



HENRIQUE DERVICHE KASPRZAK

FULL-STACK DEVELOPER · GO & REACT

LOCATION
Curitiba, Brazil

PORTFOLIO
ikki.dev.br

PHONE / WHATSAPP
+55 42 99940-5868

LINKEDIN
linkedin.com/in/henrique-kasprzak

EMAIL
ihiquedk@gmail.com

GITHUB
github.com/IkkiDK

SUMMARY

Full-stack developer and Computer Engineer (B.Sc. 2026) with two years on a smart-city platform in production. I write the Go services that bring a city's cameras and sensors into one place — plate and face recognition ingestion, watchlist alerting, PTZ control, telemetry over MQTT — and the React screens operators work from.

Outside work I ship complete systems alone: a fleet inspection and fueling system with a signed, tamper-evident evidence chain (close to 300 commits, 10 ADRs), a farm-maintenance logbook that runs as one Go binary, and an IoT capstone from sensor to browser. I value simple, direct solutions — a lean HTTP handler over a framework, understanding the problem over adding a dependency — and use AI as a serious tool without outsourcing my understanding of what I build.

HIGHLIGHTS

- Production Go services on Kubernetes: hybrid push-and-poll ingestion, webhooks, MQTT, Prometheus, memory-bounded concurrency.
- End-to-end systems delivered solo: backend, offline-first PWA, database, deploys, tests, runbooks and ADRs.
- Evidence-grade data: append-only tables, SHA-256 hash chains, Ed25519-signed anchors to WORM storage, database-free verification.
- Software/hardware integration: ESP32 and sensors, LPR/ANPR and facial cameras, PTZ control.

TECHNICAL SKILLS

Languages

Go, TypeScript, JavaScript, C, C++, Java, SQL, Bash, HTML and CSS.

Front-end

React 19, Next.js 15/16, Vite, Tailwind CSS, Recharts, Radix UI, Framer Motion, Leaflet and htmx.

DevOps & cloud

Docker, Kubernetes, GitHub Actions, Caddy, systemd, Vercel, Render and MagaLU Cloud.

IoT, hardware & domains

ESP32, Raspberry Pi, ADC sensors, computer vision, LPR/ANPR, facial recognition, PTZ camera control and urban telemetry.

Back-end & real-time

REST, WebSocket, SSE, MQTT (Paho/Mosquitto), webhooks, hybrid push/poll ingestion, Chi router, microservices and multi-tenancy.

Data & infrastructure

PostgreSQL/PostGIS, SQLite, Redis, Supabase, S3/MinIO with Object Lock (WORM), Goose migrations and pgx.

Observability & practices

Prometheus, OpenTelemetry, structured logging (Zerolog), testing, ADRs, runbooks, retry/backoff, hash chains and Ed25519 signing.

Tooling & AI

Git/GitHub, CLI, advanced use of LLMs for development and code review, local models (LM Studio), GIMP and Inkscape.

PROFESSIONAL EXPERIENCE

Metropolys — Software Developer

Smart-city platform | Oct 2024 – Present | Curitiba, Brazil

- Primary author of two Go plate-recognition (LPR/ANPR) ingestion services: a webhook service with image download and upload, vehicle-metadata extraction, restriction-list checks with priority alerting, SHA-256 event deduplication and Prometheus metrics; and, in 2026, a hybrid push-and-poll ingester that also takes in face matches — both paths derive

one deduplication key, so nothing is stored twice — which became the platform's facial service when the two were merged.

- Built watchlist alerting for plates and faces, from the camera platform's alarm to the operator's screen, including attaching the passage snapshot that only reaches the platform 14–22 seconds after the alarm.
- Delivered the product's plate and face screens in React and TypeScript: plate search and tracking, trace map with passage viewer, proximity view, monitored plates grouped by alert priority, face alerts and enrolment with categories on the same priority scale.
- Authored the Go PTZ (pan-tilt-zoom) camera-control service and its React 19 overlay — directional dial, zoom/focus/iris and custom command hooks rendered over the live video stream.
- Created Go urban-telemetry simulators (traffic lights with failure modes, public lighting, GeoJSON route-based GPS tracking, facial detection) so the whole pipeline can be tested end to end without physical hardware.
- Developed the KPI aggregation service that turns traffic, weather and infrastructure data into standardized multi-tenant indicators, including rolling rainfall totals from weather-station history; tuned production on Kubernetes (memory-bounded concurrency after an OOM, cursor resume on restart, readiness probes).
- Contributed to real-time public transit, IoT sensor and public-lighting telemetry over MQTT, the video gateway's camera sources, and built the operations dashboard's camera mosaic (draggable tiles, presets, search, Leaflet map).

