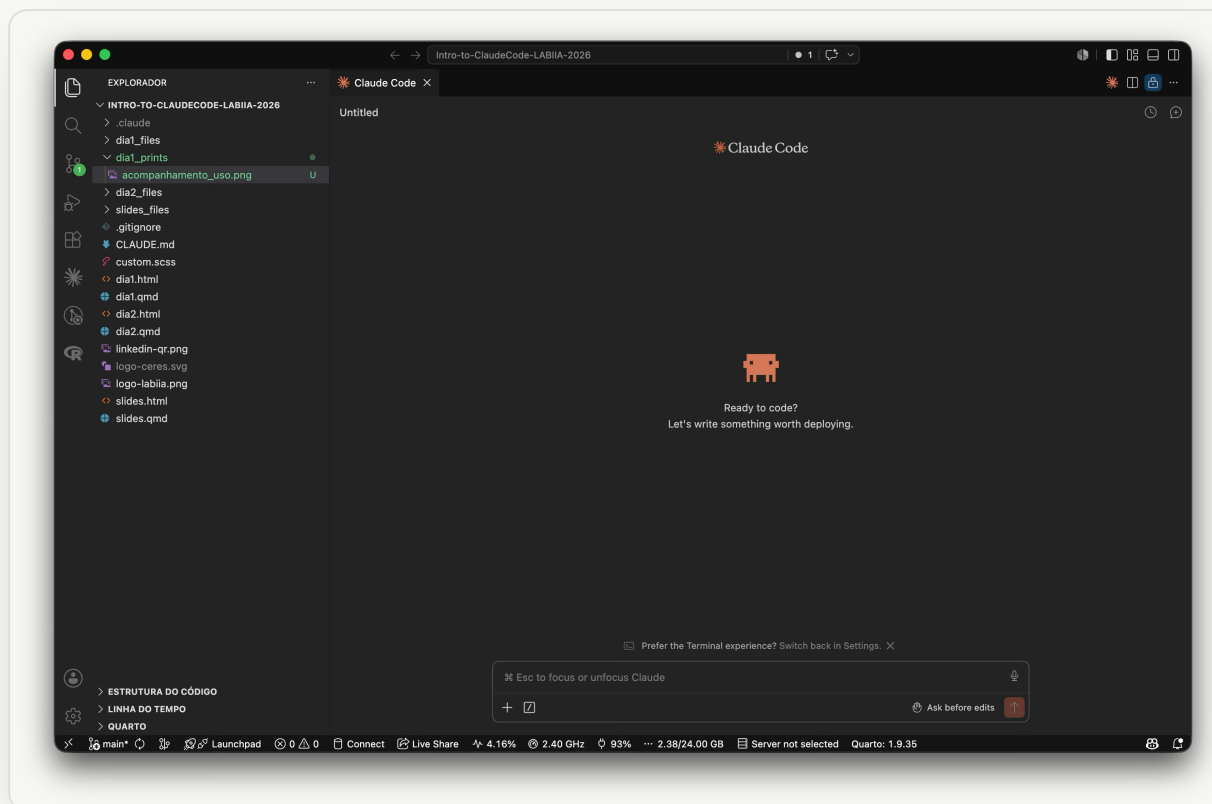
Terminal puro

O modo mais direto: abra o terminal, navegue até o projeto e execute `claude`

```
1 cd meu-projeto/  
2 claude
```



