

ANDREI MAR M. DAVA

AI AUTOMATION ENGINEER · N8N + CUSTOM CODE, BACKED BY REAL ENGINEERING

Philippines (Remote) · davaxdev@gmail.com · linkedin.com/in/andrei-mar-dava · github.com/DavaXCode · andrei.figliunicotech.online

Software engineer by trade — I build AI automation the way real software gets built: tested, traceable, and debuggable when the no-code path hits its limits. Most automators either know the tools or know the engineering; the combination is rarer, and it's what keeps a workflow running when traffic spikes, edge cases appear, or an API silently changes. By day I ship multi-tenant SaaS to distributors, hotels, and small businesses at Everyone Technologies; on the side I run n8n + custom-code systems in production on my own business, and design AI pipelines for agencies that need cost and reliability — not a demo that breaks on the second real input.

AREAS OF EXPERTISE

- n8n, Make.com, Zapier — workflow automation platforms
- AI Agent Orchestration (tools, memory, multi-turn conversations)
- Workflow orchestration & scheduling (cron, queues, retries)
- Business Process Automation
- AI/LLM Pipeline Development (Claude, OpenAI, Groq, DeepSeek)
- RAG Systems & Vector Search (pgvector, embeddings, reranking)
- Embedding model selection & comparison
- Cascading Model Routing for Cost Optimization
- Prompt Engineering & Structured Outputs
- Model Context Protocol (MCP) — built & deployed an internal MCP server (mcp-doc-server)
- Webhook security (HMAC validation, signature verification) & OAuth 2.0 flows
- Gmail, Slack, Discord, Telegram, Cal.com, Notion, Airtable
- GoHighLevel (GHL), CRM-adjacent (transferable ERP experience — HubSpot-ready)
- Twilio SMS automation
- Playwright browser automation
- Web scraping (Apify, Firecrawl)
- Queues & message brokers (Redis, RabbitMQ)
- Multi-Tenant SaaS Architecture
- Real-Time Systems (MQTT, SignalR, WebSockets)
- Full-Stack Web Development (Next.js, .NET, React)
- Cross-Platform Development (Flutter, Windows Forms)
- Cloud Deployment & DevOps (Docker, Linux, Windows Server, Nginx, IIS)
- Databases (PostgreSQL, SQL Server, MySQL)

PROFESSIONAL EXPERIENCE

Everyone Technologies

Aug 2024 – Present

Senior Software Engineer (Jan 2026 – Present) · Software Engineer (Aug 2024 – Jan 2026)

Promoted to Senior after taking end-to-end ownership of the Property Management System and co-architecting the ERP platform. Outsource SaaS team (<10 people, 2 senior engineers) shipping multi-tenant products to distributors, hospitality,

schools, restaurants, and small businesses.

- **Property Management System — end-to-end owner.** Architecture, feature delivery, production reliability. Live with 2 hotel clients in Iloilo; more in onboarding. Replaced paper-based reservation logs and manual front-desk handoffs across both properties with a digital workflow.
- **ERP platform — foundational architect.** Co-authored the original codebase that became the product. Live with 2 distributors and 1 construction firm in Iloilo; more in onboarding. Transitioning to full ownership. Digitized inventory, invoicing, and reconciliation workflows previously run on paper and Excel; cut month-end close from a multi-day manual process to same-day.
- **Reporting & business intelligence across all systems.** Built dashboards and on-demand reports (sales, inventory, occupancy, reconciliation) that turn raw operational data into decisions — replacing manual Excel pivots and giving owners same-day visibility into what’s actually happening in their business.
- **Next-generation POS — lead engineer on the rebuild.** First client secured; additional deals in the pipeline.
- **Queue Management System — real-time multi-platform system for an LGU client (Balete, Aklan).** React web kiosk, admin dashboard, Flutter mobile panel, and Windows Forms thermal printer client, synchronized over MQTT against a .NET 6 / SQL Server backend.
- Wrote custom .NET integrations where off-the-shelf connectors couldn’t handle the edge cases.
- Manages Linux server deployments, Docker containers, and DevOps workflows for all production releases.

