

Gabriel Mizael de Sousa Gomes

Guarabira-PB · contato@gabrielmizael.com
gabrielmizael.com · linkedin.com/in/gabrielmizael · github.com/gbrlmzl

RESUMO CURRICULAR

Estudante de Sistemas de Informação (7º período) na Universidade Federal da Paraíba, com dois sistemas full stack em produção, atendendo usuários reais. Atuo da construção do backlog ao deploy: interfaces em Next.js e React, APIs REST em Spring Boot e Express, containerização com Docker e infraestrutura na AWS. Escrevo testes automatizados nas três camadas da pirâmide (unitários, de integração e ponta a ponta com Cypress), todos rodando em pipelines configuradas nos repositórios no GitHub, garantindo a CI a cada push.

FORMAÇÃO ACADÊMICA

Sistemas de Informação (UFPB, 2023–atual)
Previsão de conclusão: 12/2027

HABILIDADES TÉCNICAS

- **Linguagens de Programação:** Java, JavaScript, TypeScript, SQL, HTML, CSS
- **Frontend:** React, Next.js
- **Backend:** Spring Boot, Node.js, Express
- **Banco de Dados:** PostgreSQL, Modelagem relacional
- **Autenticação e Segurança:** JWT, Spring Security
- **Testes:** Jest, Testing Library, Supertest, Cypress (E2E), testes unitários e de integração
- **Deploy e Infraestrutura:** Docker, CI/CD (GitHub Actions), AWS ECS/EC2/S3, Caddy
- **Ferramentas e Ambiente:** Git/GitHub, Linux

Idiomas: Inglês – Intermediário

PROJETOS EM PRODUÇÃO

1. CRONOS

Desenvolvimento full stack, testes e infraestrutura — Frontend em Next.js, React e TypeScript + API em Node.js com Express

cronos.gabrielmizael.com | github.com/gbrlmzl/sistema-controle-despesas-front
github.com/gbrlmzl/sistema-controle-despesas-api | github.com/gbrlmzl/sistema-controle-despesas-deploy

Plataforma full stack em produção para controle de despesas compartilhadas entre moradores de uma mesma residência, com rateio automático da conta no fim do mês. Desenvolvi o frontend, a API e o banco de dados, e sou responsável pela infraestrutura: deploy em AWS ECS sobre EC2 com Caddy como borda da aplicação.

A suíte de testes cobre as três camadas da pirâmide: na API, testes unitários e de integração com Supertest contra PostgreSQL real, incluindo rate limiting e rotação de refresh token; no frontend, Jest e Testing Library; e, de ponta a ponta, Cypress nos fluxos críticos de usuário. A entrega segue o modelo build once, promote everywhere: a imagem publicada no GHCR só vai para produção depois que a suíte E2E passa contra a stack completa.

Tecnologias e conceitos aplicados: Next.js, React, TypeScript, Node.js, Express, Prisma, PostgreSQL, Jest, Testing Library, Supertest, Cypress, Docker, GitHub Actions (CI/CD), AWS ECS/EC2 (ARM64), Caddy e Cloudflare.

2. Website da Rinha do Campus IV

Liderança técnica/Arquitetura — Frontend em Next.js, React e TypeScript + Backend em Java com Spring Boot
rinhacampusiv.org | github.com/gbrlmzl/site-rinha-v2 | github.com/gbrlmzl/api-rinha-v2

Plataforma full stack em produção para gerenciamento de um torneio de League of Legends organizado na UFPB (Rinha do Campus IV). Atuei na liderança técnica e arquitetura do projeto, definindo a estrutura do frontend e do backend, além das integrações com APIs externas. A plataforma conta com módulo de inscrição de participantes,

módulo de pagamento integrado à API do Mercado Pago com atualizações em tempo real dos pagamentos via WebSocket, e painel administrativo para gerenciamento completo dos torneios.

Tecnologias e conceitos aplicados: Next.js, React, TypeScript, Spring Boot, Java, PostgreSQL, JWT, Spring Security, integração com APIs de terceiros (Mercado Pago, Imgur, Gmail SMTP) e Docker.

PROJETOS PESSOAIS

1. Chess Mania (Java)

github.com/gbrlmzl/chess-system-java

Aplicação em Java para terminal que implementa um jogo de xadrez para dois jogadores, expandido a partir de um curso prático da Udemey. Inclui funcionalidades adicionais como ranking, cadastro de jogadores e modos de jogo (casual e competitivo).

Tecnologias e conceitos aplicados: Java, programação orientada a objetos, tratamento de erros e persistência de dados em arquivos .csv.

2. Sistema Financeiro RT (Java)

github.com/gbrlmzl/sistemaFinanceiroRT

Aplicação desktop em Java com interface gráfica (Swing/JFrame) para controle de despesas compartilhadas. Permite registrar gastos e calcular automaticamente a divisão proporcional entre usuários.

Tecnologias e conceitos aplicados: Java, POO, desenvolvimento de interfaces gráficas e manipulação de dados.

3. Sistema de Controle de Despesas (Next.js)

github.com/gbrlmzl/sistema-controle-despesas

Aplicação web full stack desenvolvida em Next.js, React, JavaScript, CSS, Node.js e PostgreSQL. Trata-se da evolução do Sistema Financeiro RT, migrado para a web com novas funcionalidades, como cadastro de múltiplos usuários e persistência em banco de dados relacional via autenticação de usuários.