JULIO CÉSAR AGUILAR VERA

AI Engineer | Full Stack Developer

Lima, Peru | +51 956-582-267 | jcesar.a.15@gmail.com

LinkedIn | GitHub

PROFESSIONAL SUMMARY

AI Engineer and Full Stack Developer with a background in Industrial Engineering and 16 years of leadership in the Peruvian Navy. I design and build AI applications and web products end to end, from architecture to production deployment.

I have built LLM-based solutions, including voice agents, RAG systems and multi-agent architectures with LangChain, vector databases (Pinecone and ChromaDB), automated answer evaluation (LLM-as-a-judge), observability with Langfuse, intelligent OCR, and applications that use LLMs to solve complex problems with reliable, traceable information. I work with Spec-Driven Development: I define measurable criteria before coding and evaluate every system against real cases.

I complement this experience with Full Stack development using Python, TypeScript, FastAPI, Next.js, NestJS, PostgreSQL and MongoDB. I founded and run **RedLegal.pe**, a legal marketplace in production with 90+ registered lawyers and 50+ client–lawyer contacts generated.

Core stack: Python, TypeScript, FastAPI, LangChain, RAG, LLM, Vapi, Langfuse, Pinecone, ChromaDB, Next.js, NestJS, PostgreSQL.

KEY ACHIEVEMENTS

- **RedLegal.pe —** Marketplace in production with real traction:

90+ registered lawyers, 50+ client–lawyer contacts generated, and a 16% sign-up conversion rate at launch.

- **Lucía (voice agent) —** 13/13 test scenarios passed at US$0.065 per call.
- 16 years leading multidisciplinary teams in the Peruvian Navy, managing operations under high pressure and coordinating resources in complex environments.

PROFESSIONAL EXPERIENCE

Full Stack AI Developer — Marchand Ramos Contratistas Generales *Sep 2026 – Present*

- Built the corporate website with Next.js and React, owning design, frontend and deployment of the company pages.
- Own the company's AI roadmap: currently designing and implementing a customer service chatbot and a RAG system for internal queries, applying the same end-to-end approach (LangChain, LLMs).

Founder & Full Stack Developer — RedLegal.pe *Jan 2026 – Present*

Legal marketplace in production that connects people with verified lawyers in Peru by practice area and city. End-to-end ownership: product, architecture, backend, frontend, deployment and growth.

- **Launched and run the platform in production:** 90+ registered lawyers and 50+ client–lawyer contacts generated.
- **Implemented Next.js 14 (App Router + SSR)** to generate 150+ dynamic landing pages by practice area × city, fully indexed by Google.
- **Designed a REST API with NestJS** using a modular architecture (dependency injection; lawyer, search and user modules).
- **Modeled lawyer ↔ practice area ↔ city relationships** in PostgreSQL with TypeORM, optimizing multi-filter queries.
- **Measured the conversion funnel with PostHog** (custom events) to detect drop-offs and make data-driven decisions; reached a 16% sign-up conversion rate at launch.

Stack: Next.js 14, NestJS, TypeScript, PostgreSQL, TypeORM, Tailwind CSS, PostHog, Vercel.

Naval Officer — Peruvian Navy *Jan 2010 – Present*

- Led multidisciplinary teams in naval operations, coordinating personnel and resources under high pressure and tight deadlines.
- Developed skills that transfer to tech: decision-making under uncertainty, effective communication with diverse stakeholders, and management of projects with multiple dependencies.

PROJECTS

AI Engineer — Lucía, AI Voice Receptionist *Nov 2026*

Spanish-speaking voice agent that answers calls for a medical practice: checks availability in Google Calendar, books the appointment, sends the payment link via WhatsApp and hands off urgent cases to a person. Personal project built with Spec-Driven Development.

- **Orchestrated the voice pipeline with Vapi** (Deepgram, GPT-4o-mini and Azure es-PE voice) and a FastAPI backend with hexagonal architecture that exposes the agent's tools: availability, booking, payment and handoff.
- **Integrated payments with Culqi and WhatsApp (Twilio):** the appointment is confirmed only when the payment webhook arrives, and is released automatically after 30 minutes if unpaid.
- **Evaluated the system with real calls:** 13/13 scenarios passed after 3 iterations, 8% WER, 478 ms tool latency and US$0.065 per call; Langfuse traces and 136 tests with pytest.

Stack: Python, FastAPI, Vapi, OpenAI, Deepgram, Google Calendar API, Culqi, Twilio, SQLModel, APScheduler, Langfuse, pytest.

AI Engineer — Contract Comparison with Vision and Agents *Nov 2026*

System that compares a contract with its amendment from scanned images and returns the changes in a structured, validated format. Capstone project for Module 4 of AI Engineering (Henry).

- **Parsing with GPT-4o vision** instead of traditional OCR, preserving the clause structure.
- **Designed 2 specialized agents:** a contextualization agent (gpt-4o-mini) that aligns the sections and an extraction agent (gpt-4o with structured outputs) that identifies the changes; output validated with Pydantic.
- **Results:** 14/14 changes detected with no false positives; latency cut from 14.5 s to 8.7 s (−40%) and cost −20% through parallel parsing (asyncio) and a smaller model.

Stack: Python, OpenAI (GPT-4o), Pydantic, Langfuse, asyncio, pytest.

