Ricardo Velasco

Los Angeles, CA · velasco.rico@gmail.com · LinkedIn: linkedin.com/in/ricardo-velasco · GitHub: github.com/generalChaos

---

Summary

Senior Full-Stack Engineer with 10+ years across IC and leadership, including 2 years as Engineering Manager while remaining hands-on. Deep in Next.js/Node/NestJS/TypeScript/AWS, event-driven patterns, performance, and CI/CD. Proven ability to ship measurable results (reliability, conversion, delivery velocity) and elevate team practices without slowing execution.

Technical Skills

Frontend: React, Next.js, React Native, TypeScript, WebSockets, Accessibility (WCAG), Performance optimization
Backend: Node.js, NestJS, Prisma, REST/GraphQL, event-driven systems, microservices, CI/CD, Docker
Data & Infra: MongoDB, PostgreSQL, AWS (Lambda, S3, EC2), Serverless, caching and CDN, observability
AI & Tooling: LLM-powered workflows, content/ops automation, prompt/agent prototypes

Experience

Senior Software Engineer → Engineering Manager — Diamond Foundry / VRAI | Nov 2020 – Sep 2025

- Re-architected the core e-commerce platform (entire site) to event-driven services and queues, absorbing a ~3x seasonal surge (~$1M/week → $3M+ for 3 consecutive weeks) with 0 P1 incidents during mission-critical windows.
- Scaled team 4→10 engineers (hired 6); established leveling, onboarding, review SLAs, and ownership routing to professionalize delivery.
- Enabled faster iterations post re-architecture—roughly ~2x faster from idea/design to production for medium features; improved deploy cadence from ~1/week to ~3-5/week via preview environments and CI gates.
- Instituted playbooks and alerting; MTTR improved from ~90-120 min to ~30-45 min during non-peak; peak periods stayed calm due to preventative guardrails.
- Standardized PR workflow (owners, 24-hr SLA, small-PR policy, preview environments), doubling reviews per PR (~1.1→~2.2) and cutting median review cycle ~40-60%; reduced “stale >48h” PRs by ~70% during peak Q4.
- Launched “Cut For You” (gem customization) end-to-end (Next.js + Node), collaborating across Design and Product Management and Backend to unlock make-to-order capability and reduce finished-goods pressure (engineering enablement).
- Introduced LLM-assisted documentation and engineering workflows (templates, checklists, and internal helpers), removing manual steps and shortening internal SLAs.

Software Engineer — Verishop | Feb 2022 – Sep 2022

- Rolled out AWS Cognito SSO and UX fixes, raising login success ~90-93% → ~97-99% and cutting median login time ~20-35%, reducing friction in sign-in flows.
- Optimized PDP performance and UX (bundle split, image strategy), reducing LCP ~3.5-4.5s → ~2.2-2.8s and page weight –200-400 KB; corresponded with ~+3-5% Add-to-Cart on PDP.

CTO & Lead Developer — 1Dash1VR | Apr 2019 – Feb 2022

- Prototyped immersive building walkthroughs (Unreal/C++) and designed novice-friendly VR UI/UX for client demos and internal reviews; owned technical direction and delivery cadence.
- Built the authoring pipeline for importing architectural assets, creating scene layouts, and packaging interactive builds for headset demos (Quest 2).

Software Engineer — Bot1 (Stashimi) | Jun 2017 – Apr 2019

- Developed the React Native feeds UI (iOS and Android) rendering aggregated content for ~2k-5k artists across social, RSS, official sites, YouTube, and news APIs; partnered with backend integration teams.

- Scaled the feed experience to 100k+ new items/day with ~1-2 min freshness using virtualized lists, incremental fetch, and batching.
- Engineered the initial Facebook Messenger posting pipeline and admin tooling, enabling new-artist onboarding → first scheduled post within ~1 hour.
- Set a posting SLA (~99% within ~60s) with alerting and operational playbooks; platform handled ~500-1,500 posts/day at peak across 200+ artists (≈50 mid-to-large) while maintaining schedule accuracy.
- Released the RN app to iOS and Android with ~5k-20k MAU and ~99% crash-free sessions.

Selected Technical Highlights

Performance: systematic profiling, route-level budgets, and bundle splitting; steady TTI/LCP improvements.
Reliability: idempotent workers, dead-letter strategies, circuit breakers; fewer incidents and faster recovery.
DX: local development speedups, lint/test/pre-commit standards, preview environments; safer PRs and easier onboarding.

Education

B.S., Computer Science — Syracuse University.