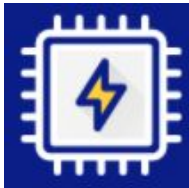


# Relatório sobre a Loja Virtual

## Introdução ao Desenvolvimento Web

Professora: Kamila Rios

Grupo 8: E-Commerce de componentes eletrônicos



### Alunos:

Daniel Carvalho Dantas  
Eduardo Figueiredo Freire Andrade  
Yann Amado Nunes Costa  
Leonardo dos Santos Marcondes

### nUSP

10685702  
11232820  
10746943  
10748476

## 1. Requerimentos

Para o nosso projeto de loja virtual, alguns componentes mostraram-se extremamente necessários para uma boa interação usuário-interface. Para que essa fosse o mais fluida e intuitiva possível, optamos por orientar o consumidor durante o seu processo de compra, deixando-o informado da situação em que se encontra no momento. Dessa forma, necessitou-se implementar páginas sequenciais que mostram de maneira lógica o processo de compra, além de indicações claras do processo.

Além disso, maneiras de criar ou deletar produtos também se mostraram essenciais para a manutenção geral do site, como uma loja de componentes, produtos novos entram em estoque assim como produtos antigos acabam, e por isso, uma maneira dinâmica de manipular esses dados necessitou-se ser planejada; mais detalhes da implementação serão vistos na descrição do projeto.

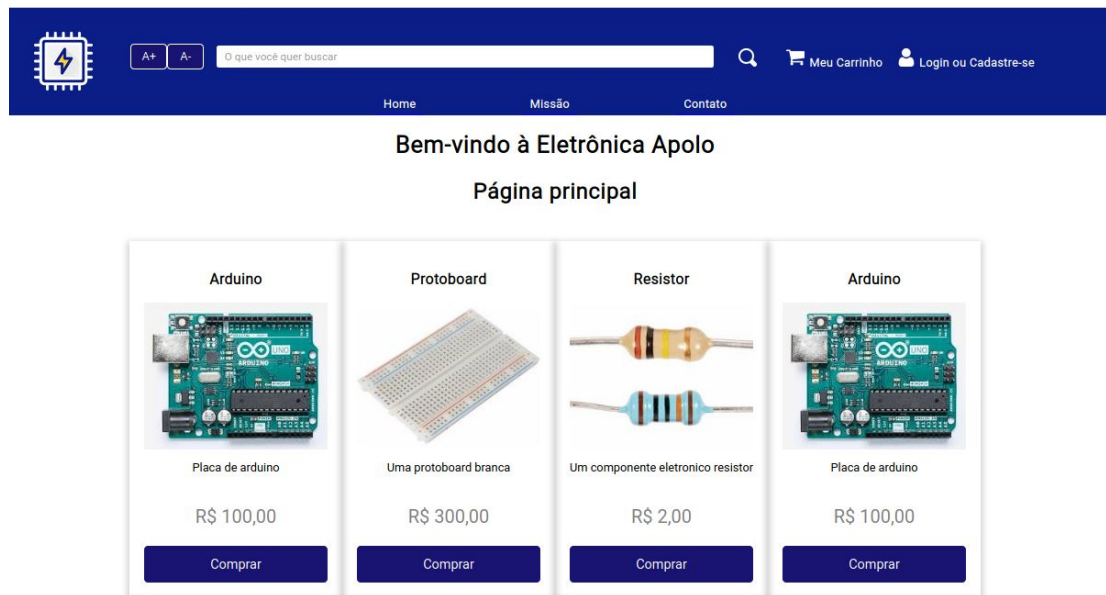
Ademais, para conseguirmos ser o mais inclusivo o possível com o público-alvo, aspectos de acessibilidade foram adicionados, como botões, uso de atls, e maneiras de navegar pela página com o teclado; conceitos os quais serão mais explorados no próximo tópico.

Como componentes de eletrônica requerem, muitas vezes, cuidados e atenções especiais ao serem usados, implementamos links dentro da página do produto os quais levam o usuário para vídeos de projetos os quais o componente em questão foi utilizado. Dessa forma, garantimos uma implementação com menos riscos e maior satisfação do cliente ao não deixá-lo desamparado quanto ao uso, além de fornecer uma explicação prática quanto aos benefícios do produto.

## 2. Descrição do projeto

Para implementar as funcionalidades propostas, a seguinte sequência de páginas foi idealizada:

1- Na própria página inicial (home) aparecem todos os produtos disponíveis na loja para serem selecionados pelo consumidor.



