Opinião

# IA e estudantes

Eu uso IA na maior parte do meu trabalho de programação porque passei décadas aprendendo os fundamentos primeiro. Estudantes e iniciantes precisam de outra abordagem: usar a IA como apoio sem entregar o raciocínio.

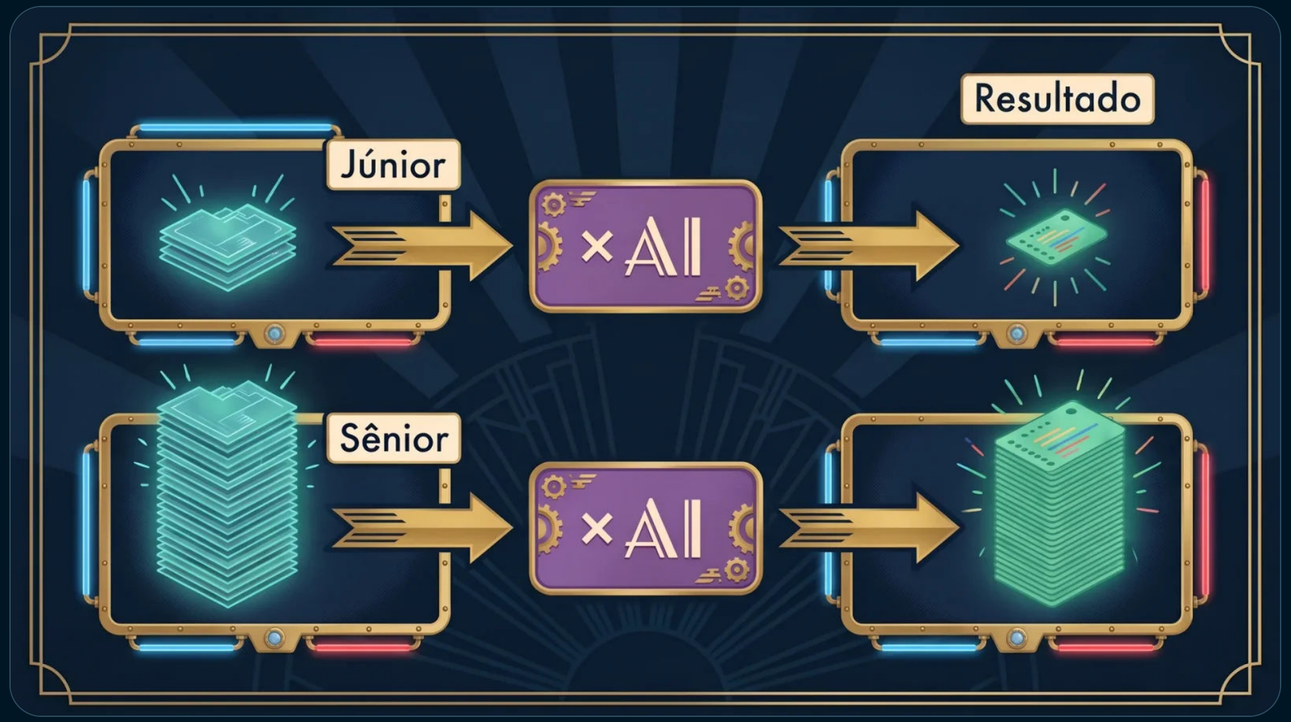
**Fabio Akita · ailair.akitaonrails.com/pt-br/students/**

> *Eu, sênior, anos de experiência, vou fazer 100% vibe codar. você, estudante, júnior, DEVE escrever o máximo possível na mão, deixar a IA fazer tarefas mais mundanas, em paralelo, e você DEVE revisar tudo manualmente. Você precisa de dezenas de milhares de horas de prática deliberada, estudo, pra se tornar alguém que, depois, vai saber o que mandar a IA fazer por você, não antes.*
>
> *Eu no X, 24 de setembro de 2026*

1. O multiplicador

# A IA multiplica o que você traz

A mesma ferramenta dá ao sênior mais trabalho bom, mais rápido, e dá ao iniciante mais do que o iniciante já tem, erros incluídos.



