



Universidade Federal do Ceará
Campus Sobral
Curso de Engenharia da Computação
Disciplina: Tecnologias Web II
Professor: Thiago Iachiley Araujo de Souza

José Alan Torres Pires

RELATÓRIO DO TRABALHO 02

Sobral-CE

2024

Sumário

1. INTRODUÇÃO.....	3
2. PROCESSO DE DESENVOLVIMENTO.....	4
3. INSTRUÇÕES PARA EXECUÇÃO DO PROJETO.....	9
4. CONSIDERAÇÕES FINAIS.....	10
5. REFERÊNCIAS BIBLIOGRÁFICAS.....	11

1. INTRODUÇÃO

O presente projeto tem o objetivo desenvolver uma aplicação web utilizando as tecnologias html, css, javascript e node.js. Tendo como finalidade revisar os conteúdos aprendidos na disciplina de tecnologias web I e também familiarizar-se com o node.js. Os objetivos específicos são detalhados logo abaixo:

- Relembrar as tags semânticas do html
- Relembrar conceitos de flexbox e estilização com css
- Relembrar os frameworks disponíveis para css
- Relembrar sobre responsividade
- Criar um servidor local com node.js
- Permitir requisições e retorná-las como uma página html

As tecnologias empregadas são descritas abaixo:

- **Html(Hypertext Markup Language):** HTML é a linguagem padrão para criação de páginas da web. Ele consiste em uma série de elementos, como tags, atributos e texto, usados para estruturar o conteúdo da página. As tags HTML são utilizadas para definir elementos como texto, imagens, links e outros. HTML fornece a estrutura básica de uma página da web e é interpretado pelo navegador para renderizar o conteúdo visualmente.
- **Css(Cascading Style Sheets):** CSS é uma linguagem de estilo usada para controlar a apresentação visual das páginas da web. Com CSS, é possível definir estilos, layout e aparência dos elementos HTML em uma página. Os estilos CSS são aplicados aos elementos HTML usando seletores e declarações de estilo. É possível definir propriedades como cor, fonte, tamanho, posicionamento, margens, bordas e animações, entre outros.
- **JavaScript:** JavaScript é uma linguagem de programação de alto nível utilizada para tornar as páginas da web dinâmicas e interativas. É interpretada pelos navegadores e frequentemente usada para manipular o conteúdo HTML, responder a eventos do usuário e realizar operações assíncronas, como fazer requisições HTTP. JavaScript é

orientada a objetos e possui uma sintaxe similar a outras linguagens de programação, como Java e C++.

- **Node.js:** Node.js é um ambiente de execução JavaScript do lado do servidor construído no motor V8 do Google Chrome. Ele permite que os desenvolvedores usem JavaScript para escrever scripts do lado do servidor, expandindo as possibilidades além do navegador. Node.js é conhecido por sua eficiência e escalabilidade, sendo adequado para construir aplicativos de rede de alto desempenho e em tempo real. Possui um sistema de módulos embutido e um grande ecossistema de pacotes disponíveis através do npm (Node Package Manager). É comumente usado para construir aplicativos da web, APIs RESTful, aplicativos de mensagens em tempo real, aplicativos de streaming de vídeo, entre outros.
- **GitHub:** Github é uma plataforma de desenvolvimento colaborativo baseada na web que permite hospedar, revisar e compartilhar projetos de software utilizando o controle de versão Git. Fundado em 2008, o GitHub se tornou uma das ferramentas mais populares e amplamente utilizadas por desenvolvedores de software em todo o mundo.
- **Bootstrap:** Bootstrap é um dos frameworks de front-end mais populares e amplamente utilizados para desenvolvimento web responsivo. Ele fornece um conjunto abrangente de ferramentas e componentes CSS e JavaScript que facilitam a criação de interfaces de usuário modernas e atrativas.

2. PROCESSO DE DESENVOLVIMENTO

Inicialmente, foi criado o servidor utilizando os módulos http e fs. Desse modo, foi criado duas condições, caso a requisição fosse o arquivo .html e caso a requisição fosse o arquivo .css. Além disso, é claro, foram criados o arquivo index.html e o arquivo .css.

