DAVID ALONSO

Frontend Engineer — React · TypeScript · Product Systems

@ masdavidalonso@gmail.com linkedin.com/in/dalonsodev github.com/dalonsodev Remote (CET/UTC+1) · Almería, Spain

SUMMARY

Some engineers need a product spec, a tech lead, and three standups a week to ship a feature.

I'm not that engineer.

My last project required implementing Uniswap V3 liquidity formulas from the protocol whitepaper, handling BigInt precision for on-chain values that exceed JavaScript's Number.MAX_SAFE_INTEGER, and designing an 8-stage fail-fast computation pipeline — no tutorial, no SDK wrapper, first principles.

The project before that was a cocktail recommendation engine built on 11 years of hospitality management: I'd watched customers freeze in front of 30-item menus a thousand times before I wrote the first line of code.

Available for remote contractor roles. Full professional English (C1/C2). CET timezone.

No micromanagement required.

EXPERIENCE

2023 - Present
Remote

Frontend Systems Engineer

Freelance

- Render-as-you-fetch with React Router v8 loaders — data available on first paint, no useEffect waterfalls, no skeleton states on primary routes
- Replaced 40 sequential REST calls with a single batched GraphQL query; added IntersectionObserver + session useRef cache so data is fetched once per viewport entry, never re-fetched
- URL as single source of truth for all filter, sort, and pagination state — shareable, bookmarkable, back-button compatible; defaults omitted from output URL

2020 - 2022
Remote

DeFi Protocol Researcher & Quantitative Practitioner

Researched and executed yield strategies across DeFi protocols; built custom risk-monitoring workflows for volatile markets. Protocol knowledge preceded DeFiScout — domain expertise made the whitepaper math achievable

2018 - 2020
Remote

Web Publisher & Programmatic SEO Architect

- Built content properties generating €1,400+/month through programmatic SEO and automated monetization pipelines

2018 - 2019
Spain / UK

Cabin Crew

Workforce International / Ryanair

- Commercial aviation regulatory certification completed in 45 days; all audits passed on first attempt

2007 - 2018
Spain

Operations Lead & Senior Management

Grupo Dani García · Grupo La Máquina · Platea Madrid · Babicka Vodka

- Directed 30+ person teams in Michelin-level environments owning P&L, inventory, and peak-service execution — and then built the software to solve the problem I watched play out on the floor every night

SKILLS

TypeScript · React · Vite · React Router · TanStack Table · GraphQL · Firebase · react-i18next · Recharts · Tailwind CSS · CSS Scroll Snap · Vitest · ESLint · Prettier · Netlify · Vercel · Astro

LANGUAGES

English — Bilingual C1/C2

Spanish — Native

ADDITIONAL INFO

Software Engineering

2023 - Present

Full-time self-directed, 2023–Present. TypeScript/JS internals, React reconciliation, clean architecture, algorithms. Applied to production — no toy projects

Aviation Cert

Workforce International / Ryanair — completed in 45 days

Execution

Completed 75Hard. Lost 50kg+. I set hard standards and don't negotiate with resistance — same approach to a deadline

PROJECTS

DeFiScout — Uniswap V3 Pool Analytics & LP Strategy Simulator

https://github.com/dalonsodev/defi-scout

DeFi liquidity providers have no accessible tool to simulate concentrated positions before committing capital. Existing tools wrap SDK calls without revealing the math.

- Protocol-level LP simulator built from the Uniswap V3 whitepaper — concentrated liquidity formulas ($L = \min(L_0, L_1)$ in √price space), tick-to-price conversion ($P(i) = 1.0001^i$), and BigInt parsing for on-chain liquidity ($\sim 10^{24}$) that silently loses precision with parseFloat.
- 8-stage fail-fast computation pipeline (simulateRangePerformance): input validation, data quality assessment, metadata validation, dual-path price inference (ETH oracle with TVL fallback), composition, liquidity, fee accumulation, APR aggregation. Each stage returns a typed discriminated union; the pipeline short-circuits on the first failure with a specific user-facing error.
- Statistical data quality engine classifies 30 days of hourly snapshots as INSUFFICIENT / LIMITED / RELIABLE / EXCELLENT based on DeFi-relevant time windows (7-day minimum for weekly volume cycles). Dynamically downgrades quality rating if anomaly rate exceeds 20%.
- HODL vs LP strategy projection — impermanent loss formula ($IL = (2\sqrt{ratio} / (1 + ratio)) - 1$), projected fee accumulation over a user-defined horizon, break-even analysis in days, volatility-adjusted range buffer from 30-day historical price spread.
- 1,000 pools loaded once and filtered in-memory — TheGraph doesn't support multi-column server-side filtering; filter latency is a useMemo call, not a round trip.
- Manual sorting (manualSorting: true) applies to the full dataset before pagination, not just the 40 visible rows.
- Batch sparkline fetching via IntersectionObserver — a single GraphQL query replaces 40 sequential REST calls; session useRef cache prevents re-fetching on scroll-back; differential updates skip already-cached IDs on every trigger.
- URL as a single source of truth for all filter, sort, and pagination state — shareable, bookmarkable, back-button compatible; clean URL strategy omits defaults from output.
- Pure function test suite with zero React dependencies — grouped by Edge Cases, Defaults, Happy Path, Round-Trip consistency, and Performance (sub-1ms assertions on URL parsing). Vitest + @vitest/coverage-v8.

SipMatch — Cocktail Recommendation Engine & Interactive Bar Menu

https://github.com/dalonsodev/DrinkWise

After 11 years managing bars and restaurants in Michelin-level environments, I'd watched the same pattern repeat thousands of times: a customer faces 30+ options, enters decision paralysis, and orders "the usual" — capping the ticket. I didn't need a product brief to define the problem. I lived it. SipMatch is a preference-based recommendation engine — 3 questions, 1–5 curated results — that functions like a knowledgeable bartender encoded in software.

- Discriminated union type Cocktail = AlcoholicCocktail | NonAlcoholicCocktail with hasAlcohol: true | false as the literal discriminant — TypeScript narrows automatically in every filter function; no instanceof checks, no runtime type guards.
- Locale-agnostic recommendation engine — useAnswerMapping converts Spanish/English UI strings to stable internal English keys before any filtering. The engine always operates on deterministic keys regardless of active locale; changing a translation string cannot break a filter.
- Dynamic Q3 options derived from Q1+Q2 results via useQ3Options — the user is never shown a spirit that leads to zero matches; options are a pure function of current filter state, not a hardcoded list.
- Q3 skip logic — if Q1+Q2 already narrow results to one cocktail, Q3 is skipped entirely; a useRef guard (isSkippingToResults) prevents the auto-advance effect from re-triggering during the skip sequence.
- Hook decomposition — useQuizLogic was originally ~200 lines of interleaved state, filtering, and auto-advance concerns; refactored into 6 single-responsibility hooks with the orchestrator at ~60 lines. Each hook independently testable.
- CSS Scroll Snap carousel — scroll-snap-type: x mandatory, 80vw cards, 10% of the adjacent card always visible as a swipe invitation.
- Zero JavaScript scroll logic; browser-native, performant on low-end mobile hardware.
- WCAG 2.1 AA — keyboard navigation on all controls, full ARIA labelling (aria-label, aria-pressed, role="group"), skip-to-main link, contrast ratio > 4.5:1 across all text.