1. O multiplicador

# A IA multiplica o que você traz

- A IA pode multiplicar erros quando os fundamentos são fracos. Com fundamentos fortes, ela consegue multiplicar trabalho útil.
- O relatório DORA 2025 do Google viu a mesma coisa em times: "a IA não conserta um time; ela amplifica o que já está lá."
- Se você é experiente e ainda está resistindo, pare. Não tem mérito nenhum em escrever um Dockerfile ou um Makefile na mão pela centésima vez. As tarefas chatas vão pra IA; os fundamentos não.
- Quem entende sistemas operacionais, bancos de dados, redes, estruturas de dados e compiladores usa agentes como exoesqueleto. Quem não entende usa como muleta. Exoesqueleto amplifica a força; muleta só esconde a fraqueza.
- E alguém tem que conferir o resultado. No teste da Veracode de 2025 com mais de 100 modelos, 45% das amostras de código falharam nos testes de segurança e introduziram vulnerabilidades do OWASP Top 10.

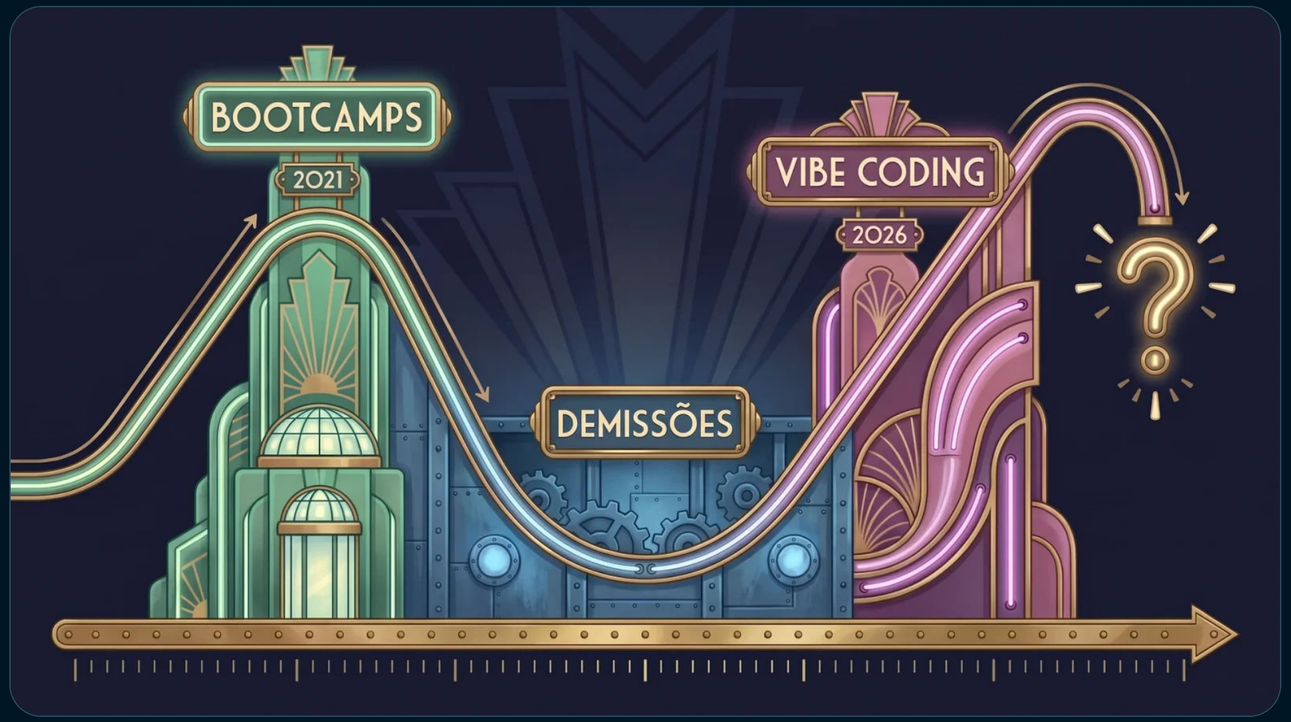
## 1. O multiplicador

> *Programador bem preparado não vai ser substituído. Disse claramente: "EU" nunca vou ser.*
>
> *Eu no X, 28 de setembro de 2026*

2. A última bolha

# A gente já viu esse atalho

Antes da IA, a promessa era o bootcamp: dois meses e você ganharia salário de Google. Acabou em fechamentos e demissões.