O código inicial do servidor, segue abaixo:

```
const http = require('http');
const fs = require('fs');
const server = http.createServer((request, response) => {
  if (request.url === '/' || request.url === '/index.html') {
    fs.readFile('index.html', (err, arquivo) => {
      response.writeHead(200, { 'Content-Type': 'text/html' });
      response.write(arquivo);
      return response.end();
    });
  } else if (request.url === '/style.css') {
    fs.readFile('style.css', (err, arquivo) => {
      response.writeHead(200, { 'Content-Type': 'text/css' });
      response.write(arquivo);
      return response.end();
    });
  }
});
```

Para a página html inicialmente foi criada a estrutura de uma página html e em seguida o header, utilizando o framework bootstrap. Segue abaixo o código:

```
<header>
  <nav class="navbar navbar-expand-lg bg-body-tertiary">
    <div class="container-fluid">
      <a class="navbar-brand" href="#"> Buggy's Delivery</a>
      <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-target="#navbarNav"
      aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
```

```

        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
            <li class="nav-item">
                <a class="nav-link active" aria-current="page"
href="#">Home</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#dtqs">Destques</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Clientes</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Contato</a>
            </li>
        </ul>
    </div>
</div>
</nav>
</header>

```

O resultado pode ser visualizado na figura 01:

Figura 01: Cabeçalho do site



Fonte: Autor.

Após isso, foi feito o primeiro teste com o servidor executando. E a página carregou normalmente. Em seguida, foram adicionada ao arquivo html, a seção main, composta por um banner principal e por cards contendo os produtos em destaque cujo código pode ser observado abaixo:

```

<div class="card" style="width: 18rem;">
    
    <div class="card-body">
        <p class="card-text">Mangá One Piece v.507</p>
        <h4>R$ 40,00</h4>
    </div>
</div>

```

Após recarregar a página com essa nova seção, notou-se que as imagens não foram carregadas. Para corrigir esse erro, foi necessário acrescentar uma condição na lógica do servidor, para caso receber uma solicitação de acesso a um arquivo de imagem, o código adicionado segue a baixo:

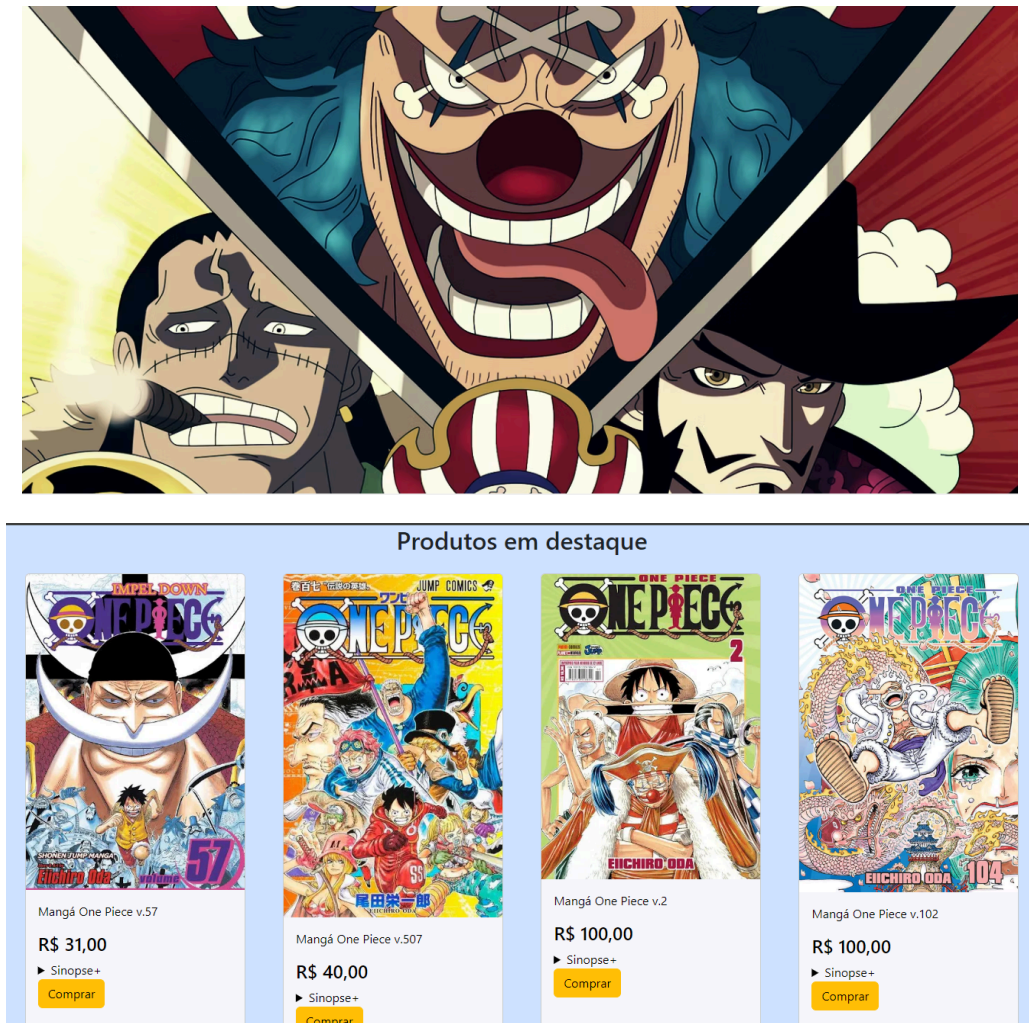
```

else if (request.url.startsWith('/assets/')) {
    const imagePath = path.join(__dirname, request.url);
    fs.readFile(imagePath, (err, imagem) => {
        if (err) {
            response.writeHead(404, { 'Content-Type': 'text/plain' });
            response.write('Imagem não encontrada');
            return response.end();
        }
        const imageExtension = path.extname(imagePath);
        const contentType = `image/${imageExtension.substring(1)}`;
        response.writeHead(200, { 'Content-Type': contentType });
        response.end(imagem);
    });
}

