



Cruzeiro do Sul Virtual
Educação a distância

Olá, estudante.

A seguir, você dará continuidade ao desenvolvimento da sua solução através dos campos específicos para a resolução dos 3 desafios propostos, lembrando que eles se complementam.

Nome: Yuri de França Cordeiro

RGM: 27205312

Documentação:

Revisite a documentação do projeto desenvolvido na PIT I, faça as atualizações e melhorias necessárias de acordo com os pontos explicitados no material teórico. Ao terminar os ajustes, suba a documentação em um repositório GIT.

Link do repositório:

Codificação:

Na Tabela a seguir insira as informações referentes ao desenvolvimento do código do *front-end* e *back-end*.

Linguagem do <i>Back-end</i>	JS, CSS, HTML
Banco de Dados	LocalStorage
Hospedagem	
Plataforma	WEB
Modo de Codificação	(x) Tradicional () Low-code () No-code
Link do repositório no GitHub com os códigos abertos	
Link da solução em funcionamento	https://drive.google.com/file/d/1QF19vEXo5bPRE-XuSnYiI4RPl39f3MxcR/view?usp=sharing
Link do vídeo narrado (no mínimo 5 min)	

Testes da Solução

Escolha 5 colegas para testar sua aplicação e preencha a tabela a seguir com as informações obtidas:

Nome: Pamela Nascimento	Data do teste: 20/10/2024
O que testou e funcionou: Adicionar os itens no carrinho	
O que testou e não funcionou – O que deve ser corrigido: Quando o nome do produto é muito grande, o valor fica muito em cima dos botões para aumentar e diminuir a quantidade.	
Funcionalidade não testada (faltou ou não foi implementada):	

Nome: Pollyana Furtado	Data do teste: 19/11/2024
O que testou e funcionou: Quando clicamos nos menus ele leva para a página correta	
O que testou e não funcionou – O que deve ser corrigido: A opção de pesquisar não funciona.	
Funcionalidade não testada (faltou ou não foi implementada):	

Nome: Rafael Vilela	Data do teste: 23/10/2024
O que testou e funcionou: Adicionar os produtos, no carrinho, aumentar/diminuir a quantidade e excluir produtos	
O que testou e não funcionou – O que deve ser corrigido: Ao finalizar a compra ele não tem uma página onde inserimos o endereço de entrega ou selecionamos para retirar na loja	
Funcionalidade não testada (faltou ou não foi implementada):	

Nome: Bianca Bernardes	Data do teste: 29/11/2024
-------------------------------	----------------------------------

O que testou e funcionou: Finalizar a compra e ver se os produtos saem do carrinho
O que testou e não funcionou – O que deve ser corrigido: Senti falta de ter um login no site, para que se caso eu quiser repetir o pedido.
Funcionalidade não testada (faltou ou não foi implementada):

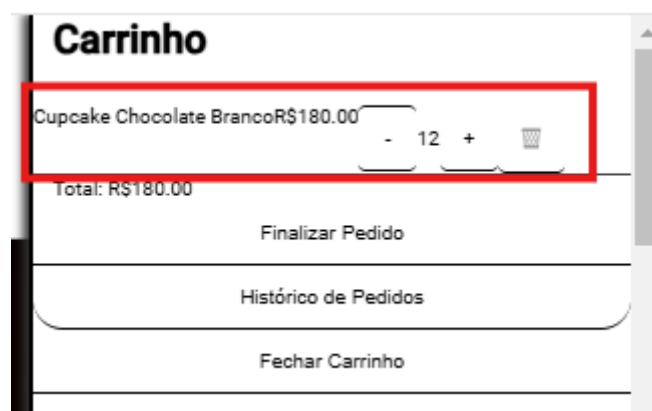
Nome: Ana Lidia	Data do teste: 15/11/2024
O que testou e funcionou: Informações sobre o produto	
O que testou e não funcionou – O que deve ser corrigido: Ao clicar no produto ele deveria falar mais sobre ele, para evitar problemas com alergia ou outras situações ruins	
Funcionalidade não testada (faltou ou não foi implementada):	

Laudo de Qualidade

Insira a seguir o laudo de qualidade do sistema, apontando os erros e as correções. Não esqueça de coletar as evidências para inseri-las no laudo:

Problemas Identificados:

1. Quando o nome do produto é muito grande, o valor fica muito em cima dos botões para aumentar e diminuir a quantidade.



Para solucionar eu criei um **justify-content: space-between;** e um **align-items: center;** e mudei o layout um pouco



2. A opção de pesquisar não funciona. Ao finalizar a compra ele não tem uma página onde inserimos o endereço de entrega ou selecionamos para retirar na loja.



