

PUCRS

Curso: Análise e Desenvolvimento de Sistemas

Disciplina: Desenvolvimento de Sistemas Frontend

Aluno: Hayleno Santos Hossoé

Entrega do Projeto Fase 2: Um arquivo zipado, salvo como nome_aluno-projeto-fase-2.zip. Esse .zip deverá conter todos os arquivos para o projeto funcionar (exceto a pasta node_modules), um arquivo PDF com breve explicação sobre como proceder para executar o projeto e uma descrição do que cada componente realiza e como esses são conectados à API. ATENÇÃO: Seu projeto deve permitir execução, a fim de ser visualizado em um navegador.

Instruções:

Nosso projeto é um CRUD de carros HotWheels que usa React e Node.js para criação de um frontend com diferentes componentes, além de formulário e funcionalidades. Utilizamos bibliotecas como React Router, para criar rotas de navegação entre os componentes, Material UI, para estilização, e Axios, para realizar requisições HTTP e consumir dados de um servidor, entre outras. Também utilizamos o Cypress para realizar testes dos componentes de forma automatizada.

O projeto pode ser visualizado e utilizado a partir de um navegador e contém a estrutura padrão de arquivos e pastas de um projeto React, dispostos da seguinte forma, com as principais pastas, subpastas e componentes:

node_modules – pasta padrão de projetos React que utilizam dependências do Node.js

public – pasta padrão de projetos React

src – pasta padrão de projetos React com componentes editáveis

package.json - arquivo de configuração utilizado para estipular e configurar dependências do seu projeto

package-lock.json – arquivo de configuração com os registros das dependências instaladas

src\App.js – arquivo principal do aplicativo, que contém o código para utilização dos demais componentes, incluindo o roteamento para eles

src\App.css – arquivo de estilo do App.js

src\index.js – arquivo de índice do projeto

src\index.css – arquivo de estilo do index.js

src\Components – subpasta que contém outras 8 subpastas com componentes .JS do aplicativo e seus respectivos arquivos .CSS de estilos

src\Components\About – subpasta do componente “About.js”, que exibe uma explicação sobre o aplicativo

src\Components\CarDetail – subpasta do componente “CarDetail.js”, que exibe as informações detalhadas do carro selecionado

src\Components\CarForm – subpasta do componente “CarForm.js”, que possui um formulário para adicionar carros ou editá-los

src\Components\CarsList – subpasta do componente “CarsList.js”, que exibe a listagem de carros e permite buscá-los, editá-los ou excluí-los

src\Components\Home – subpasta do componente “Home.js”, que exibe informações da página inicial do aplicativo

src\Components\Hooks – subpasta do componente “useCars.js”, que é um Hook personalizado configurado para utilizar a biblioteca Axios e consumir os dados da API, permitindo aos demais componentes do aplicativo operações como obter, editar, adicionar e excluir dados.

src\Components\Navbar – subpasta do componente “Navbar.js”, que contém uma barra de navegação com as opções Home, Sobre, Carros e Adicionar Carro

src\Components\NotFound – subpasta do componente “NotFound.js”, que exibe uma mensagem de “Carro não encontrado” quando se busca por um carro inexistente ou se apaga um carro quando se está na página de informações detalhadas

Server\server.js – arquivo que contém uma API que funciona como servidor para permitir acesso ao arquivo carsData.js por meio de rotas GET e POST, como solicitado, e PUT e DELETE, que foram incluídas, disponíveis no endereço <http://localhost:5000/cars>

Server\carsData.json – arquivo que contém um array com dados de carros a serem utilizados no aplicativo

cypress\e2e\about.spec.cy.js – arquivo que contém testes para Cypress do componente About.js

cypress\e2e\carDetail.spec.cy.js – arquivo que contém testes para Cypress do componente CarDetail.js

cypress\e2e\carForm.spec.cy.js – arquivo que contém testes para Cypress do componente CarForm.js

cypress\e2e\carsList.spec.cy.js – arquivo que contém testes para Cypress do componente CarsList.js

cypress\e2e\home.spec.cy.js – arquivo que contém testes para Cypress do componente Home.js

cypress\e2e\navbar.spec.cy.js – arquivo que contém testes para Cypress do componente Navbar.js

cypress\e2e\notFound.spec.cy.js – arquivo que contém testes para Cypress do componente NotFound.js

cypress\e2e\useCars.spec.cy.js – arquivo que contém testes para Cypress do componente useCars.js