# A gente já viu esse atalho

- Os formados em bootcamp cresceram umas oito vezes entre 2013 e 2018. A Lambda School anunciava taxas de colocação "de até 86 por cento"; o Consumer Financial Protection Bureau viu que os números internos dela ficavam mais perto de 50%, às vezes tão baixos quanto 30%.
- Eu comecei o Akitando em 2018 exatamente pra isso: preparar programadores de verdade pra sobreviver ao estouro da bolha. Em 2022 eu estava dizendo pro pessoal cancelar os cursos e parar de acreditar em influencer.
- Agora a mesma promessa voltou com cara nova: você não precisa aprender a programar, a IA faz por você. Ninguém vira engenheiro num bootcamp de um mês, e ninguém vira engenheiro escrevendo prompt também.
- Aí o dinheiro barato acabou. Bootcamps fecharam em 2023 e 2024, a 2U anunciou no fim de 2024 que ia sair do negócio, e a indústria de tecnologia cortou mais de 150 mil vagas só naquele ano.
- O que esses cursos ensinavam, formulários, CRUD, relatórios e cópias da mesma UI, é exatamente o que os LLMs replicam melhor. A maioria dessas vagas já foi cortada depois de 2022.

3. As evidências

# Aprender no piloto automático não fixa

Os estudos até agora apontam pro mesmo lado: quando a IA pensa por você, você não aprende.

**17 pts**

menos pontos no quiz pra engenheiros, a maioria júnior, que aprenderam uma biblioteca nova com IA

**17%**

nota menor na prova pra alunos que praticaram com o GPT-4 puro, depois que ele foi tirado

**19%**

mais lentos: desenvolvedores experientes com IA, que ainda achavam que estavam mais rápidos

3. As evidências

# Aprender no piloto automático não fixa

- O próprio estudo da Anthropic, janeiro de 2026: engenheiros, a maioria júnior, aprendendo uma biblioteca nova (Trio) com IA tiraram 50% no quiz, contra 67% de quem programou na mão, sem economia de tempo significativa. Quem usou a IA pra explicar conceitos manteve a maior parte do aprendizado; quem entregou o código pra ela ficou entre as notas mais baixas.
- O experimento da METR de 2025: desenvolvedores open source experientes levaram 19% mais tempo com IA, e depois ainda achavam que ela tinha deixado eles 20% mais rápidos.
- Num estudo da PNAS com cerca de 1.000 alunos do ensino médio, o GPT-4 puro aumentou as notas nos exercícios em 48% e depois derrubou as notas na prova em 17% quando foi tirado. Um tutor feito pra fazer eles pensarem evitou a perda em grande parte.
- Um preprint do MIT, ainda sem revisão por pares, viu que quem escreveu redações com o ChatGPT teve dificuldade pra citar o próprio texto.

4. O mercado

# O caminho de entrada ficou mais estreito

As empresas ainda contratam quem sabe revisar e decidir. Contratam bem menos iniciantes pra digitar.

**65%**

menos contratação de recém-formados nas big techs do que em 2019

**4,5%**

das vagas de desenvolvimento de software são de nível de entrada (1º tri de 2026)

**19%**

menos jovens de 22 a 25 anos do que o esperado nos empregos mais expostos à IA

# O caminho de entrada ficou mais estreito

- SignalFire, 2026: a contratação de recém-formados caiu cerca de 65% nas big techs e 76% em startups em estágio inicial em comparação com 2019, enquanto a contratação de engenharia nas big techs caiu só 11%.
- O estudo "Canaries" de Stanford: o emprego de jovens de 22 a 25 anos nos trabalhos mais expostos à IA, desenvolvedores de software incluídos, está 19% abaixo de onde estaria se tivesse acompanhado os trabalhos menos expostos, principalmente por causa de menos contratações. Os autores tomam o cuidado de chamar isso de descritivo, e não causal.
- Indeed: vagas de nível de entrada eram 4,5% dos anúncios de software no início de 2026. Os anúncios cresceram quase 15% depois que o Claude Code foi lançado em fevereiro de 2025, e 71% desse crescimento foi de vagas sênior.
- Recém-formados em ciência da computação têm uma das maiores taxas de desemprego entre todos os cursos (7,0%), mas uma das menores taxas de subemprego (19%, contra 39% no geral): quando são contratados, geralmente é num emprego que exige o diploma. O caminho de entrada típico ainda é o bacharelado.

5. A lacuna

# A IA não adivinha o que você precisa

Ela faz o que você pede, e não faz o que você não pediu. Saber o que pedir é o trabalho.



