BRYAN ZAVALA

Desarrollador Web Full-Stack

Madrid, España | 611 696 248 | dev.bryanzavala@gmail.com | LinkedIn | GitHub | Portafolio

PERFIL PROFESIONAL

Desarrollador Web Full-Stack con experiencia práctica en React,Angular TypeScript y Spring Boot, incluyendo el diseño de una API propia en Java con autenticación JWT, Docker y despliegue continuo. Durante mis prácticas en Globant participé en la optimización y despliegue de interfaces seguras. Busco oportunidad para aplicar mi base técnica en resolución de problemas reales y seguir consolidando buenas prácticas de código.

HABILIDADES TÉCNICAS

Frontend: React, Next.js, TypeScript, JavaScript (ES6+), Tailwind CSS, Astro, Angular, Framer Motion

Backend: Java 21 (Spring Boot 3, Spring Security / JWT), Node.js (Express), PHP (Laravel)

Bases de datos: PostgreSQL, SQL, Supabase, MongoDB

DevOps y herramientas: Git, GitHub Actions (CI/CD), Docker, Swagger / OpenAPI 3, Render, Vercel, API REST

Testing: Jest / React Testing Library (pruebas unitarias iniciales)

Gestión de estado y rendimiento: Zustand, React Query / TanStack Query, Lighthouse

EXPERIENCIA LABORAL

Desarrollador Web Full-Stack (Prácticas FCT) — Globant Marzo 2026 – Junio 2026

El Reto: La carga inicial de la plataforma presentaba un Largest Contentful Paint (LCP) elevado, afectando la experiencia de usuario y el SEO.

La Solución: Migración del renderizado hacia Server Components de Next.js y centralización del estado del servidor con React Query para cachear peticiones y evitar sobrecargas en la API.

El Resultado: Disminución del tiempo de carga inicial en 2 segundos, mejorando el Total Blocking Time (TBT) y protegiendo el acceso a las vistas de los usuarios mediante NextAuth y JWT (RBAC).

PROYECTOS WEB PRÁCTICOS

Portafolio Backend GitHub | Swagger Demo

Java 21 · Spring Boot 3 · Spring Security (JWT) · PostgreSQL · Docker · GitHub Actions · Render · Swagger/OpenAPI 3

El Reto: Construir una API propia en Java para centralizar el contenido del portafolio con acceso protegido de cara a futuras funcionalidades de administración.

La Solución: API REST desarrollada con Spring Boot siguiendo una arquitectura en capas (controller–service–repository), con autenticación y autorización mediante Spring Security y JWT, y documentación interactiva con Swagger/OpenAPI 3. Contenerización con Docker e integración continua con GitHub Actions para despliegue automático en Render.

El Resultado: API en producción con endpoints documentados y verificables desde Swagger, conexión segura a PostgreSQL (Aiven) y despliegues automatizados sin intervención manual.

Gestor Financiero SaaS GitHub | Demo

Remix · React · TypeScript · Supabase (PostgreSQL) · Docker · GitHub Actions · Vercel

El Reto: Automatizar los despliegues a producción garantizando que el código nuevo no rompiera la aplicación ni incluyera errores de tipado.

La Solución: Integración de flujos CI/CD con GitHub Actions y diseño de imágenes Docker multi-stage para el aislamiento del entorno.

El Resultado: Despliegues seguros en Vercel e imágenes de contenedor ultraligeras, extrayendo datos de Supabase solo cuando es estrictamente necesario.

Next Commerce GitHub | Demo

Next.js · React 19 · Zustand · TanStack Query · Jest / React Testing Library

El Reto: Evitar que cambios futuros rompieran el comportamiento de componentes clave de la tienda.

La Solución: Gestión de estado con Zustand y TanStack Query, e introducción de las primeras pruebas unitarias del proyecto con Jest y React Testing Library sobre componentes puntuales.

El Resultado: Base inicial de testing automatizado sobre la que ampliar cobertura, y detección temprana de regresiones en los componentes cubiertos.

Alexandria — Catálogo Social Multimedia (TFG en equipo) GitHub | Demo

React · Tailwind CSS · Zustand · React Query · consumo de APIs REST

El Reto: Las auditorías de Lighthouse penalizaban la velocidad (alto LCP) por la carga masiva de catálogos multimedia y re-renderizados constantes.

La Solución: Desarrollo de la interfaz con React y Tailwind CSS, refactorizando el flujo de datos con Zustand (cliente) y React Query (servidor) para el consumo de múltiples APIs REST.

El Resultado: Interfaz fluida, reducción drástica del bloqueo del hilo principal y conexión exitosa con el motor de recomendaciones del backend, desarrollado por otro miembro del equipo.

Proyectos de IA & LLMs (Formación DAW)

Python · LangGraph · RAG · OpenAI · Hugging Face · Google Colab

El Reto: Automatizar la extracción de información y construir sistemas conversacionales dinámicos.

La Solución: Orquestación de agentes en Python con LangGraph y sistemas RAG, integrando modelos de OpenAI y Hugging Face.

El Resultado: Flujos desplegados en Google Colab combinando web scraping y visión artificial (OCR), con enrutamiento basado en los prompts del usuario y gestión segura de variables de entorno.

EDUCACIÓN

- Grado Superior en Desarrollo de Aplicaciones Web (DAW) — IES María de Zayas y Sotomayor (Junio 2026)
- Bootcamp Full-Stack Developer (+400h) — Upgrade Hub (2024)
- Grado Medio Técnico Administrativo — La Salle Madrid

IDIOMAS E INFORMACIÓN ADICIONAL

- Idiomas: Español (Nativo), Inglés (B2 — comprensión técnica fluida)
- Movilidad: Disponibilidad inmediata y total movilidad geográfica (vehículo propio)