



UFC – UNIVERSIDADE FEDERAL DO CEARÁ CAMPUS DE SOBRAL

CURSO DE ENGENHARIA DA COMPUTAÇÃO

TECNOLOGIAS WEB II

PROFESSOR: THIAGO IACHILEY ARAUJO DE SOUZA

TRABALHO 02

JOANA STHEFANNY GOMES COSTA DOS SANTOS - 470890

**Sobral - CE
2024.1**

LINK DO REPOSITÓRIO GITHUB
<https://github.com/JoanaSthefanny/BabyShop>

1. INTRODUÇÃO

No contexto do desenvolvimento web, a compreensão dos fundamentos das Tecnologias Web é crucial para a criação de aplicações modernas e eficientes. Este relatório tem como objetivo revisitar os conceitos fundamentais abordados em Tecnologias Web I, enquanto é explorado o uso do Node.js como uma ferramenta para aplicar novos conceitos.

2. TECNOLOGIAS UTILIZADAS

2.1. HTTP

O Protocolo de Transferência de Hipertexto (HTTP), em inglês Hypertext Transfer Protocol, é um protocolo de comunicação utilizado para transferir informações na World Wide Web (WWW). Ele é fundamental para a comunicação entre clientes (como navegadores da web) e servidores web. Ele define um conjunto de regras e formatos para a troca de informações, facilitando o acesso e o compartilhamento de recursos na internet.

2.2. HTML

HTML, ou Hypertext Markup Language, é a linguagem de marcação padrão usada para criar páginas web. Ela fornece a estrutura básica para o conteúdo de uma página web, definindo elementos que representam diferentes tipos de conteúdo, como texto, imagens, vídeos, links e etc.

2.3. CSS

CSS, ou Cascading Style Sheets, é uma linguagem de estilo usada para controlar a apresentação e o layout de páginas web escritas em HTML e XHTML. Ele permite aos desenvolvedores web definir o visual de uma página, incluindo aspectos como cores, fontes, margens, espaçamentos, posicionamento e animações, desempenhando um papel crucial no design e na apresentação de páginas web modernas, permitindo aos desenvolvedores controlar e personalizar a aparência visual de seus sites de forma flexível e poderosa.

2.4. JavaScript

JavaScript é uma linguagem de programação de alto nível, dinâmica e interpretada, amplamente utilizada para desenvolvimento web. Ela é conhecida principalmente por adicionar interatividade, dinamismo e funcionalidades avançadas às páginas da web, sendo uma linguagem de programação essencial para o desenvolvimento web moderno, oferecendo aos desenvolvedores uma ampla gama de recursos.

2.5. Node.js

Node.js é um ambiente de execução de código JavaScript do lado do servidor, construído com o motor de JavaScript V8 do Google Chrome. Ele permite que os desenvolvedores usem JavaScript para criar aplicativos de rede escaláveis e de alto desempenho, sendo uma plataforma que permite aos desenvolvedores criar aplicações web escaláveis, eficientes e de alto desempenho usando JavaScript tanto no lado do cliente quanto no lado do servidor. Ele tem sido amplamente adotado pela comunidade de desenvolvimento web devido à sua facilidade de uso, desempenho e extensibilidade.

3. DESENVOLVIMENTO

3.1. HTML

Inicialmente, a primeira parte do código HTML define o início do documento e configura várias configurações importantes. A `<head>` é a seção que contém metadados e informações sobre o documento, sendo utilizada para configurar o ambiente de renderização.

```
index.html > html > body > header > div.Logo
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <link rel="stylesheet" href="style.css">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Babyshop</title>
9  </head>
10 <body>
```

O elemento `<body>` é onde todo o conteúdo visível da página é colocado. É dentro desta tag que é encontrado o conteúdo principal da página. A `<header>` é onde são colocados elementos como o logotipo da empresa, a busca e o carrinho de compras. O código pode ser visto na seguinte imagem:

```
11 <body>
12
13     <header>
14         <div class="Logo">
15             
16         </div>
17
18         <div class="Busca">
19             <input type="text" id="txtBusca" placeholder="Buscar..."/>
20             
21         </div>
22
23         <div class="carrinho">
24             
25         </div>
26
27     </header>
28
29 </body>
```