5. A lacuna

# A IA não adivinha o que você precisa

- Um agente de código precisa que você explique o objetivo, o contexto, as restrições, as prioridades, o que não pode quebrar, quais caminhos são aceitáveis e como o sucesso é medido. Muito programador nunca aprendeu a dizer isso.
- Mesmo num app minúsculo, qualquer coisa feita com vibe coding que vai pra produção precisa ser revisada por um programador experiente. Isso não é opcional.
- Meu jeito de falar com um agente tem quatro partes: o que eu quero, como, o que eu não quero, e como a gente confere. Você não consegue preencher isso sem conhecer o domínio.

6. O mito da linguagem

# Testes importam mais que a linguagem

Em outubro de 2026 eu dei a mesma tarefa difícil pra 4 modelos em 7 linguagens. Nenhuma linguagem saiu na frente.

**112**

tentativas: 7 linguagens, 4 modelos, 2 condições, 2 rodadas cada

**24**

resultados corretos dos dois modelos mais fracos quando eu dei os testes

**16**

resultados corretos dos mesmos modelos sem os testes

# Testes importam mais que a linguagem

- Os dois modelos de fronteira acertaram toda tentativa elegível, em todas as linguagens.
- Aprenda a escrever testes e transforme todo bug em um teste. Isso funciona em qualquer linguagem, com qualquer modelo.
- Toda falha quebrou uma regra do próprio problema. Nenhuma foi erro de tipo, então nenhum compilador teria pegado.

7. A pergunta

# Se a IA pode programar tudo, por que aprender?

Porque ela não pode fazer tudo. Ela reproduz o que já viu, e o que ela não viu é justamente onde estão os próximos empregos.



7. A pergunta

# Se a IA pode programar tudo, por que aprender?

- Os modelos aprenderam com código público. A maior parte do código do mundo é privada, 81,5% das contribuições só no GitHub, e um problema que ninguém resolveu antes não está em nenhum dado de treino.
- Legado é a mesma história. Não existe uma comunidade open source grande de COBOL: o código que roda banco, seguradora e governo é privado, e os modelos quase nunca viram ele.
- Onde algo parecido foi publicado, a IA replica rápido. Onde não foi, ela gira em loops longos e caros, e alguém tem que assumir: definir o que é "certo", construir o oráculo, escrever testes com alvos objetivos, caçar a referência que ninguém publicou e perceber quando o agente começou a girar em círculos.
- Cada um desses trabalhos precisa de alguém que entende o problema de baixo pra cima: como computadores, sistemas operacionais, compiladores, redes e bancos de dados funcionam de verdade. É o conhecimento difícil que a maioria pula.

7. A pergunta

# Se a IA pode programar tudo, por que aprender?

- Na minha opinião, essa é a oportunidade. Quanto menos gente disposta a aprender os fundamentos do jeito difícil, mais valem os que aprendem. Eles vão ser a geração que fecha as lacunas que os modelos não conseguem fechar sozinhos.

> *A habilidade que a máquina mais precisa de você é a que ela menos consegue aprender com o treino dela.*

8. Um exemplo real

# O que "a IA fez" custou de verdade

O ai-memory, meu sistema de memória open source pra agentes de código, é escrito na maior parte por IA. Tudo que mantém ele correto veio da minha review e da minha direção, semana após semana.



8. Um exemplo real

# O que "a IA fez" custou de verdade

**4.743**
palavras de regras pros agentes no AGENTS.md

**16**
invariantes que nunca podem quebrar

**22**
fronteiras de segurança garantidas, cada uma ligada a uma guarda e a testes adversariais

**3.000+**
testes unitários

**600+**
testes de integração

- Cerca de 65 diretivas explícitas, 28 notas de memória com curadoria que existem por causa de erros específicos, e mais de 60 documentos com pesquisa, post-mortems e decisões de arquitetura.
- Cada camada marca um lugar onde a IA sozinha não acertou e eu tive que saber como consertar.
- Nada disso apareceu no primeiro dia. São semanas vendo a IA fazer o grosso do trabalho todo dia e refinando, e o refinamento depende de experiência.

## 8. Um exemplo real

> *Nada disso é automático, feito pela IA. Tudo isso é onde a IA sozinha não consegue fazer e eu preciso saber com ajustar. Nenhum amador consegue fazer isso.*
>
> *Eu no X, 25 de setembro de 2026*

9. O que fazer

# Se você é estudante ou júnior

Use a IA do jeito que um sênior usa um estagiário: pras tarefas chatas. O aprendizado tem que passar pelas suas mãos.



