

# João Vitor Veras de Oliveira

---

Email: [joaovitorzv@outlook.com](mailto:joaovitorzv@outlook.com) | Telefone: (16) 9 9309-4531 | [LinkedIn](#) | [Github](#)

## SOBRE MIM

Tenho 18 anos, meu primeiro primeiro “hello world” foi em 2017, me considero um desenvolvedor autossustentável pois consigo realizar tarefas que nunca realizei antes utilizando a documentação, stackoverflow, github issues etc. Apesar de me aprofundar em desenvolvimento web, também tenho interesse em outras áreas como Backend, Mobile e Algoritmos no geral. Meu nível de inglês é avançado e todo o conteúdo sobre tecnologia que eu consumo atualmente é em inglês.

## EXPERIÊNCIA

\*Todos os projetos abaixo estão disponíveis no [meu github](#) com um readme rico em detalhes\*

**Know Your Taste** • Para implementar o OAuth 2.0 ( algo que eu nunca tinha feito antes ) utilizei o NextAuth.js que conta com vários providers built-in e o spotify é um deles, essa biblioteca também faz a validação de um CSRF token aumentando a segurança do fluxo de login, um ponto legal de implementar foi a regeneração do auth token após o tempo de expiração de 1 hora utilizando um objeto de callbacks eu pude criar uma função que faz essa verificação do tempo e caso tenha expirado é feito uma request de um novo token para a API do Spotify, para lidar com os dados requisitados da API utilizei o SWR uma library que conta com várias features interessantes, uma key feature que eu utilizei foi no componente de playlists onde eu implementei uma paginação com scroll infinito com limite de 5 playlists por request. O projeto está no ar em [know-your-taste.vercel.app](https://know-your-taste.vercel.app) porém infelizmente API do spotify em modo de desenvolvimento limita a autenticação para apenas usuários com seu email do spotify explicitamente definido na whitelist do projeto.

**Blog Pessoal** • Blog simples feito com Next.js ([pode ser acessado por aqui](#)), o blog é exportado para html estático assim pode ser servido pelo Github Pages, o conteúdo do site é gerenciado pelo GraphCMS e as requests à API são feitas em build-time utilizando o apollo-client.

**React-state-tabs** • Um component-library que está disponível como [package no NPM](#), utilizei o TSDX para facilitar a configuração de rollup, jest, treeshaking etc e focar no desenvolvimento da biblioteca que é basicamente um componente de Tabs que controla a aba selecionada e o seu conteúdo visível, a key feature deste projeto são as animações na troca de abas que podem ser estilizadas de acordo com a necessidade do usuário ou até mesmo desabilitar as animações e ter acesso apenas ao estado de qual aba está ativa.

## OBJETIVO

Fazer um trabalho bem feito dentro do prazo, aprimorar minhas habilidades através da prática e ser um teammate amigável.