

Requisitos para Rodar a Aplicação

A aplicação foi desenvolvida utilizando Spring Boot para o back-end, React.js para o front-end e MySQL como banco de dados (via XAMPP).

Para executar corretamente a aplicação, é necessário garantir que os seguintes requisitos estejam atendidos:

- Java (Versão 11 ou superior)
- Maven (Para gerenciar e criar o back-end)
- Node.js e npm/yarn (Versão 14 ou superior)
- XAMPP (Para fornecer o servidor MySQL)

Códigos

React.js:

Conexão com a API:

```
1  import axios from 'axios';
2
3  const api = axios.create({
4    baseURL: 'http://localhost:8080/api',
5    headers: {
6      'Content-Type': 'application/json'
7    }
8  });
9
10 export const carrosService = {
11   getAll: () => api.get('/carros'),
12   getById: (id) => api.get(`/carros/${id}`),
13   create: (data) => api.post('/carros', data),
14   update: (id, data) => api.put(`/carros/${id}`, data),
15   delete: (id) => api.delete(`/carros/${id}`),
16   getDisponiveis: () => api.get('/carros/disponiveis')
17 };
18
19 export const marcasService = {
20   getAll: () => api.get('/marcas'),
21   create: (data) => api.post('/marcas', data),
22   update: (id, data) => api.put(`/marcas/${id}`, data),
23   delete: (id) => api.delete(`/marcas/${id}`)
24 };
25
26 export const modelosService = {
27   getAll: () => api.get('/modelos'),
28   create: async (data) => {
29     try {
30       console.log('Payload enviado para criar modelo:', data);
31       const response = await api.post('/modelos', data);
32       console.log('Resposta bem sucedida:', response.data);
33       return response;
34     } catch (error) {
35       console.error('Detalhes do erro:', {
36         status: error.response?.status,
37         data: error.response?.data,
38         message: error.message,
39         payload: data
40       });
41       throw error;
42     }
43   }
44 };
```

Página adicionar marca:

```

1  import React, { useState } from 'react';
2  import styled from 'styled-components';
3  import { marcasService } from '../api/api';
4
5  const Form = styled.form`
6    display: flex;
7    flex-direction: column;
8    gap: 1rem;
9    max-width: 500px;
10   margin: 0 auto;
11 `;
12
13  const Input = styled.input`
14    padding: 0.5rem;
15    border-radius: 4px;
16 `;
17
18  const Button = styled.button`
19    padding: 0.5rem;
20    background: #333;
21    color: white;
22    border: none;
23    cursor: pointer;
24    &:hover {
25      background: #444;
26    }
27    border-radius: 4px;
28 `;
29
30  const ErrorMessage = styled.div`
31    color: red;
32    font-size: 0.875rem;
33 `;
34
35  const SuccessMessage = styled.div`
36    color: green;
37    font-size: 0.875rem;
38 `;
39
40  export default function BrandForm() {
41    const [nome, setNome] = useState('');
42    const [error, setError] = useState('');
43    const [success, setSuccess] = useState('');
44    const [loading, setLoading] = useState(false);
45
46    const handleSubmit = async (e) => {
47      e.preventDefault();
48      setError('');
49      ...

```

```

47     e.preventDefault();
48     setError('');
49     setSuccess('');
50
51     if (!nome.trim()) {
52         setError('Por favor, insira um nome para a marca');
53         return;
54     }
55
56     setLoading(true);
57     try {
58         await marcasService.create({ nome: nome.trim() });
59         setSuccess('Marca adicionada com sucesso!');
60         setNome(''); // Limpa o campo de entrada
61     } catch (erro) {
62         console.error('Erro ao adicionar a marca:', erro);
63         setError(erro.response?.data?.message || 'Erro ao adicionar a marca. Tente novamente.');

