

ECOSHOP: E-COMMERCE DE PRODUTOS SUSTENTÁVEIS

Caroline Ramos de Sousa⁽¹⁾, Gabriel Mendes Sousa⁽²⁾, Guilherme Rodrigues dos Santos⁽³⁾, Jorge Lucas Silva Tanan⁽⁴⁾, Samuel Pereira Brandão⁽⁵⁾, Orientador: Professor Dr. Eugênio Akihiro Nassu ⁽¹⁾RA 00250427, ⁽²⁾RA 00250615, ⁽³⁾RA 00249962, ⁽⁴⁾RA 00264010, ⁽⁵⁾RA 00250205

RESUMO

O artigo a seguir apresenta o desenvolvimento de uma aplicação e-commerce (comércio eletrônico) multiplataforma, na qual a problemática apresentada são os problemas ambientais urbanos causados pelo uso excessivo e má reciclagem de produtos prejudiciais ao meio ambiente. Para a solução, o desenvolvimento de um site em que o usuário poderá comprar e vender produtos sustentáveis, com o objetivo de vender itens, objetos, peças e até mesmo serviços dentro das comunidades que necessitam de assentamento humano e ações sustentáveis, para apoiar em relações econômicas, sociais e ambientais positivas entre áreas urbanas. O site será gratuito e conterá anúncios para o seu rendimento e manutenção.

Palavras-chave: E-Commerce de Produtos Sustentáveis, Site de Vendas de Produtos, Site multiplataforma.

1. Introdução

Os problemas ambientais urbanos são fenômenos que impactam diretamente no meio ambiente das cidades. Em muitas situações, esses fenômenos possuem causas naturais, mas que são potencializadas pela ação do homem, assim como pela acentuada transformação da natureza. Os elementos motivadores desses problemas ambientais nas cidades estão centrados em questões relacionadas à intensa modificação do espaço natural pelas atividades produtivas. Sendo alguns deles: poluição, enchentes e deslizamentos de terra. Atualmente, os motivos atrelados aos problemas ambientais urbanos continuam intimamente ligados à ação humana ao meio ambiente. Como por exemplo: poluição da água, ela é ocasionada pela emissão indevida de resíduos poluentes líquidos e sólidos. Poluição do solo, está atrelada à destinação incorreta de resíduos em terrenos diversos. A deposição incorreta de lixo, materiais de refugo e entulhos em lotes e terrenos urbanos é um elemento comum na paisagem das cidades.

Uma das principais consequências dos problemas ambientais urbanos destaca-se o prejuízo ambiental causado por esses fenômenos. Assim como, muitas perdas econômicas e humanas. A chuva ácida, por exemplo, impacta diretamente nas plantações agrícolas [1].

Para a solução, com base na ODS 11 (Objetivo de Desenvolvimento Sustentável) da Nações Unidas, que tem como objetivo tornar as cidades e os assentamentos humanos inclusivos, seguros, resilientes e sustentáveis[2], o site multiplataforma desenvolvido será um e-commerce voltado para pessoas que queiram contribuir para a melhoria do meio ambiente, onde haverá a possibilidade do usuário vender e comprar produtos

recicláveis e sustentáveis de forma autônoma e segura. Consequentemente os usuários poderão firmar relações econômicas, sociais e ambientais de forma positiva entre áreas urbanas amenizando em partes os problemas ambientais urbanos.

Diante dessa problemática, justifica-se o desenvolvimento de um site responsivo e-commerce para atender usuários em várias plataformas (*Desktop, laptop, tablet, mobile*), com o propósito de contribuir de forma positiva com a preservação do meio ambiente, sendo assim amenizando essa problemática.

2. Metodologia

Para realização desse projeto foi elaborado um *brainstorming* (chuva de ideias) [3], que serviu para o grupo discutir e debater as ideias principais bem como as problemáticas atuais que afligem o mundo atual, sendo assim, chegando em um consenso do tema, foram utilizadas pesquisas e referências bibliográficas para o respaldo da problemática levantada.

Para a codificação do *software* foram utilizadas as IDE's (Ambiente de Desenvolvimento Integrado) Visual Studio Code e Spring Tool Suite.

