

Everton Resende

AI Engineer

everton.augustor@gmail.com · +55 37 99913-3556 · linkedin.com/in/evertonresende

github.com/evertonresende · evertonresende.com

Pará de Minas, MG, Brasil · remoto (UTC-3)

Disponível para PJ, CLT ou contrato internacional

Construo e coloco em produção sistemas com LLMs, agentes e RAG, da arquitetura à infra, em TypeScript, Python, Cloudflare e AWS. **7+ anos desenvolvendo software e 5 produtos em produção** (Tazz, Maximus, Dental Smart, Coppay e Davions), entre eles um agente de IA no atendimento, com supervisão humana, e um chat clínico com RAG. Trabalho com um workflow agent-first: defino arquitetura, contexto e critérios de aceite, oriento agentes de código, reviso, depuro e respondo pelo que vai para produção.

EXPERIÊNCIA PROFISSIONAL

AI Engineer — Coppay

10/2025 – 10/2026

Fintech de conta digital e crédito · CLT · remoto

- **IA em produção:** triagem de e-mail com Workers AI, usada pelo time, e parecer de crédito assistido por LLM (gpt-oss-120b), usado pelos analistas.
- **Agente de WhatsApp em Python:** Agno + FastAPI, com triagem para especialistas, tools de leitura e handoff para humano. Atendeu clientes reais até 09/2026.
- **App e BFF:** app em Expo nas lojas, com correntistas reais, e BFF em Hono sobre Cloudflare Workers integrando Cronos, Asaas e HBI.
- **Back-office:** Next.js com RBAC e audit log.
- **Novo BaaS:** em 09/2026, BFF novo sobre o BaaS do Asaas, com uma subconta por cliente, rodando lado a lado com o legado.

TypeScript · Expo · Hono · Cloudflare Workers · Durable Objects · Postgres · Python · FastAPI

Desenvolvedor de Software — Softplan

11/2022 – 10/2025

CLT · remoto

- Produtos para o Ministério Público e as Procuradorias.
- Reconhecido 10x no Softplayer A+.

Desenvolvedor de Software — FAPAM

11/2018 – 10/2021

CLT · híbrido

- Portal acadêmico com 4.000+ alunos.

PROJETOS

Tazz — app de corridas

2026 – atual

- **App novo em Expo:** arquitetei e construí do zero o app de passageiro e motorista e os Workers novos dele: auth próprio, hub em tempo real com Durable Objects e API sobre Postgres; o app não fala direto com Firestore nem Firebase Auth. Primeiras builds de produção em 10/2026, em pré-lançamento para substituir o app Flutter em produção.
- **Construído com agentes:** orquestrei agentes num backlog de ~200 frentes em paralelo, cada uma com prova por critério de aceite e revisão por um modelo de outro provedor; portões humanos em banco, main e lojas.
- **Migração:** o backend rodava em ~158 Cloud Functions e bateu na cota fixa de 20 vCPU do GCP em São Paulo com o app em operação. Portei 100+ delas para 8 Cloudflare Workers em ~47,5 h, em strangler-fig e com agentes de código.
- **Agente de atendimento:** no WhatsApp e no app, em produção com supervisão humana: a IA escreve o rascunho, e o atendente envia, edita ou descarta. Tools de consulta, nota interna e handoff, com allowlist e auditoria, sem tool de dinheiro; no máximo 3 rodadas, guardrails em camadas, LLM-juiz e mascaramento de dados pessoais.
- **Modelos e evals:** LLM da OpenAI via AI Gateway, com fallback. Evals offline com datasets, gate e comparação de 4 modelos; rollout em shadow e canary.
- **Decisão:** sem RAG no MVP. Base curada de 58 entradas, busca por palavra-chave e gatilho objetivo para reavaliar.
- **Pagamentos:** PIX (Asaas) em produção desde 09/2026, sobre ledger de partida dobrada isolado.

TypeScript · Expo · Cloudflare Workers · Durable Objects · Queues · Postgres

- **Produto:** app em Expo e API em Hono sobre Cloudflare Workers, com Supabase e RLS. No ar, em uso por anestesistas.
 - **Chat clínico com RAG:** Responses API (gpt-5.4) com file_search sobre vector store da OpenAI, code interpreter para gráficos, streaming e pergunta por texto, imagem ou voz (gpt-4o-transcribe).
 - **Questões do TSA:** geradas com structured output e validação, a partir da base vetorial.
 - **Decisão:** construí RAG manual (classificador gpt-5-nano + busca vetorial) e troquei pelo file_search nativo, porque o classificador barrava buscas legítimas.
- TypeScript · Expo · Hono · Workers · Supabase · OpenAI Responses API

- **Multi-tenant:** SaaS em Next.js e Supabase, com isolamento por RLS no Postgres (um tenant por clínica). ~20 clínicas ativas, 0 churn.
 - **IA em produção:** transcrição e resumo de consultas desde 02/2026, com Whisper large-v3-turbo e Llama 3.3 70B na Groq.
 - **Integrações:** fila de mensagens de WhatsApp com variações geradas por LLM, NFS-e via Focus NFe e assinatura de contrato por WhatsApp. CI no GitHub Actions e backup do banco no R2.
 - **Decisão:** conciliação bancária por regras, sem IA.
- Next.js · React · Supabase · Postgres RLS · Edge Functions · Groq

- **Re-hospedagem:** refatorei o app, nascido no Lovable, em 10 Cloudflare Workers com Supabase e RLS. Entregue em 10/2026.
 - **Pagamentos:** escrow com o Asaas (livro-razão em SQL e saque por PIX) em dark launch.
 - **IA em produção:** geração de posts de blog, legendas e temas com Claude, via Cloudflare AI Gateway.
- TypeScript · React · Cloudflare Workers · R2 · Supabase RLS · Asaas · Stripe Connect

HABILIDADES

IA:	LLMs (OpenAI, Anthropic/Claude, Groq, Workers AI), agentes e tool calling, RAG (file_search, vector store), structured output, guardrails, LLM-juiz, evals, speech-to-text
Linguagens:	TypeScript, Python, SQL
Cloudflare:	Workers, Durable Objects, Queues, R2, AI Gateway, Workers AI
AWS (projetos anteriores):	EC2, ELB, Lightsail, Lambda, SQS, S3, Aurora, Rekognition
Backend e dados:	Hono, FastAPI, Node.js, Postgres, Supabase (RLS, Edge Functions), Docker, CI (GitHub Actions)
Front e mobile:	React, Next.js, React Native / Expo
Pagamentos:	PIX, Asaas, Stripe Connect, ledger de partida dobrada

OPEN SOURCE

2 PRs merged no Cal.com: [#27636](#) e [#27637](#) (02/2026).

FORMAÇÃO

Sistemas de Informação (incompleto) · FAPAM · 11/2018 – 12/2020

IDIOMAS

Português (nativo) · Inglês (confortável em entrevistas técnicas)