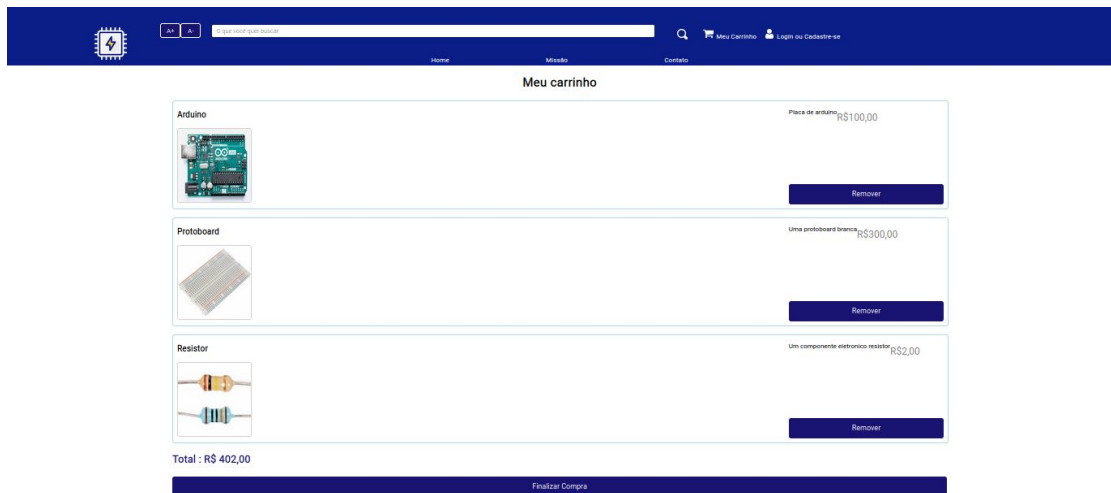
**Figura 1 - Página Principal da Loja.**

2- Ao selecionar qualquer produto, o consumidor é encaminhado para a página do mesmo, onde pode optar pela escolha das quantidades daquele produto, o que se mostra imprescindível para o nicho de eletrônicos, os quais são comprados muitas vezes em dezenas. Além disso, links para vídeos que descrevem como o projeto pode ser usado ajudam o consumidor a se situar mais daquilo que está comprando, e ainda, mostra ao consumidor conceitos importantes sobre normas de segurança acerca do componente.



**Figura 2 - Página do Produto.**

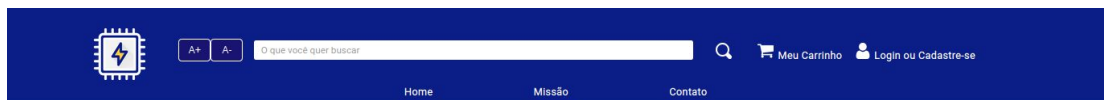
3- Após selecionar o produto e a quantidade, o consumidor é redirecionado para o seu carrinho, onde poderá optar por remover quaisquer produtos, seguir para a aba de produtos novamente, ou finalizar a sua compra.



**Figura 3 - Página do Carrinho.**

4- Em seguida, o usuário é redirecionado para a página de cadastro do endereço e finalmente ao pagamento, terminando assim, o seu processo de compra.

**Figura 4 - Página de Cadastro do Endereço.**



### Pagamento

Número do Cartão
<input type="text" value="Digite o número do cartão"/>
CVV
<input type="text" value="CVV"/>
Validade
<input type="text" value="Validade"/>
Nome do Titular
<input type="text" value="Digite o nome do titular"/>
<input type="button" value="Avançar"/>

**Figura 5 - Página de Pagamento.**

O fluxo de cadastro de endereço e pagamento indicado acima é válido para quando o cliente já efetuou o login, quando isso ainda não ocorreu ou o cliente ainda não possui cadastro, ele é redirecionado para página de login/cadastre-se da figura 6, antes de acessar as páginas das figuras 4 e 5.



