

**IFTM - INSTITUTO FEDERAL DO TRIÂNGULO MINEIRO  
ENGENHARIA DE COMPUTAÇÃO**

Rafael de Oliveira Évora  
Yuri David Silva Duarte

**OITAVA ETAPA - TUTORIAL  
BACK EXPRESS.JS E FRONT REACT**

**Uberaba  
2024**

Chegou a hora de ver como ficou o código final.

Inicialmente vamos configurar as rotas usando os comandos a baixo em src/routes/Rotas.tsx

```
import React from 'react';
import { BrowserRouter as Router, Route, Routes } from
'react-router-dom';
import TelaInicial from '../pages/TelaInicial.tsx';
import BuscaDoador from '../pages/BuscaDoador.tsx';
import CadastroDoador from '../pages/CadastroDoador.tsx';
import DoadorDoacao from '../pages/DoadorDoacao.tsx';
import Doacoes from '../pages/Doacoes.tsx';

function Rotas() {
  return (
    <div>
      <Router>
        <Routes>
          <Route path="/" element={<TelaInicial />} />
          <Route path="/BuscaDoador" element={<BuscaDoador />} />
          <Route path="/CadastroDoador" element={<CadastroDoador
/>} />
          <Route path="/BuscaDoador/DoadorDoacao"
element={<DoadorDoacao />} />
          <Route path="/Doacoes" element={<Doacoes />} />
        </Routes>
      </Router>
    </div>
  );
}

export default Rotas;
```

Para que as rotas funcionem é necessário instalar a biblioteca a baixo:

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

Em src/index.tsx nosso código ficou assim:

```
import React from 'react';
```

```
import { createRoot } from 'react-dom/client';
import Rotas from './routes/Rotas.tsx';

const container = document.getElementById('root');
const root = createRoot(container!);

root.render(
  <React.StrictMode>
    <Rotas />
  </React.StrictMode>,
);
```

Para o css funcionar corretamente, precisaremos do seguinte código em src/global.d.ts:

```
declare module '*.module.css' {
  const classes: { [key: string]: string };
  export default classes;
}
```

Nós criamos um código de cor padrão para a aplicação, ela fica armazenada em src/colors/cores.css:

```
:root{

  --azulpastel: #80cee1;
  --branco: white;
  --vermelho: #ff2401;
  --verde: #228b22;

}
```

Em src/pages/TelaInicial.tsx ficou dessa forma:

```
import React from 'react';
import { Link } from 'react-router-dom';
import './global.d.ts';
import estilo from './styles/TelaInicial.module.css';

function TelaInicial() {
  return (
    <body className={estilo.body}>
      <div className={estilo.container}>
        <h1 className={estilo.h1}>Tela Inicial</h1>
      </div>
    </body>
  );
}
```

```

        <nav className={estilo.nav}>
          <Link className={estilo.link} to="/CadastroDoador">Cadastrar
Doador</Link>
          <Link className={estilo.link} to="/BuscaDoador">Buscar
Doador</Link>
          <Link className={estilo.link} to="/Doacoes">Ver Doações</Link>
        </nav>
      </div>
    </body>
  );
}

export default TelaInicial;

```

Para o CSS desta página incluiremos em src/styles/TelaInicial.modules.css o código:

```

@import url('../colors/cores.css');

body{

  background-color: var(--azulpastel);
  margin: 0;
  padding: 0;

}

.body {

  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: Arial, sans-serif;
  height: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
  margin-top: 15vh;

}

.container {
  display: flex;
  justify-content: center;

```

```
    align-items: center;
    text-align: center;
    background-color: #fff;
    padding: 40px;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    max-width: 400px;
}

.h1 {
    margin-bottom: 20px;
    font-size: 24px;
    color: #333;
    margin-right: 30px;
}

.nav {
    display: flex;
    flex-direction: column;
    gap: 10px;
}

.link {
    padding: 10px 20px;
    font-size: 16px;
    color: white;
    background-color: #4CAF50;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
    text-decoration: none;
}

.link:hover {
    background-color: #45a049;
}
```

