BIOINFORMÁTICA NA SELVA DA INTELIGENCIA ARTIFICIAL

Um guia para quem atua em bioinformática sobreviver na selva da inteligência artificial

Jorge Muñoz
Matheus Pimenta

Roteiro

01

Introduçao - Motivação

02

Discussões de pontos chave - Opiniões

03

Conceitos

04

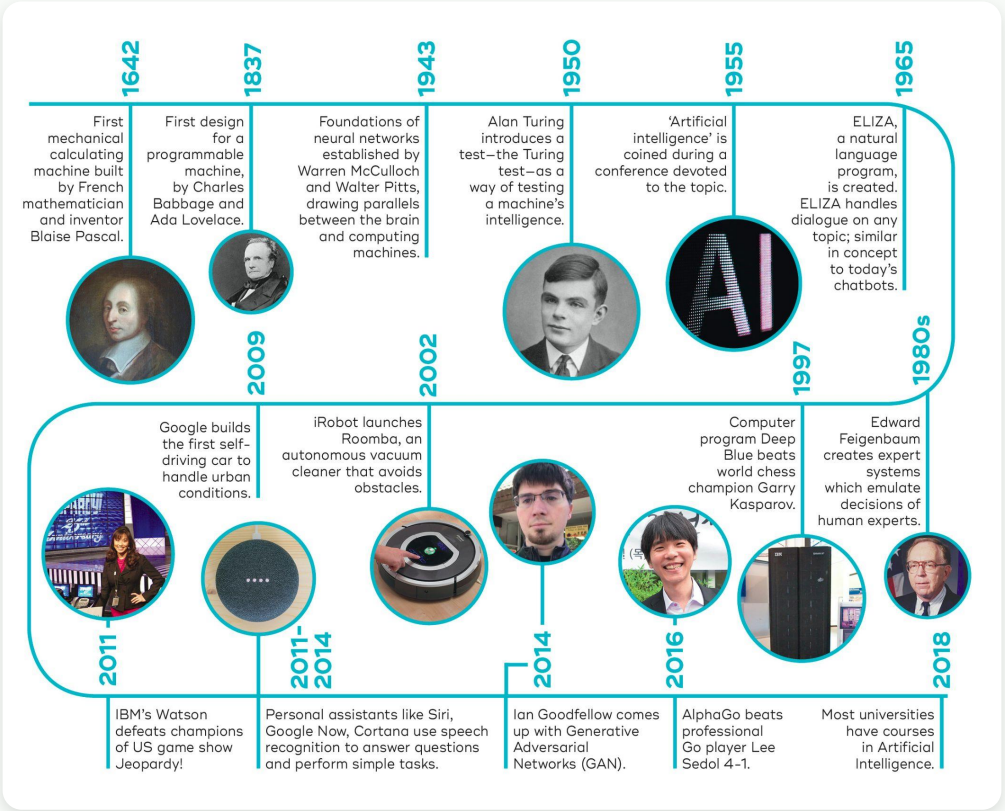
Guia pratico de como usar IA

ROTEIRO

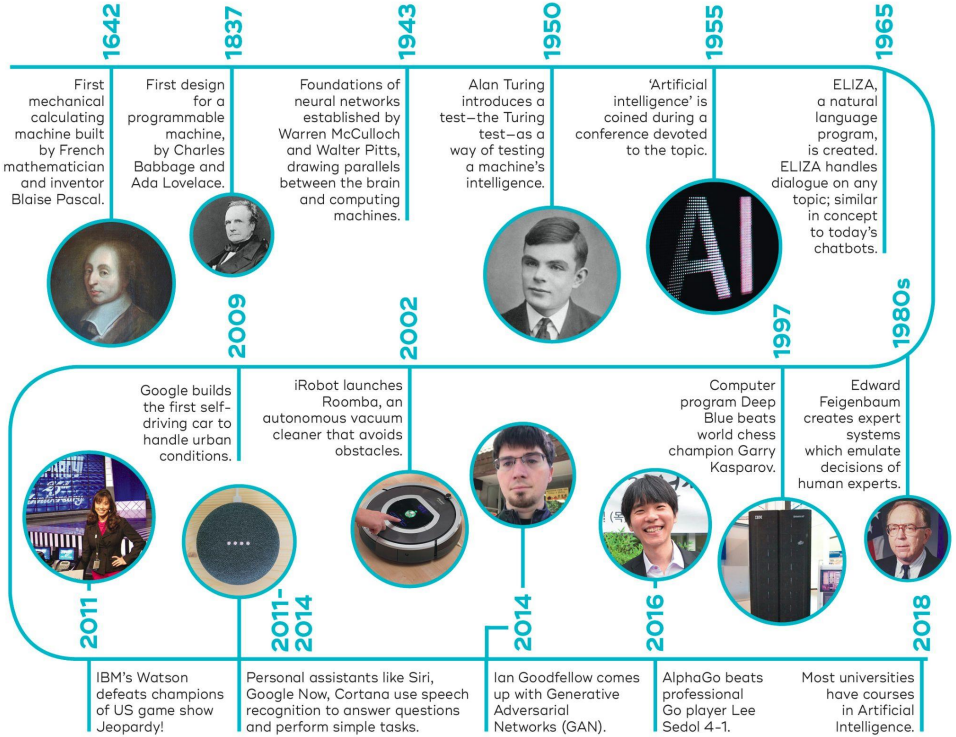
01

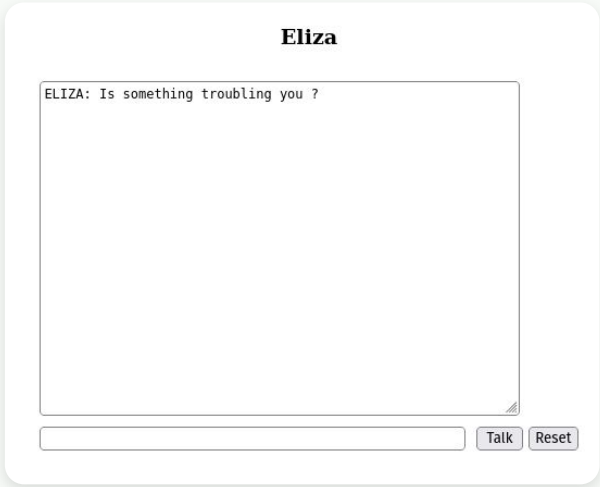
Introduçao - Motivação

HISTÓRIA DA INTELIGÊNCIA ARTIFICIAL



HISTÓRIA DA INTELIGÊNCIA ARTIFICIAL



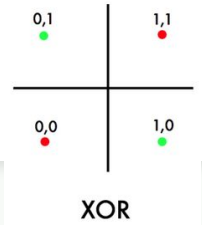


OS INVERNOS DA INTELIGÊNCIA ARTIFICIAL

Inverno: as promessas não se cumprem e o financiamento e o interesse desabam

1º inverno (~1974–1980)

- Minsky & Papert (1969): o perceptron não resolve o XOR *(mantenha a figura)*
- Relatório Lighthill (1973) e cortes da DARPA
- Pouco poder computacional e poucos dados



Retomada (anos 1980)

sistemas especialistas e backpropagation (1986), que resolve o XOR com redes multicamadas

2º inverno (~1987–1993)

- Sistemas especialistas caros, frágeis e difíceis de manter
- Colapso das máquinas Lisp e fracasso do projeto japonês de 5ª geração

Saída do inverno (2006–2016)

deep learning + GPUs + big data → AlexNet (2012), AlphaGo (2016)

Lição: a expectativa ficava acima da realidade técnica

01 · INTRODUÇÃO

DO TRANSFORMER AO BOOM (2017 → hoje)

