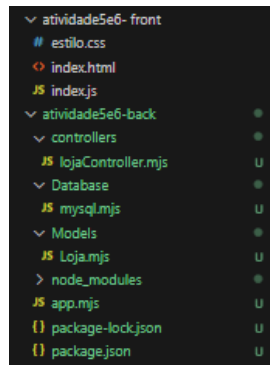


## ATIVIDADES 5 E 6

Estas atividades envolvem uma linguagem diferente das anteriores, sendo o node js. Achei uma linguagem menos complicada do que o PHP e não tem que colocar \$, o que melhora muito. A atividade foi simples, criei um novo banco de dados chamado Loja e o gerei automaticamente pelo node js, como é pedido na atividade 5, porém, utilizei também o front-end, que é pedido na atividade 6, ligando diretamente um ao outro, fazendo deles uma só atividade. Segue a divisão dos arquivos e as capturas exigidas na atividade.



Será enviado primeiramente o back-end, da atividade 5;

O controlador `lojaController` gerencia operações CRUD para o modelo `Loja`. Ele permite criar, ler, atualizar e deletar registros na loja, manipulando os dados recebidos e enviando as respostas apropriadas.

```

lojaController.mjs
atividade5e6-back > controllers > lojaController.mjs > lojaController > "delete"
1 import Loja from "../Models/Loja.mjs";
2
3 const lojaController = {
4   "new": async (req, res) => {
5     const created = await loja.create({
6       fabricante: req.body.fabricante,
7       modelo: req.body.modelo,
8       ano: req.body.ano,
9       marca: req.body.marca,
10      estoque: req.body.estoque
11    });
12    res.send(created);
13  },
14  "one": async (req, res) => {
15    const l = await loja.findOne({
16      where: { id: req.params.id }
17    });
18    res.json(l);
19  },
20  "all": async (req, res) => {
21    res.json(await loja.findAll());
22  },
23  "edit": async (req, res) => {
24    const l = await loja.findOne({
25      where: { id: req.body.id }
26    });
27    l.fabricante = req.body.fabricante;
28    l.modelo = req.body.modelo;
29    l.ano = req.body.ano;
30    l.marca = req.body.marca;
31    l.estoque = req.body.estoque;
32    await l.save();
33    res.json(l);
34  },
35  "delete": async (req, res) => {
36    const l = await loja.findOne({
37      where: { id: req.body.id }
38    });
39    await l.destroy();
40    res.json(l);
41  }
42 };
43
44 export default lojaController;
45

```

Este código importa o Sequelize, cria uma instância conectando ao banco de dados MySQL "Loja", e a exporta como padrão.

```

loja.mjs
atividade5e6-back > Database > loja.mjs > default
1 import { Sequelize } from "sequelize";
2
3 const sequelize = new Sequelize("mysql://root:root@localhost:3306/Loja");
4
5 export default sequelize

```

Este código define um modelo `Loja` com Sequelize, especificando seus atributos e tipos de dados, e sincroniza o modelo com o banco de dados.

```
JS Loja.mjs U X
atividade5e6-back > Models > JS Loja.mjs > Loja
1 import sequelize from "../Database/mysql.mjs";
2 import { DataTypes } from "sequelize";
3
4 const Loja = sequelize.define('Loja', {
5   fabricante: DataTypes.STRING,
6   modelo: DataTypes.STRING,
7   ano: DataTypes.INTEGER,
8   marca: DataTypes.STRING,
9   estoque: DataTypes.INTEGER
10 });
11
12 await Loja.sync({});
13
14 export default Loja;
15
```

O app.mjs configura um servidor Express, define rotas para operações CRUD utilizando o controlador `lojaController`, e inicia o servidor na porta 3000.

```
JS app.mjs U X
atividade5e6-back > JS app.mjs > ...
1 import express from 'express';
2 import cors from 'cors';
3 import lojaController from './controllers/lojaController.mjs';
4
5 const app = express();
6 const port = 3000;
7
8 app.use(cors());
9 app.use(express.json());
10 app.use(express.urlencoded({ extended: true }));
11
12 app.get('/lojas', lojaController.all);
13 app.get('/lojas/:id', lojaController.one);
14 app.post('/lojas', lojaController.new);
15 app.put('/lojas', lojaController.edit);
16 app.delete('/lojas', lojaController.delete);
17
18 app.listen(port, () => {
19   console.log(`Example app listening on port ${port}`);
20 });
21
```