Para o versionamento de código utilizamos o GitHub, da qual é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão, para a administração do projeto. Acesse o repositório do projeto para ter acesso a todas as informações sobre o desenvolvimento da aplicação: <https://github.com/Biellms/ProjetoIntegrador-EcoShop>

As linguagens de programação utilizadas para o desenvolvimento do projeto foram: Java, para o back-end, juntamente com o Spring Framework para o desenvolvimento de uma Rest API, e o MySQL como banco de dados.

Já no front-end, foi utilizada a linguagem de programação JavaScript, uma ferramenta de construção chamada Vite [4] juntamente com a biblioteca React [5] para a consumação da API; Para desenvolvimento da estrutura do front-end foi utilizado a linguagem de marcação HTML (*Hyper Text Markup Language*) e de estilização CSS (*Cascading Style Sheet*).

No desenvolvimento dos fluxogramas e diagrama entidade relacionamento, o Draw.io foi utilizado, da qual é uma plataforma online que serve para construir desenhos gráficos.

Nas entregas contínuas do artigo em si, e das tarefas de desenvolvimento do projeto, foi utilizada a metodologia ágil *SCRUM*, que por sua vez auxiliou o grupo a entender os principais requisitos e fornecer as funcionalidades de forma estratégica, rápida e funcional, oferecendo uma nova versão funcional do projeto a cada entrega [6].

Algumas disciplinas foram essenciais para o desenvolvimento do aplicativo, sendo elas: Sistemas distribuídos, interfaces digitais Front-end, Programação Back-end, Programação de aplicativos para dispositivos móveis e Metodologia da pesquisa científica.

3. Desenvolvimento

Para o começo do desenvolvimento do Back-end, foi estabelecido a criação de um banco de dados relacional com as seguintes tabelas “Usuários”, “Produtos” e “Categorias”, relacionando-se de tal forma: um usuário poderá ter vários produtos, um produto poderá ter um usuário e uma categoria, e por fim uma categoria poderá ter diversos produtos, conforme a figura do fluxograma abaixo:

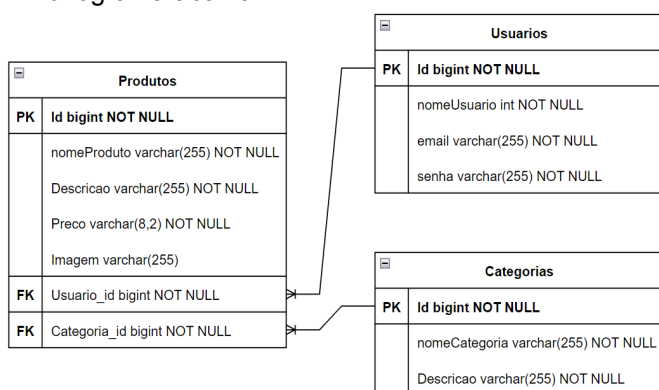


Figura 1 - Diagrama Entidade Relacionamento do EcoShop - Diagrams.net

Com base no fluxograma, foram desenvolvidas as rotas para que o usuário possa percorrer os possíveis caminhos na aplicação, tais como, ao iniciar a entrada no site é necessário efetuar o *login* e caso não possua uma conta é necessário realizar o cadastro. Em sequência, será redirecionado à página inicial da aplicação, da qual terá a liberdade de escolher entre vender ou comprar produtos, ou até mesmo efetuar as duas

funcionalidades. Caso o usuário escolha vender um produto, poderá cadastrá-lo, e consequentemente editá-lo ou removê-lo assim como as categorias. Ou então, entrar no feed de produtos que exibirá todos os produtos cadastrados por todos os usuários, adicionar um ou mais produtos que deseja comprar ao carrinho e finalizar a compra. Acesse o fluxograma através do link: <https://github.com/Biellms/ProjetoIntegrador-EcoShop/blob/main/Documentação/Fluxograma%20EcoShop.png>