AI Engineer — Internal Support Multi-Agent System *Oct 2026*

System that automatically routes employees' internal questions to the right department (HR, IT or Finance) and answers with information grounded in each department's documentation, citing the source. Capstone project for Module 3 of AI Engineering (Henry).

- Designed a multi-agent architecture: an orchestrator with structured output (Pydantic) classifies intent and conditionally routes to 3 specialized RAG agents, each with its own vector collection in ChromaDB. Out-of-scope queries are resolved without invoking agents, reducing latency and cost.
- Implemented an evaluator agent (LLM-as-a-judge) that scores each answer on relevance, completeness and accuracy. After finding it was too lenient, I calibrated the rubric with concrete criteria per level: a weak answer dropped from 8 to 6 on completeness, while a complete answer kept its 10.
- Instrumented end-to-end observability with Langfuse (traces per query, spans per agent, scores, latency and cost) and validated the system with a test set that includes ambiguous and informal queries: 100% correct routing (17/17), 8.7/10 average quality and ~US$0.002 per query.
- Published an interactive demo on Streamlit Cloud that shows how the system decides: chosen department, sources consulted and evaluation of each answer.

Stack: Python, LangChain, OpenAI, RAG, ChromaDB, Langfuse, Pydantic, Streamlit.

AI Engineer — Legal Assistant RAG *Aug 2026*

Retrieval-Augmented Generation (RAG) system for querying Peru's Constitution and main codes and laws (Civil, Criminal, Labor, Procedural, among others) in natural language, returning grounded answers with the exact article and code cited, along with the source documents used.

- Designed and built the solution end to end, focused on reliability and production architecture: an ingestion pipeline with selective OCR (automatically detects and fixes low-quality scanned pages without touching good ones), automatic classification of documents by legal area, and a hybrid retriever that combines MMR search, automatic query reformulation (MultiQuery) and simple similarity in a weighted ensemble. The knowledge base lives in Pinecone (cloud vector store), decoupled from any local machine, and the system is deployed on Streamlit Cloud.
- As part of the process, I compared a local language model (Ollama) with a cloud model (OpenAI) on the same queries: the local model generated non-existent legal provisions, while the more capable model cited real information and honestly acknowledged when it could not find a provision in the context. That validation drove the choice of the final model for a high-stakes domain like law.
- Modular architecture (services, prompts, config) following SOLID principles, with fully decoupled ingestion and query layers.

Stack: Python, LangChain, LLM, Pinecone, RAG, Tesseract OCR, Streamlit.

AI Engineer — AI-Powered CV Evaluation System *Jul 2026*

Web application that evaluates how well a CV matches a job posting using an LLM (Google Gemini), returning a structured report with relevant experience, key skills, strengths, areas for improvement and a job-fit percentage.

- Designed and built the solution end to end, focused on taking it beyond a local demo to a production-ready system: prompt engineering with specific context to guide the model, structured outputs validated with Pydantic to ensure consistent LLM responses, robust error handling so the app never fails unpredictably, and a modular architecture (models, prompts, services, ui) following the same discipline I apply in Full Stack projects.

Stack: Python, LLM, LangChain, Chroma, Streamlit.

Full Stack Developer — Text-to-Speech Document Converter *Jun 2026*

Web application that converts articles, Word documents and PDFs into MP3 audio files, built as a personal project applying what I learned in the advanced Python module of the Master's in Artificial Intelligence.

- **Designed and built the solution end to end:** text extraction from multiple sources (web article scraping, Word and PDF parsing), text-to-speech conversion with Google Text-to-Speech, and an interactive interface built with Streamlit.
- Applied a modular architecture with separation of concerns (SOLID principles), version control with Git from the first commit, and complete documentation for environment reproducibility (venv + requirements.txt).

Stack: Python, Streamlit, newspaper3k, python-docx, PyPDF2, gTTS.

SKILLS

Languages: Python, TypeScript, JavaScript.

AI Engineering: LangChain, Retrieval-Augmented Generation (RAG), multi-agent systems, voice agents (Vapi, Deepgram), LLM-as-a-judge, LLM APIs (OpenAI, Gemini), Langfuse, Pydantic, Streamlit.

Frontend: HTML5, CSS, React, Next.js, Tailwind CSS.

Backend: Python (FastAPI), Node.js, NestJS, Express, TypeORM, SQLModel.

Databases: PostgreSQL, SQLite, MongoDB, Pinecone, ChromaDB.

Analytics & Product: PostHog, Google Search Console, UTM tracking.

Tools: Git, GitHub, Postman, Vercel, Render, pytest, Ruff.

Integrations: Google Calendar API, Twilio (WhatsApp), Culqi (payments).

TECHNICAL EDUCATION

AI Engineering *Sep 2026 – Nov 2026*

Henry. Agentic AI, RAG and end-to-end AI systems with LangChain, Pinecone and FastAPI.

Master's in Artificial Intelligence *Mar 2026 – Present*

Conquer Blocks

Full Stack Web Developer *Apr 2025 – Dec 2025*

Henry. SOLID principles, Clean Code, testing and agile development.

ACADEMIC BACKGROUND

Industrial Engineering *Jun 2007 – Dec 2009*

University of Piura

Naval and Maritime Sciences *Feb 2005 – Dec 2009*

Peruvian Naval Academy

LANGUAGES

Spanish: Native

English: B2 – Upper Intermediate (technical reading, writing and conversation)