Para cadastrar um doador a tela ficará da seguinte maneira (src/pages/CadastroDoador.tsx):

```
import React from "react";
import estilo from "../styles/CadastroDoador.module.css";
import "../global.d.ts";

function CadastroDoador() {

    const handleCadastro = async (event:
    React.FormEvent<HTMLFormElement>) => {
        event.preventDefault(); // Evita o comportamento padrão do
        formulário de recarregar a página

        // Obtenha os dados do formulário
        const nomeInput = document.getElementById("nome") as
        HTMLInputElement | null;
        const cpfInput = document.getElementById("cpf") as
        HTMLInputElement | null;
        const contatoInput = document.getElementById("contato") as
        HTMLInputElement | null;
        const tipoSanguineoInput =
        document.querySelector('input[name="opcao_tipo"]:checked') as
        HTMLInputElement | null;
        const fatorRhInput =
        document.querySelector('input[name="opcao_rh"]:checked') as
        HTMLInputElement | null;

        // Verifica se os elementos foram encontrados antes de
        acessar suas propriedades
        if (nomeInput && cpfInput && contatoInput &&
        tipoSanguineoInput && fatorRhInput) {
            const nome = nomeInput.value;
            const cpf = cpfInput.value;
            const contato = contatoInput.value;
            const tipoSanguineo = tipoSanguineoInput.value;
            const fatorRh = fatorRhInput.value;

            // Monta o objeto com os dados
            const dadosDoador = {
                nome,
                cpf,
                contato,
                tipoSanguineo,
                fatorRh
            }
        }
    }
}
```

```

    });

    // Faz a requisição POST para o backend
    try {
        //console.log('Dados do doador:', dadosDoador);
        const response = await
fetch('http://localhost:5000/api/insertDoador', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json'
            },
            body: JSON.stringify(dadosDoador)
        });

        if (response.ok) {
            console.log("Doador cadastrado com sucesso!");
            // Realize aqui qualquer ação adicional após o
            cadastro bem-sucedido
            window.location.reload();
        } else {
            console.error('Falha ao cadastrar doador.');
        }
    } catch (error) {
        console.error('Erro ao fazer a requisição:',
error);
    }
    } else {
        console.error('Elementos do formulário não foram
encontrados.');
```

```

className={estilo.titulo_formulario}>Cadastro de doador</h1>
    <label htmlFor="nome"
className={estilo.labelFormulario}>Nome:
        <input
            type="text"
            name="nome"
            id="nome"
            placeholder="Insira o nome aqui"
            className={estilo.inputFormulario}
            required
        />
    </label>
    <br />
    <label htmlFor="cpf"
className={estilo.labelFormulario}>CPF:
        <input
            type="text"
            name="cpf"
            id="cpf"
            placeholder="Insira o CPF"
            className={estilo.inputFormulario}
            required
        />
    </label>
    <br />
    <label htmlFor="contato"
className={estilo.labelFormulario}>Contato:
        <input
            type="tel"
            name="contato"
            id="contato"
            className={estilo.inputFormulario}
            placeholder="Número de contato"
            required
        />
    </label>
    <br />
    <div className={estilo.opcao}>
        <div className={estilo.opcao_tipo}>
            <p>Tipo sanguíneo:</p>
            <input
                type="radio"

```



```

        name="opcao_tipo"
        id="A"
        value="A"
    />
<label htmlFor="A">A</label>
<br />
<input
    type="radio"
    name="opcao_tipo"
    id="B"
    value="B"
/>
<label htmlFor="B">B</label>
<br />
<input
    type="radio"
    name="opcao_tipo"
    id="AB"
    value="AB"
/>
<label htmlFor="AB">AB</label>
<br />
<input
    type="radio"
    name="opcao_tipo"
    id="O"
    value="O"
/>
<label htmlFor="O">O</label>
</div>
<div className={estilo.opcao_rh}>
    <p>Fator RH:</p>
    <input
        type="radio"
        name="opcao_rh"
        id="POSITIVO"
        value="+"
    />
    <label htmlFor="POSITIVO">+</label>
    <br />
    <input
        type="radio"

```

```

        name="opcao_rh"
        id="NEGATIVO"
        value="- "
    />
    <label htmlFor="NEGATIVO">-</label>
    <br />
</div>
</div>
<div className={estilo.div_botoes}>
    <button className={estilo.botaoCadastrar}
id="cadastrar" type="submit">Cadastrar</button>
    <button className={estilo.botaoLimpar}
id="limpar" type="reset">Limpar</button>
</div>
</form>
</div>
</body>
)
}

export default CadastroDoador;

```

E o CSS desta página fica assim (src\styles\CadastroDoador.module.css):

```

@import url('../colors/cores.css');

.body{

    display: flex;
    justify-content: center;

}

.area_cadastro {

    margin-top: 20vh;
    background-color: var(--azulpastel);

}