```

Página adicionar carro:

```
1 import React, { useState, useEffect } from 'react';
2 import styled from 'styled-components';
3 import { carrosService, marcasService, modelosService } from '../api/api';
4
5 const Form = styled.form`
6   display: flex;
7   flex-direction: column;
8   gap: 1rem;
9   max-width: 500px;
10  margin: 0 auto;
11  font-family: "Roboto", serif;
12  font-weight: 400;
13 `;
14
15 const Input = styled.input`
16   padding: 0.5rem;
17   border-radius: 4px;
18   border: 1px solid #ccc;
19 `;
20
21 const Select = styled.select`
22   padding: 0.5rem;
23   border-radius: 4px;
24   border: 1px solid #ccc;
25 `;
26
27 const Button = styled.button`
28   padding: 0.8rem;
29   background: #333;
30   color: white;
31   border: none;
32   cursor: pointer;
33   border-radius: 4px;
34   &:hover {
35     background: #444;
36   }
37 `;
38
39 const Label = styled.label`
40   font-size: 14px;
41 `;
42
43 const Message = styled.div`
44   padding: 1rem;
45   border-radius: 4px;
46   margin-bottom: 1rem;
47   text-align: center;
48   font-weight: bold;
```

```

48   font-weight: bold;
49   background-color: ${({ type }) => (type === 'success' ? '#d4edda' : '#f8d7da')};
50   color: ${({ type }) => (type === 'success' ? '#155724' : '#721c24')};
51   border: 1px solid ${({ type }) => (type === 'success' ? '#c3e6cb' : '#f5c6cb')};
52 `;
53
54 export default function CarForm() {
55   const [brands, setBrands] = useState([]);
56   const [models, setModels] = useState([]);
57   const [loading, setLoading] = useState(false);
58   const [formData, setFormData] = useState({
59     marca: '',
60     modelo: '',
61     ano: '',
62     preco: '',
63     cor: '',
64     quilometragem: '0',
65     urlImagem: '',
66     disponivel: true
67   });
68
69   useEffect(() => {
70     loadBrandsAndModels();
71   }, []);
72
73   const loadBrandsAndModels = async () => {
74     try {
75       const [brandsResponse, modelsResponse] = await Promise.all([
76         marcasService.getAll(),
77         modelosService.getAll()
78       ]);
79
80       console.log('Marcas disponiveis:', brandsResponse.data);
81       console.log('Modelos disponiveis:', modelsResponse.data);
82
83       setBrands(brandsResponse.data);
84       setModels(modelsResponse.data);
85     } catch (error) {
86       console.error('Erro ao carregar dados iniciais:', error);
87       alert('Falha ao carregar marcas e modelos. Por favor, recarregue a página.');

```

```

94         id: parseInt(data.modelo)
95     },
96     anoFabricacao: parseInt(data.ano),
97     cor: data.cor.trim(),
98     preco: parseFloat(data.preco),
99     quilometragem: parseInt(data.kilometragem) || 0,
100     statusDisponibilidade: data.disponivel ? 'Disponivel' : 'Indisponivel'
101 };
102
103 if (data.urlImagem && data.urlImagem.trim()) {
104     formattedData.urlImagem = data.urlImagem.trim();
105 }
106
107 console.log('Dados formatados para envio:', formattedData);
108 return formattedData;
109 };
110
111 const handleSubmit = async (e) => {
112     e.preventDefault();
113
114     try {
115         setLoading(true);
116
117         if (!formData.modelo || !formData.ano || !formData.cor || !formData.preco) {
118             alert('Por favor, preencha todos os campos obrigatórios');
119             return;
120         }
121
122         const carData = formatCarData(formData);
123
124         console.log('Tentando criar carro com os dados:', {
125             modeloSelecioneado: models.find(m => m.id === parseInt(formData.modelo))?.nome,
126             ...carData
127         });
128
129         const response = await carrosService.create(carData);
130         console.log('Resposta do servidor:', response.data);
131
132         setFormData({
133             marca: '',
134             modelo: '',
135             ano: '',
136             preco: '',

```

```

136         preco: '',
137         cor: '',
138         quilometragem: '0',
139         urlImagem: '',
140         disponivel: true
141     });
142
143     alert('Carro adicionado com sucesso!');
144     } catch (error) {
145         console.error('Erro detalhado:', {
146             message: error.message,
147             status: error.response?.status,
148             statusText: error.response?.statusText,
149             data: error.response?.data
150         });
151
152         alert('Erro ao adicionar carro. Por favor, verifique os dados e tente novamente.');
```

```

153     } finally {
154         setLoading(false);
155     }
156 };
157
```

```

158 const filteredModels = formData.marca
159   ? models.filter(model => model.marca.id === parseInt(formData.marca))
160   : models;
161
```

```

162 return (
163     <Form onSubmit={handleSubmit}>
164         <Input
165             type="text"
166             placeholder="URL imagem"
167             value={formData.urlImagem}
168             onChange={e => setFormData({ ...formData, urlImagem: e.target.value })}
169         />
170         <Select
171             value={formData.marca}
172             onChange={e => setFormData({ ...formData, marca: e.target.value, modelo: '' })}
173             required
174             disabled={loading}
175         >
176             <option value="">Selecione a Marca do Carro</option>
177             {brands.map(brand => (
178                 <option key={brand.id} value={brand.id}>
179                     {brand.nome}
180                 </option>
181             ))}
182         </Select>
183     </Form>
184 );
185
```

```

179         {brand.nome}
180       </option>
181     ))}
182   </Select>
183
184   <Select
185     value={formData.modelo}
186     onChange={e => setFormData({ ...formData, modelo: e.target.value })}
187     required
188     disabled={loading || !formData.marca}
189   >
190     <option value="">Selecione o Modelo do Carro</option>
191     {filteredModels.map(model => (
192       <option key={model.id} value={model.id}>
193         {model.nome}
194       </option>
195     ))}
196   </Select>
197
198   <Input
199     type="number"
200     placeholder="Ano de fabricação"
201     value={formData.ano}
202     onChange={e => setFormData({ ...formData, ano: e.target.value })}
203     required
204     min="1900"
205     max={new Date().getFullYear() + 1}
206     disabled={loading}
207   />
208   <Input
209     type="number"
210     step="0.01"
211     placeholder="Preço"
212     value={formData.preco}
213     onChange={e => setFormData({ ...formData, preco: e.target.value })}
214     required
215     min="0"
216     disabled={loading}
217   />
218   <Input
219     type="text"
220     placeholder="Cor"
221     value={formData.cor}
222     onChange={e => setFormData({ ...formData, cor: e.target.value })}
223     required
224     disabled={loading}

