

Emanuel Rojas Ramirez

📍 Medellín, Colombia | Disponible para oportunidades remotas o presenciales

✉️ emanuelrr1522@gmail.com | ☎ +57 301 316 4185

🔗 [Portafolio](#) | [GitHub](#) | [LinkedIn](#)

PERFIL PROFESIONAL

Desarrollador **Full-Stack** enfocado en **Frontend, motivado y autodidacta**, con experiencia práctica construyendo **aplicaciones web** completas usando **React** en el **frontend** y **Java Spring Boot** en el **backend**. Aunque estoy en búsqueda de mi primer rol profesional, he desarrollado y desplegado proyectos funcionales del mundo real que demuestran mi capacidad para diseñar **interfaces limpias**, implementar **flujos seguros** y conectarlos con **APIs REST robustas**.

Mi habilidad más fuerte es construir **interfaces de usuario responsivas y funcionales** con **React (rutas protegidas, autenticación, validación, actualizaciones en tiempo real)**, aunque también tengo un **conocimiento sólido** de **backend**, incluyendo **arquitectura por capas**, bases de datos relacionales (**MySQL, PostgreSQL**) y **pruebas/documentación** de APIs usando **Postman** y **Swagger**.

Soy **proactivo, organizado y apasionado** por el aprendizaje continuo. Busco unirme a un equipo colaborativo, aportar en proyectos reales y crecer como **desarrollador full-stack**.

HABILIDADES CLAVE

Backend: Backend: Java, Spring Boot, APIs RESTful, Arquitectura por capas, Arquitectura limpia (principiante).

Frontend: Frontend: React, React Router, Rutas protegidas, Validación de entradas, i18next, Diseño responsivo, Bootstrap (básico).

Tecnologías Web: JavaScript (ES6+), nociones básicas de TypeScript, manejo de JSON en frontend.

Pruebas y Documentación de APIs: Swagger (uso básico), Postman

Bases de Datos: H2 (en memoria), MySQL.

Herramientas de desarrollo: Git, GitHub, GitHub Projects, Visual Studio Code, IntelliJ IDEA, Maven, Spring Initializr.

Despliegue: Vercel (frontend), Render (APIs simuladas), entornos locales (localhost)

Prácticas ágiles: Scrum (básico), gestión de proyectos con Git

Aprendizaje y resolución de problemas: depuración por consola, pruebas por ensayo y error, aprendizaje basado en proyectos

PROYECTOS DESTACADOS

Portafolio Personal

React · i18next · Diseño responsivo · Vercel

- Sitio multilenguaje (EN/ES) con modo oscuro / claro
- Galería de diplomas con carrusel en escritorio y gestos táctiles en móvil
- Totalmente responsivo, muestra habilidades, proyectos e información de contacto

App Web para Restaurante

React (Frontend – Vercel), Spring Boot (Backend – GitHub), MySQL

- Manejo de login y registro, usuarios validados y guardados en API simulada (Render)
- Estado del carrito persistido con localStorage; flujo de compra seguro
- Backend REST completo (6 entidades) con endpoints CRUD (/cliente, /carrito, /plato, etc.)

Gestor de Suscripciones

React Hooks, SweetAlert, Render (base de datos simulada)

- App de una sola página para crear, editar, eliminar y listar planes de suscripción
- Filtros en tiempo real mientras se escribe
- Formularios validados con alertas personalizadas, UI limpia para seguimiento rápido

API REST para Escuela

Spring Boot, MySQL, Swagger, Postman

- Arquitectura por capas con entidades como Usuarios, Docentes, Estudiantes, Cursos, Asignaturas, Calificaciones, Asistencia
- Rutas CRUD completas (/student/save, /teacher/update/{id}, etc. – más de 30 endpoints)
- Probado con Postman y documentado con Swagger

Simulador Backend de Máquina Expendedora

Java Spring Boot, Swagger

- Flujo de venta completo:
POST /venta → registra venta (PENDIENTE) y devuelve código QR simulado
GET /venta/{id}/estado → expira automáticamente tras 3 min si no se paga
PUT /venta/{id}/verificar-pago → cambia estado a PAGADO (simulado)
GET /venta/{id}/liberar → simula liberación del producto
- Preparado para integrar pagos reales con Izipay y dispensador Arduino
- Construido con separación de DTOs, mapeo de entidades y diseño de base de datos relacional

Platzi Market API Simulation

Java, Spring Boot, PostgreSQL, Postman, Clean Architecture

- Backend simulation of a digital supermarket system, designed following hexagonal architecture principles.
- Full CRUD operations for Products, with filtering endpoints like:
GET /products/scarse-products/{quantity} → returns products with low stock
GET /products/category/{categoryId} → get products by category
- Built with DTO separation, entity mapping, and relational database design.

IDIOMAS

Español: Nativo

Inglés: C1

EDUCACIÓN Y CERTIFICACIONES

- ◆ Técnico Auxiliar en Desarrollo de Software – CESDE (2025)
- ◆ Curso Profesional de Java Spring – Platzi (2025)
- ◆ Java Orientado a Objetos – Platzi (2024)
- ◆ Introducción a Java SE – Platzi (2024)
- ◆ Curso de React JS – Platzi (2025)
- ◆ Manipulación del DOM con JavaScript – Platzi (2024)
- ◆ Fundamentos de JavaScript – Platzi (2024)
- ◆ Git y GitHub – Platzi (2025)
- ◆ Curso Definitivo de HTML y CSS – Platzi (2024)
- ◆ Responsive Design: mobile-first – Platzi (2024)
- ◆ Fundamentos de Ingeniería de Software – Platzi (2024)