```

```
.formulario {  
  
    background-color: var(--branco);  
    border: 1px solid;  
    border-radius: 10%;  
    width: 350px;  
  
}  
  
.titulo_formulario {  
  
    margin: 2vh;  
    margin-left: 60px;  
  
}  
  
.labelFormulario {  
  
    margin-left: 1vw;  
  
}  
  
.inputFormulario {  
  
    border: 1px solid;  
    border-radius: 1vw;  
    padding-left: 10px;  
  
}  
  
#contato {  
  
    width: 25vh;  
  
}  
  
#cpf {  
  
    width: 25vh;  
  
}
```

```
#nome {  
  
    width: 35vh;  
  
}  
  
.opcao {  
  
    display: flex;  
    margin-bottom: 1vw;  
  
}  
  
.opcao_tipo{  
  
    margin-left: 1vw;  
  
}  
.opcao_rh {  
  
    margin-left: 5vw;  
  
}  
  
.div_botoes {  
  
    text-align: center;  
    margin-bottom: 2vh;  
  
}  
  
.botaoCadastrar{  
  
    width: 7vw;  
    height: 5vh;  
    color: var(--branco);  
    background-color: var(--verde);  
    border: 0px;  
    border-radius: 10px;  
  
}
```

```

.botaoLimpar{

    width: 7vw;
    height: 5vh;
    color: var(--branco);
    background-color: var(--vermelho);
    border: 0px;
    border-radius: 10px;
    margin-left: 10px;

}

.botaoCadastrar: hover, .botaoLimpar: hover{

    opacity: 0.7;

}

```

Para fazermos as buscas dos doadores a tela fica da seguinte forma em src/pages/Buscador.tsx:

```

import React, { useState, ChangeEvent, FormEvent } from "react";
import { Link } from "react-router-dom";
import "../global.d.ts"
// import estilo from "../styles/BuscaDoador.module.css";

interface Doador {
    nome: string;
    cpf: string;
    contato: string;
    tipoSanguineo: string;
    fatorRh: string;
    codigo: number;
}

function BuscaDoador() {
    const initialState = {
        nome: "",
        cpf: "",
        contato: "",
        tipoSanguineo: "",
        fatorRh: ""
    };
};

```

```

const [doadores, setDoadores] = useState<Doador[]>([]);
const [searchParams, setSearchParams] = useState(initialState);
const [loading, setLoading] = useState(false);
const [noResults, setNoResults] = useState(false);
const [isEditing, setIsEditing] = useState<{ [key: number]: boolean }>({});
const [updating, setUpdating] = useState<{ [key: number]: boolean }>({});

const handleInputChange = (event: ChangeEvent<HTMLInputElement>) => {
  const { name, value } = event.target;
  setSearchParams(prevState => ({
    ...prevState,
    [name]: value
  }));
};

const handleSearch = async (event: FormEvent<HTMLFormElement>) => {
  event.preventDefault();
  setLoading(true);
  setNoResults(false);

  try {
    const response = await
fetch('http://localhost:5000/api/getOneDoador', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(searchParams)
    });

    if (response.ok) {
      const data: Doador[] = await response.json();
      setDoadores(data);
      if (data.length === 0) {
        setNoResults(true);
      }
    } else {
      console.error('Falha ao buscar doadores.');
```

```

    const handleDelete = async (codigo: number) => {
      try {
        const response = await
fetch('http://localhost:5000/api/deleteDoador', {
          method: 'POST',
          headers: {
            'Content-Type': 'application/json'
          },
          body: JSON.stringify({ codigo })
        });

        if (response.ok) {
          setDoadores(doadores.filter(doador => doador.codigo !==
codigo));
        } else {
          console.error('Falha ao excluir doador.');
```

```

        'Content-Type': 'application/json'
      },
      body: JSON.stringify(updatedDoador)
    });

    if (response.ok) {
      setIsEditing(prevState => ({ ...prevState, [codigo]: false
    })));

    } else {
      console.error('Falha ao atualizar doador.');
```

```

    }
  } catch (error) {
    console.error('Erro ao fazer a requisição:', error);
  } finally {
    setUpdating(prevState => ({ ...prevState, [codigo]: false }));
  }
};
```

```

const handleClear = () => {
  setSearchParams(initialState);
};
```

```

return (
  <div className={estilo.container}>
    <div className={estilo.searchContainer}>
      <form className={estilo.form} onSubmit={handleSearch}
onReset={handleClear}>
        <h1 className={estilo.title}>Busca de doador</h1>
        <div className={estilo.inputGroup}>
          <input
            type="text"
            name="nome"
            placeholder="Nome"
            value={searchParams.nome}
            onChange={handleInputChange}
          />
          <input
            type="text"
            name="cpf"
            placeholder="CPF"
            value={searchParams.cpf}
            onChange={handleInputChange}
          />
          <input
            type="tel"
            name="contato"
            placeholder="Contato"
            value={searchParams.contato}
          />
        </div>
      </form>
    </div>
  </div>
);
```



```

        onChange={handleInputChange}
      />
    </div>
    <div className={estilo.radioGroup}>
      <input
        type="radio"
        name="tipoSanguineo"
        id="A"
        value="A"
        checked={searchParams.tipoSanguineo === 'A'}
        onChange={handleInputChange}
      />
      <label htmlFor="A">A</label>
      <input
        type="radio"
        name="tipoSanguineo"
        id="B"
        value="B"
        checked={searchParams.tipoSanguineo === 'B'}
        onChange={handleInputChange}
      />
      <label htmlFor="B">B</label>
      <input
        type="radio"
        name="tipoSanguineo"
        id="AB"
        value="AB"
        checked={searchParams.tipoSanguineo === 'AB'}
        onChange={handleInputChange}
      />
      <label htmlFor="AB">AB</label>
      <input
        type="radio"
        name="tipoSanguineo"
        id="O"
        value="O"
        checked={searchParams.tipoSanguineo === 'O'}
        onChange={handleInputChange}
      />
      <label htmlFor="O">O</label>
    </div>
    <div className={estilo.radioGroup}>
      <input
        type="radio"
        name="fatorRh"
        id="positivo"
        value="+"
        checked={searchParams.fatorRh === '+'}

