# Marcus Vinícius Carneiro dos Santos

Desenvolvedor Backend — Node.js | NestJS | TypeScript | AWS

akyriu@hotmail.com | +55 19 99477-0193 | linkedin.com/in/marcusvinícius- | github.com/akyriuu

## Summary

Desenvolvedor Backend focado em Node.js, NestJS e TypeScript para construção de APIs REST e sistemas distribuídos orientados a eventos. Atua em processamento assíncrono de pagamentos com AWS Lambda, SQS FIFO, SNS, S3, CloudWatch e IAM provisionados via Infrastructure as Code com Serverless Framework, e em mensageria com RabbitMQ, Transactional Outbox, dead letter queue e políticas de retry com backoff exponencial. Domínio de PostgreSQL e Prisma ORM em modelagem relacional, migrations versionadas, índices compostos, constraints de unicidade, controle de concorrência otimista e pessimista e garantias de idempotência. Experiência em integrações de pagamento com Stripe, Mercado Pago e PIX, incluindo Checkout e webhooks com verificação de assinatura HMAC, além de cache e rate limiting com Redis. Complementa o back-end com React e Next.js no front-end. Aplica Clean Code, SOLID, Design Patterns e máquinas de estado, com testes em Jest, Vitest, Supertest e Testcontainers, containerização em Docker e Docker Compose, LocalStack e CI/CD com GitHub Actions. Inglês C2 (Domínio Pleno).

## Education

**Fundamentos de Inteligência Artificial no Azure — AI-900 | Certificação**
Fundação Bradesco / Microsoft | 2026

**Administrando Banco de Dados | Curso Complementar**
Fundação Bradesco | 2026

**Recursos Humanos | Técnico**
ETEC Pedro Ferreira Alves | Conclusão em Dezembro 2021

## Experience

**Desenvolvedor Backend | Freelancer - Conchal, São Paulo | Jan 2026 – Presente**

- Desenvolvi APIs REST com Node.js, TypeScript e NestJS, entregando 15+ endpoints de autenticação, gestão de produtos e processamento de pagamentos, com JWT, refresh tokens, guards de rota e validação de payloads com class-validator e Zod.
- Projetei o processamento assíncrono de pagamentos end-to-end com 2 funções Lambda e 5 recursos AWS (SQS FIFO, dead letter queue, SNS, S3 e CloudWatch) provisionados via Infrastructure as Code, retirando a chamada ao provider do ciclo da requisição HTTP.
- Garanti cobrança única por requisição combinando chave de idempotência com unique index, controle de concorrência otimista por UPDATE condicional e replay da mesma chave ao provider em cada retry.
- Reduzi em 80% o reprocessamento de mensagens em lotes com falha isolada ao adotar o contrato de partial batch failure do SQS.
- Implementei política de retry em 2 camadas (3 tentativas de cobrança e 5 recebimentos) com roteamento para dead letter queue e alarme de detecção em até 60 segundos.
- Corrigi classe de falha crítica em que erro de infraestrutura posterior à autorização marcava como recusados pagamentos já cobrados no provider, isolando a chamada ao gateway atrás de uma fronteira transacional explícita.
- Modelei outbox transacional garantindo entrega at-least-once de eventos de domínio mesmo com o broker indisponível, padrão aplicado tanto sobre SQS/SNS quanto sobre RabbitMQ.
- Integrei 2 provedores de pagamento (Stripe e Mercado Pago), cobrindo Checkout, cobranças PIX e webhooks com verificação de assinatura HMAC, garantindo 100% de rejeição de notificações não autenticadas antes da atualização de status.

- Reduzi a carga de leitura no banco com cache Redis de TTL 10s e rate limiting por IP, sustentando picos de tráfego em serviço serverless com fallback resiliente e deploy em produção na Vercel.
- Modelei e mantive schemas com Prisma ORM e PostgreSQL, aplicando migrations versionadas, índices compostos e constraints de unicidade, com validação de entrada em 100% dos endpoints de escrita.
- Entreguei 3 aplicações full-stack com React.js, Next.js e NestJS, incluindo marketplace com checkout e serviço serverless de geração dinâmica de SVG, com testes em Jest e Vitest sobre as regras de negócio críticas.
- Padronizei ambiente de desenvolvimento reproduzível com Docker Compose e LocalStack, operacional em 2 comandos, com code review e versionamento via Git/GitHub.