A classe “container” atua como um contêiner para o conteúdo principal da página. Ela é responsável por limitar a largura do conteúdo principal e centralizá-lo horizontalmente na página. A classe menu refere-se a um elemento `<div>` que envolve o menu de navegação da página. Dentro do `<div class="menu">`, temos um elemento `<nav>` que é utilizado para definir uma seção de navegação e um elemento `` que é uma lista não ordenada que contém os itens do menu. O código pode ser visto na imagem abaixo:

```
<div class="container">
    <main>
        <div class="menu">
            <nav>
                <ul id="Lista">
                    <li><b><a href="#">Home</a></b></li>
                    <li><b><a href="#">Lançamentos</a></b></li>
                    <li><b><a href="#">Destaque</a></b></li>
                    <li><b><a href="#">Avaliações</a></b></li>
                    <li><b><a href="#">Contatos</a></b></li>
                </ul>
            </nav>
        </div>
    </main>
</div>
```

A classe "lancamentos" é usada para identificar uma seção da página que exibe os produtos recém-lançados. Dentro da `<div class="lancamentos">`, é encontrado o título "Lançamentos" (definido pelo elemento `<h1>`) seguido pelo conteúdo dos produtos recém-lançados (total de 4 itens). Este conteúdo é composto por uma série de elementos `<div class="item">`, cada um representando um produto diferente. O Código pode ser visto na imagem abaixo:

```
46 <div class="lancamentos">
47     <h1>Lançamentos</h1>
48     <div class="lancamentosimgs">
49         <div class="item">
50             <a href="#">
51             <p class="preco">R$ 299,00</p>
52             <p class="Marca">Munchkin Criança Feliz - Kit Alimentação 4 Peças</p>
53             <p class="Descricao">
54                 O Munchkin Criança Feliz - Kit Alimentação 4 Peças é o conjunto completo para tornar a hora da refeição do seu filho mais fácil e divertida. Com uma colher e um garfo fáceis de segurar, um prato resistente a derramamentos e um copo ergonômico, este kit oferece tudo o que você precisa para alimentar seu pequeno com praticidade e conforto. Desenvolvido com materiais de alta qualidade e designs inteligentes, é o companheiro perfeito para as refeições.</p>
55         </div>
56     </div>
```

A classe "destaques" é usada para identificar uma seção da página que exibe os produtos em destaque. Dentro da `<div class="destaques">`, você encontrará o título "Destques" (definido pelo elemento `<h1>`) seguido pelo conteúdo dos produtos em destaque. Assim como na seção de "Lançamentos", este conteúdo é composto por uma série de elementos `<div class="item">` (totalizando 4 itens), cada um representando um produto diferente. O código pode ser visto na imagem abaixo:

```
81 <div class="destaques">
82     <h1>Destques</h1>
83     <div class="destaquesimgs">
84         <div class="item">
85             <a href="#">
86             <p class="preco">R$ 209,00</p>
87             <p class="Marca">Brinquedo macio oficial do ursinho Paddington</p>
88             <p class="Descricao">O brinquedo macio oficial do Ursinho Paddington é uma adorável e fofo companhia para o seu pequeno explorador. Inspirado no personagem querido dos livros e filmes, este brinquedo oferece conforto e diversão, sendo perfeito para abraçar e brincar durante todo o dia. Feito com materiais macios e seguros, é a escolha ideal para os fãs mais jovens do adorável ursinho viajante.</p></a>
89
90     </div>
```