```

```

        onChange={handleInputChange}
      />
      <label htmlFor="positivo">+</label>
      <input
        type="radio"
        name="fatorRh"
        id="negativo"
        value="-"
        checked={searchParams.fatorRh === '-'}
        onChange={handleInputChange}
      />
      <label htmlFor="negativo">-</label>
    </div>
    <div className={estilo.div_botoes}>
      <button type="submit"
        className={estilo.searchButton} disabled={loading}>
        {loading ? 'Buscando...' : 'Buscar'}
      </button>
      <button type="reset"
        className={estilo.clearButton}>Limpar</button>
    </div>
  </form>
</div>
<div className={estilo.resultsContainer}>
  <h1>Resultados da busca:</h1>
  {noResults ? <p>Nenhum resultado encontrado.</p> : (
    <table>
      <thead>
        <tr>
          <th>Nome</th>
          <th>CPF</th>
          <th>Contato</th>
          <th>Tipo Sanguíneo</th>
          <th>Fator RH</th>
          <th>Botão Editar</th>
          <th>Botão Doação</th>
          <th>Botão Excluir</th>
        </tr>
      </thead>
      <tbody>
        {doadores.map((doador, index) => (
          <tr key={index}>
            <td>
              {isEditing[doador.codigo] ? (
                <input
                  type="text"
                  name="nome"
                  value={doador.nome}

```

```

                                onChange={e =>
handleUpdateChange(doador.codigo, e)}
                                />
                                ) : (
                                doador.nome
                                )}
                                </td>
                                <td>{doador.cpf}</td>
                                <td>
                                {isEditing[doador.codigo] ? (
                                <input
                                type="text"
                                name="contato"
                                value={doador.contato}
                                onChange={e =>
handleUpdateChange(doador.codigo, e)}
                                />
                                ) : (
                                doador.contato
                                )}
                                </td>
                                <td>{doador.tipoSanguineo}</td>
                                <td>{doador.fatorRh}</td>
                                <td>
                                {isEditing[doador.codigo] ? (
                                <button
                                onClick={() =>
handleUpdate(doador.codigo)}
                                disabled={updating[doador.codigo]}
                                >
                                {updating[doador.codigo] ?
'Atualizando...' : 'Salvar'}
                                </button>
                                ) : (
                                <button onClick={() =>
handleEdit(doador.codigo)}>Editar</button>
                                )}
                                </td>
                                <td>
                                <Link
                                to={` /doacao/${doador.codigo}`}>Doar</Link>
                                </td>
                                <td>
                                <button onClick={() =>
handleDelete(doador.codigo)}>Excluir</button>
                                </td>
                                </tr>

```

```
        ))}  
        </tbody>  
    </table>  
    }  
    </div>  
</div>  
);  
}  
  
export default BuscaDoador;
```