Stack: .NET, Next.js, React, Flutter, React Native, C#, TypeScript, Python, PostgreSQL, SQL Server, MySQL, Docker, Linux, Windows Server, Nginx, IIS, MQTT, SignalR.

AI AUTOMATION PORTFOLIO

Production systems running on my own business, plus designed-for-client architectures that are demo-able on request.

Lead Funnel — AI-Powered Front Door for Inbound Leads

In production · Running on my own business

An AI-powered front door for inbound leads — so they never sit unread while I’m heads-down on real work.

- **Why it exists.** Leads from my portfolio contact form were sitting in my inbox for 8–24 hours, sometimes 36, because I was deep in client work. By the time I replied, the lead had cooled.
- **What it helps with.** Hot leads now get a first reply faster than I could send one manually, and I never miss a fresh lead — even if I don’t check my inbox all day.
- **How it works.** The moment a form is submitted, a Slack notification lands in my pocket. The AI reads the message, classifies it hot / warm / cold, and drafts a tier-appropriate first reply automatically. Bookings and lead details get logged so nothing falls through the cracks.

Stack: n8n, Groq, Gmail, Slack, Cal.com, Google Sheets.

AI Inbox — Email Triage & Reply Drafting

In production · Running on my own business

Turns the nightly inbox grind into a one-minute pass.

- **Why it exists.** I was burning 30–60 minutes most nights triaging email — time that should have been client work or rest.
- **What it helps with.** Cut nightly triage from 30–60 minutes to ~1 minute, measured against my own inbox over the first month of running it. The AI surfaces only the threads that actually need attention, and the drafts it writes usually need a quick edit before sending.
- **How it works.** The system reads incoming mail, classifies it (meeting request / lead / noise), and routes each one accordingly. Meeting requests get real calendar slots checked against availability. Replies are written as drafts — never sent automatically — so I stay the final approver.

Stack: n8n, Groq, Gmail, Google Calendar, Google Sheets, Slack.

Advanced RAG — Cost-Optimized AI Reporting Pipeline

Case study · Designed for a US agency engagement

A working n8n architecture I designed in response to a real prospect's cost and hallucination problem.

- **Why it exists.** Running every document through a top-tier model is expensive *and* still hallucinates. Most teams pay premium prices for answers that aren't even reliable.
- **What it helps with.** Cascading model routing (Groq → Claude Haiku → Claude Sonnet on quality-check failure) reduces per-document AI cost by ~80% versus a single premium-model baseline — calculated from published Anthropic and Groq token pricing on a representative document set. Hallucinations drop because the AI is grounded in retrieved source material instead of guessing.
- **How it works.** Clean the input before the AI sees it. Ground the model in real source material via RAG. Use a small, cheap model first; escalate only when the answer fails a programmatic quality check. Anything that still doesn't pass goes to a human-review queue — nothing gets silently published.
- **Engineering depth.** Custom n8n Code nodes (JavaScript) for token accounting, chunking strategy, and retry logic with exponential backoff — where built-in nodes hit their limits.

Stack: n8n, OpenAI, Anthropic Claude, Groq, Postgres + pgvector, Slack, Google Sheets.

AI Support Agent — Knowledge-Base Bot with Smart Escalation

Case study

A support bot that answers from *your* docs in seconds — and knows when to call a human.

- **Why it exists.** Support staff burn minutes per question hunting through docs, product pages, and old tickets. Off-the-shelf chatbots either hallucinate or refuse to help.
- **What it helps with.** Answers in seconds with the relevant source material cited, opens a real support ticket when it can't help, and pings a human on Slack when something genuinely needs judgment.
- **How it works.** The bot reads from a private knowledge base instead of guessing. Each message goes through a small model first; only the hard cases escalate to a bigger one. Conversations are remembered per user so it doesn't lose context mid-thread.
- **Engineering depth.** Per-user conversation memory persisted in Postgres with windowed context retention so threads don't lose state across sessions. Answers cite the exact knowledge-base section they were drawn from — and when the bot can't answer with confidence, it doesn't guess: it opens a real support ticket, pings a human on Slack with the unresolved question, and the resolution can be folded back into the knowledge base for next time.