```



```

223         required
224         disabled={loading}
225     />
226     <Input
227       type="number"
228       placeholder="Quilometragem"
229       value={formData.quilometragem}
230       onChange={e => setFormData({ ...formData, quilometragem: e.target.value })}
231       min="0"
232       disabled={loading}
233     />
234     <Label>
235       <input
236         type="checkbox"
237         checked={formData.disponivel}
238         onChange={e => setFormData({ ...formData, disponivel: e.target.checked })}
239         disabled={loading}
240       />
241       Disponivel
242     </Label>
243
244     <Button type="submit" disabled={loading}>
245       {loading ? 'Adicionando...' : 'Adicionar Carro'}
246     </Button>
247   </Form>
248 );
249 }
250

```

Página adicionar modelo:

```

1 import React, { useState, useEffect } from 'react';
2 import styled from 'styled-components';
3 import { modelosService, marcasService } from '../api/api';
4
5 const Form = styled.form`
6   display: flex;
7   flex-direction: column;
8   gap: 1rem;
9   max-width: 500px;
10  margin: 0 auto;
11 `;
12
13 const Input = styled.input`
14   padding: 0.5rem;
15   border-radius: 4px;
16 `;
17
18 const Select = styled.select`
19   padding: 0.5rem;
20   border-radius: 4px;
21 `;
22
23 const Button = styled.button`
24   padding: 0.5rem;
25   background: #333;
26   color: white;
27   border: none;
28   cursor: pointer;
29   &:hover { background: #444; }
30   border-radius: 4px;
31 `;
32
33 const ErrorMessage = styled.div`
34   color: red;
35   font-size: 0.875rem;
36 `;
37
38 const SuccessMessage = styled.div`
39   color: green;
40   font-size: 0.875rem;
41 `;
42
43 export default function ModelForm() {
44   const [formData, setFormData] = useState({
45     nome: '',
46     id_marca: ''
47   });
48   const [marcas, setMarcas] = useState([]);

```

```

48     const [marcas, setMarcas] = useState([]);
49     const [error, setError] = useState('');
50     const [success, setSuccess] = useState('');
51     const [loading, setLoading] = useState(false);
52
53     useEffect(() => {
54         const carregarMarcas = async () => {
55             try {
56                 const response = await marcasService.getAll();
57                 setMarcas(response.data);
58             } catch (erro) {
59                 console.error('Erro ao carregar marcas:', erro);
60                 setError('Erro ao carregar marcas');
61             }
62         };
63
64         carregarMarcas();
65     }, []);
66
67     const handleChange = (e) => {
68         const { name, value } = e.target;
69         setFormData(prev => ({
70             ...prev,
71             [name]: value
72         }));
73     };
74
75     const handleSubmit = async (e) => {
76         e.preventDefault();
77         setError('');
78         setSuccess('');
79
80         try {
81             const modeloData = {
82                 nome: formData.nome.trim(),
83                 marca: {
84                     id: parseInt(formData.id_marca, 10)
85                 }
86             };
87
88             console.log('Tentando criar modelo com dados:', modeloData);
89
90             const response = await modelosService.create(modeloData);
91             console.log('Resposta do servidor:', response.data);
92
93             setSuccess('Modelo adicionado com sucesso!');
94             setFormData({

```

```

94         setFormData({
95             nome: '',
96             id_marca: ''
97         });
98     } catch (erro) {
99         console.error('Erro ao criar modelo:', {
100             status: erro.response?.status,
101             data: erro.response?.data,
102             message: erro.message
103         });
104
105         let mensagemErro = 'Erro ao adicionar modelo.';
106         if (erro.response?.data?.message) {
107             mensagemErro += ` Detalhe: ${erro.response.data.message}`;
108         }
109         setError(mensagemErro);
110     }
111 };
112
113 return (
114     <Form onSubmit={handleSubmit}>
115         <Input
116             name="nome"
117             placeholder="Nome do Modelo"
118             value={formData.nome}
119             onChange={handleChange}
120             disabled={loading}
121         />
122
123         <Select
124             name="id_marca"
125             value={formData.id_marca}
126             onChange={handleChange}
127             disabled={loading}
128         >
129             <option value="">Selecione uma marca</option>
130             {marcas.map(marca => (
131                 <option key={marca.id} value={marca.id}>
132                     {marca.nome}
133                 </option>
134             ))}
135         </Select>
136
137         {error && <ErrorMessage>{error}</ErrorMessage>

```

```

137         {success && <SuccessMessage>{success}</SuccessMessage>}
138
139         <Button type="submit" disabled={loading}>
140             {loading ? 'Adicionando...' : 'Adicionar Modelo'}
141         </Button>
142     </Form>
143 );
144
145 }