A classe "avaliacoes" é usada para identificar uma seção da página que exibe as avaliações dos clientes sobre os produtos (total de 6 avaliações). Dentro dessa `<div class="avaliacoes">`, você encontrará o título "Avaliações" (definido pelo elemento `<h1>`) seguido pelo conteúdo das avaliações dos clientes. Cada avaliação é representada por um elemento `<div class="avaliacao">`, onde são exibidos o nome do cliente, uma imagem do avatar, a classificação em estrelas e o texto da avaliação. O código pode ser visto na imagem abaixo:

```
116     <div class="avaliacoes">
117         <div class="avaliacao">
118             <div class="section-review-userinfo">
119                 <div class="section-userinfo-username">
120                     
121                     <p class="nome-avatar"><b>Laura Mendes</b></p>
122                 </div>
123                 <div class="avaliacao_estrelas">
124                     
125                 </div>
126                 <div class="avaliacao_usuario">
127                     <p class="texto">Estou simplesmente encantada com a qualidade e a segurança desses produtos para bebês! Desde os babadores até os brinquedos educativos, cada item é pensado com tanto carinho. Sem dúvida, confio plenamente na marca para cuidar do meu pequeno tesouro.</p>
128                 </div>
129             </div>
130         </div>
131     </div>
132
133
```

O footer (rodapé) é a seção final do código HTML fornecido. contém três parágrafos <p> que fornecem informações de contato para a loja "BabyShop Honeybee". Cada parágrafo contém um ID único (WhatsApp, Instagram e Gmail). O código pode ser visto na imagem abaixo:

```
zz1
222     <footer>
223         <p id="WhatsApp">WhatsApp: (88) 9 9999-9999</p>
224         <p id="Instagram">Instagram: @honeybee</p>
225         <p id="Gmail">Gmail: honeybeebabyshop@gmail.com</p>
226     </footer>
227
228 </body>
229 </html>
```

3.2. CSS

O CSS, como dito anteriormente, é utilizado para a estilização do arquivo HTML. Inicialmente, é utilizado o Seletor Universal (*), na qual vai definir margin e padding para 0. Em seguida, tem-se os estilos para o cabeçalho (header), para a Logo e para a barra de buscas, melhorando a aparência para o usuário. O código pode ser visto na imagem abaixo:

```
# style.css > ...
1  /* Reset de estilos */
2  * {
3      margin: 0;
4      padding: 0;
5  }
6
7  /* Estilos para o cabeçalho */
8  header {
9      display: grid;
10     grid-template-columns: 1fr 2fr 1fr; /* Define três colunas */
11     align-items: center;
12     padding: 20px;
13     background-color: #f5f5f5;
14     border-bottom: 1px solid #ccc;
15 }
16
17 .Logo {
18     justify-self: start; /* Alinha o logo à esquerda */
19 }
20
21 .Logoimg{
22     width: 210px; /* Defina o tamanho desejado */
23     height: auto; /* Mantenha a proporção */
24     padding-right: 10px;
25 }
26
27 .Busca {
28     display: flex;
29     justify-content: center; /* Alinha o campo de busca ao centro */
30     background-color: #ffffff;
31     padding: 5px;
32 }
```

O seletor input[type="text"] é utilizado para estilizar especificamente os elementos de entrada do tipo texto (input). O seletor ".carrinho" é utilizado para estilizar a classe carrinho no documento HTML. A tag <h1> é uma tag de cabeçalho em HTML usada para representar as seções da página da web (total de 3). O seletor ".container main" é usado

para estilizar o contêiner principal do conteúdo principal (<main>). O código pode ser visto na imagem abaixo:

```
34 /* Estilos adicionais para os elementos */
35 input[type="text"] {
36   border: transparent;
37   background-color: #ffffff;
38   width: 100%;
39   min-height: 40px;
40   min-width: 200px;
41 }
42
43
44 .carrinho {
45   justify-self: end; /* Alinha o carrinho à direita */
46 }
47
48
49 h1 {
50   padding: 20px;
51   border-radius: 10px;
52   color: #8D4F04;
53 }
54
55
56 .container main {
57   display: grid;
58   place-items: center;
59   grid-template-columns: 1fr; /* Uma coluna */
60   gap: 20px; /* Espaçamento entre os itens */
61   padding: 20px;
62   background-color: #f5f5f5;
63 }
64
```

As propriedades de estilo ajudam a tornar o menu visualmente fácil de navegar para os usuários:

- .menu li: Define o estilo para os itens de lista do menu;
- .menu a: Define o estilo para os links dentro do menu;
- .menu li a: Define o estilo para os links dentro dos itens de lista do menu;
- .menu a:hover e .menu li a:hover: Define o estilo dos links quando é passado o mouse sobre eles;

O código pode ser visto na imagem abaixo:

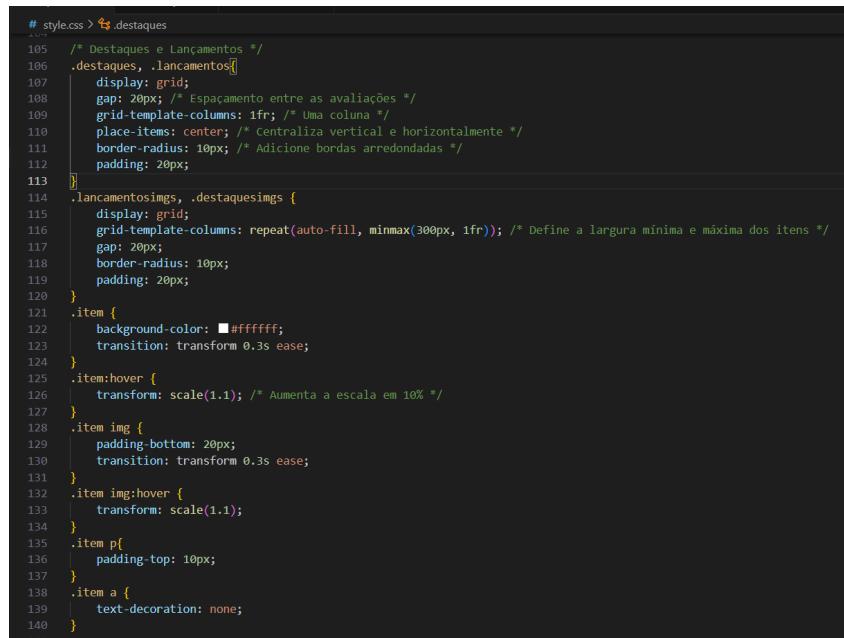
```
# style.css > ↗ .Logoimg
65
66 /* Menu */
67 .menu {
68   text-align: center;
69   background-color: #FFD230;
70   border-bottom: 3px solid #8D4F04;
71 }
72
73
74 .menu li {
75   display: inline-block;
76 }
77
78 .menu a {
79   font-size: 18px;
80   transition: transform 0.3s ease;
81   background-color: #FFD230;
82 }
83
84 .menu a:hover {
85   transform: scale(1.1); /* Aumenta a escala em 10% */
86 }
87
88 .menu li a {
89   text-decoration: none;
90   display: block;
91   padding: 20px;
92   padding-left: 100px;
93   padding-right: 100px;
94   background-color: #FFD230;
95   color: #8D4F04;
96 }
97
98 .menu li a:hover {
99   background-color: #8D4F04;
100  color: #FFD230;
101 }
```

.destaques, .lancamentos: aplicam estilos comuns para os contêineres de destaques e lançamentos. Eles são estilizados como grids, com espaçamento entre as células, bordas arredondadas, e são centralizados tanto vertical quanto horizontalmente.

.lancamentosimgs, .destaquesimgs: Estes seletores aplicam estilos aos contêineres que envolvem as imagens dos produtos de destaques e lançamentos. Eles são estilizados como grids, permitindo que as imagens sejam exibidas em colunas responsivas, com largura mínima de 300px e ocupando o espaço disponível. Eles também têm espaçamento entre as células e bordas arredondadas.

.item: Este seletor aplica estilos aos itens individuais (produto) dentro dos contêineres de destaques e lançamentos. Eles têm uma transição suave quando o mouse passa sobre eles, causando um aumento de escala. Além disso, possuem um fundo branco e espaçamento interno.