DO TRANSFORMER AO BOOM (2017 → hoje)

2017

“Attention Is All You Need”

surge o Transformer, a base de todos os modelos atuais

2018

GPT-1 (OpenAI) BERT (Google)

pré-treino em grandes volumes de texto

2019

GPT-2

a OpenAI hesita em liberá-lo por medo de mau uso

2020

GPT-3

175 bilhões de parâmetros; escalar funciona

2022

ChatGPT

a IA chega ao público, com ~100 milhões de usuários em ~2 meses

2023

A corrida começa

GPT-4, Claude (Anthropic), Llama (Meta, pesos abertos), Gemini (Google)

DO TRANSFORMER AO BOOM (2017 → hoje)

2024–2025

Modelos que "raciocinam" e agentes; o DeepSeek-R1 (jan/2025) mostra que a China compete com modelos abertos e mais baratos

2026 – Hoje

desde 1º de setembro, a OpenAI lançou a família GPT-6, a Anthropic lançou três modelos Claude 5.5, o Google liberou o Gemini 4 Argon para todos e a Meta lançou o Muse. No lado aberto, DeepSeek V4.1 Flash e V4 Pro, GLM-5.3, Kimi K3 e os modelos menores Qwen estão disponíveis com pesos abertos

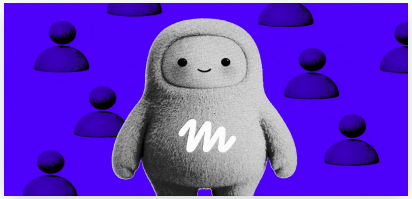
ChatGPT

Claude

Gemini

deepseek

GLM-5.3



Qwen

ROTEIRO

02

Discussões de pontos chave - Opiniões

Discussões de pontos chave - Opiniões

1. **Dica 1:** Abrace a tecnología e esteja aberto a novidades.
2. **Dica 2:** Melhore a legibilidade do código e a documentação.
3. **Dica 3:** Escreva código de forma eficiente.
4. **Dica 4:** Use a IA para aprimorar a limpeza de dados.
5. **Dica 5:** Use a IA para aprimorar a visualização de dados.
6. **Dica 6:** Use a IA para melhorar sua escrita.
7. **Dica 7:** Certifique-se de entender ou de saber como testar o que a IA gera.
8. **Dica 8:** Aprenda os fundamentos da engenharia e elaboração de prompts.
9. **Dica 9:** Considere usar APIs de IA para expandir as funcionalidades das suas aplicações.
10. **Dica 10:** Não se torne excessivamente dependente da IA.

