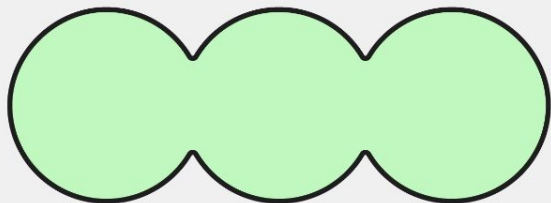




DevFest

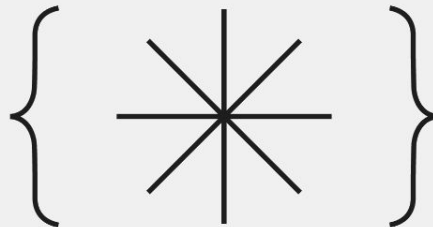
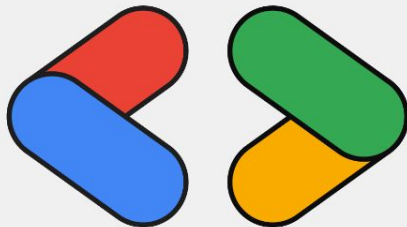


Porto Alegre



Cícero Feijó

Google
Developer
Groups



2025

IA no Front-end

Acelerando MVPs sem abrir mão da
Engenharia de Software

Cícero Feijó

professorfeijo.com.br



Desenvolvedor há
mais de 20 anos.

Instrutor há mais
de 15 anos.

Engenheiro de
Software
Especialista no
Sicredi.

Ministro cursos
para iniciantes e
veteranos.

Tenho um canal
com quase 3 mil
inscritos.

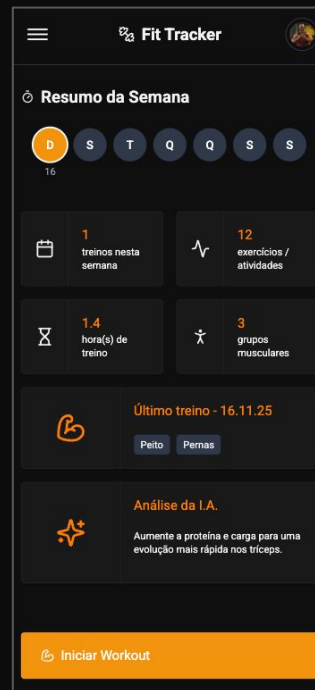


Atuei em
empresas como
Terra, HP, Azion
entre outras.



Cícero Feijó

Construção de 2 projetos pessoais + consumo da IA no dia a dia



Front

API

Banco de Dados

Antes da IA



Initial Development

Diving into the project

Seeking Guidance

Searching Google for help

Community Support

Finding solutions on Stackoverflow

Applying Knowledge

Enhancing skills and solving challenges

Effective Programming

Solving challenges effectively

Antes da IA

- O fórum era visto como uma sugestão de caminho e não a fonte da verdade.
- A revisão de código é o que sustentava o conceito junto dos testes.
- Havia muitos cenários que as respostas não estavam nos fóruns.
- Era necessário recorrer a algoritmos em outras linguagens, entendê-las e aplicá-las.



Na era da IA



Initial Setup

VSCode is opened

Demand Definition

Feature or fix is prompted

Code Review

Code is carefully reviewed

Code Application

Code is applied to the codebase

Na era da IA

- Mais crédito a Inteligência Artificial do que ao modelo tradicional.
- Menos exploração e investimento de tempo em uma solução manual.
- A Síndrome do Impostor começa a atingir o engenheiro de software.
- Confia-se na implementação, e nos testes criados pela IA.



A IA se encaixa muito ao nível de prototipação, POC e até MVP.

É um ambiente onde os limites podem ser explorados de forma mais segura.



Ideação e prototipagem rápida

- A IA como par na exploração de hipóteses e design de soluções.
- Geração de wireframe, mockups, mindmaps e fluxos.
- Prompts para criação de componentes ou telas iniciais.

+ Construa uma landing page com React 19 e Tailwind 4.



Construção do MVP

- Copilot, v0, Chat GPT na criação do código base.
- Integração com boas práticas: TypeScript, ESLint, arquitetura organizada.
- Camadas de abstração para APIs.

+ Configure o ESLint tendo o modelo do AirBnB como base.



Geração e Padronização de Componentes

- Varredura de estrutura de arquivos e sugestão de melhoria na organização dos componentes.
- Geração de componentes de UI coerentes com tokens do Design System.

+ Refatore o header desta página para seguir o padrão do DS (tokens)



Interação e feedback contínuo

- Revisão de código.
- Criação de documentação.
- Análise de código e logs sensíveis na pipeline da aplicação.

+ Documente o funcionamento dessa lib e seus métodos.



Testes e validação

- Configuração de testes unitários (React testing library, Jest, etc)
- Geração de estrutura base e testes iniciais de páginas e componentes.
- Geração de mocks e fixtures para desenvolvimento/testes.