Siga os seguintes passos para instalar e executar o projeto:

1 - Baixar e instalar o Node.js

2 – Abra sua pasta de projetos usando o Terminal e digite o comando “npx create-react-app hotwheels-crud”

3 – Entre na pasta “hotwheels-crud” e digite os comandos abaixo para instalar as bibliotecas e dependências necessárias à execução do projeto:

```
npm install react-router-dom
```

```
npm install @mui/material @emotion/react
```

```
npm install @mui/icons-material
```

```
npm install @emotion/styled
```

```
npm install -g nodemon
```

```
npm install express-generator
```

```
npm install axios
```

```
npm install express
```

```
npm install cors
```

```
npm install cypress --save-dev
```

4 – Baixe e descompacte a pasta zipada hayleno-hosoe-projeto-fase-2.zip, depois copie os arquivos e pastas para dentro da pasta do seu projeto hotwheels-crud, sobrescrevendo os arquivos e pastas com o mesmo nome.

5 – Dentro da pasta “hotwheels-crud\Server” utilize o Terminal e digite “node server.js” para iniciar a API do servidor, que permitirá ao aplicativo acessar as rotas para manipulação dos dados do arquivo carsData.json. Ele estará acessível no navegador no endereço “<http://localhost:5000/cars>”

6 – Dentro da pasta “hotwheels-crud” utilize o Terminal e digite “npm start” para iniciar o aplicativo no navegador. Ele estará acessível no navegador no endereço “<http://localhost:3000/>”

7 – Dentro da pasta “hotwheels-crud” utilize o Terminal e digite “npx cypress open” para iniciar o Cypress e realizar testes por meio do navegador.

* Observações sobre utilização de API alternativa:

Apesar de ter tentado diversas vezes, não consegui fazer uso da API indicada nas orientações da Fase 2 do Projeto da disciplina devido à falhas no download e execução da mesma. Dessa forma, desenvolvi uma API alternativa, que funciona como servidor e executa as mesmas tarefas, permitindo acesso ao arquivo carsData.js por meio de rotas GET e POST, como solicitado, e PUT e DELETE, que foram incluídas, disponíveis no endereço <http://localhost:5000/cars>. Com esta solução, o aplicativo consome a API desenvolvida e disponibiliza todas as funcionalidades solicitadas no projeto.

* Relato dos problemas enfrentados na utilização da API sugerida no projeto:

Nas orientações consta essa instrução na parte de "Recurso":

"Você deve fazer uso da API Hotwheelsl-api que se encontra em <https://github.com/adsPucrsOnline/DesenvolvimentoFrontend/> (branch dev)."

Entretanto, tentei fazer o gitclone tanto da pasta inteira da disciplina quanto a pasta "hotwheels-api", mas ocorreu erro. Inclusive, baixei a pasta toda da disciplina como um arquivo .ZIP, mas ocorreu erro ao tentar descompactá-la.

Acredito que os erros estejam ocorrendo, provavelmente, devido ao nome da subpasta "public/stylesheets", dentro de "hotwheels-api", que não é aceito como nome de pasta pelo Windows, em razão da barra "/" no nome.

Dessa forma, não pude baixar ou fazer o git clone de toda a pasta da disciplina devido a este erro.

Buscando soluções, baixei manualmente todos os arquivos e subpastas da pasta da API da Fase 2 do Projeto, "hotwheels-api", e criei a mesma estrutura de pastas, substituindo a pasta "public/stylesheets" por "public".

Em seguida, tentei executar esta outra orientação para rodar o servidor:

"Para executar a API, basta executar npm start, após o git clone."

Novamente, não foi possível, pois apareceu mensagem pedindo a instalação do nodemon e não havia orientação nas aulas da disciplina ou nas instruções da Fase 2 do Projeto sobre como instalá-lo e usá-lo. Procurei na internet e instalei por meio do comando "npm install -g nodemon".

Entretanto, nem assim funcionou a orientação, pois, ao executar o "npm start" na pasta, apareceu um longo erro com essa mensagem aparecendo repetidas vezes:

```
> hotwheels-api@0.0.0 start
> nodemon ./bin/www
```

```
[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
```

Instalei a pasta "node_modules" na pasta "hotwheels-api" da API e continuou ocorrendo o mesmo erro.