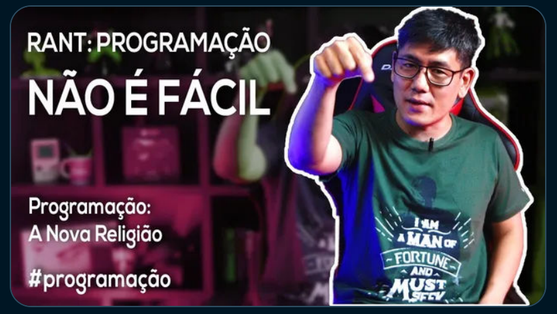
9. O que fazer

# Se você é estudante ou júnior

- Escreva o máximo que puder na mão. Quebre, debugue, conserte. Ficar travado faz parte: os pesquisadores da Anthropic escrevem que o esforço cognitivo, "mesmo ficar dolorosamente travado", provavelmente é importante pra maestria.
- Revise tudo que ela produz, linha por linha. Se você não consegue explicar uma mudança, você não está pronto pra aceitar ela.
- Sênior é alguém que já foi júnior em cada assunto que domina.
- Deixe a IA cuidar das tarefas mundanas em paralelo, e peça pra ela explicar conceitos, não pra escrever o código com o qual você deveria aprender.
- Conte as horas. Do zero ao primeiro emprego de júnior são pelo menos umas 1.400 horas de estudo e prática; na minha opinião, maestria leva dezenas de milhares. Os melhores violinistas do Ericsson tinham mais de 10.000 aos 20 anos, e pesquisas posteriores mostram que prática não é a história toda, mas nada substitui ela.

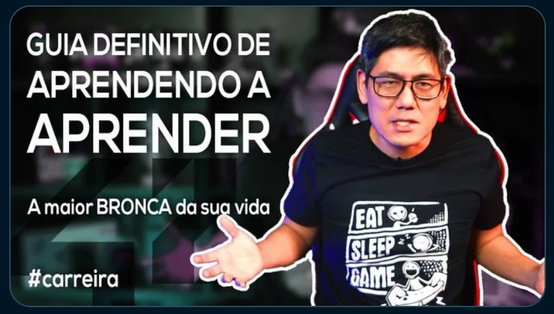
◈ Por onde começar

# O arquivo do Akitando




RANT: Programação NÃO É Fácil

fev. de 2020 · 32:22




Guia DEFINITIVO de Aprendendo a Aprender | A maior BRONCA d...

abr. de 2020 · 1:04:37



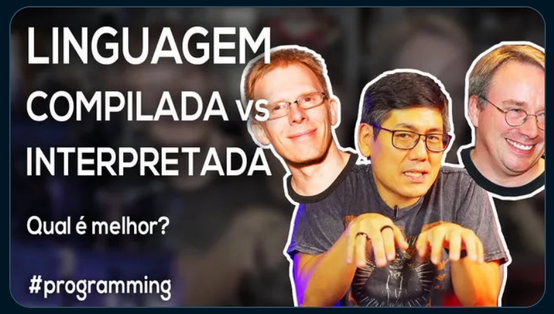

O Guia +Hardcore de Introdução à COMPUTAÇÃO

jun. de 2020 · 1:18:28



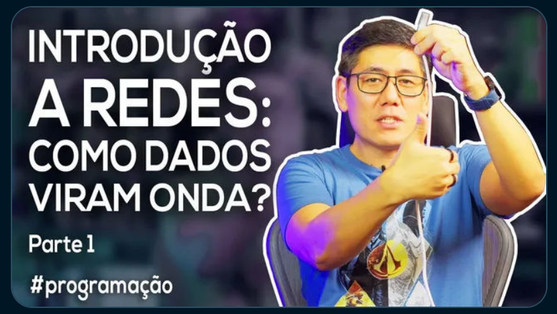

Hello World Como Você Nunca Viu! | Entendendo C

mar. de 2021 · 1:09:22




Linguagem Compilada vs Interpretada | Qual é melhor?

abr. de 2022 · 1:11:44




Introdução a Redes: Como Dados viram Ondas? | Parte 1

jul. de 2022 · 37:59

# As conclusões

**1**

A IA multiplica o que você já sabe. Com pouco pra multiplicar, você recebe muito pouco.

**2**

Atalhos vendidos como carreira já tiveram um crash. O novo é a mesma promessa.

**3**

Aprender no piloto automático não fixa. Os estudos até agora concordam.

**4**

O trabalho que continua valioso precisa de gente capaz de julgar o resultado, enquanto a contratação de nível de entrada está encolhendo.

