

MOISES MOLINA IZAQUITA - DESARROLLADOR DE SOFTWARE



✉ moiamdev@gmail.com

☎ +34 640 17 64 16

🌐 moiam.dev

➤ [GitHub](#)

📍 Alicante, España

Tecnologías

Lenguajes: Java, TypeScript, SQL

Backend: Spring Boot, Hibernate/JPA, diseño de APIs REST, OpenAPI/Swagger

Datos: MySQL/MariaDB, MongoDB, migraciones con Flyway

Frontend: Angular 15/17, RxJS, HTML5, CSS3, Tailwind

Prácticas: Sagas, idempotencia, patrón Gateway, microservicios, code review, Scrum, Git

Formación

Grado en Física, UNED

En curso (a tiempo parcial, junto al trabajo)

Certificación Java OOP
2023

Inglés C1, Central Park School
2023

Idiomas

Español — Nativo

Inglés — Avanzado (C1)

PERFIL

Desarrollador de software con más de 2 años en i2e, la empresa que desarrolla Fixner, un SaaS para empresas de servicios. Trabajo en los dos lados de sus integraciones, el producto y las plataformas de clientes construidas sobre él, manteniendo los datos coherentes entre sistemas que fallan por separado y no comparten transacción.

EXPERIENCIA

Desarrollador de Software — i2e (Ingeniería Informática Empresarial)

Jun 2024 – Actualidad

i2e desarrolla Fixner (órdenes de trabajo, facturación, TPV) y plataformas a medida para los clientes que lo usan.

Ijes Solar

Plataforma cliente para la construcción de plantas solares, que funciona como sistema distribuido con Fixner: bases de datos separadas, sincronizadas solo a través de la API REST pública de Fixner.

- A partir de un boceto general, diseñé un modelo de datos maestros en el que la plataforma, y no Fixner, es dueña de las plantillas de tareas, ampliando la tabla de órdenes de trabajo existente en lugar de duplicarla; las órdenes antiguas pasan al nuevo modelo al usarse por primera vez, mientras conviven el código nuevo y el antiguo (strangler fig).
- Detecté que el backend dependía del frontend para evitar órdenes de trabajo duplicadas; trasladé al servidor la decisión de reutilizar o crear, evitando duplicados en Fixner y registros huérfanos en local.
- Introduce una capa Gateway para la API de Fixner (una clase por recurso, política de reintentos aislada), eliminando llamadas HTTP duplicadas entre servicios, y llevé la lógica de compensación existente a una saga que mantiene los datos coherentes para unos 40 equipos de obra.
- Construí un job de reconciliación cada 6 horas que compara el catálogo de Fixner con los datos locales mediante un índice hash, detectando cambios que Fixner no puede notificar (sin webhooks); las transacciones por registro y los reintentos con backoff lo hacen idempotente y tolerante a fallos.
- Redacté el documento de diseño técnico de cada fase y desarrollé la interfaz operativa en Angular 17 con RxJS.

Fixner

Único desarrollador del TPV (retail, unos 40 clientes; hostelería)

- Reconstruí el TPV de extremo a extremo a partir de un prototipo generado con IA: cliente Angular tipado y una capa REST sin llamadas entre controladores.
- Rediseñé la facturación de hostelería: antes el carrito solo se guardaba a mano y se perdía al recargar; ahora se crea un borrador de factura en Fixner desde el primer producto, así los pedidos abiertos persisten y se pueden consultar desde distintas mesas.

Inversis (banca)

Proyecto de consultoría a través de i2e

- Desarrollé y documenté APIs REST (Java, Spring Boot, OpenAPI) durante la migración de una aplicación bancaria heredada a microservicios.
- Incorporé a dos compañeros nuevos en un contexto de alta rotación, explicándoles las herramientas, los repositorios y las decisiones previas del equipo.