



CERES
Centro para o
Estudo da Pesquisa e da
Estratificação Social

WORKSHOP CERES 2026.1



Agentes de IA para pesquisa em ciências sociais

uma introdução ao Claude Code

Felipe Lamarca

IESP-UERJ · MAPE · DOXA · NECON

Feito com Claude Code 😊

13 de maio de 2026

- 01 **Do chat ao agente:** o que muda na prática
- 02 **Claude Code na prática:** CLAUDE.md, skills e contexto
- 03 **Três aplicações:** demos ao vivo
- 04 **Limites e recursos:** o que não delegar

Do chat ao agente

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

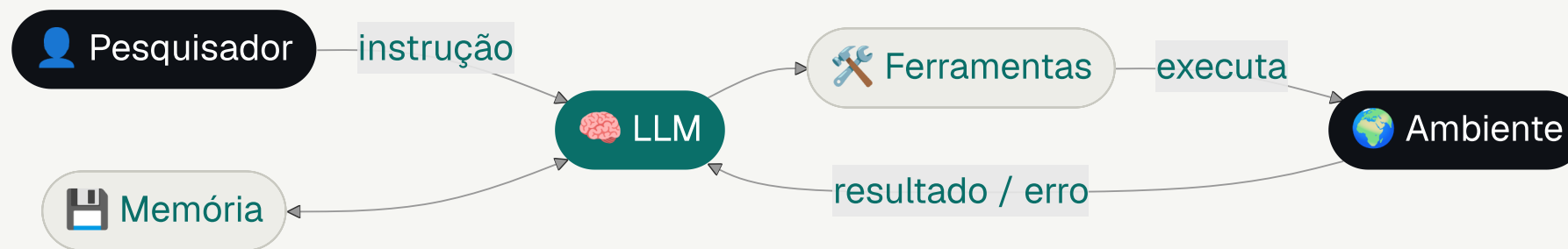
- Sem acesso ao seu ambiente de trabalho (arquivos, código, terminal)
- Sem memória entre sessões
- Cada etapa do ciclo de pesquisa – coleta, análise, documentação, comunicação – exige sua coordenação manual

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
- 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.

Claude Code na prática

Primeiros passos

Mac / Linux

```
1 npm install -g @anthropic-ai/claude-code  
2 claude
```

Windows

Baixar o app em claude.com/download · instalar **Git for Windows** · abrir a aba **Code** no app

Quanto custa? 💰

Plano	Preço	Para quem
Pro	US\$ 20/mês	Uso individual (meu plano)
Max 5×	US\$ 100/mês	Uso intensivo
Max 20×	US\$ 200/mês	Uso muito intensivo
API	Por token	Desenvolvedores
Team / Enterprise	A partir de US\$ 30/usuário	Organizações

O plano Pro tem limites de uso, mas permite um bom aproveitamento para pesquisa acadêmica

CLAUDE.md: as regras do jogo

Arquivo de instruções persistentes, lido automaticamente a cada sessão

O que colocar:

- Contexto do projeto e objetivos
- Convenções e restrições analíticas
- Decisões já tomadas (e por quê)

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

Skills: comandos customizados ⚡

`/nome-da-skill` executa um *prompt* reutilizável, portátil entre projetos

```
1 /voice          # reescreve texto no seu estilo
2 /document-notes # organiza anotações brutas
3 /review         # revisa código ou argumento
```

Na prática, é o mecanismo de ensinar o agente a executar tarefas no seu vocabulário e com suas convenções

Gerenciamento de contexto

A janela de contexto

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

INÍCIO

como coletar os dados?

 aqui está o código

ALGUMAS TROCAS DEPOIS

como coletar os dados?

 aqui está o código

 erro na linha 12

corrija o erro

 [versão corrigida]

 dados.csv

MUITAS ITERAÇÕES DEPOIS

como coletar os dados?

 aqui está o código

 erro na linha 12

corrija o erro

 [versão corrigida]

 dados.csv

agora plote os dados

 [mais código...]

