# JARBY JESSID COSME

**Ingeniero de Software e IA | Sistemas de IA en Producción, Backend y Datos**

Cali, Colombia · jessidc95@gmail.com · Inglés B2 (intermedio-alto)
linkedin.com/in/jarbyjessidcosme · github.com/JessidC · Portafolio: jarby-portfolio.netlify.app/es/
Remoto, híbrido o presencial en Colombia; remoto internacional (GMT-5)

## RESUMEN PROFESIONAL

Ingeniero de software con ~7 años de experiencia (Python, C#/.NET, TypeScript, SQL) que lleva proyectos de cero a producción. Desde 2025 diseño y hoy soy el responsable técnico único de la plataforma de IA del soporte de SIESA, con un objetivo: mejorar el servicio de punta a punta, no "poner un chatbot". Empecé por los datos, construí un RAG sobre Claude (AWS Bedrock) que reemplazó a un proveedor externo en 15 módulos del ERP (~1.000 conversaciones al día), medí lo que de verdad se resolvía y orquesté en LangGraph los agentes que lo miden. Presento casos de negocio al CEO.

## EXPERIENCIA PROFESIONAL

### Desarrollador de Software — SIESA, Cali, Colombia
mar. 2022 – Actualidad

**Plataforma de IA del servicio de soporte — responsable técnico único (2025 – Actualidad)**

- **Primero el diagnóstico:** analicé 90 días del servicio (55.350 chats, ~964 mil mensajes) y presenté a dirección el caso de negocio del reemplazo con 6 decisiones priorizadas por impacto y costo; una auditoría posterior (NPS, comentarios de clientes y trazas) mostró que el indicador oficial medía contacto, no resolución.
- **Un RAG a la medida del servicio:** diseñé, construí y opero el RAG que reemplazó al proveedor externo en el chat de soporte, sobre tres reglas: no inventar (validación de citas y cifras, ~170 reglas deterministas), escalar a una persona cuando la confianza es baja (tool calling) y no evadir al cliente. La escalación a humano en Nómina bajó de 40,2% a 32,7% por chat (p = 0,0005) y las respuestas que remitían a "contactar soporte", de 6,95% a 0,25%. *Python/FastAPI sin frameworks en el camino crítico; Claude Sonnet 4.6 y Haiku 4.5 en Bedrock; PostgreSQL + pgvector; Cohere Rerank 3.5.*
- **Calidad medida, no supuesta:** golden sets con juez LLM como compuerta de salida, replay A/B sobre trazas reales y pruebas de significancia: la correctitud subió de 71,6% a 89,6% (juez validado por humanos) y el caché de prompt bajó 16% el costo por turno. Medí dos veces la recuperación híbrida BM25 + RRF y la retiré: la palanca era el contenido (re-procesé 1.027 documentos con OCR multimodal).
- **Medir lo que se resuelve — Jarvis:** la regla oficial le daba a la IA ~25% porque acreditaba el chat a quien lo tocara. Construí Jarvis, un juez LLM diario (Haiku) que decide quién resolvió cada conversación — IA, agente o nadie— con la misma regla para ambos: la IA atiende el 68,5% de los chats atendibles. El tipificador anterior coincidía con la auditoría humana solo en un 41,1%; Jarvis lo reemplaza y re-tipifica el ticket al cierre (en sombra).
- **Orquestación en LangGraph:** implementé en LangGraph y LangChain el motor de cierre de Jarvis y los procesos nocturnos del servicio (juez de resolución, calificación, tipificación y curación con aprobación del experto): estado tipado, jueces en paralelo, checkpoints en PostgreSQL para retomar una corrida y pausas de revisión humana. Reescribí el pipeline del bot como grafo con paridad probada en sus 646 pruebas.
- **Para que otros construyan encima:** curación con aprobación humana en un clic (~150 Q&A nuevas por semana, ~USD 10 al mes), consola con permisos por módulo (44 cuentas) y una pasarela de modelos que usan otros equipos. Operé producción en AWS (Bedrock, EC2, systemd, IAM, secretos) con observabilidad por etapa (latencia p50/p95, costo por turno) y liberaciones por módulo con plan de reversa.
- **El servicio, no solo el modelo:** rediseñé la operación de atención (límite de carga por agente, asignación FIFO y balanceo): el CSAT subió de 30% a 64% (encuesta semanal activa), la primera respuesta bajó de 20,2 a 15,2 min y la carga máxima por agente, de 27 a 20 chats.