ROTEIRO

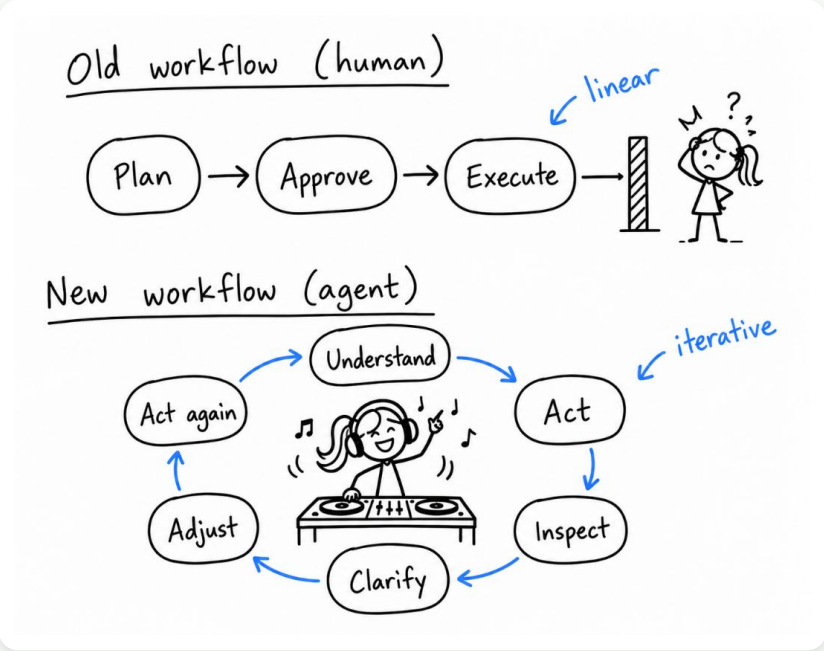
03

Conceitos

Conceitos IA atual: Fluxo de trabalho agêntico

Fluxo de trabalho agêntico

- Um LLM sozinho: entra texto, sai texto. Não executa, não verifica, não corrige.
- Fluxo agêntico: o modelo recebe um **objetivo**, não uma pergunta isolada.
- Funciona em ciclo: **Pensar → Agir → Observar → Decidir**.
- O modelo **escolhe os próximos passos em tempo real**, com base no que observa.
- Repete o ciclo até cumprir o objetivo (ou pedir ajuda).



https://www.aymannadeem.com/artificial/intelligence,/developer/tools/2026/09/24/plan-mode-is-dead.html

Conceitos IA atual: Fluxo de trabalho agêntico

Pipeline vs. Agente

Pipeline (Nextflow/nf-core):

DAG definido antes, mesmo caminho a cada execução.

Previsível e reprodutível.

vs

Agente:

caminho decidido durante a execução.

Flexível e adaptativo, porém menos previsível.

O agente se parece mais com um pós-doc no terminal do que com um script.

Reprodutibilidade é um ponto crítico em bioinformática.

Conceitos IA atual: Fluxo de trabalho agêntico

Os componentes

Modelo (LLM): o motor de raciocínio. Sozinho, só gera texto.

Ferramentas (tools): ações que o modelo pode *solicitar*, como rodar bash, ler arquivos ou buscar no PubMed.

Harness: o software que envolve o modelo e executa o ciclo.

- Executa as ferramentas e devolve os resultados.
- Gerencia o contexto, as permissões e o sandbox.
- Exemplos: Claude Code, Codex CLI, Gemini CLI.

Skills: conhecimento empacotado e reutilizável, carregado **só quando necessário**.

- Funcionam como POPs/protocolos de laboratório.
- Exemplo: “como o lab roda DESeq2”.

Peças complementares

Janela de contexto: a memória de trabalho do agente, que é finita.

MCP: padrão para conectar serviços externos (bancos de dados, Drive, APIs).

Subagentes: o agente principal delega subtarefas a agentes auxiliares.

Humano no ciclo: pausas para aprovação antes de ações arriscadas.

ROTEIRO

04

Guia pratico de como usar IA

Como usamos IA hoje? Três perfis no espectro

O conversador

- Abre o navegador, pergunta e copia a resposta
- Uso igual ao de 2023, ainda que a ferramenta tenha evoluído
- **Risco:** aceitar sem verificar (alucinações)