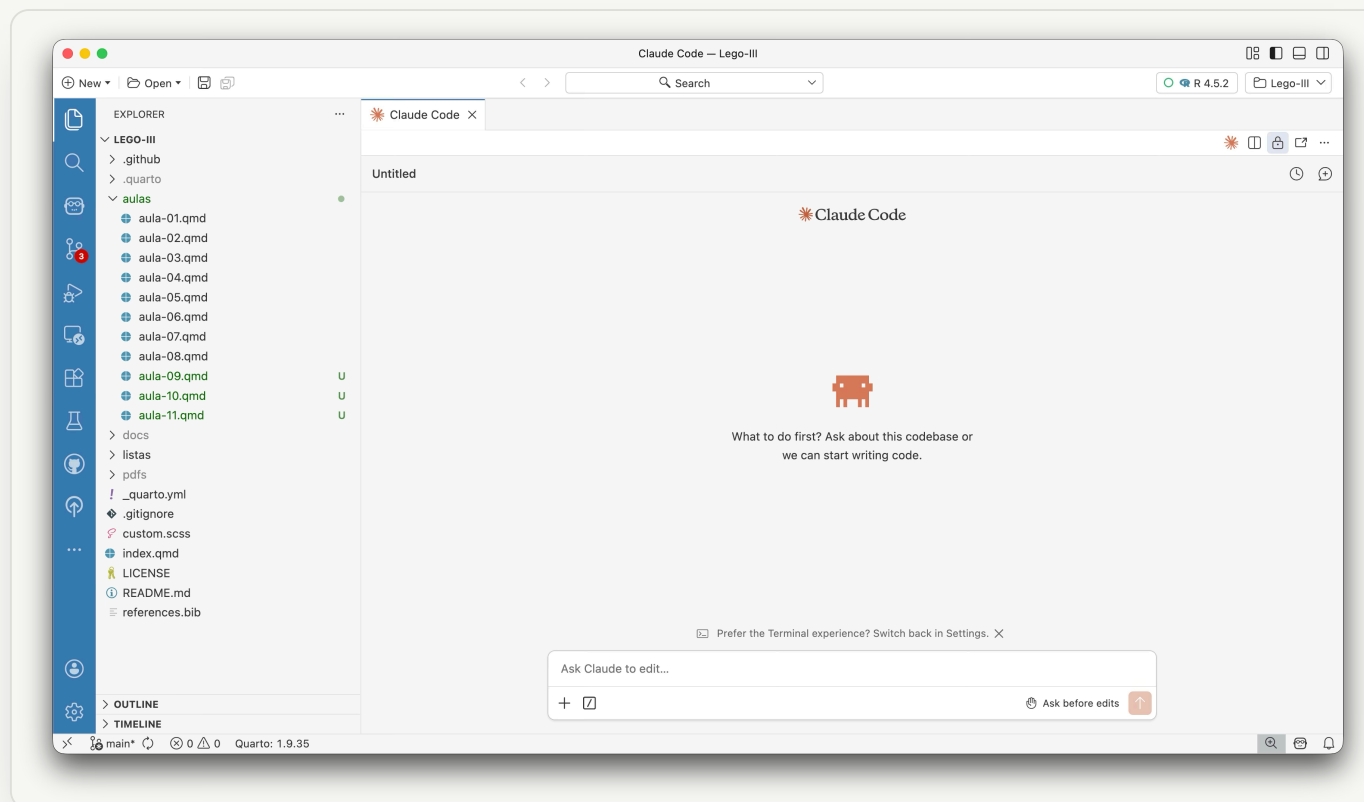
1. Abrir a aba de extensões (`Cmd+Shift+X` / `Ctrl+Shift+X`)
2. Buscar “**Claude Code**” e instalar
3. Ícone aparece na barra lateral



Para quem usa **R** — baseado no VSCode, o mecanismo é o mesmo

1. Instalar a extensão “**Claude Code**”

2. Abrir o painel lateral



De volta à distribuição normal

Com o Claude Code instalado e integrado ao VSCode, voltemos ao exemplo da demo inicial:

- Abrir a pasta com o script gerado antes da instalação
- Usar o painel do VSCode para pedir modificações: ajustar parâmetros, adicionar labels, exportar a figura
- O agente edita o arquivo real, roda no terminal integrado, você acompanha o diff em tempo real

A diferença em relação ao chat: o agente está *dentro* do seu ambiente de trabalho

Configurando o ambiente

CLAUDE.md: as regras do jogo

Arquivo de instruções persistentes, lido automaticamente pelo agente a cada nova sessão

O que colocar:

- Contexto do projeto e objetivos
- Convenções técnicas (linguagem, estrutura de pastas)
- **Decisões analíticas já tomadas** — e por quê

É a **memória permanente** do agente no projeto: o que sobrevive ao `/clear`

Um CLAUDE.md para pesquisa

```
1 # CLAUDE.md
2
3 ## Contexto
4 Projeto de análise de gastos parlamentares (CEAP 2023).
5 Dataset: `data/ceap_2023.csv` – não modificar o arquivo original.
6
7 ## Convenções
8 - Usar Python 3.11+ com pandas e matplotlib
9 - Salvar gráficos em `output/figures/`
10 - Cada análise em script separado
11
12 ## Decisões analíticas
13 - Hipótese: deputados de partidos menores gastam mais em passagens
14 - Filtrar somente despesas do exercício corrente
15 - Unidade de análise: deputado × categoria de gasto
16 - Excluir valores negativos (estornos)
```

A janela de contexto

Tudo que o modelo “vê” ao mesmo tempo cresce a cada iteração

INÍCIO DA SESSÃO

carregue e explore os dados

 aqui está o código



MUITAS ITERAÇÕES DEPOIS

carregue e explore os dados

 aqui está o código

 erro na linha 42

corrija o erro e plote

 [versão corrigida + gráfico]

 LIMITE DA JANELA

Estratégias de gerenciamento

Estratégia	O que faz
<code>/clear</code>	Reinicia a sessão, preserva o CLAUDE.md
Compaction automática	O Claude resume o contexto sozinho quando necessário
Sessões focadas	Uma tarefa por sessão
CLAUDE.md	Memória que sobrevive ao <code>/clear</code>

