

Rafhael Oliveira

Itajubá, MG
Rafhaelx@gmail.com
(35) 9 9196-8159

<https://www.linkedin.com/in/rafael-oliveira/>
<https://github.com/Underewarr/>

Objetivo Profissional: Sou um apaixonado por tecnologia e desenvolvimento web, com experiência como desenvolvedor full stack. Trabalhei em diversos projetos pessoais e possuo sólidos conhecimentos em programação funcional, POO e DevOps. Meu interesse é focado em escalabilidade de aplicações e melhores práticas de desenvolvimento.

Contribuir e ajudar em comunidades de tecnologia é algo que adoro fazer, e estou sempre buscando aprender com projetos pessoais, aprimorando constantemente meus métodos de desenvolvimento de software.

Trabalho bem em equipe e sempre busco aprender com os erros para evitar repeti-los. Sou um entusiasta em tecnologia, buscando aprender novos conceitos e aplicá-los, na prática, visando melhorias contínuas em meu trabalho.

Estou em constante busca por conhecimento para me tornar um especialista na área de desenvolvimento, focado em entregar soluções de alta qualidade para os projetos em que me envolvo.

Formação

- Desenvolvimento Web Full Stack - Trybe, conclusão em Dezembro de 2022.

O programa conta com mais de 1.500 horas de aulas e aborda introdução ao desenvolvimento de software, front-end, back-end, ciência da computação, engenharia de software, metodologias ágeis e habilidades comportamentais.

Experiências Profissionais

- Desenvolvedor Front-end
Jan 2019 / Jan 2020
 - A
 - Buscar soluções e implementar correções no website.
- Manutenção de computadores e redes + Vendas de planos OI Telefonía (MEP Informatica)
Jan 2017 / Jan 2019
 - Abrir a loja e organizar documentos e planilhas assim como lidar com o dinheiro, comprar produtos e manter o estoque.
 - Vendas de planos de internet para a OI Telefonía.

Principais tecnologias

Frontend:

- HTML, CSS para criar interfaces elegantes e responsivas.
- JavaScript e Typescript para desenvolvimento de funcionalidades dinâmicas e escaláveis.
- React, Router, Redux e Hooks para construir aplicações modernas e de alto desempenho.
- Cypress e React Library Test para realizar testes automatizados e garantir a qualidade do código.

Backend:

- Git e Github para versionamento e colaboração no desenvolvimento de projetos.
- Linux e Windows Servers para implantação e administração de aplicações.
- Node.js e Next.js como base para a criação de servidores e APIs.
- SQL e MongoDB para gerenciar dados de forma eficiente e segura.
-

Idiomas

- Inglês - Intermediário - Self Learn

Principais projetos

Front-End:

- 1. Trade-Technology: Consumindo API de Futebol**
 - Desenvolvimento de uma aplicação que consome uma API com lista de todas as ligas, campeonatos e times de futebol do mundo.
 - Tecnologias Utilizadas: React, Typescript, Axios, Redux, Hooks, React-Bootstrap.
 - [Link para o GitHub](#)
- 2. Recipe App: Aplicação de Delivery de Bebidas**
 - Criação de uma aplicação de delivery de bebidas.
 - Tecnologias Utilizadas: React, Redux, Hooks, React-Bootstrap.
 - [Link para o GitHub](#)
- 3. Trybe Wallet: Carteira de Controle de Gastos com Conversor de Moedas**
 - Implementação de uma carteira para controle de gastos com recurso de conversor de moedas.
 - Tecnologias Utilizadas: React, Redux, React-Bootstrap.
 - [Link para o GitHub](#)
- 4. StarWars-Planet-Search: Lista com Filtros de Planetas de Star Wars**
 - Desenvolvimento de uma lista com filtros de planetas do universo de Star Wars usando Context API e Hooks.
 - Tecnologias Utilizadas: React, Context API.
 - [Link para o GitHub](#)
- 5. Pokemon React Game: Jogo de Pokémon com Context API e Hooks**
 - Criação de um jogo de Pokémon utilizando Context API e Hooks.
 - Tecnologias Utilizadas: React, Context API.
 - [Link para o GitHub](#)

Back-End:

- 1. Smith - Leitura, Atualização e Exclusão (CRUD)**
 - Desenvolvimento de uma aplicação com funcionalidades de CRUD (Create, Read, Update e Delete).
 - Tecnologias Utilizadas: Javascript, Node.js, Express, Sequelize com MySQL e Docker.
 - Aplicada Arquitetura de Software, com as camadas de Modelo, Serviço e Controladores.
 - [Link para o GitHub](#)
- 2. Blogs API: Aplicação para Postar Conteúdo em um Blog**
 - Criação de uma API para postar conteúdo em um blog.
 - Tecnologias Utilizadas: Node.js, Express, Sequelize com MySQL e Docker.
 - Aplicada Arquitetura de Software, com as camadas de Modelo, Serviço e Controladores.
 - [Link para o GitHub](#)
- 3. App de Delivery para Distribuidora de Bebidas**
 - Desenvolvimento de uma aplicação de delivery para uma distribuidora de bebidas.
 - Tecnologias Utilizadas: Javascript, Node.js, Express, Sequelize com MySQL e Docker.
 - Aplicada Arquitetura de Software, com as camadas de Modelo, Serviço e Controladores.
 - [Link para o GitHub](#)
- 4. NossoCha: App para Receber e Gerenciar Presentes de Casamentos**
 - Criação de um aplicativo para receber e gerenciar presentes de casamentos.
 - Tecnologias Utilizadas: Typescript, Node.js, Express, Sequelize com MySQL.
 - Aplicada Arquitetura de Software, com as camadas de Modelo, Serviço e Controladores.
 - Aplicado o padrão de projeto Serverless Functions.
 - [Link para o GitHub](#)
- 5. Serverless Workflow | Mercadopago SDK | Checkout Pro | Checkout Transparent**

- Implementação de um workflow serverless com integração ao MercadoPago SDK, Checkout Pro e Checkout Transparent.
- Tecnologias Utilizadas: Typescript, Node.js, Express, Sequelize com MySQL e Docker.
- Aplicada Arquitetura de Software, com as camadas de Modelo, Serviço e Controladores.
- [Link para o GitHub](#)