

Eduardo Cruz

Desenvolvedor Full-Stack

Cidade: Rio de Janeiro, Brasil

Telefone: (21) 98270-5996

E-mail: eduardovscruuz@gmail.com

LinkedIn: <https://www.linkedin.com/in/eduardovscruuz>

Github: <https://www.github.com/eduardovscruuz>

Portifólio: <https://eduardovscruuz.github.io/portfolio/>

Resumo Profissional e Habilidades:

Como desenvolvedor full-stack, possuo habilidades tanto na área de front-end quanto de back-end. No front-end, tenho experiência com HTML5, CSS3, JavaScript, TypeScript, React, Styled-Components e Sass. No back-end, tenho experiência em Node, Next, Express, APIs Rest/Restful, PostgreSQL, MongoDB, Git, Sequelize e Docker. Combinando essas habilidades, sou capaz de desenvolver soluções criativas para desafios complexos, utilizando as mais recentes tecnologias e ferramentas no desenvolvimento web.

Formação:

- **Análise e Desenvolvimento de Sistemas - UNINTER**
Jun/2023 – Dez/2025
- **Direito - UCAM**
Jan/2018 – Dez/2022

Cursos:

1. **Desenvolvimento Full Stack** - Dev Club
2. **Fundamentos da Programação** - Awari
3. **NodeJS e MongoDB** - Awari
4. **Typescript Zero to Hero** - Glaucia Lemos
5. **Desenvolvimento Web** - ProgramadorBR

Idiomas:

- **Português** - Fluente/Nativo
- **Inglês** - Avançado

Experiência:

- **Desenvolvedor Freelancer | Dez/22 ~ Atualmente**

Experiência no desenvolvimento de projetos web responsivos, utilizando habilidades sólidas em HTML, CSS e JavaScript, além de dominar tecnologias como React, Sass e Material UI para criar interfaces performáticas e responsivas. Também sou proficiente no uso de ferramentas

como Node.js, Express.js e Docker para criar e consumir APIs REST com bancos de dados SQL e NoSQL, incluindo Postgres e MongoDB.

Como parte do meu processo de desenvolvimento, adoto práticas de código limpo e legível, utilizando gerenciadores de pacotes como Yarn e NPM, bem como o Sequelize para interagir com bancos de dados SQL. Além disso, sou hábil em consumir APIs para integrar dados em sistemas web.

- **Auxiliar Administrativo na Laboclin | Mai/21 ~ Atualmente**

Como responsável pelo atendimento digital da empresa, contribuí de forma significativa para melhorar o setor. Unifiquei os atendimentos das filiais e ajudei a criar a central de atendimento única, em conformidade com a LGPD. Além disso, implementei o software CutePDF para aplicar senhas em arquivos sensíveis, evitando a necessidade de conferência manual. Essas mudanças resultaram em maior eficiência, economia de tempo e segurança nos processos da empresa.

Principais Projetos:

- **DevBurger** - Hamburgueria online com páginas para: Cadastro, login, realização de pedidos e pagamento por parte do cliente. Conta também com páginas exclusivas para o administrador, que pode realizar a exclusão e adição de produtos, assim como visualizar movimento de caixa. No **front-end**, a interface foi construída em React, usa LocalStorage do navegador para manter as sessões ativas, e tokens validados, no desenvolvimento foram utilizados hooks como useState, useContext (alternativa nativa do react para Redux), e bibliotecas como Styled-components, Router Dom, Axios, e para otimizar o desempenho e a modularização do projeto, a Material UI. Tudo integrado ao **back-end**, onde construí uma API Rest, usando NodeJS e Express. A API utiliza PostgreSQL e MongoDB para armazenar dados. Usei ORM (Sequelize e Mongoose), YUP para validação de dados, Bcrypt para criptografia de senhas e JWT para gerar tokens de acesso.
- **To-Do-List** - Aplicação onde o usuário pode, usando o navegador, cadastrar tarefas, editar os detalhes dessa tarefa, realizar a marcação daquelas ao qual completou, como também pode excluir, projeto foi construído com HTML, CSS, Javascript, React, Styled-Components, React Router Dom, Axios, uuidv4, e integração com uma fakeApi para armazenamento dos dados.
- **React Users (CRUD)** - O projeto é uma aplicação web desenvolvida com React para a interface do usuário e Node.js para o back-end. Ele permite que o cliente cadastre e gerencie usuários, com nome e idade, com uma página de cadastro e outra de listagem de usuários com opção de exclusão. O servidor back-end é construído com Express.js e integra com o banco de dados MongoDB utilizando o pacote Mongoose. O projeto também utiliza outras bibliotecas e ferramentas, como Axios para lidar com requisições HTTP, react-router-dom para navegação entre páginas e .env para gerenciamento de variáveis de ambiente.