**5**

A IA não adivinha seu objetivo, seus limites ou seus trade-offs. Saber isso é o trabalho.

**6**

No meu benchmark, dar testes pro modelo mudou os resultados. Escolha a linguagem que serve pro problema.

**7**

A IA replica o que é público. As lacunas que ela deixa, no trabalho privado e novo, são o trabalho, e os fundamentos difíceis são como você preenche elas.

**8**

Escreva na mão, deixe a IA com as tarefas chatas, revise tudo e coloque as horas.

> *A única coisa que importa é realidade objetiva.*

# Leia mais: meus posts e as fontes

## A IA multiplica o que você traz

- Do Zero à Pós-Produção em 1 Semana: Como usar IA em Projetos de Verdade | Bastidores do The M.Akita Chronicles
- VS Code é o novo Cartão Perfurado
- Parem de inventar desculpas e façam mais deploy! Com a IA, a premissa mudou. Entendam.
- Google DORA 2025: a IA amplifica o que já existe
- Veracode: falhas de segurança em código gerado por IA

## A gente já viu esse atalho

- RANT: IA acabou com os programadores?
- CFPB: as promessas de colocação da Lambda School
- Course Report: o mercado de bootcamps, de 2013 a 2018
- Inside Higher Ed: os fechamentos de bootcamps
- Higher Ed Dive: a 2U sai dos bootcamps
- TechCrunch: as demissões em tecnologia de 2024
- Akitando #132: a bolha de startups estourou (2022)

## Aprender no piloto automático não fixa

- Novo LLM Benchmark v4: retestando TODOS os principais LLMs (Parte 1)
- VS Code é o novo Cartão Perfurado
- Como Driveclub e shadPS4 Quase Derrotaram a IA e a Mim: Como Aprender
- Anthropic: como a assistência de IA afeta habilidades de programação
- Bastani et al., PNAS: IA generativa pode prejudicar o aprendizado
- METR: desenvolvedores experientes 19% mais lentos
- MIT Media Lab: Your Brain on ChatGPT (preprint)

# Leia mais: meus posts e as fontes

### O caminho de entrada ficou mais estreito

- RANT: IA acabou com os programadores?
- SignalFire: State of Tech Talent 2026
- Indeed: o mercado de trabalho pende pra senioridade
- Indeed: vagas de software desde o Claude Code (julho de 2026)
- Stanford Digital Economy Lab: o abismo da IA pros trabalhadores jovens (agosto de 2026)
- Fed de Nova York: o mercado de trabalho pra recém-formados
- BLS: desenvolvedores de software

### A IA não adivinha o que você precisa

- Por que LLMs vão falhar na Sua Empresa?
- Por que as LLMs não te dão o resultado esperado | Por que eu prefiro Claude Code hoje
- Vibe Code: Eu fiz um appzinho 100% com GLM 4.7 (TV Clipboard)

### Testes importam mais que a linguagem

- Existem Linguagens Melhores pra programar com LLMs?
- O benchmark: código, dados e método

### Se a IA pode programar tudo, por que aprender?

- Os Limites das LLMs: Boas em reproduzir o que existe. Caras pro que não existe
- Engenharia Reversa com IA: a Adobe morreu?
- VS Code é o novo Cartão Perfurado
- GitHub Octoverse 2025: 81,5% privado
- arXiv (2026): LLMs e COBOL, código que raramente é público
- ARC Prize: tarefas inéditas, 0% pra LLMs puros

### O que "a IA fez" custou de verdade

- AI-MEMORY 2.0 - o melhor sistema de memória para agentes e times
- Software Nunca Está 'Pronto': 4 Projetos, a Vida Pós-Deploy, e Por Que One-Shot Prompt É Mito

# Leia mais: meus posts e as fontes

### Se você é estudante ou júnior

- Como Driveclub e shadPS4 Quase Derrotaram a IA e a Mim: Como Aprender
- RANT: o Akita abriu as pernas pra IA??
- Terminando Minha Maratona de IA: Sucesso ou Fracasso?
- Anthropic: "ficar dolorosamente travado"
- Ericsson et al. (1993): prática deliberada
- Macnamara e Maitra (2019): a replicação
- Akitando #72: programar não é fácil

# Cada ponto, com todos os links, na web

ailair.akitaonrails.com/pt-br/students/

Fabio Akita · akitaonrails.com · x.com/AkitaOnRails