.item img, .item p, .item a: Estes seletores estilizam as imagens os parágrafos e os links, respectivamente dentro dos itens de produto. O código pode ser visto na imagem abaixo:



```
# style.css > 4. destaque
105 /* Destaques e Lancamentos */
106 .destaques, .lancamentos{
107   display: grid;
108   gap: 20px; /* Espacamento entre as avaliações */
109   grid-template-columns: 1fr; /* Uma coluna */
110   place-items: center; /* Centraliza vertical e horizontalmente */
111   border-radius: 10px; /* Adicione bordas arredondadas */
112   padding: 20px;
113 }
114 .lancamentosimgs, .destaquesimgs {
115   display: grid;
116   grid-template-columns: repeat(auto-fill, minmax(300px, 1fr)); /* Define a largura mínima e máxima dos itens */
117   gap: 20px;
118   border-radius: 10px;
119   padding: 20px;
120 }
121 .item {
122   background-color: #ffffff;
123   transition: transform 0.3s ease;
124 }
125 .item:hover {
126   transform: scale(1.1); /* Aumenta a escala em 10% */
127 }
128 .item img {
129   padding-bottom: 20px;
130   transition: transform 0.3s ease;
131 }
132 .item img:hover {
133   transform: scale(1.1);
134 }
135 .item p{
136   padding-top: 10px;
137 }
138 .item a {
139   text-decoration: none;
140 }
```

.preco, .Marca, .Descricao: Estes seletores estilizam os elementos de texto que representam o preço, a marca e a descrição dos produtos, definindo cores e tamanhos de fonte específicos.

.lancamentosimgs > div, .destaquesimgs > div: Estes seletores estilizam os contêineres individuais que envolvem os itens de produto. Eles têm uma borda sólida e arredondada, com preenchimento interno e alinhamento centralizado.

.lancamentos img, .destaques img: Estes seletores estilizam as imagens de produtos dentro dos contêineres de destaques e lançamentos, garantindo que elas tenham uma largura máxima de 100% e uma altura automática para uma exibição responsiva. O código pode ser visto na imagem abaixo:

```

202
142 .preco {
143   color: #8C52FF;
144   font-size: 20px;
145   font-weight: bold;
146 }
147 .Marca{
148   color: #8D4F04;
149   font-weight: bold;
150 }
151 }
152 .descricao{
153   color: #8D4F04;
154   text-align: justify;
155 }
156 }
157 .lancamentosimsg > div, .destaquesimsgs > div {
158   border: 1px solid #rgb(222, 222, 222);
159   border-radius: 10px;
160   padding: 20px;
161   text-align: center;
162 }
163 }
164 .lancamentos img, .destaques img {
165   max-width: 100%;
166   height: auto;
167 }
168 }

```

O “.avaliacoes” é um seletor que aplica estilos ao contêiner geral que envolve todas as avaliações. O “.avaliacao” é um seletor que estiliza cada avaliação individual. O “.section-usuario-username” é um seletor que estiliza o contêiner que envolve o nome do usuário e a imagem do avatar. O “.texto” é um seletor que estiliza o texto das avaliações. O “.imagem-avaliacao-estrelas” é um seletor que estiliza a imagem das estrelas de avaliação. O código pode ser visto na imagem abaixo:

```

202 /* Avaliações */
172 .avaliacoes {
173   display: grid;
174   gap: 20px;
175   grid-template-columns: 1fr;
176   place-items: center;
177   border-radius: 10px;
178   background-color: #e9e9e9;
179   max-width: 100%;
180   padding: 20px;
181 }
182 }
183 .avaliacao {
184   border: 1px solid #e4e4e4;
185   padding: 20px;
186   border-radius: 10px;
187   max-width: 75%;
188   background-color: #ffc074;
189 }
190 }
191 }
192 }
193 .section-usuario-username {
194   display: flex;
195   align-items: center; /
196 }
197 .texto {
198   text-align: justify;
199 }
200 .imagem-avaliacao-estrelas {
201   width: 100px;
202   height: auto;
203 }

```

Por fim, o footer está relacionado à estilização do rodapé da página. O código pode ser visto na imagem abaixo:

```

204
205 footer{
206   display: flex;
207   flex-direction: column;
208   align-items: center;
209   background-color: #ebebeb;
210   color: #8D4F04;
211   font-family: 'Times New Roman', Times, serif;
212   padding: 20px;
213   line-height: 1.5;
214 }
215
216

```

3.3. JavaScript

Para iniciar um servidor HTTP e interagir com o sistema de arquivos em um aplicativo Node.js, primeiro, importamos os módulos necessários. Utilizamos const http = require('http'); para importar o módulo HTTP, que nos permite criar servidores HTTP. Em seguida, usamos const fs = require('fs'); para importar o módulo File System, que permite interagir com o sistema de arquivos do servidor. Por fim, importamos o módulo Path usando const path = require('path');, o qual fornece utilitários para lidar com caminhos de arquivos e diretórios. Com esses módulos importados, é permitido que o Node.js possa lidar com solicitações HTTP e manipular arquivos no sistema de arquivos do servidor.

