JULIO CÉSAR AGUILAR VERA

AI Engineer | Full Stack Developer

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

PERFIL PROFESIONAL

AI Engineer | Full-Stack Developer con background en Ingeniería Industrial y 16 años de liderazgo en la Marina de Guerra del Perú. Diseño y desarrollo aplicaciones de IA y productos web de extremo a extremo, desde la arquitectura hasta el despliegue en producción.

He construido soluciones basadas en LLMs, incluyendo agentes de voz, sistemas RAG y arquitecturas multiagente con LangChain, bases de datos vectoriales (Pinecone y ChromaDB), evaluación automática de respuestas (LLM-as-a-judge), observabilidad con Langfuse, OCR inteligente y aplicaciones que integran LLMs para resolver problemas complejos con información confiable y trazable. Trabajo con Spec-Driven Development: defino criterios medibles antes de programar y evalúo cada sistema con casos reales.

Complemento esta experiencia con desarrollo Full Stack utilizando Python, TypeScript, FastAPI, Next.js, NestJS, PostgreSQL y MongoDB. Fundé y opero **RedLegal.pe**, un marketplace legal en producción con +90 abogados registrados y +50 contactos generados.

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

LOGROS DESTACADOS

- **RedLegal.pe —** Marketplace en producción con tracción real:

+90 abogados registrados, +50 contactos generados entre clientes y abogados y 16% de conversión a registro en el lanzamiento.

- **Lucía (agente de voz) —** 13/13 escenarios aprobados y US$ 0,065 por llamada.
- 16 años liderando equipos multidisciplinarios en la Marina de Guerra del Perú, gestionando operaciones bajo alta presión y coordinando recursos en entornos complejos.

EXPERIENCIA PROFESIONAL

Full Stack AI Developer — Marchand Ramos Contratistas Generales *set. 2026 – actualidad*

- Desarrollo del sitio web corporativo con Next.js y React, a cargo del diseño, frontend y despliegue de las páginas de la empresa.
- A cargo de la hoja de ruta de IA de la empresa: diseño e implementación en curso de un chatbot de atención al cliente y un sistema RAG para consultas internas, aplicando el mismo enfoque end-to-end (LangChain, LLMs).

Fundador & Full Stack Developer — RedLegal.pe *ene. 2026 – actualidad*

Marketplace legal en producción que conecta personas con abogados verificados en Perú por especialidad y ciudad. Responsable end-to-end: producto, arquitectura, backend, frontend, despliegue y crecimiento.

- **Lancé y opero la plataforma en producción:** +90 abogados registrados y +50 contactos generados entre clientes y abogados.
- **Implementé Next.js 14 (App Router + SSR)** para generar 150+ landing pages dinámicas por especialidad × ciudad, con indexación completa en Google.
- **Diseñé una API REST con NestJS** con arquitectura modular (inyección de dependencias; módulos de abogados, búsqueda y usuarios).
- **Modelé las relaciones abogados ↔ especialidades ↔ ciudades** en PostgreSQL con TypeORM, optimizando las consultas de filtrado múltiple.
- **Medí el embudo de conversión con PostHog** (eventos personalizados) para detectar abandonos y decidir con datos; en el lanzamiento alcanzó 16% de conversión a registro.

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

Oficial de Marina — Marina de Guerra del Perú *ene. 2010 – actualidad*

- Liderazgo de equipos multidisciplinarios en operaciones navales, coordinando personal y recursos bajo condiciones de alta presión y plazos estrictos.
- Desarrollo de habilidades transferibles al sector tech: toma de decisiones bajo incertidumbre, comunicación efectiva con stakeholders diversos, y gestión de proyectos con múltiples dependencias.

P R O Y E C T O S

AI Engineer — Lucía, Recepcionista de Voz con IA *nov. 2026*

Agente de voz en español que atiende las llamadas de un consultorio: consulta disponibilidad en Google Calendar, agenda la cita, envía el link de pago por WhatsApp y deriva las urgencias a una persona. Proyecto personal construido con Spec-Driven Development.

- **Orquesté el pipeline de voz con Vapi** (Deepgram, GPT-4o-mini y voz Azure es-PE) y un backend FastAPI con arquitectura hexagonal que expone las herramientas del agente: disponibilidad, reserva, pago y derivación.
- **Integré pagos con Culqi y WhatsApp (Twilio):** la cita se confirma solo al recibir el webhook de pago y se libera automáticamente a los 30 minutos si no se paga.
- **Evalué el sistema con llamadas reales:** 13/13 escenarios aprobados tras 3 iteraciones, WER de 8%, 478 ms de latencia en herramientas y US$ 0,065 por llamada; trazas en Langfuse y 136 tests con pytest.

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

AI Engineer — Comparador de Contratos con Visión y Agentes *nov. 2026*

Sistema que compara un contrato con su enmienda a partir de imágenes escaneadas y devuelve los cambios en un formato estructurado y validado. Proyecto integrador del Módulo 4 de AI Engineering (Henry).

- **Parseo con visión de GPT-4o** en lugar de OCR tradicional, conservando la estructura de cláusulas.
- **Diseñé 2 agentes especializados:** uno de contextualización (gpt-4o-mini) que alinea las secciones y otro de extracción (gpt-4o con structured outputs) que identifica los cambios; salida validada con Pydantic.
- **Resultados:** 14/14 cambios detectados sin falsos positivos; latencia reducida de 14,5 s a 8,7 s (−40%) y costo −20% con parseo en paralelo (asyncio) y modelo mini.

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

AI Engineer — Multiagente de Soporte Interno *oct. 2026 – oct. 2026*

Sistema que enruta automáticamente las consultas internas de los colaboradores al área correcta (Recursos Humanos, TI o Finanzas) y responde con información fundamentada en la documentación de cada área, citando la fuente. Proyecto integrador del Módulo 3 de AI Engineering (Henry).