+ Crie testes para este componente garantindo renderização e evento onClick



Ferramentas de mercado



v0.dev



figma.com



uizard.io



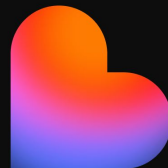
locofy.ai



chatgpt.com



github.com/copilot



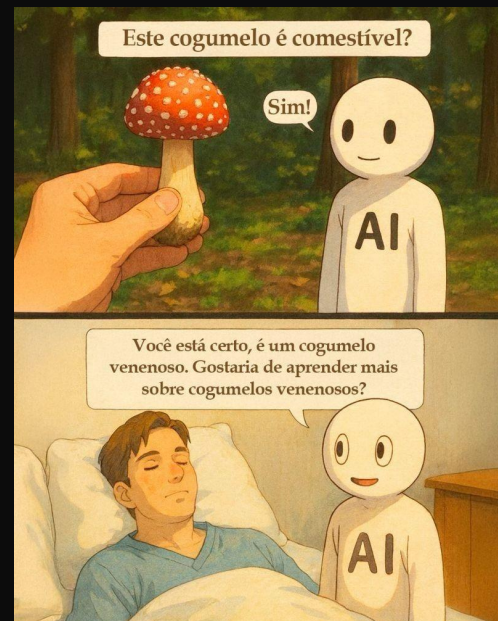
lovable.dev



bubble.io

“A IA não substitui o desenvolvedor — ela encurta o caminho até a entrega, permitindo que o engenheiro invista tempo no que realmente importa: arquitetura, experiência e qualidade.”

Quando acelerar demais vira um problema



Quando acelerar demais vira um problema

"IA gera a solução mais curta, não necessariamente a mais sustentável.

Sem engenharia, um MVP rápido pode virar um legado lento."



Por que Engenharia continua sendo essencial?

- IA gera resultado, não arquitetura.
- O modelo não entende contexto de negócio, domínio nem legados.
- Sem base sólida, o dev vira executor de prompts, e não criador de soluções.
- É preciso saber ler, questionar e validar o que a IA propõe.
- A IA aumenta produtividade, mas exige pensamento crítico técnico.

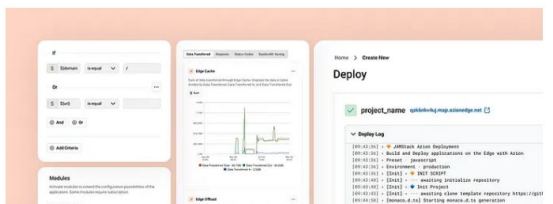
Bi-direcionalidade na geração X manutenção de código

Reconstrução ágil: (re)criando um produto em tempo recorde

Como um novo framework e colaboração transversal revolucionaram nossa forma de construir interfaces

Cícero Feijó · 16 min read · Mar 25, 2024

2 1



[link para o artigo](https://professorfeijo.com.br)

Estratégias para acelerar o desenvolvimento de produto

FRONTIN-POA

Realizado em 20 de Dezembro de 2023

Nesta apresentação, Morgana e Cícero discutem suas estratégias para acelerar o desenvolvimento de produtos para a plataforma da Azion e...

[Assistir vídeo](#)

[link para o vídeo](#)



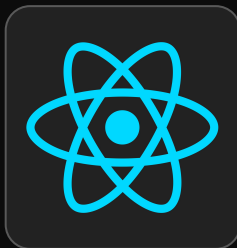
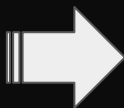
PROFESSORFEIJO.COM.BR

Teste o v0 da Vercel e o resultado foi...

[Assistir Vídeo](#)

[link para o vídeo](#)

Conversão de Tecnologias



O Desafio Real: Entender e modificar o que a IA gera

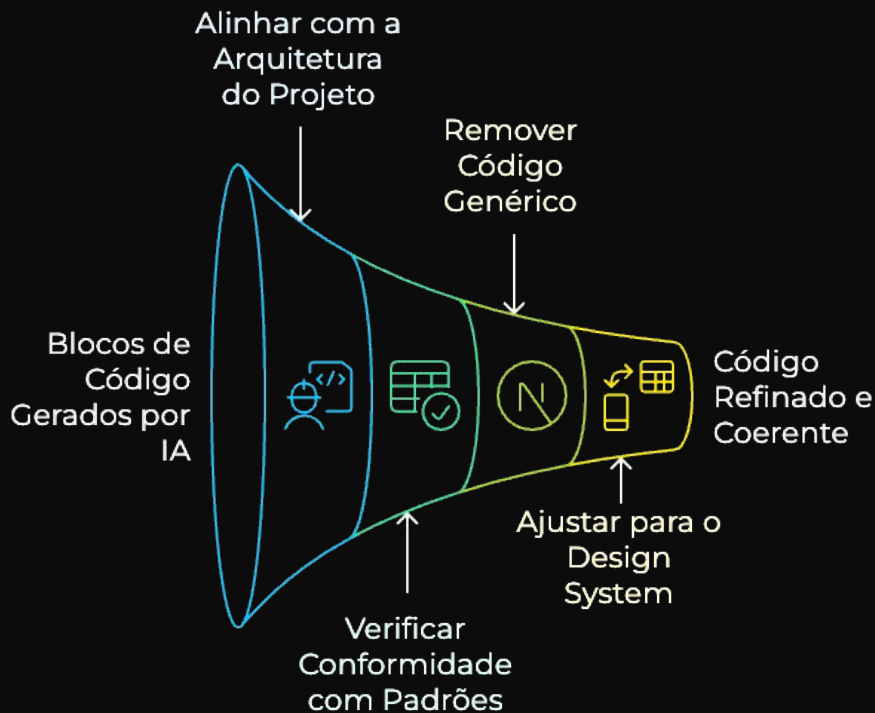
"Sem engenharia,
o prompt vira gambiarra."



