David Chicaiza

Software Engineer · AI Agents & Agentic Systems · 8+ years

Prague, Czech Republic (CET, remote) · dscm0725@gmail.com · linkedin.com/in/david-chicaiza-12a737a1 · +420 722 983 945

SUMMARY

I build agent systems that keep working when nobody is watching. My own runs today: three long-lived agents on heartbeats and a pool of up to 10 Claude Code workers that take tickets from a board, ship to main and stop to ask only when a decision is mine. At Make I built parts of the AI Agents product: sub-agents, tool output filtering and the agent chat. Eight-plus years across the stack, mostly TypeScript and React, with a year managing a team of eight.

EXPERIENCE

Long-running agent fleet · independent project

May 2026 – present

Private repo, one-page case study on request

- Run agents unattended on a Mac and a VPS. A dispatcher keeps up to 10 Claude Code workers busy on tickets from a board, each in its own git worktree, and three long-lived agents (product, personal assistant, home) wake on heartbeats. Every worker turn has to end in a question, a done report or a wait, so silence reads as a fault.
- Numbers as of 4 Nov 2026: always-on agents since May. The VPS sync loop has not gone quiet for more than 11 hours in 89 days. The worker pool went live on 1 Nov and shipped 277 tickets in its first 4 days, 209 of them on the fleet's own tooling, and a risk rule answered 61 questions and closed 55 tickets without me.
- Built for failure: dead-man switch, secret scan on every sync commit, a machine brake and per-stage worker health checks. 108 incidents logged in the first 36 hours of tracking, and 71 of 72 lost workers came back on their own. The bigger failures each left a fix: a supervisor loop that ran the VPS out of memory, a worker stampede that took the Mac to load 828, a dead-man switch that raised 28 false alarms in 12 hours.
- Cut it back when parts stopped earning their place: an October audit took nine chat agents down to one and removed 969 files and 23 cron jobs. The agents that came back run as terminals with a board instead of a chat server.
- TypeScript, Claude Code, MCP servers, Convex, Herdr, GitHub, Tailscale.

Software Engineer, AI Agents team · Make.com

Nov 2025 – Dec 2026

Prague

- Took sub-agents (agents that call other agents as tools) from my own prototype to a shipped feature in six months with full ownership, front end to back end. Customers can now build writer/reviewer and other orchestration patterns themselves.
- Built tool output filtering: people choose which fields of a tool's output reach the agent, so the context window holds data and not metadata. Cleaner context, lower token cost.
- Built the agent chat people use to try and debug an agent: tool calls, reasoning steps, errors and context usage, without running it headless.
- Day to day: TypeScript, React, complex stateful UI, context engineering, prompt and tool design with function calling and structured outputs, feature-flagged rollouts, CI/CD, telemetry (Datadog, Snowflake).

Engineering Manager, Data Transformations · Ataccama

Jan 2024 – 2025

Prague · data management platform

- Led a team of eight through a major on-prem to SaaS transition: introduced team rituals, workshops, and incident simulations. Team eNPS went from -20 to +60.
- Owned roadmap planning and cross-team dependencies; wrote role guidelines and career growth tracks that other teams picked up.
- Stayed hands-on in the front end the whole time, and went back to full-time engineering in 2025 on purpose: I wanted to specialize in AI.

Frontend Engineer, Product Foundation · Ataccama Feb 2021 – Feb 2024
Prague

- Core contributor to Ataccama ONE Gen 2: built the foundational frameworks and tools other teams built features on, including the audit page (logs every user event for compliance), a no-code landing pages system, and the Upgrade Center.
- Led projects and mentored junior engineers; worked closely with product and design.
- Technologies: TypeScript, React, MobX, GraphQL/Apollo.

Frontend Developer · Kiwi.com Nov 2018 – Feb 2021
Prague

- Relocated to Prague to join Kiwi.com; built a CRM app that gave support agents visibility into customer bookings, plus Kiwi.com's surveys web app.
- Technologies: TypeScript/Flow, React, Redux, GraphQL/Relay Modern, Next.js, GitLab CI.

Earlier roles 2015 – 2020
Colombia / US, remote

- **Dexon Software** (2017–2020): full-stack development of internal business applications in modern JavaScript.
- **InfraSite Solutions** (2018): hardware and asset monitoring applications for a Florida infrastructure firm (React/Redux).
- **LinkedAI, Car-B** (2018–2019): frontend development for an AI data labeling platform and a mobility app (React/Redux).
- **Universidad de Los Andes** (2015, 2017): Undergraduate Teaching Assistant, IT Product Design & Innovation and Business Intelligence.

SKILLS

Languages & frameworks: TypeScript (expert), JavaScript, React, Node.js, Next.js, GraphQL, SQL

Agentic systems: long-running unattended agents (heartbeats, worker pools, health checks, dead-man switches, risk-gated approvals), multi-agent orchestration (orchestrator plus specialized sub-agents), plan-and-execute loops with recovery, tool and command layer design, MCP servers, prompt engineering with function calling and structured outputs, evaluating and iterating on agent behavior from telemetry, context engineering

Platform & ops: CI/CD, feature flags, observability (Datadog, Snowflake), Docker, Linux server administration, GitOps

Ways of working: engineering management background: roadmap planning, mentoring, incident and on-call process design, product collaboration

SPEAKING & COMMUNITY

Master of Ceremonies · Node Congress (GitNation) · 2024 & 2025 · Hosted and moderated the online editions of the conference.
Music · I write and release my own songs (a couple are on Spotify).

EDUCATION

Universidad de Los Andes, Bogotá, Colombia · B.Sc. Systems and Computing Engineering (Ingeniería de Sistemas y Computación) · 2012 – 2017

LANGUAGES

Spanish (native) · English (fluent) · Czech (A2)