```

O resultado com essa nova seção pode ser visualizado na figura 02:

Figura 02: Adicionando banner e produtos



Fonte: Autor.

Em seguida, foi adicionada a seção de avaliações dos clientes, composta por cards construídos da maneira abaixo:

```
<div class="card text-bg-info mb-3" style="max-width: 18rem;">
  <div class="card-body">
    <h5 class="card-title">Alan Pires</h5>
    <div class="rating">
      <span class="star">&#9733;</span>
      <span class="star">&#9733;</span>
      <span class="star">&#9733;</span>
      <span class="star">&#9733;</span>
      <span class="star">&#9734;</span>
    </div>
    <p class="card-text">Ótimo atendimento, entrega rápida e eficiente.
```



```
Parabéns!!!</p>
  </div>
</div>
```

O resultado pode ser visto na figura 03:

Figura 03: Adicionando banner e produtos



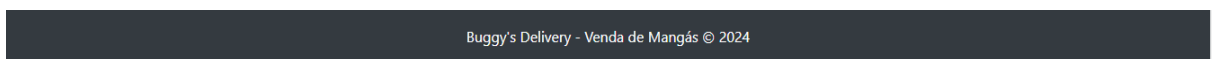
Fonte: Autor.

E por último, foi feito a seção de rodapé, com a seguinte codificação:

```
<footer id="contato" class="footer mt-auto py-3">
  <div class="container">
    <span class="text-muted">Buggy's Delivery - Venda de Mangás &copy;
2024</span>
  </div>
</footer>
```

O resultado pode ser visto na figura 04:

Figura 04: Adicionando footer.



Fonte: Autor.

Vale ressaltar que foi utilizado o github para versionamento do código visto que o projeto foi feito utilizando os computadores do laboratório de informática e também o computador do autor.

3. INSTRUÇÕES PARA EXECUÇÃO DO PROJETO

Para execução do projeto, siga os passos abaixo:

1. Certifique-se que o node está instalado na sua máquina, caso não acesse o site <https://nodejs.org/en/download> e faça o download.
2. Baixe o arquivo com o código fonte e descompacte ele.
3. Após isso, abra o terminal e acesse o diretório com os arquivos do projeto.
4. Após isso, digite o comando `node server.js`, ele irá rodar o servidor.
5. Em seguida, abra o navegador e digite <http://localhost:3000/index.html>.
6. Pronto, a página web será carregada.

4. CONSIDERAÇÕES FINAIS

Por fim, percebe-se que o nodejs é uma ferramenta de suma importância para o desenvolvimento web tendo papel crucial na criação de servidores web. Ademais, a utilização do bootstrap permite a criação de um design responsivo e atual para a página web, bem como facilita o processo de desenvolvimento. Além disso, o github se mostra como uma ferramenta essencial no controle e versionamento do código para os desenvolvedores e deve ser empregado sempre quando estiver trabalhando em equipe.

5. REFERÊNCIAS BIBLIOGRÁFICAS

Bootstrap. Disponível em: <https://getbootstrap.com/>. Acesso em: 28 de Março de 2024.

Alura. Node.js: Guia completo para aprender sobre Node.js. Disponível em: https://www.alura.com.br/artigos/node-js?utm_term=&utm_campaign=%5BSearch%5D+%5BPerformance%5D+-+Dynamic+Search+Ads+-+Artigos+e+Conte%C3%BAdos&utm_source=adwords&utm_medium=ppc&hsa_acc=7964138385&hsa_cam=11384329873&hsa_grp=111087461203&hsa_ad=687448474447&hsa_src=g&hsa_tgt=dsa-2276348409543&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=Cj0KCQjw2a6wBhCVARIsABPeH1vuvsoOVSN0plbakHF1tD0Tz_Wff3wZFkA2bsp1waulcZSb5OFmy34aAgfoEALw_wcB. Acesso em: 28 de Março de 2024.

Origamid. Flexbox: Guia Completo. Disponível em: <https://origamid.com/projetos/flexbox-guia-completo/>. Acesso em: 28 de Março de 2024.

.