

Ricardo Brossard

Full-Stack Developer · Applied AI · TypeScript · Python

Concepción del Uruguay, Entre Ríos, Argentina (UTC-3) — 100% Remote · info@ricardobrossard.com · +54 3442 679907 · linkedin.com/in/ricardo-brossard · github.com/ricardobing · ricardobrossard.com/en/projects

PROFESSIONAL SUMMARY

Systems engineer and full-stack developer with applied AI: I bring artificial intelligence into business applications that people use every day, and I own the whole product, from understanding the problem to production and support. I work autonomously and take responsibility for the result end to end. My rule with AI: it interprets and talks, an explicit formula computes the numbers, and a person decides when there is doubt.

FEATURED PROJECTS

Lexinton Propiedades — Sales management system integrated with Tokko, listing portals, WhatsApp and Google

Real estate agency in Buenos Aires · Live since May 2026 · 10 users · 4 systems in 2 years for the same client · panel.lexinton.com.ar

The problem: inquiries from buyers and renters arrived through several channels and all ended up in a shared mailbox. Each one depended on somebody copying it by hand into a spreadsheet, and the owner did not know how many came in or who was handling each client.

What I built: a system that gathers all those inquiries into a single panel with zero manual entry, each one already tied to the property it asks about and assigned to its real estate agent. It reads the mailbox every 15 minutes, so a new inquiry shows up in the panel without anyone copying it. It connects to Tokko Broker, the software where the agency manages its listings and agents. It answers WhatsApp through Meta's official API, and the follow-up stops when the client replies, so the conversation hands over to a person.

Result:

- 10 users and more than 3,500 inquiries processed with zero manual entry
- 98% of those that arrived through listing portals last month were linked to their property automatically
- For the first time the owner sees the business in numbers: inquiries by channel, the status of each one and who is handling it
- 207 valuations and 186 viewings booked from the panel
- 7 automated flows, such as the WhatsApp follow-up

Stack: Next.js 14 (App Router) · React · TypeScript · Supabase (PostgreSQL, Auth, RLS, Realtime, Edge Functions in Deno) · Tokko Broker API · WhatsApp Cloud API · Gmail API · Google Calendar API (OAuth) · Vitest · Playwright · Vercel

Quality: 100 unit tests and 33 end-to-end tests · every inquiry keeps its history: who handled it and what happened at each step

Tasador — Property valuations with AI and a price that can be explained

Lexinton Propiedades, Buenos Aires · Full application (backend and frontend) tested with real properties · code published · github.com/ricardobing/tasador-rag

The problem: to decide the asking price of a property, the real estate agent compares it by hand against similar listings: an hour or more per valuation. Two valuations of the same property end up at different prices, with no written record of why.

What I built: a system that produces the valuation report in about a minute, with the comparable listings that justify the price. The AI finds the similar listings and decides which ones are worth comparing, but the price is computed by a formula, never by the model. Every figure the AI takes from a listing is checked against the original text, so no invented number reaches the report.

Result:

- The report comes out in about a minute, against an hour or more of manual searching, with the same criteria for every valuation
- Every price is backed by its comparables, and when there are not enough the system says it cannot value the property
- It costs between 2 and 5 US cents per report

Stack: Python 3.12 · FastAPI · LangGraph · RAG · PostgreSQL 16 + pgvector · Redis · LiteLLM · Next.js 15 (React 19, TypeScript) · Docker Compose · pytest · Playwright

Quality: 492 backend tests and 55 end-to-end tests · 113 test cases to choose the comparable-listings search

TomaNota — Assistant that answers shops' WhatsApp and takes their orders

My product · tomanota.lat · open demo at app.tomanota.lat/probar

The problem: the neighborhood pizzeria takes orders on WhatsApp while the owner is busy behind the counter. At peak hours messages go unanswered, and orders get written on paper with mistakes.

What I built: an assistant that answers that WhatsApp on its own: it talks to the customer, puts the order together with the real prices from the menu and sends it to the shop's order screen. If the customer asks for a person, or the owner jumps into the chat,

the assistant stops replying. A language model holds the conversation and uses system functions to check the menu, validate the address and close the order. If the model fails, a rules engine takes over and the shop does not lose the order.

Result:

- The shop takes WhatsApp orders at peak hours without leaving the counter
- Every delivery address is assigned to its zone automatically, so deliveries go out without anyone checking a map
- The shop updates its menu, delivery zones and QR code from its own panel

Stack: Node.js 22 · TypeScript · Fastify · Vercel AI SDK over an OpenAI-compatible API · tool calling · Supabase (PostgreSQL, RLS, Realtime) · Redis · WhatsApp Cloud API · React 18 · Vite · Vitest

Quality: 847 automated tests · 1,203 simulated conversations against the real model

Zizu — Multi-shop delivery platform with online payments

Client in Argentina · Live in production · delivered in 4 milestones agreed with the client · zizu.com.ar

The problem: in small towns, orders are taken by phone and WhatsApp. Nobody coordinates the customer, the shop and the courier in one place.

What I built: an app where the customer orders, the shop prepares, the courier delivers and the admin oversees, all of them watching the same order update live. Every payment is confirmed against Mercado Pago, and a repeated payment notice never credits twice. The payments ledger cannot be edited even with direct database access, so the total collected always matches what happened.

Result:

- Shops went from taking orders by phone and WhatsApp to receiving and collecting them online on their own platform
- Customer, shop, courier and admin coordinate the same order live, from an app that installs on the phone

Stack: React 18 · Vite · TypeScript · Supabase (PostgreSQL, RLS, Realtime) · Mercado Pago · Playwright · Vercel

Quality: 152 end-to-end tests and 57 unit tests

STACK

Frontend: React · Next.js · Vite · TypeScript · Tailwind CSS · TanStack Query · PWA

Backend: Node.js · TypeScript · Fastify · Python · FastAPI · Edge Functions (Deno) · REST APIs · webhooks (HMAC signature, idempotency) · OAuth2

Databases: PostgreSQL · Supabase · Row Level Security · triggers · auditing · Redis · pgvector

AI — tools: Claude (Anthropic) · LangGraph · RAG · tool calling · Vercel AI SDK · embeddings

AI — judgment: every figure verified against its source · fallback to fixed rules · handover to a person · evaluation with test cases

AI-assisted development: I direct AI coding agents (Claude Code); I define the architecture and the test cases, and the automated tests decide whether a change gets in.

Integrations: Tokko Broker API · WhatsApp Cloud API · Gmail API · Google Calendar API · Mercado Pago

Quality and infrastructure: Vitest · pytest · Playwright (end-to-end tests) · Docker · GitHub Actions (CI/CD) · Vercel

PROFESSIONAL EXPERIENCE

Independent Consultant — AI, Full Stack and Automation

2022 — Present · Clients in Argentina and Spain

- I work directly with owners and their teams: I map the process with the people who run it, agree the scope in stages and own the project end to end until production.
- For a real estate agency in Buenos Aires, the first stage was automating valuation inquiries with Make, Apps Script and a language model. Then I built their custom system.
- For a client in Spain, a database of amateur football leagues that pulls the data from the web every week and loads it automatically (Python, PostgreSQL).

Freelance Systems Engineer

2017 — 2022 · Automations with Python and Apps Script, API integrations, scraping and internal systems for small businesses.

EDUCATION

Information Systems Engineer — UTN, Facultad Regional Concepción del Uruguay

LANGUAGES

- **Spanish:** Native
- **English:** advanced technical reading and professional written communication (docs, email, Slack)