O Arquiteto

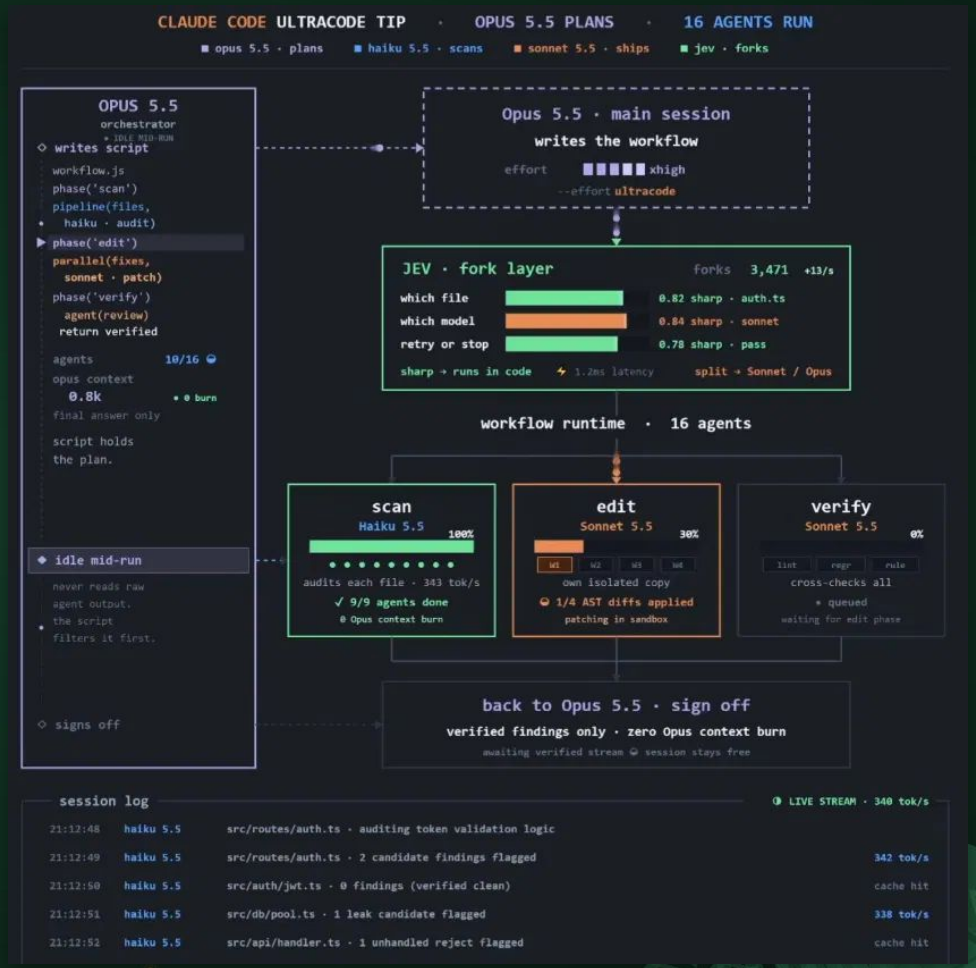
- Agentes, subagentes, orquestração, dezenas de skills e servidores MCP, specs, loops
- Uma engenharia enorme, muitas vezes para um problema pequeno
- **Risco:** passar mais tempo montando o andaime do que resolvendo o problema

O Pragmático

- Usa o harness da própria empresa do modelo (Claude Code, Codex, Antigravity...)
- Investe em um bom prompt: contexto claro, objetivo claro, critério de pronto
- Deixa o harness fazer o trabalho pesado

04 · GUIA PRÁTICO

O exagero do Arquiteto



“Just Prompt”: o manifesto

Quem são: um grupo de programadores cansados de ouvir, toda semana, que tudo o que aprenderam sobre IA na semana anterior já está obsoleto

A crítica, (o que dizem que você precisa → o que isso realmente significa):

Harness engineering	→	um harness para te vender
Agent orchestration	→	mais tokens queimados
Vinte servidores MCP	→	**dezenove que você nunca chama**

A ideia central: quem consegue explicar um problema a um colega consegue escrever um prompt, e nenhuma quantidade de “harness” ajuda quem não consegue

> **A** habilidade que importa é **saber explicar o problema**, e isso pesquisadores já treinam a vida inteira.

“Just Prompt” : o manifesto

Por exemplo, converter um pipeline de Snakemake para Nextflow:

Conversador: cola trechos no chat e junta as respostas na mão.

Arquiteto: monta 12 agentes, um especialista em cada regra do pipeline.

Pragmático: abre o Claude Code no repositório e escreve: “Converta este pipeline para Nextflow, mantenha os mesmos outputs e rode os testes até passarem.”