O arquivo package-lock.json descreve as dependências do projeto, incluindo cors, express, mysql2 e sequelize, necessárias para o funcionamento do back-end da aplicação.

```

() package-lock.json U X
atividade5e6-back > () package-lock.json > ...
1 {
2   "name": "atividade5e6-back",
3   "version": "1.0.0",
4   "lockfileVersion": 3,
5   "requires": true,
6   "packages": {
7     "": {
8       "name": "atividade5e6-back",
9       "version": "1.0.0",
10      "license": "ISC",
11      "dependencies": {
12        "cors": "^2.8.5",
13        "express": "^4.19.2",
14        "mysql2": "^3.9.3",
15        "sequelize": "^6.37.2"
16      },
17      "devDependencies": {}
18    },
19    "node_modules/@types/debug": {
20      "version": "4.1.12",
21      "license": "MIT",
22      "dependencies": {
23        "@types/ms": "*"
24      }
25    }
26  }

```

(Este é somente o início do código).

O `package.json` define as dependências (cors, express, mysql2, sequelize) e os scripts (dev para iniciar o servidor) do projeto, além de incluir informações básicas como nome, versão, descrição, autor e licença.

```

() package.json U X
atividade5e6-back > () package.json > ...
1 {
2   "dependencies": {
3     "cors": "^2.8.5",
4     "express": "^4.19.2",
5     "mysql2": "^3.9.3",
6     "sequelize": "^6.37.2"
7   },
8   "name": "atividade5e6-back",
9   "version": "1.0.0",
10  "description": "Atividades",
11  "main": "app.js",
12  "scripts": {
13    "dev": "node app.mjs",
14    "test": "echo \"Error: no test specified\" && exit 1"
15  },
16  "author": "Samyra",
17  "license": "ISC",
18  "devDependencies": {}
19 }
20

```

Os módulos do node também estão contidos na pasta do arquivo.

## FRONT, ATIVIDADE 6

```

# estilo.css X
atividade5e6- front > # estilo.css > ...
1  * {
2      margin: 0;
3      padding: 0;
4  }
5
6  body {
7      background-color: write(211, 210, 208);
8      color: black;
9      padding: 20px;
10 }
11
12 div {
13     position: fixed;
14     top: 0;
15     right: 0;
16     bottom: 0;
17     left: 0;
18     background-color: rgba(230, 220, 220, 0.651);
19     color: brown;
20     display: none;
21     justify-content: center;
22     align-items: center;
23 }
24
25 .mostrarModal {
26     display: flex;
27 }

```

CSS da página (há mais códigos

abaixo, só não foi capturado).

O HTML cria uma página para listar produtos de uma loja, com um botão para adicionar novos produtos e um formulário modal para editar os detalhes dos produtos. Inclui também a estrutura básica de cabeçalho, corpo principal e rodapé, com links para estilos CSS e scripts JavaScript externos.

```

index.html X
atividade5e6- front > index.html > html > body
2  <html lang="en">
3
13 <body>
14     <header>
15         <h1>AQUI TEM DE TUDO UM POUCO.</h1>
16         <h2><center>Por pura diversão.</center></h2>
17     </header>
18     <main>
19         <h2><center>Listagem</center></h2>
20         <button id="btNovo">Novo</button>
21         <table border="1">
22             <thead>
23                 <tr>
24                     <th>#</th>
25                     <th>FABRICANTE</th>
26                     <th>MODELO</th>
27                     <th>ANO</th>
28                     <th>MARCA</th>
29                     <th>ESTOQUE</th>
30                     <th>AÇÕES</th>
31                 </tr>
32             </thead>
33             <tbody></tbody>
34         </table>
35     </main>
36     <div id="modal" classe="modal">
37         <form id="modalForm">
38             <input type="hidden" name="id" id="id"><br>
39             <label for="fabricante">Fabricante</label><br>
40             <input type="text" name="fabricante" id="fabricante"><br>
41             <label for="modelo">Modelo</label><br>
42             <input type="text" name="modelo" id="modelo"><br>
43             <label for="ano">Ano</label><br>
44             <input type="number" name="ano" id="ano"><br>
45             <label for="marca">Marca</label><br>
46             <input type="text" name="marca" id="marca"><br>
47             <label for="estoque">Estoque</label><br>
48             <input type="number" name="estoque" id="estoque"><br>
49             <button type="submit">Salvar</button>
50             <button type="button" id="btClose">Fechar</button>
51         </form>
52     </div>
53     <footer>
54         <p>©copy; Samyra</p>
55     </footer>
56     <script src="index.js"></script></body>
57 </html>
58

