Francisco Marcano

QA Solutions Architect · Staff Engineer

Architecture and Tooling · AI agents for engineering teams

Buenos Aires, Argentina · fjmarcanos@gmail.com
LinkedIn (linkedin.com/in/franciscojmarcano) · franmarcano.com

Summary

I change how engineering teams do quality. I start from what slows the teams down and build what removes it: self-service internal platforms, AI agents and the tooling around them, automation frameworks and quality gates in the delivery pipeline. I run internal tools as products: discovery from real usage data, early-adopter pilots, building in public, and the buy-in of engineering leads to change the process.

Nearly ten years in software quality, since 2021 in fintech.

Experience

Stori · Mexican fintech · Oct 2022 – Present

QA Solutions Architect · Jan 2026 – Present

Architecture & Tooling team: quality strategy, testing infrastructure and observability across engineering.

- **Test automation any engineer can run.** Built an AI agent that writes end-to-end tests: an engineer describes a flow in plain language, and the agent drives a real phone in the device cloud (Appium over MCP) or a browser (Playwright) and writes the test once, as Gherkin step definitions, opened as a pull request in the team's repository. Every run after that is plain code in CI, with no model. On mobile, a new test takes about 3–4 minutes instead of about 85, and the login regression runs in 18.5 s versus about 5 minutes by hand.
- **Agent and harness engineering.** An internal AI agent that works as an engineering teammate with the company's context: SRE support, triage, incident response and automation. Built on AWS Bedrock AgentCore with the Strands Agents SDK (ReAct, RAG over pgvector), with its tools served over MCP and RBAC on what each user can reach; the harness is optimized against evals. Triage reads OpenTelemetry traces in OpenSearch, CloudWatch logs and crash reports, plus known error patterns drawn from real support threads and read-only infrastructure tools. It is also the AI gateway on Bedrock for operations workflows (n8n, with a human approval step).
- **Custom MCP servers behind single sign-on.** MCP servers authenticated through the company's identity provider (OIDC SSO) that give agents deterministic tools: realistic test data (users, cards, deposit accounts) from one sentence, and automated UI and backend flows. Packaged with agent skills as a QA plugin for Cursor and Claude Code; it became the most-installed plugin in the internal plugin marketplace.

Test Automation Engineer · Oct 2022 – Dec 2025

- **Automation as a service.** Turned QA from a team that runs tests into a platform teams use on their own: found where releases stalled, won engineering leads' buy-in to change the process, and built the self-service pieces: an internal developer platform for quality (test execution on AWS Device Farm, test data, results and observability, behind SSO; Next.js, Go, AWS ECS and Lambda), custom APIs and serverless services (test users, OTPs, Device Farm orchestration) and cross-team automation SDKs.
- **Test cases as code instead of Xray.** Designed and built the core of an in-house release-quality platform that replaced Xray, a paid license. Test cases live versioned in each repo, and releases run as campaigns per team and platform, with a sign-off gate that blocks releases below the pass-rate bar. It grew from a 4-team pilot into the system of record for every mobile release team.
- **Mobile performance observability.** Lambdas that mine the Device Farm logs of the runs teams already had (Android logcat, iOS syslog) for network calls, UI freezes and garbage-collection events, a decorator library that times key flows, and AI analysis of the results: performance visibility on every run at almost no extra cost.
- **Automation frameworks.** Backend, UI (Playwright, Appium) and performance testing (Grafana k6, load tests for banking transactions), run in CI/CD so teams test and release on their own.

Ripio · crypto platform · Jan 2021 – Nov 2023
QA Automation Technical Leader · May 2021 – Nov 2023
QA Automation Engineer · Jan 2021 – May 2021

- Led the move from manual to automated testing. Each release took about 3 days of manual testing and only QA could run automation; with 90% of critical flows automated, release testing dropped to about 2 hours.
- Built Python automation frameworks (Appium, pytest, BDD, PostgreSQL test data) for fraud engines, security features (login, 2FA, OTP) and blockchain flows, run in the cloud on AWS (Lambda, ECS, Device Farm, Cognito) and Docker and wired into CI/CD so teams tested and released on their own.
- Performance testing with Grafana k6 for trading systems under volatile, high-traffic markets.
- Led a team of 4 QA engineers: OKRs, focus and priorities, and the process across people, product and tooling.

Earlier experience

Securityhnos · QA Engineer, web and backend · Jun 2020 – May 2021

Othila (creative agency) · QA Analyst, web applications · Nov 2018 – Aug 2020

PremCos · QA Analyst, web and desktop · Feb 2017 – Nov 2018

PremCos · e-commerce analyst · Jan 2016 – Feb 2017

Independent projects

FlyWorkers · flyworkers.com · 2026

A platform to hire AI agents for a role, such as support, software engineering or QA, each with its own cloud machine. Designed and built the core: per-worker VMs on Fly.io running the open-source pi agent harness, authentication with Supabase and billing with Stripe. The landing page and public pricing are live.

Skills

AI and agents: AWS Bedrock, Bedrock AgentCore, Strands Agents SDK, pi agent harness, harness optimization with evals, RAG with pgvector, MCP servers behind OIDC SSO and RBAC, agent skills and plugins (Cursor, Claude Code, Codex), n8n

Platform engineering: internal developer platforms, self-service automation (automation as a service), custom APIs, serverless services, release gates and test cases as code, CI/CD

Quality engineering: test architecture, mobile automation (Appium), web automation (Playwright), API testing, performance and load testing (Grafana k6), mobile performance observability

Cloud: AWS (ECS, Lambda, API Gateway, Device Farm, Bedrock, Cognito, DynamoDB, CloudFormation), Docker, observability (OpenTelemetry, OpenSearch, CloudWatch), Fly.io, Supabase, Stripe

Programming languages: Go, TypeScript, Python

Web and data: Next.js, PostgreSQL

Education

Universidad de Palermo · Artificial Intelligence Engineering (in progress) · 2024 – 2029

Languages

Spanish (native) · English (professional working proficiency)