O código apresenta a criação de um servidor HTTP utilizando a função createServer() do módulo HTTP em Node.js. Essa função recebe uma função de retorno de chamada que será acionada sempre que uma requisição HTTP for feita para o servidor. Em seguida, são realizadas verificações utilizando if e else if para determinar o tipo de requisição. Se a URL requisitada for a raiz do servidor ou a página inicial (index.html), a função serveFile() é chamada para servir o arquivo index.html com o tipo de conteúdo text/html. Se a URL for para o arquivo CSS, a função serveFile() é chamada para servir o arquivo style.css com o tipo de conteúdo text/css. Para tratamento de imagens, a lógica encaminha para a função serveFile() para servir a imagem com o tipo de conteúdo determinado pela função getImageContentType(), que vai ser explicada posteriormente.

Por fim, se nenhuma das condições anteriores for atendida, a lógica no bloco else é acionada para lidar com rotas não encontradas, retornando uma resposta 404 (não encontrado).

```
JS server.js > [o] server > ⌂ http.createServer() callback
1  const http = require('http');
2  const fs = require('fs');
3  const path = require('path');
4
5  const server = http.createServer((req, res) => {
6      // Lógica do servidor
7      if (req.url === '/' || req.url === '/index.html') {
8          // Lógica para servir a página inicial
9          serveFile('index.html', 'text/html', res);
10     } else if (req.url === '/style.css') {
11         // Lógica para servir o arquivo CSS
12         serveFile('style.css', 'text/css', res);
13     }
14
15     // Lógica para servir arquivos de imagem para cada pasta
16     else if (req.url.startsWith('/Logo/')) {
17         const imagePath = path.join(__dirname, req.url);
18         serveFile(imagePath, getImageContentType(imagePath), res);
19     } else if (req.url.startsWith('/Produtos/')) {
20         const imagePath = path.join(__dirname, req.url);
21         serveFile(imagePath, getImageContentType(imagePath), res);
22     } else if (req.url.startsWith('/Avatares/')) {
23         const imagePath = path.join(__dirname, req.url);
24         serveFile(imagePath, getImageContentType(imagePath), res);
25     }
26
27     else {
28         // Lógica para lidar com rotas não encontradas
29         res.writeHead(404);
30         res.end('Página não encontrada');
31     }
32 });
33
```

'const PORT = 3000' define a porta na qual o servidor irá escutar e 'server.listen(PORT, () => { ... });' inicia o servidor e faz com que ele escute as requisições na porta especificada. Quando o servidor é iniciado, imprime uma mensagem no console indicando a URL do servidor.

```
33
34 const PORT = 3000;
35 server.listen(PORT, () => {
36   console.log(`Servidor rodando em http://localhost:${PORT}`);
37 });
38
```

A ‘function serveFile(filePath, contentType, res)’ define uma função serveFile() para servir arquivos. Esta função lê o arquivo do sistema de arquivos, define o cabeçalho de resposta com o tipo de conteúdo apropriado e envia o conteúdo do arquivo como resposta.

```
39 // Função para servir um arquivo
40 function serveFile(filePath, contentType, res) {
41   fs.readFile(filePath, (err, data) => {
42     if (err) {
43       res.writeHead(500);
44       res.end('Erro interno do servidor');
45     } else {
46       res.writeHead(200, { 'Content-Type': contentType });
47       res.end(data);
48     }
49   });
50 }
```

A ‘function getImageContentType(imagePath)’ define uma função getImageContentType() para determinar o tipo de conteúdo de uma imagem com base em sua extensão de arquivo. Retorna o tipo de conteúdo apropriado com base na extensão do arquivo.

```
51
52 // Função para determinar o tipo de conteúdo da imagem
53 function getImageContentType(imagePath) {
54   const extname = path.extname(imagePath).toLowerCase();
55   switch (extname) {
56     case '.png':
57       return 'image/png';
58     case '.jpg':
59     case '.jpeg':
60       return 'image/jpeg';
61     case '.gif':
62       return 'image/gif';
63     case '.svg':
64       return 'image/svg+xml';
65     default:
66       return 'application/octet-stream';
67   }
68 }
```