Para as requisições e validação do usuário dentro da aplicação, foi desenvolvido a seguinte API Rest: *crud* do usuário com validação de *token* e com senha criptografada, *crud* de produtos e *crud* de categorias, relacionando-se de acordo com a figura 1 mencionada anteriormente. Com a utilização do Spring Framework, foi desenvolvido o back-end da aplicação e suas principais regras de negócio, sendo elas: 1 - O usuário deve se cadastrar com um e-mail válido e a senha deve conter no mínimo 8 caracteres; 2 - A senha é armazenada no banco de dados já criptografada utilizando o Spring Security. Segue a imagem do método responsável pela criptografia no back-end:

```

    ecoshop - UsuarioService.java

    1 private String criptografarSenha(String senha) {
    2     BCryptPasswordEncoder encoder = new BCryptPasswordEncoder();
    3
    4     return encoder.encode(senha);
    5 }
  
```

Figura 2 - Método de criptografia - Spring Tool Suite

3 - O *token* é gerado em Base64 e somente quando o usuário fizer o *login* e consequentemente as requisições da aplicação só estarão disponíveis se o *token* estiver válido. Ou seja, o usuário só conseguirá utilizar a API da aplicação se estiver logado, caso contrário, não poderá comprar, vender ou editar os produtos. Segue abaixo método de geração de token no back-end:

```

    ecoshop - UsuarioService.java

    1 private String gerarBasicToken(String usuario, String senha) {
    2
    3     String tokenBase = usuario + ":" + senha;
    4
    5     byte[] tokenBase64 = Base64.encodeBase64(tokenBase.getBytes(
    6         Charset.forName("US-ASCII")));
    7     return "Basic " + new String(tokenBase64);
    8 }
  
```

Figura 3 - Método de geração de Basic Token - Spring Tool Suite

Com a criação da API Rest, incluindo seus devidos métodos https: (*get*, *post*, *put* e *delete*), foi usada para testes de requisições cliente/servidor, o Postman, uma ferramenta própria para fazer requisições da API, deste modo, os métodos do *crud* dos serviços estavam acessíveis, como no exemplo da figura abaixo, uma requisição *get* que traz todos os usuários cadastrados em formato JSON já com a senha criptografada:

```
{
  "id": 2,
  "nome": "Administrador",
  "usuario": "admin@email.com.br",
  "senha": "$2a$10$S.haBPGoVqd4GcU9IBLg.e8SVRu.z2KgPE/F.sT",
  "produto": []
}
```

Figura 4 - Get Usuários do EcoShop - Postman

O *build* e o *deploy* da aplicação, foi utilizado um servidor gratuito chamado HEROKU, desta forma, a API deixou de funcionar apenas localmente e começou a ser tornar acessível para qualquer usuário que queira acessar a API [7]. Com a aplicação Back-end hospedada na nuvem, o desenvolvimento da aplicação front-end e sua integração juntamente com o back-end hospedado torna-se simplificada, pois é possível desenvolver o front-end localmente e utilizar a API hospedada na nuvem de forma simultânea.

Para o desenvolvimento do Front-end utilizamos uma ferramenta de construção chamada Vite para otimizar a aplicação, a primeira coisa que ele faz é dividir o código em duas partes: código fonte e dependências. As dependências geralmente são arquivos de Javascript que não sofrerão alterações, então elas são pré-compiladas usando o ESBUILD, que é escrito em Go e é 10 a 100 vezes mais rápido do que *bundlers* feitos com Javascript[8]. E junto ao ViteJS, utilizamos a biblioteca React como template da qual é flexível e facilita a interface com outras bibliotecas e *frameworks*.

Para a consumação da API do e-commerce, criamos um *service* com o uso do Axios que é um cliente HTTP baseado em Promises para fazer as requisições de nossa API [9]. Conforme a figura abaixo, onde é implementado o axios com a URL da API hospedada no HEROKU, implementada dentro da *service* do front-end:

```
FrontEnd - Service.ts
1 export const api = axios.create({
2   baseURL: 'https://ecoshopstore.herokuapp.com'
3 })
```