### Entrar

E-mail
<input type="text" value="Digite seu e-mail"/>
Senha
<input type="text" value="Digite sua senha"/>
<input type="button" value="Login"/>

E-mail
<input type="text" value="Digite seu e-mail"/>
Senha
<input type="text" value="Digite sua senha"/>
Confirmar senha
<input type="text" value="Confirme sua senha"/>
<input type="button" value="Cadastrar"/>

**Figura 6 - Página de Login/Cadastro.**

Os ícones de Missão e Contato direcionam o cliente para as páginas indicadas nas figuras 7 e 8, as quais hospedam informações relevantes sobre a história, papel social e mercadológico da Eletrônica Apolo, além dos dados para contato, caso o cliente deseje realizar algum orçamento ou solicitação especial.



## Missão

Somos a Eletrônica Apolo e nossa missão é trazer o melhor dos produtos eletrônicos para nossos clientes. Estamos na ativa desde ontem, quando o front foi terminado para a matéria de introdução a desenvolvimento web.

Temos uma grande variedade de Hardware para aqueles que se interessam, desde uma grande variedade de micros como Arduinos, Raspberry Pi, além de vários shields para eles, para que nossos clientes possam desenvolver os mais diversos projetos profissionais e pessoais. Além disso, temos um vasto estoque de resistores, capacitores, indutores, protoboards e até mesmo osciloscópios.

Além de componentes básicos para circuitos RLC's, protoboards e microcontroladores, oferecemos uma grande variedade de componentes de suporte, como pilhas, cabos, chaves e fontes, essenciais para projetos básicos.

**Figura 7 - Página da Missão.**



## Contato

Nosso e-mail: [eletronicaapolo@eletro.com.br](mailto:eletronicaapolo@eletro.com.br)  
 Nosso telefone: (16)3333-3333

**Figura 8 - Página do Contato.**

Para o cadastro de produtos, o usuário com permissão de administrador, a qual é concedida a todos aqueles que se cadastram com o email [@eletronica.apolo.com.br](mailto:@eletronica.apolo.com.br) será reencaminhado para uma página que o permitirá realizar a inserção de produtos; será nesse local em que o front e o back end irão interagir para que a loja se mantenha sempre atualizada quanto aos seus produtos, permitindo cadastrar os conteúdos: nome, valor, quantidade, link para um projeto, código e descrição, conforme indicado na figura 9. Caso o usuário se cadastre sem essa terminação, ele será cadastrado como um usuário normal e será redirecionado para a parte de produtos (Aba Home).