Stack: n8n, Telegram, Groq, Postgres, Notion, Slack.

mcp-doc-server — MCP Server for AI-Assisted Project Documentation

In production · Internal dev tool, used daily

An MCP server that lets Claude Code (or any MCP client) document and query its own projects — turning scattered code knowledge into structured, retrievable context.

- **Why it exists.** AI coding agents re-read source files every session, burning tokens and context window on knowledge that doesn't change between conversations. Architecture decisions, module purposes, and conventions get re-discovered every time — wasteful and lossy.
- **What it helps with.** Claude queries structured documentation (ADRs, modules, KB articles, glossary, changelog) instead of grepping source. Project onboarding takes one tool call instead of a multi-file read storm — cheaper sessions, faster context, decisions that don't get lost between conversations.
- **How it works.** Exposes ~50 MCP tools for documenting and retrieving project knowledge — register projects/modules, record architectural decisions, scan endpoints into OpenAPI, snapshot AI workflows, track changelogs linked to git

commits. Claude calls these tools directly during a coding session; every doc write is anchored to the exact commit it documents.

- **Engineering depth.** Full Model Context Protocol implementation end-to-end. Structured persistence with retrieval-optimized schema, git-commit linking for doc freshness, dashboard UI for human inspection. Built because no off-the-shelf tool combined MCP-native access with project-scoped doc structure.

Stack: Node.js, TypeScript, MCP SDK, PostgreSQL, Express.

EDUCATION

Bachelor of Science in Information Technology

Aug 2020 – Jul 2024

University of Antique — Specialization: Software Development.

- Capstone: *PARA* — *Real-Time Public Transportation Tracking Using IoT.*

EARLIER EXPERIENCE

IT Support — OJT (500 hours)

Feb 2024 – May 2024

BPO company (Philippines) · Required academic on-the-job training

Built batch-script automation that turned PC re-imaging from a multi-day manual job into a near-hands-off process — the first real automation work I shipped.

- **Automated PC re-imaging workflow.** Reduced post-reimage setup from ~1 week of manual work to ~2 days. After a fresh image, scripts ran debloaters, silently installed company software (Sophos, etc.), configured ports, hid system folders, and applied standard policies — leaving only hostname change and OM registration as manual steps.
- **Reusable across the IT support team.** Scripts replaced a repetitive checklist that every technician was doing by hand on every machine.
- **Excel productivity tooling for asset tracking.** Wrote VBA/VB6 macros to make wide hostname & serial-number spreadsheets easier to read — including a hover-highlight tool that lit up the active row and column so technicians could compare values across distant columns (e.g. col A vs col AB) without losing their place. Small tool, but it eliminated a recurring source of data-entry mistakes on asset audits.

Stack: Windows batch scripting, PowerShell, silent installers, Group Policy basics.

Student Projects & Thesis Builds

2021 – 2024

University of Antique · self-directed and peer-collaborative work

Web and mobile builds across undergrad — own thesis work plus helping peers ship theirs. Not commercial deployments; included to show continuous building before turning professional in Aug 2024.

- **PARA — Real-time public transportation tracking** (own capstone). IoT-integrated location tracking with live commuter visibility.
- **E-commerce mobile app for construction materials.** Easy canvassing/quoting flow for buyers comparing suppliers.
- **Alumni social platform (web).** Community feed + profile system for university alumni network.
- **Online appointment system with queue management.** Booking + real-time queue display for service-based businesses.

Stack: React, Flutter, Node.js, PostgreSQL, MySQL.

Also built: *Kuwenta* — offline AI finance tracker (Flutter, on-device LLM). APK on request; Play Store launch planned 2026.