EDUARDO DE SOUZA SANTOS — DESENVOLVEDOR FULL STACK

santos934510@gmail.com | (16) 99772-4259 | GitHub: github.com/Eduardoss45 | LinkedIn: linkedin.com/in/eduardo-souza432

SUMMARY / RESUMO PROFISSIONAL

Desenvolvedor Full Stack desde 2023, com foco em backend utilizando Node.js com TypeScript (principal) e Java (Spring como stack secundária), atuando em projetos freelance na construção de sistemas distribuídos e APIs, com ênfase em arquitetura, desempenho e confiabilidade.

SKILLS / HABILIDADES TÉCNICAS

Backend:

- Node.js (TypeScript), Java (Spring)
- REST, gRPC
- Mensageria (RabbitMQ), processamento assíncrono, workers, testes unitários

Frontend:

- React, Next.js (SSR, SPA)

Infraestrutura e Observabilidade:

- Docker, configuração de serviços locais
- Testes de performance com k6

Arquitetura e Comunicação:

- Microserviços e monólitos modulares
- Comunicação assíncrona, event-driven
- Controle de concorrência, fallback entre serviços
- Circuit breaker, Dead Letter Queues (DLQ)
- Aplicação prática de DDD e Clean Architecture

Dados:

- PostgreSQL, MySQL, MongoDB, Redis

PROFESSIONAL EXPERIENCE / EXPERIÊNCIA PROFISSIONAL

Sistema de gestão e administração de clientes — *Desenvolvedor Full Stack 02/2025 – 08/2025*

- Implementei arquitetura de microserviços com comunicação assíncrona utilizando NestJS e RabbitMQ, reduzindo acoplamento e habilitando escalabilidade horizontal.
- Desenvolvi comunicação em tempo real com WebSockets e processamento assíncrono com filas e workers, melhorando latência e consistência em fluxos de eventos.
- Apliquei mecanismos de resiliência em fluxos síncronos e assíncronos (circuit breaker, DLQ e fallback), além de centralização de erros, aumentando a confiabilidade do sistema.
- Melhoria de ~37% no tempo de resposta após desacoplamento e otimização do fluxo assíncrono (medido em testes de carga).

Migração completa de sistema legado — *Desenvolvedor Full Stack 12/2023 – 08/2024*

- Evoluí backend legado em Django para arquitetura de microserviços com NestJS, reduzindo acoplamento e facilitando manutenção e evolução independente dos serviços.
- Migrei frontend de React (JavaScript) para React com TypeScript, estabelecendo contratos tipados e reduzindo inconsistências na integração com o backend.
- Padronizei UI, gerenciamento de estado e validação (Tailwind, shadcn/ui, Zustand, Zod), melhorando manutenibilidade e consistência da aplicação.

Integração de Gateway de Pagamento em E-commerce — *Desenvolvedor Full Stack 02/2023 – 07/2023*

- Integrei gateway de pagamento com Stripe via Express, implementando fluxos REST para criação e confirmação de pagamentos.
- Implementei tratamento de webhooks para sincronização de estados de pagamento, evitando inconsistências entre sistema e provedor.
- Estruturei validações de fluxo e persistência para garantir integridade e prevenir duplicidades em operações financeiras.

EDUCATION / FORMAÇÃO ACADÊMICA

Formação técnica em Análise e Desenvolvimento de Sistemas, com foco em lógica de programação, bancos de dados, modelagem de sistemas e desenvolvimento full stack.

PROJECTS / PROJETOS PESSOAIS

QuickHost — Sistema de Hospedagem: github.com/Eduardoss45/quickhost

- Desenvolvi sistema de hospedagem estilo Airbnb com arquitetura de microserviços, utilizando API Gateway em NestJS e comunicação assíncrona via RabbitMQ.
- Modelei serviços independentes por domínio (auth, booking, accommodation, chat, notifications), com bancos isolados e comunicação baseada em eventos.
- Implementei autenticação com JWT (access/refresh) e comunicação em tempo real com WebSockets para notificações e chat.
- Containerizei o ambiente com Docker Compose, garantindo execução end-to-end e reprodutibilidade da arquitetura distribuída.

Finances API — Sistema Financeiro: github.com/Eduardoss45/finance-api

- Desenvolvi API para gestão financeira com Spring Boot, com foco em segurança, consistência transacional e arquitetura em camadas.
- Implementei regras críticas de domínio, incluindo controle de concorrência com lock pessimista (SELECT FOR UPDATE), prevenção de saldo negativo e imutabilidade de transações.
- Apliquei mecanismos transversais como auditoria via AOP, rate limiting em endpoints sensíveis e tratamento padronizado de erros (RFC 7807).
- Containerizei a aplicação com Docker Compose e utilizei PostgreSQL e Testcontainers para testes integrados, garantindo maior confiabilidade em cenários reais.

Doc Flow – Motor Assíncrono de Conversão de Documentos: github.com/Eduardoss45/doc-flow

- Desenvolvi backend para processamento assíncrono de documentos com Flask, estruturado como monólito modular orientado a mensageria (Celery, RabbitMQ, Redis).
- Implementei pipeline assíncrono desacoplado entre API e workers, com persistência de estado em PostgreSQL e notificações em tempo real via WebSockets.
- Apliquei isolamento por cliente com UUID e cookies HTTP-only, incluindo segregação de armazenamento e controle de recursos com expiração automática (TTL).
- Containerizei o ambiente com Docker Compose, orquestrando API, workers, broker, cache e banco, garantindo execução previsível e reprodutível.