

EDUARDO DE SOUZA SANTOS — DESENVOLVEDOR BACKEND

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

SUMMARY / RESUMO PROFISSIONAL

Desenvolvedor com experiência desde 2023, com foco em backend utilizando Node.js com TypeScript (principal) e Java (Spring Boot como stack secundária), atuando em projetos freelance no desenvolvimento de APIs e sistemas distribuídos, com ênfase em arquitetura, performance 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 Backend 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 Backend 12/2023 – 08/2024

- Evolui backend legado em Django para arquitetura de microserviços com NestJS, reduzindo acoplamento e permitindo evolução independente dos serviços.
- Estabeleci padronização de contratos entre frontend e backend ao migrar aplicações React para TypeScript, reduzindo inconsistências de integração e melhorando previsibilidade de APIs.
- Apoiei padronização da camada de interface e validação (Tailwind, shadcn/ui, Zustand, Zod), contribuindo para consistência e manutenção da aplicação.

Integração de Gateway de Pagamento em E-commerce — Desenvolvedor Backend 02/2023 – 07/2023

- Integrei gateway de pagamentos (Stripe) utilizando Express, estruturando fluxos REST para criação e confirmação de transações com foco em consistência de estados.
- Implementei processamento de webhooks para sincronização assíncrona de eventos, garantindo atualização consistente entre provedor externo e sistema interno.
- Apliquei mecanismos de validação, idempotência e persistência transacional para evitar duplicidades e garantir integridade em operações financeiras críticas.

EDUCATION / FORMAÇÃO ACADÊMICA

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

PROJECTS / PROJETOS PESSOAIS

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).
- Utilizei Docker Compose, PostgreSQL e Testcontainers para execução e testes de integração do sistema.

Gateway Failover API — Sistema de Pagamentos Multi-Gateway

- github.com/Eduardoss45/gateway-failover-api
- Desenvolvi sistema de pagamentos com arquitetura baseada em múltiplos gateways, implementando fallback automático entre provedores para garantir alta disponibilidade nas transações.
- Modelei domínio de compras com persistência de transações e relacionamento com produtos, aplicando controle de consistência e rastreabilidade por gateway utilizado.
- Implementei autenticação com tokens e controle de acesso por roles (ADMIN, MANAGER, FINANCE, USER), com separação de responsabilidades por camada (controllers, services e middleware).
- Estruturei testes automatizados com banco isolado e migrations executadas em ambiente de teste, garantindo validação de fluxos críticos como compras, reembolsos e fallback.

Order Pipeline — Arquitetura de Processamento de Eventos (Fila vs Stream)

- github.com/Eduardoss45/order-processing-patterns
- Desenvolvi projeto comparativo de arquitetura de processamento de pedidos utilizando dois modelos de mensageria (RabbitMQ e Kafka), aplicados ao mesmo domínio para análise de diferenças entre filas e streams.
- Estruturei dois pipelines independentes com orquestração própria, incluindo mecanismos de retry e dead letter queue (DLQ) em ambos os modelos.
- Criei packages compartilhados em TypeScript com validações reutilizáveis via Zod e logger customizado, utilizados pelos dois sistemas sem dependência de ferramentas de monorepo.
- Implementei o projeto em Node.js puro com TypeScript, garantindo isolamento entre implementações e consistência de contratos entre os fluxos.