Tarefas grandes = múltiplas sessões pequenas e focadas

Skills: comandos customizados ⚡

`/nome-da-skill` executa um *prompt* reutilizável, definido num arquivo `.md`

- Arquivo em `.claude/commands/` (projeto) ou `~/ .claude/commands/` (global)
- Portáteis entre projetos: copie a pasta e as skills vão junto
- Encapsulam o seu vocabulário e as suas convenções analíticas

Na prática: é o mecanismo de **ensinar o agente a trabalhar do jeito que você trabalha**

Criando uma skill

Criar o arquivo `.claude/commands/summarize.md`:

- 1 Você vai receber um texto acadêmico em português.
- 2 Produza um resumo estruturado com:
- 3 1. Pergunta de pesquisa (1 frase)
- 4 2. Método principal (1 frase)
- 5 3. Resultado central (2 frases)
- 6 4. Limitações reconhecidas pelos autores (bullet points)
- 7
- 8 Escreva de forma direta, sem jargão desnecessário.

Uso: `/summarize` — cole o abstract ou o artigo inteiro

Skills de pesquisa

Skill	O que faz
<code>/voice</code>	Reescreve texto no seu estilo de escrita
<code>/summarize</code>	Resume artigos: pergunta, método, achado, limites
<code>/referee-2</code>	Revisão em instância nova do terminal — “não peça ao mesmo Claude que escreveu o código para revisá-lo”
<code>/blindspot</code>	Identifica análises que você não pensou em fazer

`/referee-2` e `/blindspot` são do workflow de Scott Cunningham
(github.com/scunning1975/MixtapeTools)

A pasta `.claude`

```
projeto/
├── CLAUDE.md          ← instruções permanentes do projeto
└── .claude/
    ├── commands/      ← suas skills customizadas
    │   ├── voice.md
    │   └── summarize.md
    └── settings.json   ← permissões e configurações
```

E na sua máquina, fora de qualquer projeto:

```
~/.claude/
└── commands/          ← skills globais, disponíveis em todo projeto
    └── voice.md
```

Demo ao vivo: CLAUDE.md num repositório real

Um repositório existente que **nunca teve Claude Code** — o agente vai inferir o contexto:

1. Abrir Claude Code na raiz do projeto
2. Pedir: *“leia os arquivos deste repositório e crie um CLAUDE.md adequado”*
3. Revisar o que o agente inferiu sobre o projeto

Demo ao vivo: criando uma skill

Em outro repositório, criar `.claude/commands/summarize.md` ao vivo

Depois: `/summarize` + colar um abstract da área para testar imediatamente

Workflows de pesquisadores

Como dois acadêmicos configuram seus ambientes — os repositórios são públicos:

Scott Cunningham github.com/scunning1975/MixtapeTools

Manoel Galdino github.com/mgaldino/agents-workflow

A pasta `.claude` deles é pública — você pode copiar, adaptar e construir em cima