```

Navbar:

```

1  import React from 'react';
2  import styled from 'styled-components';
3  import { Link } from 'react-router-dom';
4
5  const Nav = styled.nav`
6    background: #333;
7    padding: 1rem;
8    margin-bottom: 2rem;
9    border-radius: 6px;
10 `;
11
12 const NavLink = styled(Link)`
13   color: white;
14   text-decoration: none;
15   margin-right: 1rem;
16   &:hover { color: #ddd; }
17 `;
18
19 export default function Navbar() {
20   return (
21     <Nav>
22       <NavLink to="/">Carros</NavLink>
23       <NavLink to="/">|</NavLink>
24       <NavLink to="/add-car">Adicionar Carro</NavLink>
25       <NavLink to="/">|</NavLink>
26       <NavLink to="/add-model">Adicionar Modelo</NavLink>
27       <NavLink to="/">|</NavLink>
28       <NavLink to="/add-brand">Adicionar Marca</NavLink>
29     </Nav>
30   );
31 }

```

Página menu de carros:

```

1  import React, { useState, useEffect } from 'react';
2  import styled from 'styled-components';
3  import { carrosService } from '../api/api';
4
5  const BarraPesquisa = styled.input`
6    width: 98%;
7    padding: 0.5rem;
8    margin-bottom: 1rem;
9    border-radius: 6px;
10 `;
11
12 const FiltroSelect = styled.select`
13   padding: 0.5rem;
14   margin-bottom: 1rem;
15   margin-right: 1rem;
16   border-radius: 6px;
17 `;
18
19 const GradeCarros = styled.div`
20   display: grid;
21   grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
22   gap: 1rem;
23 `;
24
25 const CartaoCarro = styled.div`
26   border: 1px solid #ddd;
27   padding: 1rem;
28   border-radius: 4px;
29   cursor: pointer;
30 `;
31
32 const Titulo = styled.h3`
33   color: #ab0201;
34 `;
35
36 const Imagem = styled.img`
37   width: 100%;
38   height: 40%;
39 `;
40
41 const Modal = styled.div`
42   position: fixed;
43   top: 0;
44   left: 0;
45   right: 0;
46   bottom: 0;
47   background-color: rgba(0, 0, 0, 0.5);
48   display: ${props => (props.show ? 'flex' : 'none')};

```

```

47   background-color: rgba(0, 0, 0, 0.5);
48   display: ${props => (props.show ? 'flex' : 'none')};
49   justify-content: center;
50   align-items: center;
51 `;
52
53 const ModalConteudo = styled.div`
54   background-color: white;
55   padding: 2rem;
56   border-radius: 6px;
57   width: 400px;
58   position: relative;
59 `;
60
61 const BotaoFechar = styled.button`
62   position: absolute;
63   top: 10px;
64   right: 10px;
65   background: transparent;
66   border: none;
67   font-size: 1.5rem;
68   color: #ab0201;
69   cursor: pointer;
70 `;
71
72 const ModalFormulario = styled.div`
73   display: flex;
74   flex-direction: column;
75   gap: 1rem;
76 `;
77
78 const Input = styled.input`
79   padding: 0.5rem;
80   border-radius: 6px;
81   border: 1px solid #ddd;
82 `;
83
84 const Label = styled.label`
85   font-size: 14px;
86 `;
87
88 const Button = styled.button`
89   padding: 0.5rem;
90   background: #333;
91   color: white;
92   border: none;
93   cursor: pointer;

```

```

93     cursor: pointer;
94     &:hover { background: #444; }
95     border-radius: 4px;
96 `;
97
98 const Icone = styled.i`
99     position: absolute;
100     bottom: 10px;
101     right: 10px;
102     font-size: 1.2rem;
103     color: #ab0201;
104     cursor: pointer;
105 `;
106 export default function ListaCarros() {
107     const [carros, setCarros] = useState([]);
108     const [pesquisa, setPesquisa] = useState('');
109     const [filtroAno, setFiltroAno] = useState('');
110     const [filtroMarca, setFiltroMarca] = useState('');
111     const [filtroModelo, setFiltroModelo] = useState('');
112     const [filtroDisponibilidade, setFiltroDisponibilidade] = useState('todos');
113     const [carroSelecioneado, setCarroSelecioneado] = useState(null);
114     const [mostrarModal, setMostrarModal] = useState(false);
115     const [modoEdicao, setModoEdicao] = useState(false);
116
117     useEffect(() => {
118         async function buscarCarros() {
119             try {
120                 const resposta = await carrosService.getAll();
121                 console.log('Estrutura completa dos carros:', resposta.data);
122                 setCarros(resposta.data || []);
123             } catch (erro) {
124                 console.error('Erro ao buscar carros:', erro);
125                 setCarros([]);
126             }
127         }
128
129         buscarCarros();
130     }, []);
131
132     const carrosFiltrados = carros
133         .filter((carro) => {
134             const correspondePesquisa = pesquisa
135                 ? carro.modelo?.nome.toLowerCase().includes(pesquisa.toLowerCase()) ||
136                   carro.modelo?.marca?.nome.toLowerCase().includes(pesquisa.toLowerCase()) ||
137                   carro.anoFabricacao.toString().includes(pesquisa)
138               : true;
139