3.4. Prints do Site

A seguir, para visualizar o projeto no node.js, é preciso abrir o terminal da pasta que se encontra os arquivos e digitar “node server.js”, apertando Enter logo em seguida. Dessa forma, irá mostrar que o servidor está rodando em http:localhost:3000, ou seja, que ele está rodando na porta 3000.

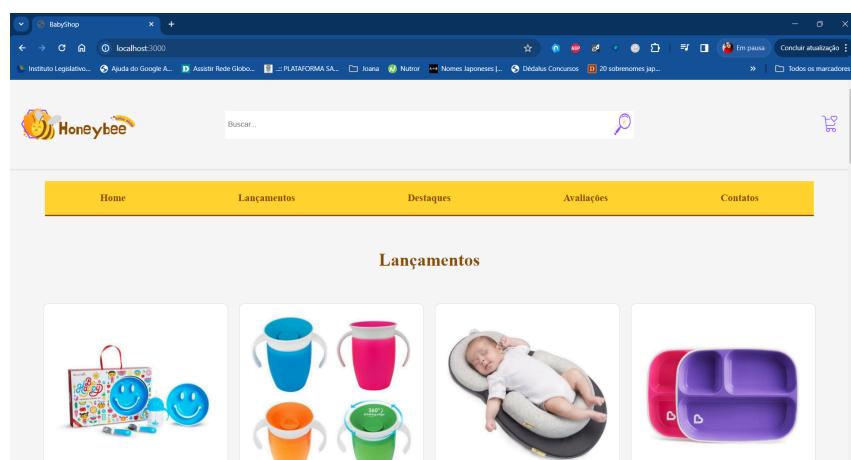
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS GITLENS

