

Teste Econverse: Vaga Desenvolvedor Front-End

Objetivo do Teste

Desenvolver o front end de um e-commerce utilizando um layout. Nesse projeto irá consumir um JSON para alimentar as informações na parte de vitrine de produtos, visando os seguintes requisitos funcionais:

RF01 - Desenvolvimento da Página em React e TypeScript

- Implementar a página conforme o layout especificado no Figma: [Layout do teste](#)
- Para acessar os elementos do Figma, copie o layout para sua conta e tenha acesso de edição.
- Utilizar React e TypeScript como stacks

RF02 - Montagem da Vitrine de Produtos a partir de um JSON

- Consumir as informações dos produtos a partir do seguinte JSON: [JSON de Produtos](#)
- Desenvolver a interação ao clicar em um produto, seguindo o layout. Ao clicar, abrir um modal com as principais informações do produto presente no arquivo JSON correspondente.

RF03 - Utilização de Pré-processador (Sass, Less ou Stylus) e Outras Considerações

- Utilizar um pré-processador de CSS (Sass, Less ou Stylus).
- Respeitar o layout pixel a pixel, incluindo tamanho das fontes, cores e botões.
- Não utilizar bibliotecas UI como Bootstrap, Foundation ou similares.
- Adotar HTML semântico.
- Organizar o projeto de forma estruturada.
- Implementar lógica do código e componentização.

Resolução do teste

Desenvolvi o projeto utilizando React, Vite, Typescript e Scss. O projeto segue o layout especificado nos requisitos do teste, criando um e-commerce com uma vitrine de produtos. Os dados dos produtos são consumidos a partir de um arquivo JSON. Quando o usuário clica em um produto, um modal é aberto com as principais informações desse produto, conforme o layout. Utilizei Hooks para gerenciar o estado, componentização para modularizar o código e Scss para estilização.

Para resolver o problema de CORS (Cross-Origin Resource Sharing), que ocorre quando o navegador bloqueia requisições feitas de um domínio para outro, configurei um proxy no Vite durante o desenvolvimento. Com o proxy configurado, ajuste a URL da requisição no componente Products.tsx para usar o endpoint proxy.