- Diseñé una arquitectura multiagente: un orquestador con salida estructurada (Pydantic) clasifica la intención y enruta de forma condicional a 3 agentes RAG especializados, cada uno con su propia colección vectorial en ChromaDB. Las consultas fuera de alcance se resuelven sin invocar agentes, reduciendo latencia y costo.
- Implementé un agente evaluador (LLM-as-a-judge) que puntúa cada respuesta en relevancia, completitud y precisión. Al detectar que era demasiado permisivo, calibré la rúbrica con criterios concretos por nivel: una respuesta débil bajó de 8 a 6 en completitud, mientras una respuesta completa mantuvo 10.
- Instrumenté observabilidad de punta a punta con Langfuse (trazas por consulta, spans por agente, scores, latencia y costo) y validé el sistema con un set de pruebas que incluye consultas ambiguas e informales: 100% de enrutamiento correcto (17/17), calidad promedio de 8.7/10 y ~US$ 0.002 por consulta.

- Publiqué una demo interactiva en Streamlit Cloud que muestra cómo decide el sistema: área elegida, fuentes consultadas y evaluación de cada respuesta.

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

AI Engineer — Assistance Legal RAG *ago. 2026 – ago. 2026*

Sistema de Retrieval Augmented Generation (RAG) que permite consultar en lenguaje natural la Constitución y los principales códigos y leyes del Perú (Civil, Penal, Laboral, Procesal, entre otros), devolviendo respuestas fundamentadas con cita exacta de artículo y código, junto con los documentos fuente utilizados.

- Diseñé y desarrollé la solución end-to-end, con foco en confiabilidad y arquitectura de producción: pipeline de ingesta con OCR selectivo (detecta y corrige automáticamente páginas escaneadas de baja calidad sin tocar las que ya están bien), clasificación automática de documentos por materia legal, y un retriever híbrido que combina búsqueda MMR, reformulación automática de consultas (MultiQuery) y similitud simple en un Ensemble ponderado. La base de conocimiento vive en Pinecone (vector store en la nube), desacoplada de cualquier máquina local, y el sistema despliega en Streamlit Cloud.
- Como parte del proceso, comparé el rendimiento de un modelo de lenguaje local (Ollama) contra uno en la nube (OpenAI) ante las mismas consultas: el modelo local generó contenido normativo inexistente, mientras que el modelo más capaz citó información real y reconoció honestamente cuando no encontraba una norma en el contexto. Esa validación definió la elección del modelo final para un dominio de alta responsabilidad como el legal.
- Arquitectura modular (services, prompts, config) siguiendo principios SOLID, con capas de ingesta y consulta completamente desacopladas.

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

AI Engineer — Sistema de Evaluación de CVs con IA *jul. 2026 – jul. 2026*

Aplicación web que evalúa la compatibilidad entre un CV y una oferta de trabajo usando un LLM (Google Gemini), devolviendo un informe estructurado con experiencia relevante, habilidades clave, fortalezas, áreas de mejora y porcentaje de ajuste al puesto.

- Diseñé y desarrollé la solución end-to-end, con foco en llevarla más allá de una demo local hacia un sistema apto para producción: prompt engineering con contexto específico para guiar al modelo, salidas estructuradas y validadas con Pydantic para garantizar respuestas consistentes del LLM, manejo de errores robusto para que la app nunca falle de forma impredecible, y arquitectura modular (models, prompts, services, ui) siguiendo la misma disciplina que aplico en proyectos Full Stack.

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

Full Stack Developer — Conversor de Documentos Texto a Voz *jun. 2026 – jun. 2026*

Aplicación web que convierte artículos, documentos Word y PDFs en archivos de audio MP3, desarrollada como proyecto personal aplicando lo aprendido en el módulo avanzado de Python del Máster en Inteligencia Artificial.

- **Diseñé y desarrollé la solución end-to-end:** extracción de texto desde múltiples fuentes (scraping de artículos web, parsing de Word y PDF), conversión de texto a voz con Google Text-to-Speech, e interfaz interactiva construida en Streamlit.
- Apliqué arquitectura modular con separación de responsabilidades (principios SOLID), control de versiones con Git desde el primer commit, y documentación completa para reproducibilidad del entorno (venv + requirements.txt).

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

TECNOLOGÍAS

Lenguajes: Python, TypeScript, JavaScript.

IA Engineering: LangChain, Retrieval-Augmented Generation (RAG), sistemas multiagente, agentes de voz (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.

Base de Datos: PostgreSQL, SQLite, MongoDB, Pinecone, ChromaDB.

Analytics y Producto: PostHog, Google Search Console, UTM tracking.

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

Integraciones: Google Calendar API, Twilio (WhatsApp), Culqi (pagos).

EDUCACIÓN EN TECNOLOGÍA

AI Engineering *set. 2026 – nov. 2026*
Henry. Agentic AI, RAG y sistemas de IA end-to-end con LangChain, Pinecone y FastAPI.

Master en Inteligencia Artificial *mar. 2026 – actualidad*
Conquer Blocks

Full Stack Web Developer *abr. 2025 – dic. 2025*
Henry. Principios SOLID, Clean Code, testing y desarrollo ágil.

FORMACIÓN PROFESIONAL

Ingeniería Industrial *jun. 2007 – dic. 2009*
Universidad de Piura

Ciencias Marítimas Navales *feb. 2005 – dic. 2009*
Escuela Naval del Perú

IDIOMAS

Español: Nativo

Inglés: B2 – Intermedio (lectura técnica, escritura y conversación)