Eu adicionei algumas funções, como:

```
function searchItems() {
  const searchQuery = document.getElementById("search-input").value.toLowerCase().trim();
  const suggestionsContainer = document.getElementById("suggestions");

  // Limpa as sugestões a cada nova busca
  suggestionsContainer.innerHTML = "";

  if (!searchQuery) {
    suggestionsContainer.style.display = "none"; // Esconde o menu se não
    houver consulta
    return;
  }

  const items = document.querySelectorAll(".menu .box");
  const matchingItems = Array.from(items).filter(item => {
    const itemName = item.getAttribute("data-name").toLowerCase();
    return itemName.includes(searchQuery);
  });

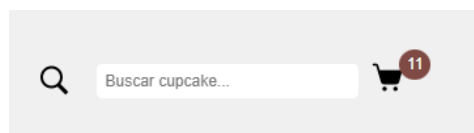
  if (matchingItems.length > 0) {
```

```
// Exibe o contêiner de sugestões
suggestionsContainer.style.display = "block";

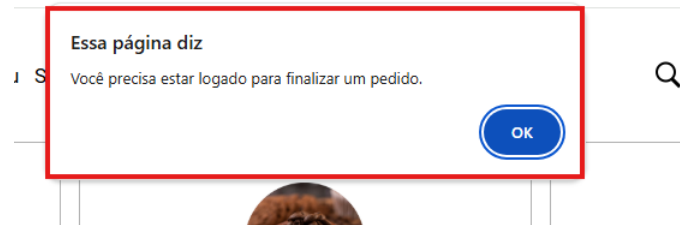
matchingItems.forEach(item => {
  const suggestion = document.createElement("div");
  suggestion.textContent = item.getAttribute("data-name");
  suggestion.onclick = () => selectItem(item); // Função para selecionar
o item
  suggestionsContainer.appendChild(suggestion);
});
} else {
  // Se não encontrar nada, exibe uma mensagem
  const noResult = document.createElement("div");
  noResult.textContent = "Nenhum item encontrado";
  noResult.style.color = "#999";
  suggestionsContainer.appendChild(noResult);
  suggestionsContainer.style.display = "block"; // Exibe o menu mesmo
sem resultados
}
}

document.getElementById("search-icon").addEventListener("click", function
() {
  const searchInput = document.getElementById("search-input");

  if (!searchInput.style.display || searchInput.style.display === "none") {
    searchInput.style.display = "block";
    searchInput.focus();
  } else {
    searchInput.style.display = "none";
    searchInput.value = "";
    document.getElementById("suggestions").style.display = "none";
  }
});
```



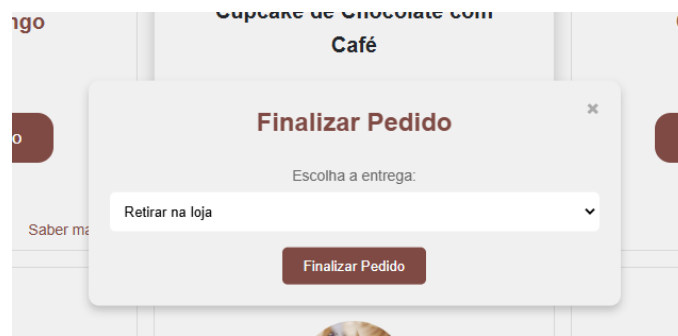
3. Ao finalizar a compra ele não tem uma página onde inserimos o endereço de entrega ou selecionamos para retirar na loja. Aparecia apenas esse alerta quando eu clicava



Criei essa função:

```
function finalizeOrder() {  
  const deliveryMethod = document.getElementById("delivery-  
method").value;  
  
  if (!deliveryMethod) {  
    alert("Por favor, selecione o método de entrega.");  
    return;  
  }  
  
  const userEmail = localStorage.getItem("userEmail");  
  
  if (!userEmail) {  
    alert("Você precisa estar logado para finalizar um pedido.");  
    return;  
  }  
  
  const order = {  
    id: Date.now(),  
    items: [...cartItems], // Todos os itens do carrinho  
    total: cartItems.reduce((sum, item) => sum + item.price * item.quan-  
tity, 0), // Total do pedido  
    deliveryMethod,  
    date: new Date().toLocaleString() // Data do pedido  
  };  
  
  // Salvando o pedido no localStorage  
  const allOrders = JSON.parse(localStorage.getItem("userOrders")) || {};  
  const userOrders = allOrders[userEmail] || [];  
  userOrders.push(order);  
  
  allOrders[userEmail] = userOrders;  
  localStorage.setItem("userOrders", JSON.stringify(allOrders));  
  
  alert("Pedido finalizado com sucesso!");  
  
  // Limpar o carrinho após finalizar o pedido  
  cartItems = [];
```

```
updateCart();  
closeModal();  
}
```