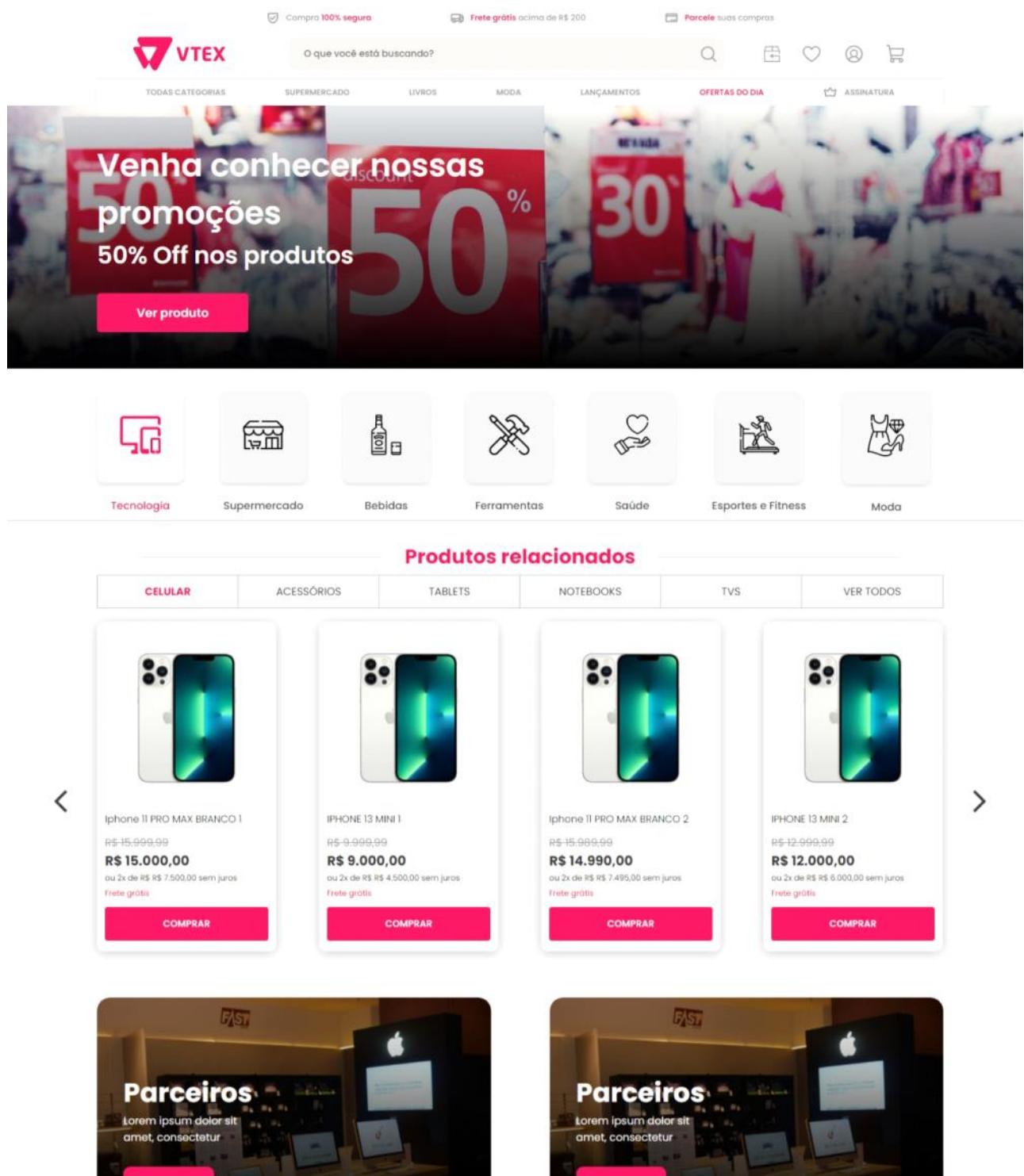
O problema com o CORS surgiu ao tentar consumir o JSON diretamente através de um link sem tratamento: <https://app.econverse.com.br/teste-front-end/junior/tecnologia/lista-produtos/produtos.json>

```
EconverseTeste > vite.config.ts > [vite] default
1  import { defineConfig } from 'vite'
2  import react from '@vitejs/plugin-react'
3  import svgr from 'vite-plugin-svgr'
4
5  // solução do problema de CORS (Cross-Origin Resource Sharing) - Proxy
6  // https://vitejs.dev/config/
7  export default defineConfig({
8    server: {
9      proxy: {
10        '/api': {
11          target: 'https://app.econverse.com.br',
12          changeOrigin: true,
13          rewrite: (path) => path.replace(/^\/api/, ''),
14        },
15      },
16    },
17    plugins: [react(), svgr()],
18  })
```

```
EconverseTeste > src > components > products > Products.tsx > [vite] Products
1  import './products.scss';
2  import Title from '../global/Title';
3  import { useState } from 'react';
4  import LinkCard from '../global/LinkCard';
5  import Modal from '../Modal';
6  import Carousel from '../carousel/Carousel';
7  import useFetch from '../hooks/useFetchJsonProducts';
8  import { ProductType, APIResponse } from '../typesJson';
9
10 const Products = () => {
11   // Define os estados locais usando useState
12   const [isModalOpen, setIsModalOpen] = useState<boolean>(false); // Estado para controlar se o modal está aberto
13   const [modalData, setModalData] = useState<ProductType | null>(null); // Estado para armazenar os dados do produto selecionado
14
15   // Usa o hook personalizado para buscar os dados de produtos
16   // link json do testes: https://app.econverse.com.br/teste-front-end/junior/tecnologia/lista-produtos/produtos.json
17   const { data, loading, error } = useFetch<APIResponse>({
18     "/api/teste-front-end/junior/tecnologia/lista-produtos/produtos.json"
```

```
EconverseTeste > src > components > products > Products.tsx > Products
1 import './products.scss';
2 import Title from '../global/Title';
3 import { useState } from 'react';
4 import LinkCard from '../global/LinkCard';
5 import Modal from './Modal';
6 import Carousel from '../carousel/Carousel';
7 import useFetch from '../hooks/useFetchJsonProducts';
8 import { ProductType, APIResponse } from '../typesJson';
9
10 const Products = () => {
11   // Define os estados locais usando useState
12   const [isModalOpen, setIsModalOpen] = useState<boolean>(false); // Estado para controlar se o modal está aberto
13   const [modalData, setModalData] = useState<ProductType | null>(null); // Estado para armazenar os dados do produto selecionado
14
15   // Usa o hook personalizado para buscar os dados de produtos
16   // link json do testes: https://app.econverse.com.br/teste-front-end/junior/tecnologia/lista-produtos/produtos.json
17   const { data, loading, error } = useFetch<APIResponse>({
18     "/api/teste-front-end/junior/tecnologia/lista-produtos/produtos.json"
19   });
20 }
```

Resultado final (E-commerce VTEX)





Produtos relacionados

[Ver todos](#)



Navegue por marcas



Sobre nós CONHEÇA COMO COMPRAR INDICAÇÃO E DESCONTO f @ v	INFORMAÇÕES ÚTEIS FALE CONOSCO DÚVIDAS PRAZOS DE ENTREGA FORMAS DE PAGAMENTO POLÍTICA DE PRIVACIDADE TROCAS E DEVOLUÇÕES	FORMAS DE PAGAMENTO 	CADASTRE-SE E RECEBA NOSSAS NOVIDADES E PROMOÇÕES <p>Excepteur sint occaecat cupidatat non ent, sunt in culpa qui officia Lorem ipsum</p> <div><input type="text" value="SEU E-MAIL"/> <input type="button" value="OK"/></div>
--	---	--------------------------------	--

COPYRIGHT © 2019. TODOS OS DIREITOS RESERVADOS. TODAS AS MARCAS E SUAS IMAGENS SÃO DE PROPRIEDADE DE SEUS RESPECTIVOS DONOS. É VEDADA A REPRODUÇÃO, TOTAL OU PARCIAL, DE QUALQUER CONTEÚDO SEM EXPRESSA AUTORIZAÇÃO.