### Incluir Produtos

Nome do Produto

Valor do Produto

Quantidade

Link

Código do Produto

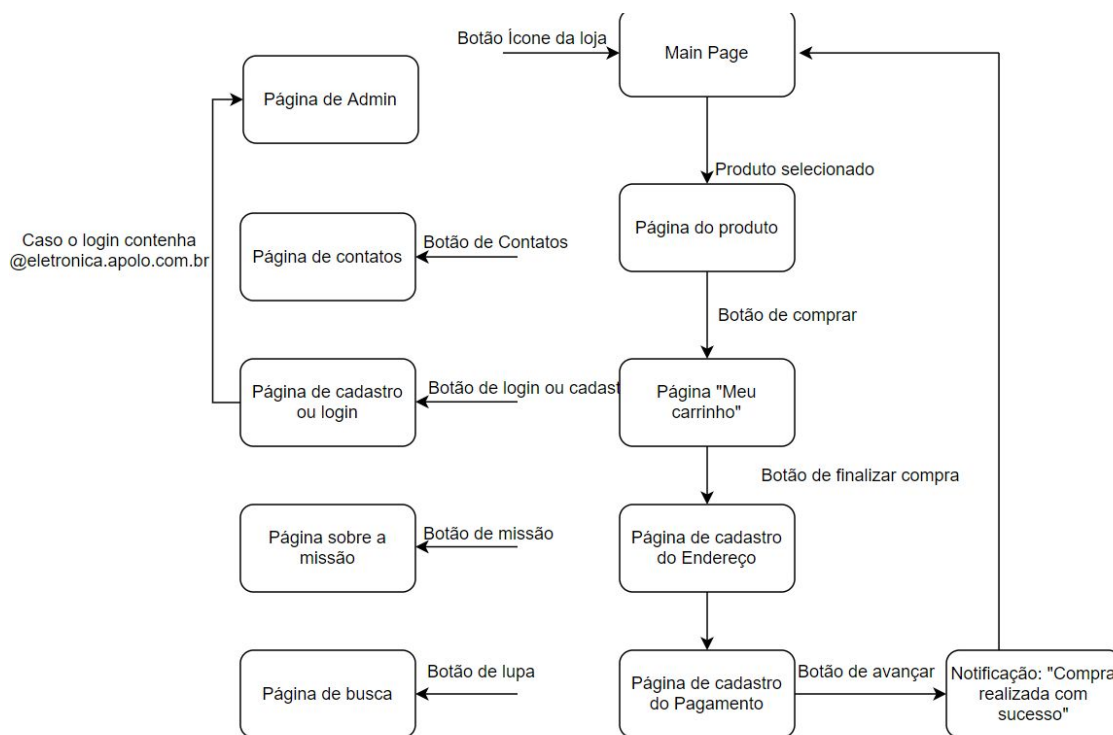
Descrição do Produto

Descreva o produto ...

Cadastrar

**Figura 9 - Formulário para Inclusão de Produtos.**

A figura 10 contém o fluxograma operacional do e-commerce desenvolvido, nele constam as ações que um cliente pode efetuar no site e o resultado de tal ação.



**Figura 10 - Fluxograma Operacional do Site.**

### Aspectos de acessibilidade:

Todas as imagens feitas no site possuem o atributo “alt”, o qual permite que usuários com deficiência visual de alto nível possam entender com maior coesão os conteúdos da página, além disso, botões que aumentam ou diminuem o tamanho da fonte foram incluídos em locais intuitivos e de fácil acesso, tanto por mouse quanto por teclado, para que usuários com outras dificuldades possam navegar com menor esforço pela aplicação. O botão pode ser visto ao lado do logo da loja, note em especial o posicionamento estratégico para fácil acesso, usuários de teclado necessitam, simplesmente, pressionar a tecla tab 2 vezes para chegar no botão de aumento da fonte.

```
<img className="carProdImg" src={produto.imagemProduto} alt={produto.nomeProduto}/>
```

### Exemplo de um Comando “alt”



**Figura 11 - Botões para mudança de fonte**

**Obs:** O arquivo index.html encontra-se no seguinte caminho :

Trabalho-Web-React-main\trabalho\public

### 3. Comentários sobre o código

#### Front End :

Para o frontend, o framework utilizado foi o react (<https://pt-br.reactjs.org/>) devido às suas características principais de poder mudar dados sem ter que recarregar a página (facilitando assim a implementação de novos elementos e manutenção da página), da sua rapidez, e do seu leque variado de bibliotecas disponíveis, o que permitiu um ótimo andamento do projeto durante a sua criação. Uma dessas bibliotecas usadas para facilitar a prototipagem foi o “React-Helmet” (<https://www.npmjs.com/package/react-helmet>), a qual permite uma maneira inteligente de lidar com problemas de duplicação de códigos Header entre elementos pais e elementos filhos da seguinte maneira: Caso o usuário esteja em uma página que contenha o elemento header pai mas não contenha o elemento header filho, ela irá sobrepor o comando pai sobre o filho; caso o usuário esteja na página do elemento filho, ela irá sobrepor, agora, o comando filho sobre o pai. Essa biblioteca permite que os programadores organizem melhor o seu código sem se preocupar com dependências entre os vários elementos contidos dentro do site.

```
import {Helmet} from "react-helmet"
```

#### Comando para importar o React Helmet.

Para definir as rotas das páginas, o React Router Dom foi utilizado. A biblioteca foi escolhida para fazer a interação entre os componentes do site e o browser, através de mudanças na URL, a renderização de diferentes componentes, e manter a página integrada com os links do site.

```
import {BrowserRouter} from "react-router-dom"
```

#### Comando para importar o React Router Dom.

A fim de melhorar a legibilidade e a documentação do código, o typescript (<https://www.typescriptlang.org/>) foi empregado. Esse programa serve para adicionar linguagem estática ao Javascript, o que permite o editor de texto (Em especial o Visual Studio Code <https://visualstudio.microsoft.com/pt-br/>) a indicar possíveis erros ou quais os componentes aquele elemento possui dentro dele, tornando o processo de implementação do código com menos erros, maior rapidez e praticidade.

Por último, para poder mandar as requisições para o backend foi utilizado o Axios, uma biblioteca do Javascript, a qual consiste em uma solução simples para fazer a ponte entre o front e o back end. Ela permite que requisições HTTP sejam feitas a

partir do node.js para o navegador, transmitindo assim, as informações requisitadas do backend.