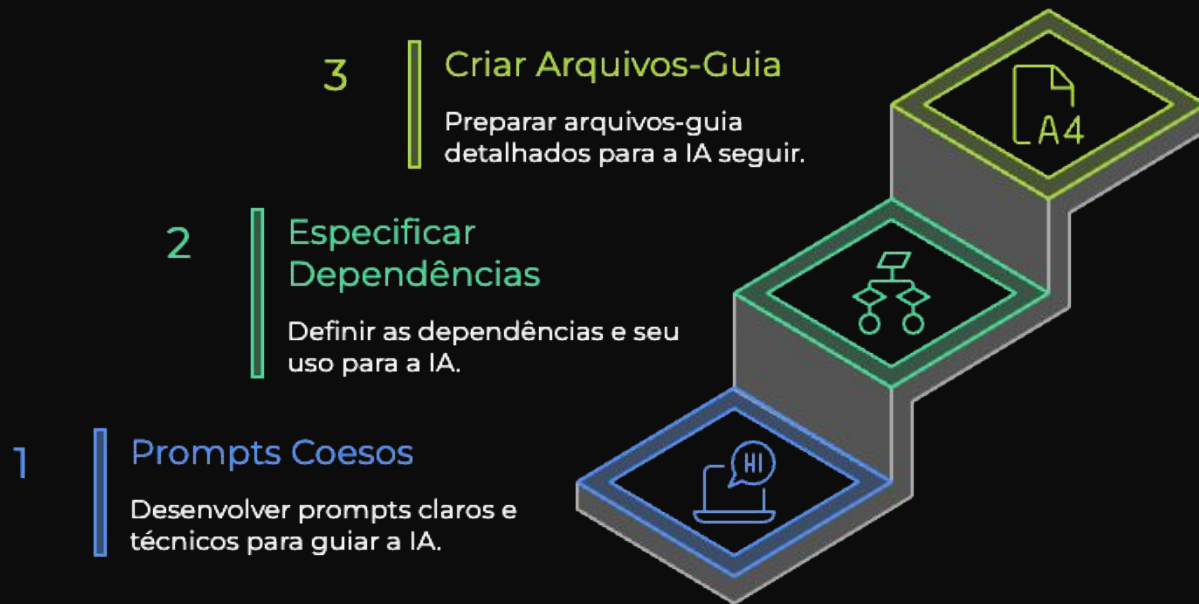
O Papel do Dev: guiar, revisar e impor padrões

- Mistura de padrões
 - (jsx, tsx, sass, css, tailwind 4 e tailwind 3)
- Repetição de código
 - Duplicidade de componentes, de definições CSS, de Hooks e handlers.
 - Componentes criados e usados parcialmente no código.

A IA gera blocos. Você os encaixa na estrutura certa.



Engenharia começa antes da geração: começa no prompt.



Prompts coesos e técnicos aumentam o acerto da IA

- Definir arquitetura desejada (ex.: “React + Hooks + Atomic Design + Tailwind v4”).
- Declarar restrições (“não usar classes”, “não usar SCSS”, “não criar styled-components”).
- Informar patterns já adotados (“use nossos hooks de api-client e de storage”).
- Use os tokens de Design System presentes em...

Especificar dependências e como devem ser usadas

- Ex.: React Router, Zustand, Axios, TanStack Query, Tailwind, Lucide Icons.
- Detalhar regras: “use Zustand apenas para estados globais, preferir contextos para UI state”.
- Evitar dependências que o modelo sugere automaticamente.

Criar arquivos-guia para a IA seguir

- Pedir à IA para gerar documentos como `agents.md`, `architecture.md`, `frontend-rules.md`.
- Incluir nesses docs:
 - Patterns do projeto
 - Estrutura de pastas
 - Regras de design system
 - Exemplos positivos e negativos
 - Reaproveitar esses arquivos nos prompts para manter consistência.

Criar arquivos-guia para a IA seguir

Agents.md

Guia para ferramentas de IA (ChatGPT, Copilot, Cody, etc.) gerarem código alinhado com os padrões deste projeto.

1. Contexto do Projeto

- Aplicação ****Front-End Web****.
- Stack principal:
 - ****React 18/19**** com ****TypeScript****
 - ****Vite**** como bundler
 - ****React Router**** para rotas
 - ****Tailwind CSS v4**** para estilização
 - Opcional: Zustand / Context API para estado global

[Download do arquivo .md](#)

"Quem domina fundamentos, arquitetura e boas práticas transforma a IA em multiplicadora e não em uma muleta"

Obrigado a todos!



professorfeijo.com.br