Quem opera a IA faz diferença (o caso ART)

Set/2026: agentes Claude encontram um sistema enzimático com repetições parecidas com CRISPR em bacteriófagos

Foram cerca de

950
agentes,

21
horas e

210 milhões
de tokens

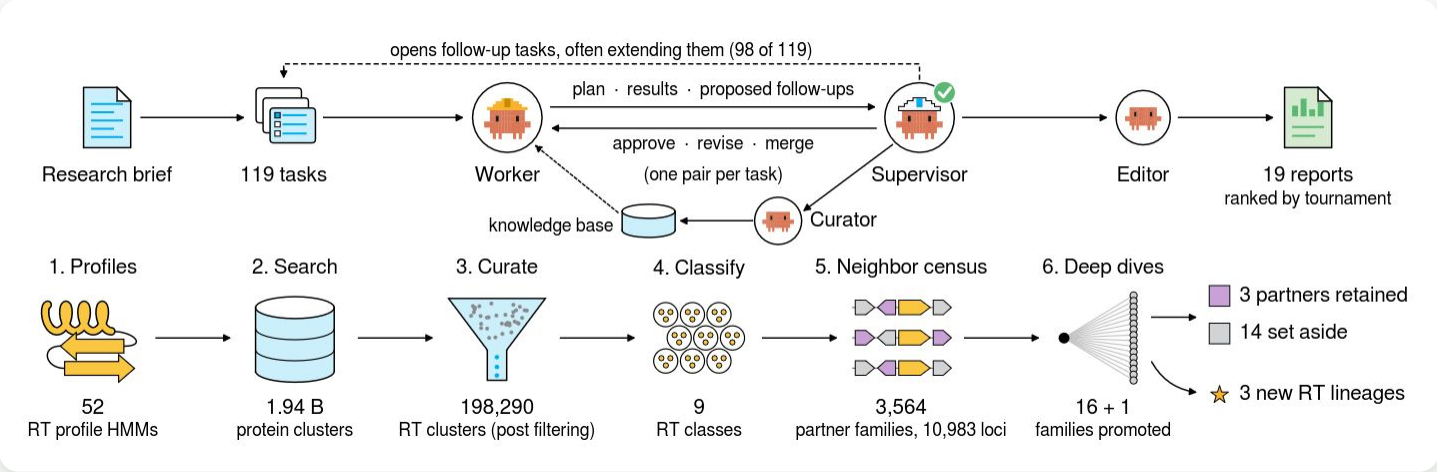
Quem conduziu: especialistas em mineração de genomas e CRISPR, que fizeram o prompt inicial e toda a validação no laboratório

Quantas vezes deu certo: 1 em 11 campanhas

O próprio time usa o que aprende avaliando as hipóteses para ajustar as instruções do Claude, ensinando-o a imitar o gosto científico deles

