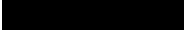


Experience

  May 2025 – present  Brazil

Software Engineer

- Replaced line-by-line manual review of medical invoices with an automated extraction pipeline for a billing platform serving **50,000+ physicians**, by combining OCR with LLM-based parsing (Ollama, AWS Bedrock) to turn unstructured invoices into structured, review-ready billing records.
- Eliminated unauthenticated service-to-service traffic across the platform by implementing an AWS Cognito authentication flow and a Python client service that validates and refreshes JWTs on every internal call.
- Took a fintech from zero to production for healthcare professionals by owning early backend architecture, service boundaries, and the core billing workflows now used by the platform's entire physician base.

  December 2022 – May 2025  Brazil

Software Engineer

- Reduced platform latency by **15%** for **50,000+ users** on production financial systems by profiling hot request paths and optimizing database queries and front-end render cycles.
- Resolved **100+ production defects** in payment and transaction flows, raising transaction reliability and cutting recurring support escalations, by owning bug triage end to end from report to release.
- Turned my own feature proposals into shipped product by pitching data-backed specs in planning and refinement sessions, then implementing them through to release.

Projects

Fraud Detection API — *Go, Gin, nginx, Docker*  May 2026 – June 2026

- Served k-nearest-neighbour fraud decisions over **3M labeled reference transactions** inside a hard **1 CPU / 350 MB** budget for Rinha de Backend 2026, by dropping Postgres + pgvector for an in-process int8-quantized IVF vector index baked into the Docker image at build time.
- Cut each request from a 3M-vector scan to a bounded cluster scan by k-means clustering the corpus into **2,048 cells** and auto-tuning `nprobe` against a brute-force oracle until it met the target recall.
- Fit two API instances under the memory cap by `mmap`-ing the read-only index outside the Go heap so both processes share the same physical pages, load-balanced behind nginx.

Rideshare Platform — *Go, Node.js, Java, React Native, Docker*  December 2024

- Shipped a full rideshare system solo across **4 services** – Go trip-matching engine, Node.js API gateway, Java notification service, React Native app – covering the entire rider-to-driver lifecycle from ride request through payment.
- Kept trip matching responsive under concurrent load by indexing driver positions geospatially and running the matching loop in a bounded Go worker pool rather than a goroutine per request, with every service containerized behind contract-first REST APIs.

Hoi4Presence – Open-Source Discord Integration — *Python, PyInstaller, Discord RPC*  September 2022 – present

- Built and still maintain a Discord Rich Presence integration for Hearts of Iron IV, reaching **2,720 downloads** across **12 releases** with **13 stars** and **9 community forks** over 4 years of active maintenance.
- Brought install friction to zero for non-technical players by shipping self-contained PyInstaller executables with a bundled installer, uninstaller, and self-updater, backed by an automated test suite that keeps releases safe across game patches.

Transactional Email Microservice — *Java, Spring Boot, AWS SES, SendGrid, Maven*  January 2025

- Delivered a Spring Boot microservice that dispatches transactional email through Amazon SES or SendGrid, letting consuming services switch providers via API-key configuration with **zero code changes**.

Education

  May 2022 – 2026  Brazil

Bachelor of Science in Computer Science

Skills

- **Languages:** Go, Java, TypeScript, JavaScript, Python, SQL
- **Backend:** Spring Boot, Node.js, Gin, Gorilla Mux, REST APIs, microservices, PostgreSQL
- **Frontend:** Angular, React, React Native
- **Cloud & Tools:** AWS (Cognito, Bedrock, SES), Docker, nginx, Git, Azure, Jira, Vim
- **Spoken:** Portuguese (native), English (B2)