

# Desenvolvimento de Sistema Frontend

## Fase 2- Matias Cazarotto

1 – Abrir a pasta ‘projeto\_fase\_2’ no VSCode.

2 – No terminal, acessar o diretório ‘hotwheels’ e executar o comando ‘npm install’ ou ‘npm install--force’ se necessário.

3 – Acessar o diretório ‘hotwheels-api’ e executar o mesmo comando ‘npm install’ ou ‘npm install--force’ se necessário.

4 – Ainda no diretório ‘hotwheels-api’, executar o comando ‘npm start’.

5 – No diretório ‘hotwheels’, executar o comando ‘npm start’.

Após estes passos, o aplicativo deverá ser aberto no navegador;

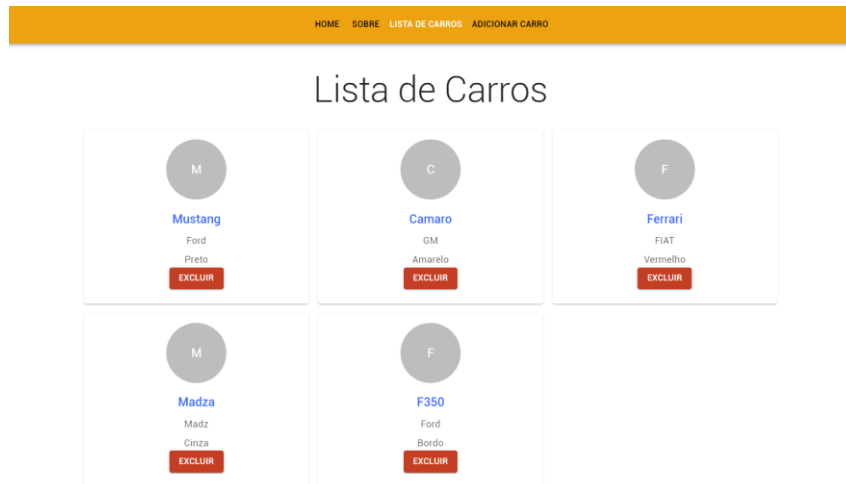


## Descrição dos componentes

- **carAdd**: Permite adicionar um novo carro no aplicativo. É buscado os carros via a API useHotWheels (get). E para adicionar um novo carro, é utilizado o método post, incrementando o carro na lista.

A screenshot of the 'Adicionar Carro' (Add Car) form in the web application. It features an orange navigation bar at the top with links: HOME, SOBRE, LISTA DE CARROS, and ADICIONAR CARRO. The title 'Adicionar Carro' is centered above the form. The form consists of four input fields stacked vertically, labeled 'Nome', 'Marca', 'Cor', and 'Ano'. Below these fields is a grey button labeled 'SALVAR'.

- **listCar**: Permite a visualização, edição e exclusão de cada veículo cadastrado no sistema. É feito uma requisição na API useHotWheels (get) solicitando o array com os carros, utilizando o map, é mostrado todos os carros. Também neste componente, é utilizado a função deleteCar da API useHotWheelsApi no qual utiliza o método delete para fazer a exclusão.



- **carEdit**: Componente que é utilizado dentro da page CarEdit.jsx, no qual faz toda a funcionalidade de edição dos campos, pegando as informações diretamente da API e modificando-as com a função updateCar, que utiliza o a api useHotWheelsApi.jsx chamando o método de requisição PUT via updateCar.

A imagem mostra o formulário de edição de um carro. No topo, há uma barra de navegação laranja com os links: HOME, SOBRE, LISTA DE CARROS (destacado) e ADICIONAR CARRO. Abaixo, o formulário contém quatro campos de texto com rótulos cinza: 'Nome' (contendo 'Mustang'), 'Marca' (contendo 'Ford'), 'Cor' (contendo 'Preto') e 'Ano' (contendo '1970'). Abaixo dos campos, há um botão azul 'SALVAR'.

- **Navbar**: barra de navegação do sistema. Onde é feito as transições de telas entre Home/Sobre/Lista de Carros/Adicionar Carro utilizando o React Router.

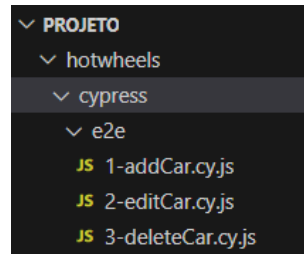


Por fim, foi utilizado o Cypress para testes. No qual, para fins de aprendizado fiz 3 arquivos nos quais:

1 – Adiciona 2 novos carros à lista;

2 – Edita os carros criados anteriormente;

3 – Exclui os carros editados anteriormente;



Para a execução do sistema de testes, deve-se utilizar o comando 'npx cypress open'. Sendo assim, o ciclo de criação, edição e exclusão pôde ser testado em sua totalidade.