EDUARDO DE SOUZA SANTOS — DESENVOLVEDOR FULL STACK

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

SKILLS / HABILIDADES TÉCNICAS

Frontend: React.js, Next.js, Vue.js
Backend: Node.js, NestJS, Express, AdonisJS, Spring Boot, Flask
Databases: PostgreSQL, MySQL, MongoDB, Firebase
ORM: Prisma, TypeORM, Hibernate, Lucid, SQLAlchemy
Messaging, Cache & Async: Redis, RabbitMQ, Celery
Infra & DevOps: Docker, Docker Compose, CI/CD
Testing: Jest (unit), JUnit (unit), Pytest (unit), Cypress (e2e)
Languages: TypeScript, JavaScript, Java, Python

SUMMARY / RESUMO PROFISSIONAL

Desenvolvedor Full Stack desde 2023, com foco em backend utilizando Node.js e TypeScript. Experiência na construção de APIs REST e sistemas com mensageria, processamento assíncrono e comunicação em tempo real, incluindo controle de concorrência e integração entre serviços, com ênfase em arquitetura e desempenho.

PROFESSIONAL EXPERIENCE / EXPERIÊNCIA PROFISSIONAL

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

- Redesenhei a arquitetura monolítica originalmente em Express para microserviços com NestJS e RabbitMQ, separando domínios em serviços independentes para reduzir acoplamento, permitir deploy isolado e viabilizar escalabilidade horizontal.
- Implementei comunicação em tempo real com WebSockets, substituindo polling síncrono, para reduzir latência na propagação de eventos e garantir consistência imediata em fluxos de chat e notificações.
- Modelei um sistema de autenticação e autorização baseada em perfis, centralizando regras de acesso para mitigar exposição indevida de recursos e simplificar a governança de segurança.

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

- Redesenhei backend legado em Django para microserviços com NestJS, reduzindo dívida técnica e permitindo manutenção e evolução independentes por serviço.
- Modernizei frontend React 17 (JavaScript) para React 19 com TypeScript, aumentando previsibilidade e segurança em refatorações.
- Padronizei UI, gerenciamento de estado e validação utilizando Tailwind CSS e shadcn/ui para UI, Zustand para estado e Zod + React Hook Form para validação, eliminando inconsistências e melhorando DX e manutenibilidade.

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

- Integrei gateway de pagamento com Stripe via Express, orquestrando fluxos REST para criação e confirmação de pagamentos online.
- Implementei tratamento de webhooks da Stripe, sincronizando estados de pagamento e evitando inconsistências entre sistema e provedor.
- Estruturei validações de fluxo e persistência para garantir integridade transacional e mitigar falhas e duplicidades financeiras.

EDUCATION / FORMAÇÃO ACADÊMICA

Formação técnica em Análise e Desenvolvimento de Sistemas (3 anos), com foco em lógica de programação, bancos de dados, modelagem de sistemas e desenvolvimento full-stack.

PROJECTS / PROJETOS

QuickHost — Sistema de Hospedagem com NestJS: *GitHub*

- Desenvolvimento de sistema de hospedagem estilo Airbnb baseado em arquitetura de microserviços, com API Gateway em NestJS, serviços desacoplados por domínio e comunicação assíncrona via RabbitMQ.
- Implementação de autenticação com JWT (access + refresh) e controle de acesso via guards, além de notificações e chat em tempo real via WebSocket, com persistência independente por serviço.
- Orquestração de múltiplos serviços com responsabilidades isoladas (auth, booking, accommodation, chat, notifications), utilizando bancos separados por domínio e comunicação baseada em eventos com UUIDs.
- Ambiente totalmente containerizado com Docker Compose, garantindo execução end-to-end do sistema, isolamento de dependências e reprodutibilidade da arquitetura distribuída.

Finances API — Sistema Financeiro com Spring Boot: GitHub

- Desenvolvimento de API REST para gestão financeira com Spring Boot, focada em segurança, consistência transacional e arquitetura em camadas, com autenticação baseada em JWT (access + refresh) e controle de acesso por roles.
- Implementação de regras críticas de domínio financeiro, incluindo controle de concorrência com lock pessimista (SELECT FOR UPDATE), prevenção de saldo negativo e garantia de imutabilidade de transações.
- Camadas transversais aplicadas com auditoria via AOP, rate limiting em endpoints sensíveis e tratamento padronizado de erros seguindo RFC 7807 (Problem Details).
- Ambiente containerizado com Docker Compose, integração com PostgreSQL e estratégia de testes com Testcontainers, garantindo maior confiabilidade e isolamento em cenários reais.

Doc Flow – Motor Assíncrono de Conversão de Documentos com Flask: GitHub

- Desenvolvimento de backend para processamento e conversão assíncrona de documentos, estruturado como monólito modular orientado a mensageria, com Flask, Celery, RabbitMQ e Redis.
- Implementação de pipeline assíncrono desacoplado entre API e workers, com persistência de estado em PostgreSQL e notificações em tempo real via WebSocket (Socket.IO) baseadas em eventos publicados por pub/sub.
- Aplicação de isolamento por cliente sem autenticação, utilizando UUID, cookies HTTP-only e segregação física de storage, além de controle de recursos com limites por cliente e expiração automática de dados (TTL).
- Ambiente totalmente containerizado com Docker Compose, orquestrando múltiplos serviços (API, workers, broker, cache e banco), garantindo execução previsível, escalabilidade horizontal e reprodutibilidade do sistema.