 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
Sessões focadas	Uma tarefa por sessão
CLAUDE.md	Memória que sobrevive ao <code>/clear</code>

Tarefas grandes = múltiplas sessões pequenas e focadas

Três aplicações práticas

As demos de hoje



Dados eleitorais

Análise e modelagem



Skill de voz

Escreve como você



Site pessoal

Do currículo ao deploy

Demo 1: análise de dados eleitorais

A ideia é simples: usar o agente para construir, do zero, um protótipo de pesquisa empírica sobre eleições.

Roteiro ao vivo:

1. Carregar e explorar dados eleitorais do TSE
2. Análise exploratória: distribuições, padrões regionais, visualizações
3. Ajustar modelos de regressão e interpretar os coeficientes

Demo 2: skill de “voice”

Você tem um estilo de escrita. O modelo pode aprendê-lo.

Como criar: (i) reunir *writing samples* seus (artigos, e-mails, textos); (ii) criar o arquivo de skill `/voice`

Uso: pedir ao Claude que revise ou reescreva textos no seu estilo

Demo 3: site pessoal a partir do currículo

- Input: currículo em PDF ou texto simples
- Output: site HTML/CSS completo e responsivo
- Publicável no GitHub Pages sem configuração manual

O mesmo raciocínio vale para páginas de projeto: o gap entre ter um artigo com código e ter uma página sobre ele é menor do que parece – o Claude Code lê o repositório e produz o HTML.

Para além do workshop

Problemas parcialmente contornáveis

Problema	Mitigação
Erros e alucinações	Revisão sistemática; agentes paralelos para verificação cruzada
Degradação do contexto	Sessões focadas por tarefa; compactação automática
Deriva de escopo	CLAUDE.md com restrições analíticas explícitas
Reprodutibilidade	Documentar decisões, não apenas o código gerado

Problemas incontornáveis

- A interpretação substantiva dos achados é do pesquisador
- As decisões de *design*: o que medir, como operacionalizar, quais hipóteses testar
- A validação: o agente sabe se o código roda; você verifica se o achado faz sentido
- O uso de IA na pesquisa ainda está sendo estudado – há literatura nova saindo com frequência

Em última instância, o pesquisador responde por tudo que assina – inclusive pelo que o agente produziu.

Bônus: `/insights`

Skill que analisa o histórico de sessões e gera um relatório personalizado de uso

O que o relatório traz:

- Áreas de projeto e padrão de interação
- O que está funcionando – e o que gera atrito
- Sugestões de adições ao `CLAUDE.md` e de skills a criar
- Workflows mais ambiciosos a explorar

“Claude confidently blamed an ‘expired’ Gemini API key – the real culprit was just a wrong model name”

Recursos úteis

Documentação oficial: claude.ai/code

Seções recomendadas:

- *How Claude Code uses memory*
- *Claude Code hooks*
- *Model Context Protocol (MCP)*

Obrigado!

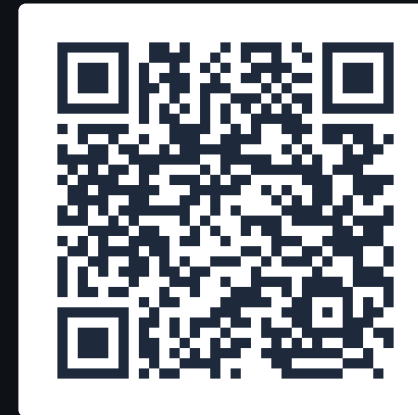
Felipe Lamarca

IESP-UERJ · MAPE · DOXA · NECON

✉ felipe.lamarca@hotmail.com

🌐 felipelamarca.com

📁 github.com/felipelmc/Presentations/tree/main/Intro-to-ClaudeCode-CERES-2026



🔗 [LinkedIn](#)