Go TypeScript React 19 PostgreSQL/PostGIS Redis MQTT WebSocket SSE Docker Kubernetes
Prometheus

FEATURED PROJECTS

Xanadu Fleet — Fleet inspection, fueling & maintenance system

Sole author, 2024 – present · close to 300 commits · 10 Architecture Decision Records

- Versioned digital checklists (58 truck items, 19 machine items, trailers attached) executed at vehicle hand-out and hand-back; every inspection pins the exact checklist version it ran against.
- Offline-first PWA with an IndexedDB outbox and a framework-neutral REST sync API; ingest tolerates out-of-order sync, quarantines clock-skewed records, and takes photos over a separate presigned-upload channel with a declared-hash check, so a record is never held back by its images.
- Corrections are new records that supersede the original — never edits — with the supersession link sealed into a per-tenant SHA-256 hash chain over append-only tables.
- Evidence anchoring: a background worker Ed25519-signs each record hash into an Object Lock (WORM) bucket, anchor lag doubles as the backup alarm, a signed digest is written daily, and an evidence-verify tool checks the log with no database at all.
- Fueling module (diesel and ARLA, receipt photo sealed at capture), a manager's cockpit with fuel and division reports, and reconciliation against the company's ERP — result per branch by joining freight documents.
- CI runs `go vet`, `go build` and `go test` against a fresh Postgres on every push; 50+ Goose migrations, on-demand deploy workflow, a baseline CSP, rate-limited and audited PII exports, runbooks for deploy, disaster recovery and evidence anchoring, and three user manuals.

Go PostgreSQL htmx Offline-first PWA S3 Object Lock Ed25519 Goose GitHub Actions MagaLU Cloud

Manutenção Lavoura — Farm-equipment maintenance logbook

Sole author, 2026 · one Go binary and one SQLite file, live in production

- Records what was serviced, when and at which meter reading for a tractor, combine, planter, seeder, sprayer and truck; each visit lists priced services with a "repeat after N hours/km", and a spend view totals cost per machine and per service, grouping names regardless of case, accents and spacing.
- The new-record form works offline through a service-worker cache and an on-device outbox; photos are resized on the phone, queued separately and uploaded right after their record, so a record is never blocked by its images.
- Installed apps update themselves: the service-worker cache is named after a hash of the files it caches, so any build that changes them triggers a refill and reload with no version to bump by hand.
- Deployed as a static binary (`CGO_ENABLED=0`, pure-Go SQLite) under systemd behind Caddy on a VPS; migrations run in transactions at startup and refuse a half-applied schema; the release workflow re-runs tests, cross-compiles, snapshots the database and installs over SSH.

Go SQLite htmx PWA Caddy systemd GitHub Actions

Mini Estufa — Capstone · Real-time full-stack IoT

Greenhouse monitoring integrating hardware, sensors and a web app

- Go API (sole author) receiving telemetry from the hardware (ESP32) over HTTP and broadcasting it in real time over WebSocket, with `/api/sensor/push`, `/api/sensor/latest`, `/ws` and `/health` handlers.
- React 19 + Vite + Tailwind dashboard with Recharts graphs (temperature, air humidity, light level, soil moisture and pump/light status) and history persisted in Supabase, with date-range filtering and timezone handling.
- Physical assembly, ADC sensor reading and calibration, irrigation-pump automation and an MQTT/Mosquitto pipeline; API on Render, dashboard on Vercel.

Go gorilla/websocket React 19 Vite Tailwind Recharts Supabase MQTT Render Vercel

Xanadu Transportes — Institutional website

Sole author · rebuilt on Next.js 15 in 2026 for prerendering and SEO

- Fully prerendered App Router site with server components by default, `next/image`, Framer Motion sections and a Cloudinary hero video with a lighter cut for phones.
- WhatsApp-only calls to action tracked as events; Google Analytics 4 loaded only after LGPD cookie consent, with a privacy-policy page.
- Permanent redirect from the Vercel preview domain to the canonical domain, plus canonical and Open Graph metadata, so search engines index exactly one address.

Next.js 15 React 19 Framer Motion Cloudinary Google Analytics 4 Vercel

EDUCATION, COURSES & LANGUAGES

Universidade Positivo

B.Sc. in Computer Engineering | 2021 – 2026 | graduated

Portuguese: native.

English: reading, writing and conversation.

Go (Golang): Udemy, 60 hours

React with REST APIs: Udemy, 30 hours

GitHub: Udemy, 20 hours

Dale Carnegie: communication, leadership and interpersonal skills