```

O index.js gerencia uma tabela de dados de lojas, permitindo adicionar, editar e excluir entradas. Ele carrega os dados via fetch, atualiza a tabela e lida com a interação do usuário com botões para cada ação (editar, excluir, adicionar e salvar).

```
1  const urlBase = 'http://localhost:3000';
2  let dados = [];
3
4  function desenhaTabela() {
5    const tbody = document.querySelector('tbody');
6    tbody.innerHTML = '';
7    for (let i = 0; i < dados.length; i++) {
8      const tr = document.createElement('tr');
9      const btEx = document.createElement('button');
10     const btEd = document.createElement('button');
11
12     const td1 = document.createElement('td');
13     const td2 = document.createElement('td');
14     const td3 = document.createElement('td');
15     const td4 = document.createElement('td');
16     const td5 = document.createElement('td');
17     const td6 = document.createElement('td');
18     const td7 = document.createElement('td');
19
20     btEx.innerText = '-';
21     btEx.setAttribute('data-id', dados[i].id);
22     btEx.addEventListener('click', (e) => {
23       const id = e.target.getAttribute('data-id');
24       alternaModal();
25       enviaDadosParaDelecao(id);
26     });
27
28     btEd.innerText = '.';
29     btEd.setAttribute('data-index', i);
30     btEd.addEventListener('click', (e) => {
31       const index = e.target.getAttribute('data-index');
32       alternaModal();
33       preencheFormParaEdicao(index);
34     });
35
36     td1.innerText = dados[i].id;
37     td2.innerText = dados[i].fabricante;
38     td3.innerText = dados[i].modelo;
39     td4.innerText = dados[i].ano;
40     td5.innerText = dados[i].marca;
41     td6.innerText = dados[i].estoque;
42     td7.appendChild(btEd, btEx);
43
44     tr.appendChild(td1, td2, td3, td4, td5, td6, td7);
45     tbody.appendChild(tr);
46   }
47 }
```

```
49 function carregaDados() {
50   fetch(`${urlBase}/lojas`)
51     .then((res) => res.json())
52     .then((json) => {
53       dados = json;
54       desenhaTabela();
55     })
56     .catch((error) => console.error("Erro ao carregar dados:", error));
57 }
58
59 function enviaDadosParaCadastro() {
60   const dados = new FormData(document.querySelector('form'));
61   const opcoes = {
62     method: 'post',
63     body: new URLSearchParams(dados)
64   };
65   fetch(`${urlBase}/lojas`, opcoes)
66     .then((res) => res.json())
67     .then((json) => {
68       alert('Loja ' + json.modelo + ' cadastrada!');
69       carregaDados();
70     })
71     .catch((error) => console.error("Erro ao cadastrar dados:", error));
72   alternaModal();
73 }
74
75 function enviaDadosParaDelecao(id) {
76   const dados = new FormData();
77   dados.append('id', id);
78   const opcoes = {
79     method: 'delete',
80     body: new URLSearchParams(dados)
81   };
82   fetch(`${urlBase}/lojas`, opcoes)
83     .then((res) => res.json())
84     .then((json) => {
85       alert('Loja ' + json.modelo + ' deletada!');
86       carregaDados();
87     })
88     .catch((error) => console.error("Erro ao deletar dados:", error));
89   alternaModal();
90 }
91 }
```

```
92 function preencheFormParaEdicao(index) {
93   document.querySelector('#id').value = dados[index].id;
94   document.querySelector('#fabricante').value = dados[index].fabricante;
95   document.querySelector('#modelo').value = dados[index].modelo;
96   document.querySelector('#ano').value = dados[index].ano;
97   document.querySelector('#marca').value = dados[index].marca;
98   document.querySelector('#estoque').value = dados[index].estoque;
99 }
100
101 function enviaDadosParaEdicao() {
102   const dados = new FormData(document.querySelector('form'));
103   const opcoes = {
104     method: 'put',
105     body: new URLSearchParams(dados)
106   };
107   fetch(`${urlBase}/lojas`, opcoes)
108     .then((res) => res.json())
109     .then((json) => {
110       alert('Loja ' + json.modelo + ' editada!');
111       carregaDados();
112     })
113     .catch((error) => console.error("Erro ao editar dados:", error));
114   alternaModal();
115 }
116
117 function alternaModal() {
118   document.querySelector('#modal').classList.toggle('mostrarModal');
119 }
120
121 document.querySelector('#btNovo').addEventListener('click', alternaModal);
122 document.querySelector('#btClose').addEventListener('click', alternaModal);
123 window.addEventListener('load', carregaDados);
124 document.querySelector('form button[type="submit"]').addEventListener('click', (e) => {
125   e.preventDefault();
126   if (document.querySelector('#id').value) {
127     enviaDadosParaEdicao();
128   } else {
129     enviaDadosParaCadastro();
130   }
131   document.querySelector('#id').value = '';
132   e.target.parentNode.reset();
133 });
134 }
```