Figura 5 - Consumação da API Rest - VSCode

A aplicação é separada em componentes acessados por rotas operadas pelo React Router Dom, sendo assim, toda a aplicação é carregada de forma dinâmica e não é necessário recarregar o site a todo instante que acessar alguma outra página [10]. Conforme a figura a seguir todos os componentes já estão incorporados ao acessar o site, além de haver um *path* que por padrão redireciona o usuário diretamente para a *login* caso o *token* não esteja válido implementado na linha 4. E uma página de erro caso o usuário passe alguma informação errada na URL ou possua algum erro de servidor, implementado na linha 12:

```
ecoshop - App.jsx
1 <BrowserRouter>
2   <Navbar />
3   <Routes>
4     <Route path="/" element={<Login />} />
5     <Route path="/login" element={<Login />} />
6     <Route path="/cadastro" element={<Cadastro />} />
7     <Route path="/home" element={<Home />} />
8     <Route path="/comprar" element={<Home />} />
9     <Route path="/vender" element={<Home />} />
10    <Route path="/carrinho" element={<Home />} />
11    <Route path="/sobre" element={<Home />} />
12    <Route path="*" element={<NotFound />} />
13  </Routes>
14  <Footer />
15 </BrowserRouter>
```

Figura 6 - Rotas das páginas - VSCode

Além disso, para o desenvolvimento do carrinho de compras do qual o usuário poderá adicionar os produtos que queira comprar, a utilização do *Context API* foi primordial [11]. Através dela é possível compartilhar *states* diretamente entre dois componentes sem fazer a utilização de *props*, sendo fácil armazenar informações e modificá-las sem fazer uso de um banco de dados, pois é criado um contexto que engloba toda a aplicação (ou alguma parte dela) tornando seus dados e métodos acessíveis de qualquer lugar, conforme a figura abaixo, onde o contexto do carrinho engloba todo o componente da aplicação (*App*) entre a linha 5 e linha 7:

```
ecoshop - main.jsx
1 import { CartProvider } from './contexts/providers/cartcontext'
2
3 ReactDOM.createRoot(document.getElementById('root')).render(
4   <React.StrictMode>
5     <CartProvider>
6       <App />
7     </CartProvider>
8   </React.StrictMode>
9 )
```

Figura 7 - ContextAPI do carrinho englobando todo o App - VSCode

Para a responsividade das páginas a qualquer dispositivo, seja: *desktop*, *tablet* ou *mobile* foi utilizado o sistema de *grid*, através dele, foi realizado a separação entre linhas e colunas da página em si, com isso a separação dos conteúdos e tópicos foram de fácil entendimento e manutenção, além do uso do *grid*, também foi estabelecida a utilização dos recursos do CSS3 chamados *flex-box* e *media query*, com eles, toda a parte do *layout* foi ajustada de forma responsiva. Todas as páginas foram adequadas a cada tamanho de dispositivos, fazendo com que a mesma possa ser acessível independente de qual seja o dispositivo que o usuário esteja utilizando para acessar a aplicação.

Com a finalização da construção de toda parte visual do site responsivo, foi realizada a hospedagem da aplicação no servidor *Netlify* [12], sendo assim, a aplicação foi dividida em dois servidores diferentes, um servidor para hospedar o back-end e o outro servidor para a hospedagem do

front-end, ambos conectados e integrados em servidores diferentes se comunicando entre si.

4. Considerações Finais

De acordo com a dissertação apresentada, essa solução contribuirá no incentivo mútuo na reciclagem de materiais de forma a compensar monetariamente os envolvidos e conscientizá-los sobre a importância da preservação do meio ambiente. Em vista de tudo que foi abordado, essa ferramenta atinge os objetivos ao facilitar o consumo de produtos já utilizados, porém que ainda são úteis e motivam as pessoas a utilizarem produtos quando possível.

Com a utilização dos tópicos do semestre atual, com as disciplinas: Programação Back-End, Programação Front-End e Sistemas Distribuídos, foi obtido sucesso na construção do sistema.

Ao longo do tempo, será avaliado o desempenho e atualizações futuras, tais como: método de pagamento, entregas, e melhorias na venda de serviços sustentáveis. Tendo em vista que a todo momento há incremento de novas tecnologias, existe ainda a possibilidade de novas aplicações sempre com a intenção de levar o usuário à satisfação pessoal em compras e vendas.

Como implementações futuras, será desenvolvido outras funcionalidades para a EcoShop dando-a capacidade de torná-la um site viável e possível como por exemplo a integração das entidades "Carrinho", "Método de pagamento" e "Entrega" em nosso back-end, para gerir da melhor forma possível o funcionamento do e-commerce e transformá-lo em uma aplicação de sucesso. O carrinho é essencial para armazenar os produtos sem que o usuário os percam ao recarregar a página e ter que fazer o processo de adicioná-los novamente. Assim como o método de pagamento que é indispensável em uma aplicação web de vendas, a entrega favorece mais ainda para que o usuário esteja motivado a comprar sem se preocupar em encontrar com outros usuários ou até mesmo sair de casa. E até mesmo utilizar APIs já existentes no mercado.