## Back End :

Para o backend selecionamos o modelo de usuários, carrinho e produto. Os usuários podem fazer login e se cadastrar dentro do banco de dados do próprio site. Diferentes de outros dados, senhas são conteúdos sensíveis as quais podem ser facilmente achadas dentro do banco de dados do site; para adicionar segurança a esse conteúdo sensível, um processo de encriptação foi implementado para poder solucionar esse problema.

```
{ "_id" : ObjectId("5fd3d3b88e5f5f721f6e0c2f"), "email" : "yannzao@usp.br", "senha" : "$2b$10$srrowhdiNFs/vrCtAD0GuhNNFzU1I/3Jl1QXhIGnp9Q06trun9mq", "admin" : false, "criadoEm" : ISODate("2020-12-11T20:16:56.736Z"), "__v" : 0 }
```

**Figura 12 - Senha Criptografada.**

Além disso, os cadastros podem ser feitos como administrador, caso o email cadastrado contenha a terminação "@eletronica.apolo.com.br". Usuários cadastrados com esse email terão permissão de administrador e serão redirecionados para a página do mesmo,

Os arquivos para os controllers de usuário, carrinho e produto foram estruturados em promises, contudo, no arquivo de usuário, para o cadastro e login, foi utilizado o async await. Essa estrutura foi utilizada nos vídeos disponibilizados e, por não conseguir deixar o funcionamento da mesma forma em promises, elas não foram usadas para essas duas funções.

## 4. Procedimentos de Build

Para iniciar o repositório, são necessárias primeiramente as seguintes instalações. (Obs: o usuário deve seguir as instruções adequadas para o seu sistema operacional. Como o windows é o SO mais comum, ele será apresentado):

**git:** Para instalar o git:

1. Entre no site [git-scm.com](https://git-scm.com) ;
2. Clique em downloads e siga as instruções ;

**npm:** Para instalar o npm:

1. Acesse o site <https://nodejs.org/en/download/> ;
2. Faça o download do windows installer e siga as instruções ;
3. Abra o prompt de comando e verifique se a aplicação foi devidamente instalada usando o seguinte comando: `node -v`. Caso a instalação tenha sido feita com sucesso, a sua versão do Node.js deve ter sido apresentada. Certifique-se de que esta seja a versão mais recente da aplicação.

**Mongo:** Para instalar o mongo:

1. Acesse o site <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>;



2. Clique em MongoDB Download Center e siga as instruções

Após instalar os programas supracitados, basta executar as seguintes operações no prompt de comando:

**git clone:** Clonagem dos arquivos para o repositório desejado em seu computador.

**Terminal:** git clone <https://github.com/Eduardofig/Trabalho-Web-React.git> ;

## Front End :

**npm install:** Entrando no diretório trabalho (após entrar dentro da pasta inicial, será a segunda de cima para baixo) e aplicando esse comando, são instaladas as dependências do projeto. Esse comando só funciona no diretório onde está o arquivo package.json.

**npm start:** Inicia o projeto em seu navegador na home page. É importante ressaltar que tanto o npm install quanto o npm start devem ser feitos no diretório trabalho.

**yarn start (Alternativa):** O comando yarn start serve como pack manager, similar ao npm start. Porém, ela garante maior rapidez na inicialização do projeto, além de permitir que mais imagens sejam inicializadas. Para empregá-lo, basta utilizar o comando *npm install -g yarn* no prompt de comando antes do comando principal.

## Back End :

**npm install:** Entrando no diretório back end e aplicando esse comando, são instaladas as dependências do projeto. Esse comando só funciona no diretório onde está o arquivo package.json

**npm start:** Inicia o projeto em seu navegador na home page. É importante ressaltar que tanto o npm install quanto o npm start devem ser feitos no diretório back end.

**mongo:** Para a database do mongo, insira o seguinte código em um outro terminal (diferente daquele que foi utilizado no npm start) ou dentro do virtual studio code:

```
mongo "mongodb+srv://eletronica-apollo.63ug4.mongodb.net/Eletronica-Apollo"
--username admin
```

A senha para acessar é: pivete.

Após informada a senha é possível verificar as collections dessa forma:

- 1 - show dbs
- 2 - use electronica-apollo
- 3 - show collections

Depois desse procedimento, utilizando os comandos `db.users.find()` e `db.products.find()` é possível verificar os novos usuários e produtos registrados.