Está começando no item 2, pois em fase de teste, foi excluído o item 1.

**AQUI TEM DE TUDO UM POUCO**  
**Por pura diversão**  
**Listagem**

Novo

| # | FABRICANTE | MODELO     | ANO  | MARCA   | ESTOQUE | AÇÕES   |
|---|------------|------------|------|---------|---------|---|
| 2 | Yh         | XTZ 125    | 2023 | Yamaha  | 4       |  |
| 3 | APPLE      | IPHONE 15  | 2024 | APPLE   | 33      |  |
| 4 | SAMSUNG    | GALAXY S23 | 2023 | Samsung | 52      |  |
| 5 | TOYOTA     | COROLLA    | 2023 | TOYOTA  | 15      |  |

Fabricante

samyra

Modelo

blusa

Ano

2024

Marca

Marisa

Estoque

7

Salvar

Fechar






adição de novos produtos

AQUI TEM DE TUDO UM POUCO

Por pura diversão

Listagem

Novo

| # | FABRICANTE | MODELO     | ANO  | MARCA   | ESTOQUE | AÇÕES   |
|---|------------|------------|------|---------|---------|---|
| 2 | Yh         | XTZ 125    | 2023 | Yamaha  | 4       |  |
| 3 | APPLE      | IPHONE 15  | 2024 | APPLE   | 33      |  |
| 4 | SAMSUNG    | GALAXY S23 | 2023 | Samsung | 52      |  |
| 5 | TOYOTA     | COROLLA    | 2023 | TOYOTA  | 15      |  |
| 6 | samyra     | blusa      | 2024 | Marisa  | 7       |  |






item 6 adicionado.

AQUI TEM DE TUDO UM POUCO

Por pura diversão

Listagem

Novo

| # | FABRICANTE | MODELO     | ANO  | MARCA   | ESTOQUE | AÇÕES   |
|---|------------|------------|------|---------|---------|---|
| 2 | Yh         | XTZ 125    | 2023 | Yamaha  | 4       |  |
| 3 | APPLE      | IPHONE 15  | 2024 | APPLE   | 33      |  |
| 4 | Motorola   | GALAXY S23 | 2023 | Samsung | 52      |  |
| 5 | TOYOTA     | COROLLA    | 2023 | TOYOTA  | 15      |  |
| 6 | samyra     | blusa      | 2024 | Marisa  | 7       |  |

edição do item 4.

AQUI TEM DE TUDO UM POUCO

Por pura diversão

Listagem

Novo

| # | FABRICANTE | MODELO     | ANO  | MARCA   | ESTOQUE | AÇÕES   |
|---|------------|------------|------|---------|---------|---|
| 2 | Yh         | XTZ 125    | 2023 | Yamaha  | 4       |  |
| 4 | Motorola   | GALAXY S23 | 2023 | Samsung | 52      |  |
| 5 | TOYOTA     | COROLLA    | 2023 | TOYOTA  | 15      |  |
| 6 | samyra     | blusa      | 2024 | Marisa  | 7       |  |

exclusão do item 3.