Além disso, será fundamental desenvolver uma área da qual o usuário poderá gerir suas vendas, onde ficará acessível todos os seus produtos vendidos para ter uma noção de como estão indo as coisas e até mesmo ver reclamações e avaliações. E o mesmo serve para os usuários que estão apenas comprando, desenvolvendo uma área para acompanhar seus pedidos.

Algo importante também é visar a verificação dos produtos cadastrados pelo usuário, para não ocorrer casos de produtos cadastrados totalmente fora da ideologia sustentável, uma boa resolução deste problema seria criar uma validação da qual o produto do usuário antessaria por uma inspeção por no de ir para o feed de produtos passa equipe ao

ser cadastrado, e ficaria algum tempo inativo até ser confirmado sua aptidão em ser um produto sustentável.

O site será totalmente gratuito e futuramente conterá anúncios utilizando o Google AdSense para o seu rendimento e manutenção.

5. Referências

- [1] Educação Mundo. Problemas ambientais urbanos: [site]. Mateus Campos; 2022 [acesso em 13 out 2022]. Disponível em: <https://mundoeducacao.uol.com.br/geografia/os-problemas-ambientais-urbanos.htm#:~:text=Os%20principais%20problemas%20ambientais%20urbanos%20s%C3%A3o%3A%20polui%C3%A7%C3%A3o%2C%20ilhas%20de%20calor,muito%20comuns%20nas%20cidades%20brasileiras>
- [2] Nações Unidas Brasil. ODS 11 Cidades e comunidades sustentáveis: [site]. Nações Unidas Brasil; 2022 [acesso em 13 out 2022]. Disponível em: <https://brasil.un.org/pt-br/sdgs/11>
- [3] Prada Charles. Brainstorming: o que é e como aplicar na geração de novas ideias [publicação online]; 2018 [acesso em 13 out 2022]. Disponível em: <https://www.euax.com.br/2018/09/brainstorming/>
- [4] Vite. Getting Started [site]. 2022 [acesso em 13 out 2022]. Disponível em: <https://vitejs.dev/guide/>
- [5] React. Uma biblioteca JavaScript para criar interfaces de usuário [site]. 2022 [acesso em 13 out 2022]. Disponível em: <https://pt-br.reactjs.org/>
- [6] B. DE PAULA GILLES. Tudo sobre Metodologia Scrum: o que é e como essa ferramenta pode te ajudar a poupar tempo e gerir melhor seus projetos [publicação online]. Treasy; 2016 [acesso em 13 out 2022]. Disponível em: <https://www.treasy.com.br/blog/scrum/>
- [7] Dev Center Heroku. Documentation [site]. 2022 [acesso em 13 out 2022]. Disponível em: <https://devcenter.heroku.com/categories/reference>
- [8] Micheletto Camilo. Vite.js - O build tool que vai facilitar a sua vida [publicação online]; 2021. [acesso em 13 out 2022]. Disponível em: <https://dev.to/lixetletto/vite-js-o-build-tool-que-vai-facilitar-a-sua-vida-15ho>
- [9] DEVMEDIA: Consumindo uma API com React.JS e Axios [publicação online]. 2021. [acesso em 17 out 2022]. Disponível em: <https://www.devmedia.com.br/consumindo-uma-api-com-react-js-e-axios/42900>
- [10] DEVMEDIA: React.JS: Criando rotas com React Router Dom [publicação online]. 2021 [acesso em 17 out 2022]. Disponível

em:<https://www.devmedia.com.br/react-js-criando-rotas-com-react-router-dom/42901>

[11] DEVMEDIA: React.JS: Passando dados com Context API. [publicação online].2021 [acesso em 16 out 2022]. Disponível em:

<https://www.devmedia.com.br/react-js-passando-dados-com-context-api/42904>

[12] Docs Netlify. Welcome to Netlify [site]. 2022 [acesso em 13 out 2022]. Disponível em:

https://docs.netlify.com/?_ga=2.243673733.1826432577.1665009617-1753150236.1665009617