

Zigor López

Software Engineer

Bilbao, Spain

SUMMARY

Full-stack software engineer with a backend centre of gravity. Nine years building products for regulated industries — banking, insurance, e-mobility and digital identity — first in Java and Spring, now in Node and NestJS, along with the front ends that sit on them.

EXPERIENCE

Software Engineer

DEHN digital solutions GmbH | Remote | Sep 2022 — Present

Migrating a legacy platform to a microservices architecture, across both the services and the front ends that consume them.

- Replaced a legacy Spring and Angular platform with NestJS microservices, now released and running in production.
- Owned the full vertical on the migrated pieces — the NestJS services and the Vue front ends built on them — rather than one side of an interface.
- The rewrite removed a substantial amount of accumulated technical debt and improved the platform's performance.

Software Engineer

Wise Security Global | Remote | Dec 2021 — Aug 2022

Building a distributed, blockchain-based identity provider.

- Java services for a distributed identity provider, integrating with an application on the Solana blockchain.

Software Engineer

everis | Bilbao, Spain | Apr 2021 — Nov 2021

Open banking product development and deployment.

Software Engineer

Entelgy | Bilbao, Spain | Nov 2019 — Mar 2021

Backend development in Spring under a microservices architecture, for the e-mobility industry.

- Spring services for the platform that controls electric-vehicle charging infrastructure.

Full-stack Developer

DOMINION GLOBAL | Bilbao, Spain | May 2017 — Oct 2019

Full-stack development of Alert, a RegTech product for the banking and insurance industry.

- Built KYC, regulatory reporting and risk scoring for a compliance platform used by banks and insurers.
- Worked across the whole stack — a Spring Boot monolith and the React front end on top of it.

SKILLS

Languages: TypeScript, JavaScript, Rust, Java, SQL

Product front-end: React, Next.js, Design systems, Accessibility, Vue, Angular

Backend & data: NestJS, Node.js, Spring Boot, PostgreSQL, Kafka

Platform & practice: Docker, Terraform, AWS, GitHub Actions, OpenTelemetry, Supply-chain security

PROJECTS

cv | Product | Production

Solo — product

This site: a CV that renders itself twice, once for a person and once for a parser.

Stack: Next.js, Tolgee, OpenBao, Kafka

- The PDF is generated at build time rather than in the browser. Client-side html-to-PDF libraries rasterise the page, destroying the selectable text the entire ATS layout exists to produce.
- Nothing it publishes carries a way to reach a person directly. The contact form reaches the same inbox through a secret held in OpenBao, so no address appears in the page, the PDF or the structured data.

gpool | Product | Production

Solo — platform, API & web

A real-time football pool platform: private pools, live scoring, and an auditable trail of every point.

Stack: Next.js, NestJS, Postgres, Kafka

- Secrets are fetched at boot by a wrapper process, never baked into an image or committed to an env file, so a leaked app token exposes one kv path rather than the whole store.
- Email is an event, not a call. The API publishes to a topic and a shared service owns delivery, idempotency and dead-lettering.
- CI regenerates the web API contract from the running API and fails on drift, so client and server cannot disagree silently.

gpool.zigordev.com

trading-bot | Product | In development

Solo — Rust & TypeScript

A trading platform split between a Rust hot path and a TypeScript control plane with an operator console.

Stack: Rust, TypeScript, Next.js, Postgres

- Backtesting and live execution consume the same strategy crate, so a result that cannot be reproduced offline is a bug rather than a mystery.

kini | Product | Proof of concept

Solo — API & web

The first football pool platform, and the one that taught me what gpool needed to be.

Stack: Next.js, NestJS, Postgres, Tolgee

- Kept as its own repository rather than folded into gpool, so the rewrite could change its mind about the architecture without rewriting history.

platform-ops | Core

Solo — infrastructure

The shared operations stack every other project attaches to instead of rebuilding.

Stack: Terraform, Docker Compose, OpenBao, Redpanda

- Per-application OpenBao policies scoped to exactly one kv path, so an app token is useless against any other app.
- The same compose manifests describe local and production, so a local reproduction is a real reproduction.

notifications | Core

Solo — backend

A Kafka-backed email service the other products delegate delivery to, with idempotency and crash recovery built in.

Stack: NestJS, Kafka, Postgres, Handlebars

- Invalid payloads fail as non-retryable and go to a durable dead-letter record, because retrying a malformed message forever is a worse outcome than losing it loudly.
- A versioned topic name, so a contract change is a new topic rather than a silent break for every producer.

design-system | Core

Solo — design & code

A technology-agnostic token and component library three products share without agreeing on a CSS framework.

Stack: React, CSS custom properties, oklch

- Themes override colour only. Shape, type scale and spacing stay on shared base tokens, so every product renders identical geometry and differs solely in paint.
- Vendored into each consumer rather than published as a package: the products are private and a copy keeps the dependency graph flat.

EDUCATION

B.Sc. Mathematics

UNED | 2026 — Present

B.Sc. Computer Science

University of the Basque Country | 2013 — 2017

LANGUAGES

Spanish: Native

Basque: Native

English: Fluent