JARBY JESSID COSME

Software & AI Engineer | Production AI Systems, Backend & Data

Cali, Colombia · jessidc95@gmail.com · English B2 (upper-intermediate)
linkedin.com/in/jarbyjessidcosme · github.com/JessidC · Portfolio: jarby-portfolio.netlify.app
Available for full-time remote US nearshore roles (GMT-5, overlaps with EST/CST)

PROFESSIONAL SUMMARY

Software engineer with ~7 years of experience (Python, C#/.NET, TypeScript, SQL) who takes projects from zero to production. Designing since 2025 and now sole technical owner of SIESA's AI support platform, with one goal: improve the service end to end, not "add a chatbot". Started from the data, built a RAG system on Claude (AWS Bedrock) that replaced a third-party vendor across 15 ERP modules (~1,000 conversations a day), measured what was actually being resolved and orchestrated in LangGraph the agents that measure it. Presents business cases to the CEO.

PROFESSIONAL EXPERIENCE

Software Developer — SIESA, Cali, Colombia
Mar 2022 – Present

AI support platform — sole technical owner (2025 – Present)

- **Diagnosis first:** analyzed 90 days of service data (55,350 chats, ~964K messages) and presented the replacement business case to leadership with 6 decisions ranked by impact and cost; a later service audit (NPS, customer comments and traces) showed the official indicator measured contact, not resolution.
- **A RAG built for the service:** designed, built and operate the RAG platform that replaced the vendor in the support chat, on three rules: never make things up (citation and numeric validation, ~170 deterministic guards), hand off to a person on low confidence (tool calling) and never brush the customer off. Cut human escalation in Payroll from 40.2% to 32.7% per chat ($p = 0.0005$) and replies sending customers to "contact support" from 6.95% to 0.25%. *Framework-free Python/FastAPI on the critical path; Claude Sonnet 4.6 / Haiku 4.5 on Bedrock; PostgreSQL + pgvector; Cohere Rerank 3.5.*
- **Quality measured, not assumed:** golden sets with an LLM judge gating releases, A/B replay on real traces and significance tests: answer correctness rose from 71.6% to 89.6% (human-validated judge) and prompt caching cut cost per turn by 16%. Measured hybrid BM25 + RRF retrieval twice and retired it: the lever was the content (reprocessed 1,027 documents with multimodal OCR).
- **Measuring what gets resolved — Jarvis:** the official rule credited each chat to whoever touched it, giving the AI ~25%. Built Jarvis, a daily LLM judge (Haiku) that decides who resolved each conversation — AI, agent or nobody — with the same rule for both: the AI handles 68.5% of attendable chats. The previous classifier matched human audits only 41.1% of the time; Jarvis replaces it and re-types the ticket at close (in shadow).
- **LangGraph orchestration:** implemented in LangGraph and LangChain Jarvis's closing engine and the service's nightly processes (resolution judge, scoring, ticket typing and expert-approved curation): typed state, parallel judges, PostgreSQL checkpoints to resume a run and human-review interrupts. Rewrote the bot pipeline as a graph with parity proven on its 646 tests.
- **Built for others to build on:** one-click human-approved curation (~150 new Q&As a week, ~US$10/month), an admin console with per-module permissions (44 accounts) and a model gateway other teams use. Ran production on AWS (Bedrock, EC2, systemd, IAM, secrets) with per-stage observability (p50/p95 latency, cost per turn) and module-by-module releases with rollback plans.
- **The service, not just the model:** redesigned support operations (per-agent WIP limits, FIFO assignment, load balancing): CSAT rose from 30% to 64% (weekly active survey), first response fell from 20.2 to 15.2 min and peak load per agent from 27 to 20 chats.

- **What's next:** designing an online triage channel that will route each request to AI, L1, L2 or L3 (data-estimated ceiling: ~6 points of escalation) and building RAG v2, which will act on customer systems (restarts, updates) with per-action permission levels. Trained 16 people across 3 teams on AI adoption.

Internal AI assistant & automation (2024 – 2025)

- Built the internal AI assistant (OpenAI API with tool calling over SQL Server, the in-house Jira-style platform's requirements and Freshdesk tickets): support agents, consultants and operations teams asked in natural language instead of opening several systems. Orchestration prototyped in LangChain; the production path went directly against the API.
- Automated the operational loop with n8n + Microsoft Teams (SLA/TTR/FCR bonus-violation alerts for L2 and L3 agents) and Power BI dashboards fed from the Jira-style platform.

Data, automation and platforms (Mar 2022 – Present)

- Replaced manual reporting with a Python pipeline (Freshdesk → SQL Server with transactional loads and hash integrity checks → Power BI via REST API; validated against the manual load, 9,607 vs. 9,606 rows; runs 4 times a day) and build medallion ETL pipelines (GitHub → SQL Server); cut a daily SQL Server load from 16.7 to 7.6 min (−54%) with identical results.
- Lead the in-house service and project management platform (Jira-style) used by 500+ employees; integrated 3 applications. Migrated 3 monoliths to microservices (.NET 8 / ASP.NET Core) in an 8-person agile team and modernized React/Angular modules.

Independent AI Consultant & Instructor — Part-time, remote 2026 – Present

- Founded and build Tocaturno (tocaturno.com), a production SaaS for 24/7 WhatsApp reception (bookings, reminders and handoff to a person for sensitive cases). Design AI agents and automations for clients (WhatsApp, n8n, MCP servers) and teach hands-on AI classes.

Freelance Software Developer — Self-employed, remote Jun 2019 – Feb 2022

- Built 6+ full-stack web applications for Colombian SMBs (.NET Core, React, JavaScript; 1,500+ end users), including a PayU e-commerce platform (US$25K+ in first-year sales) and a multi-site inventory system (stock discrepancies −90%).

TECHNICAL SKILLS

- **AI & LLM:** RAG • AI Agents • LangGraph / LangChain (state graphs, checkpoints, human-in-the-loop) • Tool Calling • MCP • Embeddings & Semantic Search (pgvector) • Reranking • LLM Evaluation (LLM-as-Judge, golden sets, A/B) • LLMOps • Guardrails & Answer Validation • Multimodal OCR • Claude (AWS Bedrock) • Azure OpenAI / AI Foundry • OpenAI API • n8n
- **Backend & Frontend:** Python • FastAPI • C# • .NET 8 • ASP.NET Core • REST APIs • WebSockets • Microservices • System Design • Clean Architecture / SOLID • Testing (pytest, xUnit/NUnit) • React • TypeScript • Angular
- **Data & Cloud:** SQL Server (T-SQL) • PostgreSQL / pgvector • ETL Pipelines • Medallion Architecture • Power BI • AWS (Bedrock, EC2, S3, IAM) • Azure DevOps • GCP (Cloud Run) • Linux • Docker • CI/CD • Git
- **Practices & Leadership:** End-to-end ownership • Measured releases (baseline, A/B, rollback) • Observability (traces, latency, cost) • Security (IAM, secrets, JWT, RBAC) • AI-assisted development (Claude Code, Cursor) • Stakeholder communication & executive reporting • Technical mentoring

EDUCATION

- **B.Sc. in Systems Engineering** — UNAD — 2024 – Present (expected 2027) • **Technologist in Systems Analysis and Development** — SENA — 2020 – 2022