A IA *amplifica quem já sabe o que procurar*

Contexto vence busca

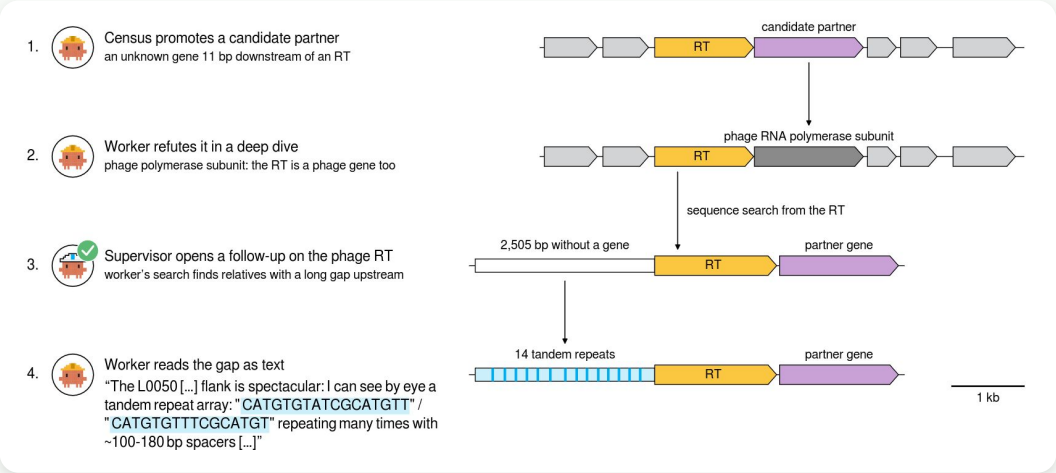


https://www-cdn.anthropic.com/22573675ada52a8ca8a97a1a4b4326b2f208a071.pdf

- Com a região de DNA entregue direto ao modelo, a detecção ficou em torno de 90%. Quando o modelo precisava encontrá-la sozinho usando ferramentas, caiu para até 32%
- Nas repetições, os agentes chegaram perto do lugar certo, mas não leram o DNA vizinho

Não espere que a IA descubra o que você já sabe. Entregue o contexto.

Contexto vence busca



https://www-cdn.anthropic.com/22573675ada52a8ca8a97a1a4b4326b2f208a071.pdf

- Com a região de DNA entregue direto ao modelo, a detecção ficou em torno de 90%. Quando o modelo precisava encontrá-la sozinho usando ferramentas, caiu para até 32%
- Nas repetições, os agentes chegaram perto do lugar certo, mas não leram o DNA vizinho

Não espere que a IA descubra o que você já sabe. Entregue o contexto.

Primeiros passos com um harness

1. Instale o harness da empresa do modelo (Claude Code, Codex, Antigravity...)

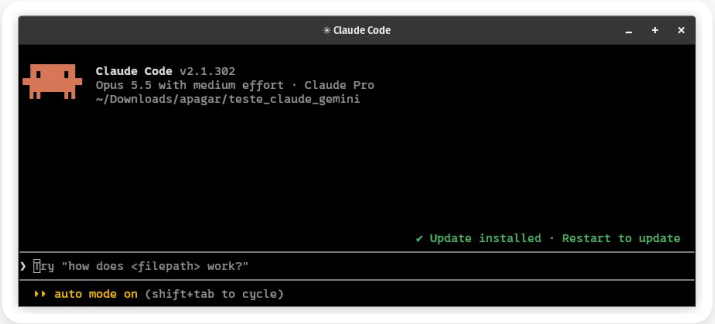
2. **Abra na pasta do projeto:** `cd meu_projeto && claude`. O harness enxerga só aquela pasta, o que já limita o escopo.

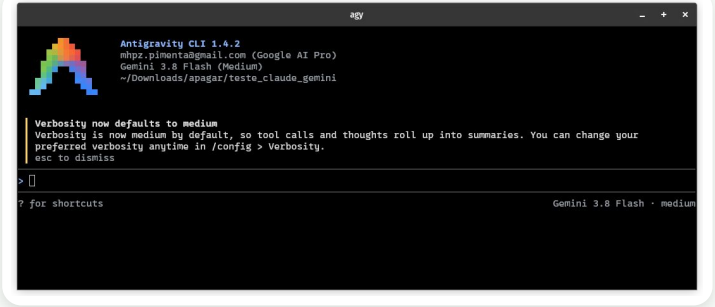
3. `/init`**:** o Claude lê o projeto e cria um `CLAUDE.md` com a "memória" dele

4. **Faça um commit no git antes de começar:** assim tudo pode ser revertido

5. **Permissões:** comece aprovando cada comando e só depois libere mais autonomia

6. **Modo plan (Shift+Tab):** o Claude planeja sem alterar nada, e você aprova antes de ele executar





