

AUGUSTO OLIVEIRA

AI Engineer · Production RAG Systems

Natal, Brazil (UTC-3) · jose.aug18@outlook.com · +55 84 98785-5144

linkedin.com/in/augusto-oliveira-4a8068235 · github.com/Augusto240

Open to full-time remote (contract / EOR) · 4+ hours overlap with US Eastern

SUMMARY

Software engineer with production experience building and operating a Retrieval-Augmented Generation system for a Brazilian federal ministry. I own the parts most RAG projects skip: evaluation harnesses, distributed tracing, safety guardrails and latency analysis. Python, LangChain, Milvus, PostgreSQL, Docker. Fluent English, remote-ready.

EXPERIENCE

AI Engineer — CCSL — Federal Institute of Rio Grande do Norte (IFRN)

Jan 2025 – Aug 2026 · Natal, Brazil

Public-sector R&D centre. Primary engineer on SmartFAQ, a citizen-facing AI assistant for Brazil's Ministry of Communications.

- Designed and shipped a production RAG pipeline in Python: LangChain, Milvus, BGE-M3 embeddings and Llama 3.3 70B via Ollama, served through Flask microservices with Celery/Redis queuing and PostgreSQL as the system of record.
- Built the evaluation harness — context precision and context recall metrics plus an LLM-as-a-judge, including calibration tests to measure the judge's own effect on the scores.
- Instrumented end-to-end observability: one request ID propagated across four services, ten structured trace events with per-phase latency, and a response-provenance field separating true vector-RAG answers from fallbacks and errors.
- Audited the codebase and found five undocumented paths that returned answers without querying the vector index. Enforced a vector-only, fail-closed pipeline that returns HTTP 503 instead of fabricated text, covered by automated conformance tests.
- Diagnosed and fixed a full production outage: a synchronous safety-model warm-up was blocking the API from binding its port. Moved it to a background thread with a keyword fallback, so the service stays available when the guardrail model fails.
- Profiled the system and documented the ten largest latency bottlenecks with file-and-line evidence and estimated impact, identifying LLM inference (50–80% of request time) and serialised URL validation as the dominant costs.
- Shipped a governance CRUD (Flask-Admin) with role-based access, soft delete and automatic deduplication hashing; added drift detection between PostgreSQL and the vector index, plus a filesystem lock preventing index corruption under concurrency.
- Built the ingestion layer: Selenium crawlers for JavaScript-rendered government portals across multiple domains, feeding an ETL that produced the structured question-and-answer knowledge base.

Full-Stack Developer — NOCS Lab — IFRN

May 2024 – Dec 2025 · Parnamirim, Brazil

- Built a vehicle access control system end to end: Angular/TypeScript front end, Node.js/Express API, MySQL, containerised with Docker.
- Integrated cameras and access-control hardware over MQTT for real-time events between field devices and the platform.
- Implemented OAuth2/JWT authentication with role-based access control, including institutional SSO integration, and automated database and image backups with restore testing.

TECHNICAL SKILLS

AI & LLM: RAG architecture, retrieval evaluation (context precision/recall, LLM-as-a-judge), guardrails and safety filtering, prompt engineering, LangChain, Milvus, Ollama, sentence-transformers / BGE-M3

Backend: Python, Flask, Django, Celery, Redis, Node.js, Express, REST APIs

Data & Pipelines: PostgreSQL, MySQL, ETL, Selenium, BeautifulSoup, pandas, ZenML, Airflow, MLflow

Infrastructure: Docker, Git, GitLab CI/CD, structured logging and distributed tracing, Linux

Frontend: TypeScript, Angular, React

EDUCATION & LANGUAGES

Technology in Internet Systems — Federal Institute of Rio Grande do Norte (IFRN). Expected 2027; thesis written, pending defence.

Languages: Portuguese (native) · English (fluent)

Also: twelve technical reports on the system's architecture, audits and performance; co-writing a research paper on the platform.