Luan Fabri

Engenheiro front-end na Loopscape

Jaú, SP, Brasil · luandaniel966@gmail.com · linkedin.com/in/luannzin · github.com/luannzin

Desenvolvedor front-end no cargo, full-stack na prática. Na Loopscape desde 2023, trabalho no Loopvet, um SaaS para clínicas veterinárias, com React, Next.js e TypeScript no front-end e Node.js, PostgreSQL, MongoDB e Rust no back-end.

Boa parte do trabalho tira gargalos de outras áreas, da importação de dados de clientes a bugs de produção que chegam ao time antes de o cliente ligar para o suporte. Também construí o front-end de um SaaS de gestão de igrejas de uma empresa irmã e liderei seu primeiro time de front-end. Fora do trabalho, criei apps open source em Rust e TypeScript que só precisam de um código ou de um link, sem conta e sem servidor para rodar.

EXPERIÊNCIA

Loopscape
Loopvet, SaaS para clínicas veterinárias, e um SaaS de gestão de igrejas de uma empresa irmã

Desenvolvedor front-end pleno — Fev 2025 – atual
Desenvolvedor front-end júnior — Jun 2023 – Fev 2025
Estagiário de front-end — Mar 2023 – Jun 2023

Ferramentas internas

- Criei uma aplicação web que converte os dados exportados pelas clínicas dos sistemas anteriores para o formato de importação do Loopvet, um trabalho que antes ficava com um desenvolvedor back-end sênior. Comecei com um script e escrevi o núcleo da conversão; um desenvolvedor júnior construiu a maior parte da interface.
- Desenvolvi uma ferramenta de auditoria de todas as requisições da API salvas no MongoDB: busca, agrupamento, contagem, tempo de resposta, queries lentas, erros e estado do banco.
- Treinei um modelo open source (PyTorch) com essas requisições para identificar ataques, loops de requisição, usuários travados e problemas de performance. Roda sem parar como uma TUI em Rust numa TV e envia alertas e um resumo matinal ao Discord. Com isso, o time corrige bugs antes de o cliente ligar para o suporte.
- Entreguei ao time de vendas uma ferramenta que mostra quais leads precisam de contato e quais ainda não responderam.

Produto (Loopvet)

- Implementei o onboarding e um sistema de tours interativos, depois empacotado como biblioteca React. O tour é escrito em passos de linguagem natural (“Clique em Novo Cliente”) e localiza cada elemento na tela, sem ids no código, então qualquer pessoa do time publica um tour sem depender de desenvolvedor.
- Escrevi o microsserviço de PDF em Fastify para o back-end em AdonisJS e depois o reescrevi em Rust com o Typst como motor de layout: um binário único que renderiza um documento por núcleo.
- Integrei terminais de pagamento, WhatsApp e assinatura digital ao produto, e lancei um site público para validação de receitas digitais.
- Reconstruí em Next.js o site de vendas e as landing pages de campanhas, com foco em performance e SEO.
- Conduzi várias refatorações de código legado, incluindo a migração de React 16 para uma stack moderna com TypeScript, e reduzi o tempo de build da aplicação principal.

SaaS de gestão de igrejas

- Construí o front-end do zero: o painel de gestão e o app PWA, ambos multi-tenant (Next.js, Tailwind CSS, shadcn/ui).
- Liderei o primeiro time de front-end e participei da contratação dos desenvolvedores.

OPEN SOURCE

- Deskline (deskline.luannzin.com): acesso remoto sem servidor para rodar, em Rust (egui, iroh). Conecta com um código e um pedido de aceite, e tem transferência de arquivos, som, chat e até 4K a 240 fps com encode na GPU. Criado para substituir o AnyDesk ao ajudar família e amigos.
- Sightline (sightline.luannzin.com): compartilhamento de tela por link, ao vivo no navegador e com som. React e WebRTC, com sinalização por relays públicos do Nostr: um site estático, sem servidor no meio.
- Hostline (hostline.luannzin.com): compartilhamento do localhost por link. Uma CLI em Rust gera um link e um QR code, e o navegador de quem abre carrega a porta direto da máquina de quem compartilha, por um data channel WebRTC, com hot reload.
- PartyLine (github.com/luannzin/partyline): game show no terminal para jogar com os amigos, em Rust (ratatui, iroh). Os jogadores entram numa sala por código de convite, na rede local ou pela internet.
- echo (useecho.dev): app de notas que identifica prazos, tarefas e palavras recorrentes em cada linha. Postgres de verdade, compilado para WebAssembly, roda no navegador, e nenhum recurso exige chave de API de IA.
- better-intl (github.com/luannzin/better-intl): i18n tipado e baseado em arquivos para Next.js, usado por este site em inglês e português.

OUTROS PROJETOS

- **ERP de comunicação multi-tenant**: arquitetura do front-end em Next.js, com cache e dados isolados por tenant.
- **ERP acadêmico para a Unoeste**: aplicação full-stack com Next.js, Drizzle ORM e PostgreSQL.

HABILIDADES

Linguagens: TypeScript, JavaScript, Rust, Python, SQL
Front-end: React, Next.js, Vite, Tailwind CSS, shadcn/ui, Base UI, PWA
Back-end: Node.js, AdonisJS, Fastify, APIs REST
Dados: PostgreSQL, MongoDB, Drizzle ORM
Nativo e redes: Rust (egui, ratatui), WebRTC, peer-to-peer (iroh, Nostr), Typst
IA: funcionalidades com LLM em produção, treino de modelos open source (PyTorch)
Outros: arquitetura multi-tenant, observabilidade, performance, CI/CD, Git, Bun

FORMAÇÃO

Sistemas para Internet Fev 2022 – Dez 2024
Fatec Jahu (Faculdade de Tecnologia de Jahu)

IDIOMAS

Português (nativo), inglês (fluente)

Palavras-chave: Desenvolvedor full-stack, engenheiro de software, desenvolvedor front-end, React, Next.js, TypeScript, JavaScript, Node.js, Rust, Python, PostgreSQL, MongoDB, API REST, AdonisJS, Fastify, SaaS, multi-tenant, PWA, WebRTC, peer-to-peer, geração de PDF, Typst, performance, observabilidade, LLM, IA, PyTorch, ferramentas internas, migração de dados, integração de pagamentos, onboarding, refatoração, open source, Tailwind CSS, CI/CD, liderança técnica, contratação