## 5. Problemas

Durante a construção do site, os maiores problemas foram provenientes do CSS e da barra de navegação. Para a nossa aplicação, diversos elementos necessitam ficar alinhados não só em uma tela convencional, porém deve ser responsivo para qualquer tipo de tamanho de tela; essa questão mostrou-se um grande, pois o alinhamento dos elementos em CSS puro não são relativos entre si, ou seja, não possuem uma unidade a qual liga todos eles em conjunto e dita a posição que eles deveriam ter quando considerados outros elementos, o que tornou tanto a implementação bruta quanto a responsiva bastante trabalhosa. Apesar de existirem diversos elementos para solucionar essa questão (como o uso de bootstraps), o grupo optou por procurar entender o funcionamento das ferramentas disponíveis, ao contrário do uso de bibliotecas já prontas, a fim de melhorar o aprendizado dos membros.

O grupo enfrentou certa dificuldade para implementar os botões de acessibilidade responsáveis por aumentar e diminuir o tamanho da fonte do texto, para que as tags necessárias sofressem alteração. Para vencer essa dificuldade, foi necessário entrar em diversos tratamentos de erro (try/catch) até que obtivéssemos a resposta desejada.

No backend, como informado anteriormente, houve uma dificuldade em manter a estrutura de promises, pois o sistema de hashing das senhas dentro da estrutura não estava funcionando corretamente, e por isso, o grupo acabou optando por manter a estrutura do vídeo disponibilizado com `async await`, a qual mostrou-se extremamente intuitiva e mais fácil de ser manipulada e aplicada.

Como a estrutura do código é integrada e feita em grupo, problemas como códigos duplicados e mudanças em partes do código que afetaram a estrutura do site como um todo estavam presentes. Devido a falta de experiência, muitas vezes mudanças que afetaram parte do código eram feitas e que acabavam por afetar outra estrutura com dependências da primeira; gerando assim muitas procuras por erros e bugs no código.

## 6. Comentários

O projeto, apesar de não estar completo de um ponto de vista de mercado, conseguiu êxito na grande maioria das diretrizes propostas. Entretanto, devido ao tempo curto intrínseco à matéria, alguns aspectos que o grupo gostaria de ter implementado tiveram que ser negligenciados, como por exemplo, a página de compra dos produtos, a qual requer uma página individual para cada produto

cadastrado, o que tomaria tempo em demasiado. Portanto, para que o projeto pudesse se tornar uma aplicação por completo, e não apenas uma ilustração das capacidades dos sistemas web, diversos escopos do projeto que já estão implementados deveriam ser expandidos, como páginas adicionais, sistemas para remoção e alteração de produtos, páginas interativas, dentre outros, deveriam ser aplicadas. Além disso, devido ao tempo, não foi possível que um usuário visse um produto imediatamente após seu cadastro, o que exigiria outra conexão além das já feitas entre back e front end para o *map* dos produtos na página principal.

Entender a forma que os diferentes usuários interagem pela internet também foi uma experiência enriquecedora, descobrir as formas pelas quais as pessoas lidam com determinados problemas e como nós como desenvolvedores podemos fazer para amenizar os impactos de diversas condições se mostrou um desafio prazeroso, que permitiu a formação de um pensamento mais aberto quanto a necessidade da acessibilidade para todos, não só na internet, mas sim na vida como um todo.

Durante a realização do projeto, algumas dificuldades surgiram como quando da implementação do css, algo que tomou algum tempo, já que a correta formatação da página exigiu diversos ajustes e definições de parâmetros que para iniciantes nem sempre são óbvios ou imediatos. Nesse ínterim, deve-se destacar o desenvolvimento da barra de navegação, o qual representou grande desafio, uma vez que exigiu o posicionamento de diversos blocos, como botões de login/cadastre-se, barra de busca, logotipo, etc. sem contudo que a página perdesse a responsividade.

Ainda tratando de dificuldades, deve-se salientar o desafio de implementação de todo o backend em um curto intervalo de tempo, algo que apesar de compreensível, dado o contexto de término de aulas, exigiu rapidez e comprometimento extra com a disciplina nesse final de semestre.

Dessa forma, o projeto serviu para consolidar os conhecimentos adquiridos durante a matéria, de maneira a colocar na prática as questões vistas de maneira teórica, o que possibilitou ao grupo aprender com os seus erros e a verem em primeira mão todas as nuances envolvidas no desenvolvimento de um projeto Web, além da necessidade do trabalho em equipe durante um projeto de tamanho escopo.

