

Jorge Leal

Estudiante de Ingeniería de Software & Entusiasta del Análisis de Datos

Teléfono: 773-178-3694

Email: jlealcornejo@gmail.com

Portafolio: jorgeleal.site

Perfil

Desarrollador Fullstack especializado en plataformas web, sistemas backend de alto rendimiento y dashboards orientados a datos, con experiencia práctica construyendo productos SaaS multi-tenant, arquitecturas híbridas (SSG/SSR), CMS headless y sistemas optimizados para rendimiento.

Fuerte enfoque en integrar frontends interactivos con backends escalables, despliegues automatizados en la nube y aplicación de modelos de IA y estadística para analítica, predicción y mejora de la experiencia de usuario.

Idiomas

Español: Nativo

Inglés: Intermedio

Educación

Universidad Autónoma Metropolitana, Unidad Xochimilco

Licenciatura en Comunicación Social

Graduado: Octubre 2015

Universidad Abierta y a Distancia de México

Ingeniería en Software

Inicio: Enero 2025 | Actualmente cursando

Proyectos

PixMinder | Rust (Axum), PostgreSQL, Redis

Servicio inteligente de optimización de imágenes diseñado como una alternativa CDN de alto rendimiento para reemplazar las transformaciones nativas de Cloudinary.

Backend desarrollado en Rust con Axum siguiendo Clean / Hexagonal Architecture y estricta separación de capas.

Optimización de imágenes sensible al contexto con selección automática de formato (AVIF, WebP, JPEG, PNG), calidad adaptativa, redimensionamiento consciente de DPR y modo pass-through para SVG.

Sistema de caché multinivel (memoria, PostgreSQL, Redis) que logra latencias de caché por debajo de 10 ms y compresión entre 15–30% superior a la nativa de Cloudinary.

Actualmente sirviendo tráfico en producción para Qubly CRM y LaZalza, con mejoras de +7–13 puntos en Lighthouse Performance y reducción de LCP de 300–600 ms.

Stack: Rust, Axum, Tokio, PostgreSQL (Neon), Redis (Upstash), SQLx, Moka, MozJPEG, WebP, AVIF (ravif), Cloudinary, Chart.js, Docker, Fly.io

Qubly CRM | SvelteKit, Go (Fiber), PostgreSQL (Neon)

Plataforma SaaS multi-tenant para pequeños negocios en México que permite crear una tienda online en minutos con pedidos directos a WhatsApp y sin comisiones por venta.

Arquitectura híbrida con frontend en SvelteKit (SSG, SSR, PWA) y backend en Go siguiendo principios de Clean Architecture.

Sistema completo de e-commerce con suscripciones recurrentes, Stripe Connect, webhooks y sistema de referidos.

Chatbot con IA usando Groq (Llama), optimizado en costos mediante detección de intención, construcción eficiente de contexto y cacheo de respuestas.

Stack: TypeScript, SvelteKit, Go, Fiber, PostgreSQL (Neon), Stripe, Groq AI, Tailwind CSS, Docker, Fly.io

Kidyboard | SvelteKit, FastAPI, PostgreSQL (Neon)

Sistema de gestión multi-rol para centros de entretenimiento infantil con dashboard administrativo, ventas, inventarios, servicios y reportes en tiempo real.

Frontend en SvelteKit con renderizado híbrido y arquitectura monorepo; backend en FastAPI siguiendo Clean Architecture con PostgreSQL serverless.

Dashboard operativo con métricas en tiempo real (ventas, ticket promedio, inventario, timers activos), reportes comparativos y predicción estadística para planeación de ventas y capacidad.

Funcionalidades avanzadas como timers en tiempo real vía WebSockets, exportación de reportes a Excel/PDF, generación de menús digitales (PDF y video) y control de acceso granular por roles.

Stack: TypeScript, SvelteKit, FastAPI, PostgreSQL (Neon), Chart.js, Tailwind CSS, Docker, Fly.io

Databoard | FastAPI, NiceGUI, Plotly, PostgreSQL (Neon)

Dashboard analítico multi-tenant para monitoreo en tiempo real de métricas web en múltiples sitios y proyectos.

Arquitectura asíncrona con aislamiento de datos por tenant, cacheo optimizado y manejo resiliente de errores.

Visualizaciones avanzadas con Plotly: series de tiempo, heatmaps horarios, mapas de geolocalización y KPIs estilo Spotify.

Incluye selección multi-sitio, segmentación de audiencia, ajuste de zona horaria, exportación CSV y predicción estadística de métricas (EWMA, análisis de tendencia).

Stack: Python, FastAPI, NiceGUI, Plotly, PostgreSQL (Neon), Tailwind CSS, Docker, Fly.io

LaZalza | Next.js, KeystoneJS, PostgreSQL

Sitio de noticias y contenido editorial sobre música, cine y cultura popular.

Frontend con Next.js 14 App Router e ISR (5 minutos).

CMS headless con KeystoneJS, PostgreSQL.

Integración con Cloudinary y logotipo interactivo en 3D.

Stack: TypeScript, React, Tailwind CSS, GraphQL, Docker, Cloudinary

Raccoonin Records | Next.js, KeystoneJS, PostgreSQL

Plataforma para un sello discográfico independiente con arquitectura híbrida.

Migración de CSR a ISR y sistema de posts con paginación infinita.

Embeds multimedia, metalinks de streaming y sitemap dinámico.

Stack: TypeScript, Next.js 14, React, Tailwind CSS, GraphQL, KeystoneJS, Cloudinary, Three.js, Docker

Beat Catalogue | SvelteKit, FastAPI, PostgreSQL

Catálogo musical con SSR híbrido y modal estilo Spotify.

Backend con FastAPI + PostgreSQL y panel administrativo custom.

Reproductor de audio con shuffle, contador de reproducciones y tema persistente.

Stack: TypeScript, SvelteKit, Tailwind CSS, FastAPI, SQLAlchemy, Cloudinary, Three.js

BSTCGenerator | FastAPI, MusicGen, Gemini

Plataforma de generación de audio para productores con arquitectura costo-eficiente.

Backend desacoplado en FastAPI y worker de IA en Google Colab (GPU) usando MusicGen.

Sistema de embeddings y recomendaciones basado en feedback y análisis de patrones.

Stack: Python, FastAPI, SQLite, Librosa, MusicGen, Gemini, Vanilla JS, GitHub Codespaces, Google Colab

Stack Tecnológico

Frontend: Next.js 14, React 18, SvelteKit, Tailwind CSS, NiceGUI, Astro, Three.js

Backend: Rust (Axum), Go (Fiber), KeystoneJS, Node.js, Python (FastAPI)

Bases de datos: PostgreSQL (Neon), SQLite, Redis

DevOps / Deploy: Docker, Fly.io, Cloudflare, GitHub Codespaces

Media / Assets: Cloudinary, CDN propio (PixMinder)

Pagos: Stripe

IA / Datos / Visualización: Plotly, MusicGen, Gemini, Groq AI

Otras herramientas: GraphQL (URQL), Axios