```



```

139
140     const correspondeAno = filtroAno ? carro.anoFabricacao.toString() === filtroAno : true;
141     const correspondeMarca = filtroMarca ? carro.modelo?.marca?.nome === filtroMarca : true;
142     const correspondeModelo = filtroModelo ? carro.modelo?.nome === filtroModelo : true;
143     const correspondeDisponibilidade =
144         filtroDisponibilidade === 'todos' ||
145         filtroDisponibilidade === carro.statusDisponibilidade;
146
147     return (
148         correspondePesquisa &&
149         correspondeAno &&
150         correspondeMarca &&
151         correspondeModelo &&
152         correspondeDisponibilidade
153     );
154 })
155 .sort((a, b) => {
156     if (a.modelo?.nome < b.modelo?.nome) return -1;
157     if (a.modelo?.nome > b.modelo?.nome) return 1;
158     return 0;
159 });
160
161 const abrirModal = (carro) => {
162     setCarroSelecionado(carro);
163     setModoEdicao(false);
164     setMostrarModal(true);
165 };
166
167 const fecharModal = () => {
168     setMostrarModal(false);
169     setCarroSelecionado(null);
170 };
171
172 const editarCarro = () => {
173     setModoEdicao(true);
174 };
175
176 const atualizarCarro = async () => {
177     try {
178         await carrosService.update(carroSelecionado.id, carroSelecionado);
179
180         setCarros((carrosAnteriores) =>
181             carrosAnteriores.map((carro) =>

```

```

180         setCarros((carrosAnteriores) =>
181             carrosAnteriores.map((carro) =>
182                 carro.id === carroSelecionado.id ? carroSelecionado : carro
183             )
184         );
185
186         setModoEdicao(false);
187         setMostrarModal(false);
188         alert('Carro atualizado com sucesso!');
189     } catch (erro) {
190         console.error('Erro ao atualizar carro:', erro);
191         alert('Não foi possível atualizar o carro. Tente novamente mais tarde.');
```

```

224         </option>
225     ))
226     ): (
227         <option disabled>Carregando marcas...</option>
228     )}
229 </FiltroSelect>
230
231 <FiltroSelect value={filtroModelo} onChange={e => setFiltroModelo(e.target.value)}>
232     <option value="">Filtrar por Modelo</option>
233     {carros.length > 0 ? (
234         [...new Set(carros.map(carro => carro.modelo?.nome).filter(Boolean))]
235         .sort()
236         .map(modelo => (
237             <option key={modelo} value={modelo}>
238                 {modelo}
239             </option>
240         ))
241     ): (
242         <option disabled>Carregando modelos...</option>
243     )}
244 </FiltroSelect>
245
246 <FiltroSelect value={filtroDisponibilidade} onChange={e => setFiltroDisponibilidade(e.target.value)}>
247     <option value="todos">Todos os carros</option>
248     <option value="Disponivel">Disponiveis</option>
249     <option value="Indisponivel">Indisponiveis</option>
250 </FiltroSelect>
251
252 <GradeCarros>
253     {carrosFiltrados.length > 0 ? (
254         carrosFiltrados.map(carro => (
255             <CartaoCarro key={carro.id} onClick={() => abrirModal(carro)}>
256                 {console.log('Dados do carro:', carro)}
257                 <Imagem src={carro.urlImagem} />
258                 <Titulo>{carro.modelo?.nome || 'N/A'}</Titulo>
259                 <p>{carro.modelo?.marca?.nome || 'N/A'}</p>
260                 <p><b>Ano:</b> {carro.anoFabricacao || 'N/A'}</p>
261                 <p><b>Preço:</b> R$ {carro.preco || 'N/A'},00</p>
262                 <p><b>Cor:</b> {carro.cor || 'N/A'}</p>
263                 <p><b>Quilometragem:</b> {carro.quilometragem || 'N/A'}</p>
264                 <p><b>Disponibilidade:</b> {carro.statusDisponibilidade || 'N/A'}</p>
265             </CartaoCarro>
266         ))
267     ): (
268         <p>Não há carros disponiveis</p>
269     )}