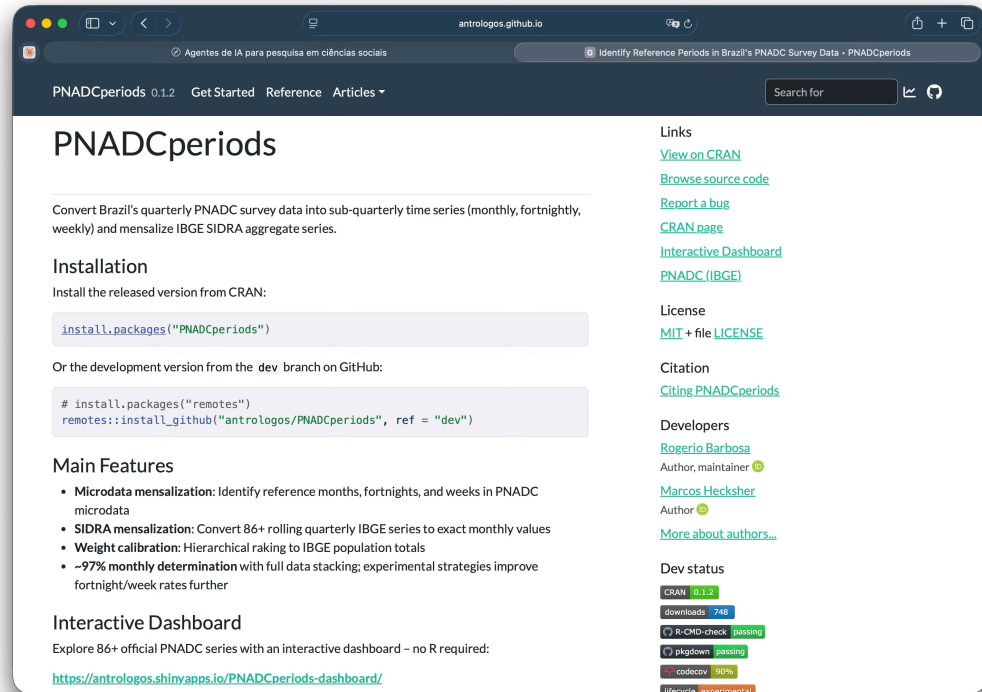
Sites de projeto e sites pessoais

O que é possível criar

Dois exemplos de páginas de projeto geradas com Claude Code:

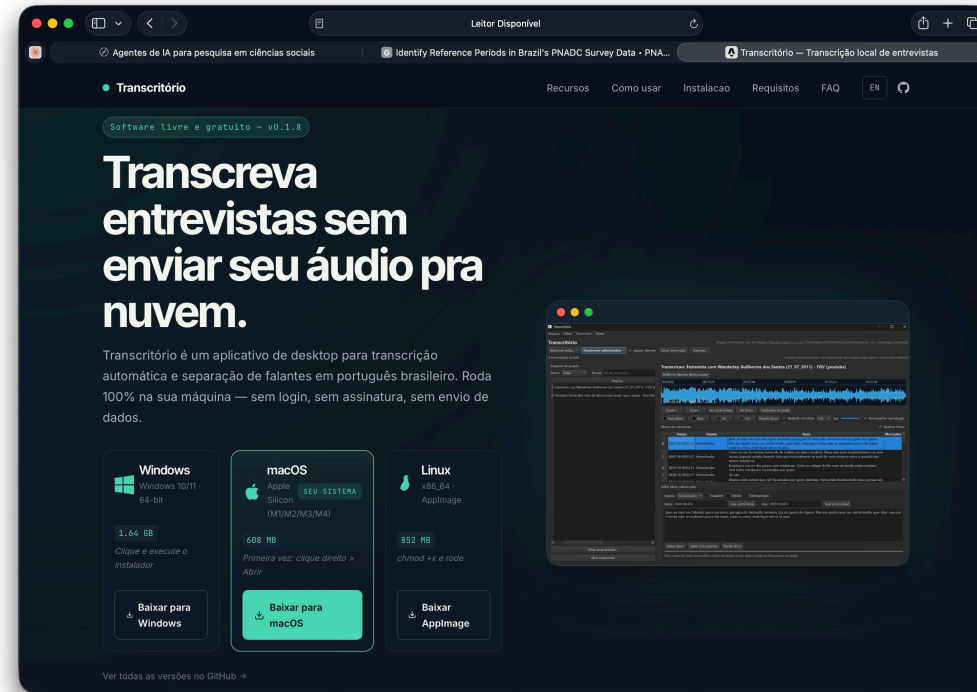
Documentação de pacote R

`antrologos.github.io/PNADCperiods/`



Landing page de projeto

`antrologos.github.io/Transcritorio/pt/`



Hoje: do currículo ao site pessoal

Input

- Currículo em texto simples ou PDF
- Uma instrução ao agente

Output

- Página HTML/CSS completa e responsiva
- Publicável no GitHub Pages sem configuração manual

O mesmo raciocínio vale para qualquer material: artigo, pacote, projeto de lab — o gap entre “tenho o conteúdo” e “tenho um site” é menor do que parece

Roteiro da demo ao vivo

1. Abrir o Claude Code no terminal
2. Fornecer o currículo como contexto
3. Pedir a geração da página HTML/CSS
4. Revisar o resultado no browser
5. Publicar no GitHub Pages

O agente lê, estrutura, estiliza e entrega — você revisa e publica

O que vem no Dia 2

Amanhã vamos usar o Claude Code para pesquisa empírica de verdade:

- **Análise de dados:** CEAP — a cota parlamentar dos deputados federais
- **Uso de APIs:** Câmara dos Deputados, dados abertos, sem autenticação
- **Boas práticas:** reprodutibilidade, o que não delegar, limites e ética

Até amanhã 🖐️

O que levar pra casa

- **CLAUDE.md sempre:** antes de começar qualquer projeto novo, escreva o contexto e as decisões analíticas
- **Skills para tarefas recorrentes:** `/voice`, `/summarize` — o investimento se paga na segunda vez
- **Sessões focadas:** uma tarefa por sessão rende mais do que uma sessão para tudo

Recursos: **claude.ai/code** · Workflows: Cunningham (`scunning1975/MixtapeTools`) · Galdino (`mgaldino/agents-workflow`)

Material do curso: **github.com/felipelmc/Presentations**