- **Lo que sigue:** diseño un triage en línea que enrutará cada solicitud a IA, L1, L2 o L3 (techo estimado con datos: ~6 puntos de escalación) y construyo la v2 del RAG, que actuará sobre los sistemas de los clientes (reinicios, actualizaciones) con niveles de permiso por acción. Formé en IA a 16 personas de 3 equipos.

**Asistente interno de IA y automatización (2024 – 2025)**

- Construí el asistente interno de IA (API de OpenAI con tool calling sobre SQL Server, los requerimientos de la plataforma tipo Jira y los tickets de Freshdesk): soportistas, consultores y operaciones consultaban en lenguaje natural en vez de abrir varios sistemas. Orquestación prototipada en LangChain; la versión en producción fue directa contra la API.
- Automaticé el bucle operativo con n8n + Microsoft Teams (alertas SLA/TTR/FCR de bonificación para agentes L2 y L3) y dashboards de Power BI desde la plataforma tipo Jira.

**Datos, automatización y plataformas (mar. 2022 – Actualidad)**

- Reemplacé la reportería manual por un pipeline en Python (Freshdesk → SQL Server con cargas transaccionales e integridad por hash → Power BI por API REST; validado contra la carga manual, 9.607 vs. 9.606 filas; corre 4 veces al día) y construyo ETL con capas medallion (GitHub → SQL Server); bajé una carga diaria de SQL Server de 16,7 a 7,6 min (−54%) con resultados idénticos.
- Lidero la plataforma interna de gestión de servicio y proyectos (tipo Jira, desarrollo propio) que usan más de 500 empleados; integré 3 aplicativos. Migré 3 monolitos a microservicios (.NET 8 / ASP.NET Core) en un equipo ágil de 8 y modernicé módulos React/Angular.

**Consultor e Instructor de IA Independiente — Part-time, remoto** 2026 – Actualidad

- Fundé y desarrollo Tocaturno (tocaturno.com), un SaaS en producción de recepción por WhatsApp 24/7 (citas, recordatorios y paso a una persona en casos delicados). Diseño agentes y automatizaciones para clientes (WhatsApp, n8n, servidores MCP) e imparto clases prácticas de IA.

**Desarrollador de Software Freelance — Independiente, remoto** jun. 2019 – feb. 2022

- Desarrollé más de 6 aplicaciones web full-stack para pymes colombianas (.NET Core, React, JavaScript; más de 1.500 usuarios finales), entre ellas un e-commerce con PayU (más de USD 25.000 en ventas el primer año) y un sistema de inventarios multisede (discrepancias de stock −90%).

## HABILIDADES TÉCNICAS

- **IA y LLM:** RAG • Agentes de IA • LangGraph / LangChain (grafos de estado, checkpoints, revisión humana) • Tool calling • MCP • Embeddings y búsqueda semántica (pgvector) • Reranking • Evaluación de LLM (LLM-as-Judge, golden sets, A/B) • LLMOps • Guardas y validación de respuestas • OCR multimodal • Claude (AWS Bedrock) • Azure OpenAI / AI Foundry • OpenAI API • n8n
- **Backend y frontend:** Python • FastAPI • C# • .NET 8 • ASP.NET Core • APIs REST • WebSockets • Microservicios • Diseño de sistemas • Clean Architecture / SOLID • Pruebas (pytest, xUnit/NUnit) • React • TypeScript • Angular
- **Datos y nube:** SQL Server (T-SQL) • PostgreSQL / pgvector • Pipelines ETL • Arquitectura medallion • Power BI • AWS (Bedrock, EC2, S3, IAM) • Azure DevOps • GCP (Cloud Run) • Linux • Docker • CI/CD • Git
- **Prácticas y liderazgo:** Entrega de punta a punta • Liberaciones medidas (línea base, A/B, reversa) • Observabilidad (trazas, latencia, costo) • Seguridad (IAM, secretos, JWT, roles y permisos) • Desarrollo asistido por IA (Claude Code, Cursor) • Stakeholders e informes ejecutivos • Mentoría técnica

## EDUCACIÓN

- **Ingeniería de Sistemas** — UNAD — 2024 – Actualidad (en curso, grado esperado 2027) • **Tecnólogo en Análisis y Desarrollo de Sistemas de Información** — SENA — 2020 – 2022