Anatomia de um bom prompt

1 Objetivo:
o que você quer e por quê

2 Contexto biológico:
organismo, tipo de dado, pergunta científica

3 Mapa da pasta:
o que tem em cada diretório e o formato dos arquivos

4 Exemplos:
uma entrada e a saída esperada

5 Guardrails:
o que NÃO pode ser feito

6 Critério de pronto:
como saber que terminou e que está certo

```
"Tenho dados de RNA-seq de Arabidopsis (3 réplicas, controle vs. seca) em data/raw/ (FASTQ
paired-end). Quero uma análise de expressão diferencial com DESeq2. scripts/ tem minha
tentativa anterior. NÃO altere nada em data/raw/, NÃO instale pacotes sem perguntar, NÃO
mude limiares estatísticos sem justificar. Pronto = relatório com volcano plot, tabela de
genes com padj < 0.05 e versões de todos os pacotes registradas."
```

Guardrails para bioinformática

`data/raw/` é **somente leitura**, sempre

Não inventar parâmetros, versões de bancos de dados ou referências: na dúvida, perguntar

Não baixar genomas ou bancos grandes sem confirmar

Não “ajustar” código ou dados para o teste passar

Registrar versões, seeds e comandos para garantir reprodutibilidade

Dados sensíveis: dados de pacientes só anonimizados e conforme a LGPD e o comitê de ética

Planeje antes de executar

Rascunho livre:

escreva a ideia num bloco de notas, do jeito que sair, e peça: *“Melhore e estruture este prompt; pergunte o que estiver faltando”*

Modo plan:

o Claude propõe as etapas, e você corrige o plano, não o código

`grill-me`:

uma skill que faz o Claude te entrevistar, uma pergunta por vez, até o plano ficar sólido. Ótima para quem ainda não sabe explicar bem o problema.

Dez minutos de planejamento economizam horas de trabalho

Agentes, quando o problema é grande

- **Estrutura:** um agente supervisor, com subagentes especializados
- **O supervisor:** divide o trabalho, delega, revisa o que volta e redistribui o que falhou
- **Os subagentes:** por exemplo, um para controle de qualidade, um para alinhamento, um para a análise estatística e um revisor que tenta refutar os resultados



Como pedir:

```
"Atue como supervisor. Divida esta análise em tarefas independentes, delegue cada uma a um subagente, revise os resultados e só me entregue o que passou na sua revisão."
```

Aviso: para tarefas pequenas, um bom prompt resolve. Agentes são para problemas grandes.

Memória e reuso

`CLAUDE.md` / `AGENTS.md`:

o “manual do projeto”, com convenções, estrutura de pastas, comandos e o que não fazer. O Claude Code lê o CLAUDE.md e outros harnesses usam o AGENTS.md; dá para manter um só e referenciar no outro.

Skills (`SKILL.md`):

procedimentos reutilizáveis, como “meu pipeline padrão de QC” ou “formato do relatório do laboratório”

Superpowers:

um conjunto pronto de skills com uma metodologia inteira (planejar, testar, revisar). Vale experimentar, mas comece simples.

Se você repetiu a mesma instrução três vezes, ela vira `CLAUDE.md` ou skill

Desconfie e valide

- **Reproduza um resultado conhecido primeiro:** o time da Anthropic faz isso, pedindo ao Claude que reproduza resultados já estabelecidos com dados públicos para checar os métodos
- **Rode mais de uma vez:** um acerto isolado pode ser sorte
- **Peça autocrítica:** *"Tente refutar este resultado"*
- **Revise os diffs e rode testes:** nunca aceite código que você não leu

> *A IA não substitui o bioinformata. Ela multiplica o que ele já sabe, inclusive os erros."*

Exemplo prático

01



Rodar um RNAseq:

https://github.com/djburks/RNA-Seq-Data-Analysis-Pipeline-for-Plants

02



Fazer uma revisão bibliográfica:

(Claude + Undermind)

Obrigado!

Perguntas?

Jorge Muñoz · Matheus Pimenta