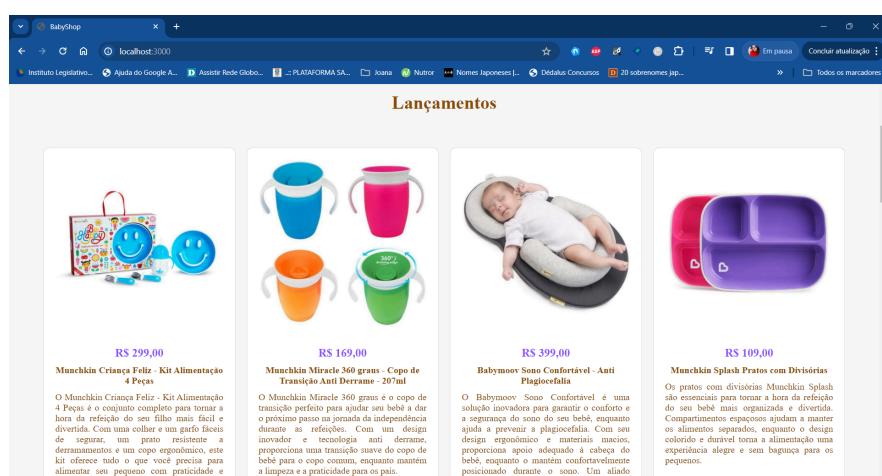
PS C:\Users\joana\Documents\GitHub\BabyShop> node server.js
Servidor rodando em http://localhost:3000

```

Segundo o link disponibilizado, tem-se a visualização do site. Na imagem abaixo mostra a Home da página, com a header e o Menu.



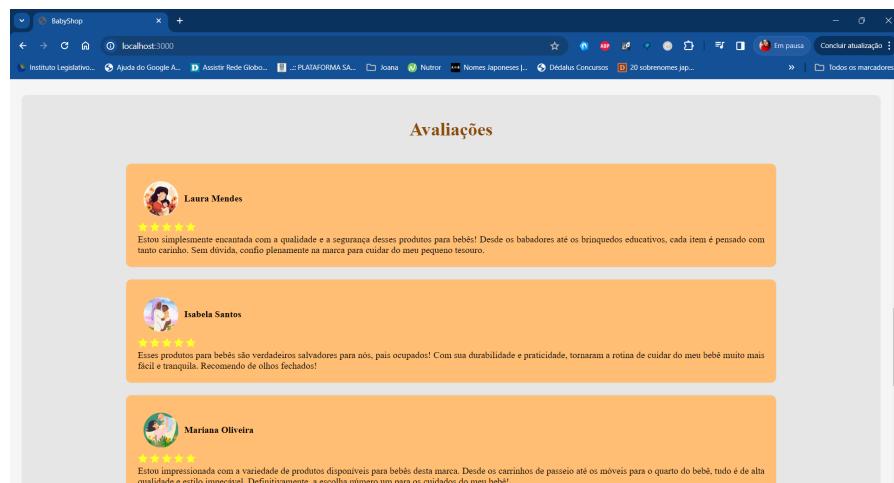
Descendo um pouco, tem-se a seção de Lançamentos. Importante ressaltar que por possuir hover nos links tanto do Menu quanto nos itens dos lançamentos e destaque, quando o mouse passar por cima, ele irá mudar. Seção Lançamentos na imagem abaixo:



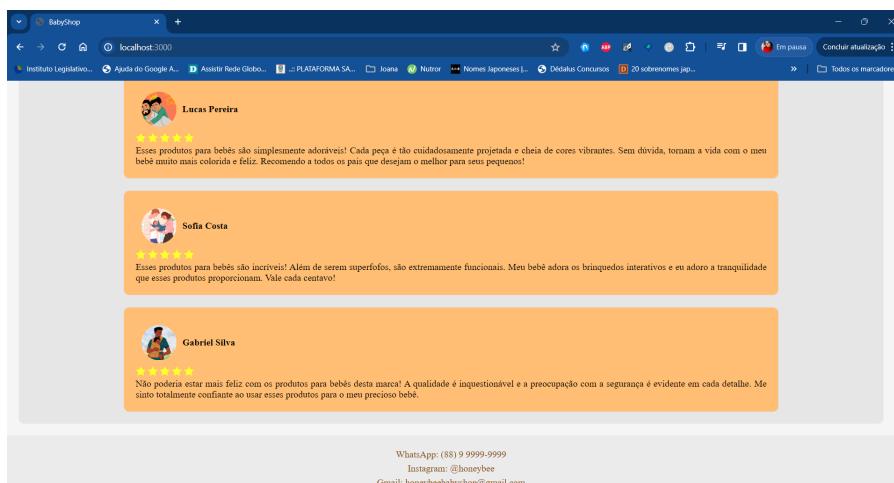
A Seção Destaques faz uso da mesma lógica da Seção Lançamentos. Ela se encontra na imagem abaixo:



Descendo mais um pouco teremos as avaliações, como visto na imagem abaixo:



Por fim, descendo até o fim da página, tem-se o Rodapé do site com as informações de contato da loja.



4. REFERÊNCIAS

ARIANE. O que é CSS? Guia Básico de CSS. Hostinger, 12 de dez. de 2022.. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>>. Acesso em: 30 de mar. de 2024.

CARLOS. O que é Node.js?. Hostinger, 07 de mar de 2024. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-node-js?ppc_campaign=google_search_generic_hosting_all&bidkw=defaultkeyword&lo=1001540&gad_source=1&gclid=CjwKCAjwqmwBhBVEiwAL-WAYTVBW_cWuOU4RuQhUgIR-ge4EqqUWROOhpiBDThaNiStrqf71qyYHhoCEuIQAvD_BwE>. Acesso em: 30 de mar. de 2024.

DE SOUSA, Ivan. O que é HTTP?. Rock Content 23 de jan. de 2019. Disponível em: <<https://rockcontent.com/br/blog/http/>>. Acesso em: 30 de mar. de 2024.

MOZILLA Developer Network. Primeiros passos em JavaScript - O que é JavaScript?. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript>. Acesso em: 30 de mar. de 2024.

MOZILLA Developer Network. HTML: Hypertext Markup Language. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 30 de mar. de 2024.