```

```

268         <p>Não há carros disponíveis</p>
269     })
270 </GradeCarros>
271
272 <Modal show={mostrarModal}>
273     <ModalConteudo>
274         <BotaoFechar onClick={fecharModal}>x</BotaoFechar>
275         <h2>{modoEdicao ? 'Editar Carro' : 'Detalhes do Carro'}</h2>
276         {!modoEdicao ? (
277             <>
278                 <Imagem src={carroSelecioneado?.urlImagem} />
279                 <Titulo>{carroSelecioneado?.modelo?.nome || 'N/A'}</Titulo>
280                 <p>{carroSelecioneado?.modelo?.marca?.nome || 'N/A'}</p>
281                 <p><b>Ano:</b> {carroSelecioneado?.anoFabricacao || 'N/A'}</p>
282                 <p><b>Preço:</b> R$ {carroSelecioneado?.preco || 'N/A'},00</p>
283                 <p><b>Cor:</b> {carroSelecioneado?.cor || 'N/A'}</p>
284                 <p><b>Quilometragem:</b> {carroSelecioneado?.quilometragem || 'N/A'}</p>
285                 <p><b>Disponibilidade:</b> {carroSelecioneado?.statusDisponibilidade || 'N/A'}</p>
286                 <Icone><i class="fi fi-rr-edit" onClick={editarCarro}></i></Icone>
287             </>
288         ) : (
289             <ModalFormulario>
290                 <Label>Preço:</Label>
291                 <Input
292                     value={carroSelecioneado?.preco || ''}
293                     onChange={(e) => setCarroSelecioneado({ ...carroSelecioneado, preco: e.target.value })}
294                     placeholder="Preço"
295                 />
296                 <Label>Ano de Fabricação:</Label>
297                 <Input
298                     value={carroSelecioneado?.anoFabricacao || ''}
299                     onChange={(e) => setCarroSelecioneado({ ...carroSelecioneado, anoFabricacao: e.target.value })}
300                     placeholder="Ano"
301                 />
302                 <Label>Cor:</Label>
303                 <Input
304                     value={carroSelecioneado?.cor || ''}
305                     onChange={(e) => setCarroSelecioneado({ ...carroSelecioneado, cor: e.target.value })}
306                     placeholder="Cor"
307                 />
308                 <Label>Quilometragem:</Label>
309                 <Input
310                     value={carroSelecioneado?.quilometragem || ''}
311                     onChange={(e) => setCarroSelecioneado({ ...carroSelecioneado, quilometragem: e.target.value })}
312                     placeholder="Quilometragem"
313                 />
314                 <Label>Status de Disponibilidade:</Label>

```

```

314                 <Label>Status de Disponibilidade:</Label>
315                 <FiltroSelect
316                     value={carroSelecioneado?.statusDisponibilidade || ''}
317                     onChange={(e) =>
318                         setCarroSelecioneado({
319                             ...carroSelecioneado,
320                             statusDisponibilidade: e.target.value,
321                         })
322                     }
323                 >
324                     <option value="Disponivel">Disponivel</option>
325                     <option value="Indisponivel">Indisponivel</option>
326                 </FiltroSelect>
327
328                 <Button onClick={atualizarCarro}>Salvar</Button>
329             </ModalFormulario>
330         )}
331     </ModalConteudo>
332 </Modal>
333 </>
334 );
335 }
336

```

Página app:

```

1  import React from 'react';
2  import styled from 'styled-components';
3  import { BrowserRouter, Routes, Route } from 'react-router-dom';
4  import Navbar from './components/navbar';
5  import CarList from './components/CarList';
6  import CarForm from './components/CarForm';
7  import ModelForm from './components/ModelForm';
8  import BrandForm from './components/BrandForm';
9
10
11  const AppContainer = styled.div`
12    max-width: 1200px;
13    margin: 0 auto;
14    padding: 20px;
15    font-family: "Roboto", serif;
16    font-weight: 400;
17  `;
18
19  export default function App() {
20    return (
21      <BrowserRouter>
22        <AppContainer>
23          <Navbar />
24          <Routes>
25            <Route path="/" element={<CarList />} />
26            <Route path="/add-car" element={<CarForm />} />
27            <Route path="/add-model" element={<ModelForm />} />
28            <Route path="/add-brand" element={<BrandForm />} />
29          </Routes>
26        </AppContainer>
27      </BrowserRouter>
28    );
29  }
30
31
32
33
34

```

Página index.js:

```

1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import App from './App';
4
5  const root = ReactDOM.createRoot(document.getElementById('root'));
6  root.render(
7    <React.StrictMode>
8      <App />
9    </React.StrictMode>
10  );
11

```

Página index.html:

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3    <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1" />
6      <link rel="preconnect" href="https://fonts.googleapis.com">
7      <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
8      <link href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap" rel="stylesheet">
9      <link rel="stylesheet" href="https://cdn-uicons.flaticon.com/2.6.0/uicons-regular-rounded/css/uicons-regular-rounded.css">
10     <title>Sistema Carros</title>
11   </head>
12   <body>
13     <div id="root"></div>
14   </body>
15 </html>

```

SpringBoot:

Conexão com banco:

```
1 spring.datasource.url=jdbc:mysql://localhost:3306/sistematicarros
2 spring.datasource.username=root
3 spring.datasource.password=
4 spring.jpa.hibernate.ddl-auto=update
5 spring.jpa.show-sql=true
```

ModeloRepository:

```
1 package com.carros.repositories;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 @Repository
6 public interface ModeloRepository extends JpaRepository<Modelo, Long> {
7 }
8
```

MarcaRepository:

```
1 package com.carros.repositories;
2
3 import com.carros.models.Marca;
4
5 public interface MarcaRepository extends JpaRepository<Marca, Long> {
6 }
7
```

CarroRepository:

```
1 package com.carros.repositories;
2
3 import com.carros.models.Carro;
4
5 public interface CarroRepository extends JpaRepository<Carro, Long> {
6     List<Carro> findByStatusDisponibilidade(StatusDisponibilidade statusDisponibilidade);
7 }
8
```

Classe modelo:

```

1 package com.carros.models;
2
3 import jakarta.persistence.*;
4
5 @Entity
6 @Table(name = "modelo")
7 public class Modelo {
8     @Id
9     @GeneratedValue(strategy = GenerationType.IDENTITY)
10    private Long id;
11
12    @Column(nullable = false)
13    private String nome;
14
15    @ManyToOne
16    @JoinColumn(name = "id_marca", nullable = false)
17    private Marca marca;
18
19    public Long getId() {
20        return id;
21    }
22
23    public void setId(Long id) {
24        this.id = id;
25    }
26
27    public String getNome() {
28        return nome;
29    }
30
31    public void setNome(String nome) {
32        this.nome = nome;
33    }
34
35    public Marca getMarca() {
36        return marca;
37    }
38
39    public void setMarca(Marca marca) {
40        this.marca = marca;
41    }
42 }
43

```

Classe Marca:

```

1 package com.carros.models;
2
3 import jakarta.persistence.*;
4
5 @Entity
6 @Table(name = "marca")
7 public class Marca {
8     @Id
9     @GeneratedValue(strategy = GenerationType.IDENTITY)
10    private Long id;
11
12    @Column(nullable = false)
13    private String nome;
14
15    public Long getId() {
16        return id;
17    }
18
19    public void setId(Long id) {
20        this.id = id;
21    }
22
23    public String getNome() {
24        return nome;
25    }
26
27    public void setNome(String nome) {
28        this.nome = nome;
29    }
30 }

```

Classe Carro:


```

1 package com.carros.models;
2
3 import jakarta.persistence.*;
4
5 @Entity
6 @Table(name = "carro")
7 public class Carro {
8     @Id
9     @GeneratedValue(strategy = GenerationType.IDENTITY)
10    private Long id;
11
12    @ManyToOne
13    @JoinColumn(name = "id_modelo", nullable = false)
14    private Modelo modelo;
15
16    @Column(name = "ano_fabricacao", nullable = false)
17    private Integer anoFabricacao;
18
19    @Column(nullable = false)
20    private String cor;
21
22    @Column(nullable = false)
23    private BigDecimal preco;
24
25    @Column(nullable = false)
26    private Integer quilometragem;
27
28    @Column(name = "url_imagem")
29    private String urlImagem;
30
31    @Column(name = "status_disponibilidade", nullable = false)
32    @Enumerated(EnumType.STRING)
33    private StatusDisponibilidade statusDisponibilidade;
34
35    public Long getId() {
36        return id;
37    }
38
39    public void setId(Long id) {
40        this.id = id;
41    }
42
43    public Modelo getModelo() {
44        return modelo;
45    }
46

```

```

46     }
47
48     public void setModelo(Modelo modelo) {
49         this.modelo = modelo;
50     }
51
52     public Integer getAnoFabricacao() {
53         return anoFabricacao;
54     }
55
56     public void setAnoFabricacao(Integer anoFabricacao) {
57         this.anoFabricacao = anoFabricacao;
58     }
59
60     public String getCor() {
61         return cor;
62     }
63
64     public void setCor(String cor) {
65         this.cor = cor;
66     }
67
68     public BigDecimal getPreco() {
69         return preco;
70     }
71
72     public void setPreco(BigDecimal preco) {
73         this.preco = preco;
74     }
75
76     public Integer getQuilometragem() {
77         return quilometragem;
78     }
79
80     public void setQuilometragem(Integer quilometragem) {
81         this.kilometragem = quilometragem;
82     }
83
84     public StatusDisponibilidade getStatusDisponibilidade() {
85         return statusDisponibilidade;
86     }
87
88     public void setStatusDisponibilidade(StatusDisponibilidade statusDisponibilidade) {
89         this.statusDisponibilidade = statusDisponibilidade;
90     }
91
92     public String getUrlImagem() {
93         return urlImagem;
94     }
95
96     public void setUrlImagem(String urlImagem) {
97         this.urlImagem = urlImagem;
98     }
99 }
100

```

Enum:

```

1 package com.carros.models;
2
3 public enum StatusDisponibilidade {
4     Disponível,
5     Indisponível
6 }
7

```

ModeloController:

```

1 package com.carras.controllers;
2
3 import com.carras.models.Modelo;
4
5
6
7
8
9
10
11 @RestController
12 @CrossOrigin(origins = "**")
13 @RequestMapping("/api/modelos")
14 public class ModeloController {
15
16     @Autowired
17     private ModeloRepository modeloRepository;
18
19     @GetMapping
20     public List<Modelo> listarTodos() {
21         return modeloRepository.findAll();
22     }
23
24     @GetMapping("/{id}")
25     public ResponseEntity<Modelo> buscarPorId(@PathVariable Long id) {
26         return modeloRepository.findById(id)
27             .map(ResponseEntity::ok)
28             .orElse(ResponseEntity.notFound().build());
29     }
30
31     @PostMapping
32     public Modelo criar(@RequestBody Modelo modelo) {
33         return modeloRepository.save(modelo);
34     }
35
36     @PutMapping("/{id}")
37     public ResponseEntity<Modelo> atualizar(@PathVariable Long id, @RequestBody Modelo modelo) {
38         if (!modeloRepository.existsById(id)) {
39             return ResponseEntity.notFound().build();
40         }
41         modelo.setId(id);
42         return ResponseEntity.ok(modeloRepository.save(modelo));
43     }
44
45     @DeleteMapping("/{id}")
46     public ResponseEntity<Void> deletar(@PathVariable Long id) {
47         if (!modeloRepository.existsById(id)) {
48             return ResponseEntity.notFound().build();
49         }
50         modeloRepository.deleteById(id);
51         return ResponseEntity.noContent().build();
52     }
53 }
54

```

MarcaController:

```

1 package com.carros.controllers;
2
3+ import com.carros.models.Marca;[]
10
11 @RestController
12 @CrossOrigin(origins = "**")
13 @RequestMapping("/api/marcas")
14 public class MarcaController {
15
16-   @Autowired
17   private MarcaRepository marcaRepository;
18
19-   @GetMapping
20   public List<Marca> listarTodas() {
21       return marcaRepository.findAll();
22   }
23
24-   @GetMapping("/{id}")
25   public ResponseEntity<Marca> buscarPorId(@PathVariable Long id) {
26       return marcaRepository.findById(id)
27           .map(ResponseEntity::ok)
28           .orElse(ResponseEntity.notFound().build());
29   }
30
31-   @PostMapping
32   public Marca criar(@RequestBody Marca marca) {
33       return marcaRepository.save(marca);
34   }
35
36-   @PutMapping("/{id}")
37   public ResponseEntity<Marca> atualizar(@PathVariable Long id, @RequestBody Marca marca) {
38       if (!marcaRepository.existsById(id)) {
39           return ResponseEntity.notFound().build();
40       }
41       marca.setId(id);
42       return ResponseEntity.ok(marcaRepository.save(marca));
43   }
44
45-   @DeleteMapping("/{id}")
46   public ResponseEntity<Void> deletar(@PathVariable Long id) {
47       if (!marcaRepository.existsById(id)) {
48           return ResponseEntity.notFound().build();
49       }
50       marcaRepository.deleteById(id);
51       return ResponseEntity.noContent().build();
52   }
53 }

```

CarroController:

```

1 package com.carros.controllers;
2
3 import com.carros.models.Carro;
4
5
6
7
8
9
10
11 @RestController
12 @CrossOrigin(origins = "**")
13 @RequestMapping("/api/carros")
14 public class CarroController {
15
16
17     @Autowired
18     private CarroRepository carroRepository;
19
20     @GetMapping
21     public List<Carro> listarTodos() {
22         return carroRepository.findAll();
23     }
24
25     @GetMapping("/{id}")
26     public ResponseEntity<Carro> buscarPorId(@PathVariable Long id) {
27         return carroRepository.findById(id)
28             .map(ResponseEntity::ok)
29             .orElse(ResponseEntity.notFound().build());
30     }
31
32     @PostMapping
33     public Carro criar(@RequestBody Carro carro) {
34         return carroRepository.save(carro);
35     }
36
37     @PutMapping("/{id}")
38     public ResponseEntity<Carro> atualizar(@PathVariable Long id, @RequestBody Carro carro) {
39         if (!carroRepository.existsById(id)) {
40             return ResponseEntity.notFound().build();
41         }
42         carro.setId(id);
43         return ResponseEntity.ok(carroRepository.save(carro));
44     }
45
46     @DeleteMapping("/{id}")
47     public ResponseEntity<Void> deletar(@PathVariable Long id) {
48         if (!carroRepository.existsById(id)) {
49             return ResponseEntity.notFound().build();
50         }
51         carroRepository.deleteById(id);
52         return ResponseEntity.noContent().build();
53     }
54
55     @GetMapping("/disponiveis")
56     public List<Carro> listarDisponiveis() {
57         return carroRepository.findByStatusDisponibilidade(StatusDisponibilidade.Disponivel);
58     }
59 }
60

```

WebConfig:

```

1 package com.carros.config;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.Configuration;
5 import org.springframework.web.servlet.config.annotation.CorsRegistry;
6 import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
7
8 @Configuration
9 public class WebConfig implements WebMvcConfigurer {
10
11     @Override
12     public void addCorsMappings(CorsRegistry registry) {
13
14     }
15 }
16

```

Sistema Carro:

```
1 package com.carros;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class SistemaCarroApplication {
8     public static void main(String[] args) {
9         SpringApplication.run(SistemaCarroApplication.class, args);
10    }
11 }
```