João Vitor Veras de Oliveira

Email: joaovitorzv@outlook.com | **Telefone**: (16) 9 9309-4531
Github: github.com/joaovitorzv | LinkedIn

SOBRE MIM

Moro no Interior de São Paulo (Franca/SP), tenho 18 anos, ensino médio completo. Me considero um desenvolvedor autodidata, meu primeiro contato com tecnologia foi em 2017, no entanto, não passava de um hobbie. Em 2019 decidi estudar de verdade para construir uma carreira na área, desde então aprendi muitas tecnologias, mas foi em 2021 que eu aprendi a aprender, hoje me considero um Desenvolvedor Front-end com um perfil T-Shaped, tenho interesse e me sinto confortável em aprender coisas novas com facilidade.

RESUMO TÉCNICO

Proficiência com JavaScript, TypeScript, HTML, CSS e Python.

Experiência em desenvolver aplicações web com ReactJS e NextJS, utilizando diversas tecnologias e bibliotecas como ApolloClient, GraphQL, React Testing Library, Formik, SASS, StyledComponents, Material UI (entre outras)

Experiência com Git Flow, CI/CD (Github Actions), métodos ágeis e Experiência *prévia* com PostgreSQL, MongoDB, Redis, Express

SIDE PROJECTS

Blog Pessoal • Blog simples feito com Next.JS, por ser hosteado no Github Pages foi necessário exportar o site para HTML estático utilizando o “next export”. o conteúdo do site é gerenciado pelo GraphCMS, o único motivo de eu ter escolhido um CMS foi porque eu queria implementar algo utilizando GraphQL, utilizei o *apollo-client* para fazer as requests à API em build-time, o conteúdo dos posts são retornados em raw e renderizados com o *rich-text-react-renderer* dessa forma eu tenho controle sobre o que está sendo renderizado e consigo estilizar imagens, cores, links etc com base no style-guide do blog. No geral a aplicação é bem simples, mas uma feature bacana de implementar foi o syntax-highlighting com o *prismjs*, para detectar qual linguagem está sendo usada nos snippets de código precisei definir minha própria “regra de negócio”, basicamente uma string com o “dollar sign” ao redor e com regex foi possível detectar a linguagem, passa-la para o prismjs e remover essa string do snippet deixando apenas o código

Taskee • Um kanban board que permite você adicionar, mover ou deletar tasks, na página do board, existem stacks com seu respectivo status, para adicionar novas tasks existe um botão em cada stack que aciona um form, que, caso seja preenchido corretamente, é feito um *dispatch* no *state* da aplicação adicionando a nova task. Todas as ações disponíveis na aplicação (mover, deletar, adicionar) estão em um objeto com funções anônimas que acionam o dispatch, esse objeto é compartilhado entre os componentes, que o utilizam, tornando assim mais fácil a relação com o controle de estados da aplicação. Para lidar com o *drag and drop* foi utilizado a lib *react-beautiful-dnd* que deixa mais fácil a interação do usuário ao alterar o status de alguma task, vale lembrar que após qualquer atualização no state é acionado um service, que persiste todas as tasks e seus status no *localStorage*

React Testing Library • Meu objetivo com esse projeto foi desenvolver algo simples mas que eu pudesse fazer uma cobertura de testes e aprender a testar aplicações web utilizando a biblioteca *react testing library*. O projeto consiste em uma página com uma *tab navigation,* onde, você pode encontrar o seu CEP pesquisando pelo nome da rua ou verificar de qual endereço determinado CEP pertence. Dentre outras funcionalidades da aplicação que foram testadas os testes que precisavam interagir com alguma requisição eu precisei “mockar” as respostas da API, para isso utilizei o MSWJS (Mock Service Worker)