4. Senti falta de ter um login no site, para que se caso eu quiser repetir o pedido eu possa. Criei uma função para ter o login:

```
document.getElementById('login-link').addEventListener('click', function () {  
    document.getElementById('modal-auth').style.display = 'flex';  
    document.getElementById('modal-title-auth').textContent = 'Login';  
    document.getElementById('login-form').style.display = 'block';  
    document.getElementById('register-form').style.display = 'none';  
});  
  
document.getElementById('register-link').addEventListener('click', function () {  
    document.getElementById('modal-auth').style.display = 'flex';  
    document.getElementById('modal-title-auth').textContent = 'Registrar';  
    document.getElementById('login-form').style.display = 'none';  
    document.getElementById('register-form').style.display = 'block';  
});  
  
// Fechar o modal  
document.getElementById('close-modal-auth').addEventListener('click', function () {  
    document.getElementById('modal-auth').style.display = 'none';  
});  
  
// Alternar para a tela de registro  
document.getElementById('go-to-register').addEventListener('click', function () {  
    document.getElementById('modal-title-auth').textContent = 'Registrar';  
    document.getElementById('login-form').style.display = 'none';  
    document.getElementById('register-form').style.display = 'block';  
});  
  
// Alternar para a tela de login  
document.getElementById('go-to-login').addEventListener('click', function () {
```



```
document.getElementById('modal-title-auth').textContent = 'Login';
document.getElementById('login-form').style.display = 'block';
document.getElementById('register-form').style.display = 'none';
});

let isLoggedIn = localStorage.getItem('isLoggedIn') === 'true';

window.onload = function () {
  // Atualiza o estado de autenticação
  isLoggedIn = localStorage.getItem('isLoggedIn') === 'true';

  if (isLoggedIn) {
    // O usuário está logado
    document.getElementById('login-link').style.display = 'none';
    document.getElementById('register-link').style.display = 'none';
    document.getElementById('logout-btn').style.display = 'inline-block';
  } else {
    document.getElementById('login-link').style.display = 'inline-block';
    document.getElementById('register-link').style.display = 'inline-block';
    document.getElementById('logout-btn').style.display = 'none';
  }
};

// Evento de login
document.getElementById('login-btn').addEventListener('click', function () {
  const email = document.getElementById('login-email').value;
  const password = document.getElementById('login-password').value;

  const storedEmail = localStorage.getItem('userEmail');
  const storedPassword = localStorage.getItem('userPassword');

  // Verificar se o email e a senha coincidem com os dados registrados
  if (email === storedEmail && password === storedPassword) {
    localStorage.setItem("isLoggedIn", "true"); // Salva o estado de login
    alert('Login bem-sucedido!');
    atualizarEstadoAutenticacao();
    document.getElementById('modal-auth').style.display = 'none';
  } else {
    alert('Email ou senha incorretos!');
  }
});

// Evento de logout
document.getElementById('logout-btn').addEventListener('click', function () {
  localStorage.setItem('isLoggedIn', 'false'); // Atualiza o estado no localSto-
rage
  localStorage.removeItem('userEmail'); // Remove o email do usuário lo-
gado
  cartItems = []; // Limpa o carrinho (opcional)
```

```
updateCart();
atualizarEstadoAutenticacao();
});

// Função para atualizar o estado de autenticação
function atualizarEstadoAutenticacao() {
  const isLoggedIn = localStorage.getItem('isLoggedIn') === 'true';

  if (isLoggedIn) {
    // Ajusta o estado do DOM para usuário logado
    document.getElementById('login-link').style.display = 'none';
    document.getElementById('register-link').style.display = 'none';
    document.getElementById('logout-btn').style.display = 'inline-block';
  } else {
    // Ajusta o estado do DOM para usuário deslogado
    document.getElementById('login-link').style.display = 'inline-block';
    document.getElementById('register-link').style.display = 'inline-block';
    document.getElementById('logout-btn').style.display = 'none';
  }
}

// Registro de usuário
document.getElementById('register-btn').addEventListener('click', function () {
  const name = document.getElementById('register-name').value;
  const email = document.getElementById('register-email').value;
  const password = document.getElementById('register-password').value;

  if (name && email && password) {
    localStorage.setItem('userName', name);
    localStorage.setItem('userEmail', email);
    localStorage.setItem('userPassword', password);
    localStorage.setItem('isLoggedIn', 'true'); // Marca como logado
    alert('Registro bem-sucedido!');
    atualizarEstadoAutenticacao();
    document.getElementById('modal-auth').style.display = 'none';
  } else {
    alert('Por favor, preencha todos os campos de registro!');
  }
});

// Lógica para o registro (armazenar os dados no localStorage)
document.getElementById('register-btn').addEventListener('click', function () {
  const name = document.getElementById('register-name').value;
  const email = document.getElementById('register-email').value;
  const password = document.getElementById('register-password').value;

  // Armazenar os dados de registro no localStorage
  if (name && email && password) {
    localStorage.setItem('userName', name);
```

```
localStorage.setItem('userEmail', email);
localStorage.setItem('userPassword', password);
alert('Registro bem-sucedido!');
isLoggedIn = true;
localStorage.setItem('isLoggedIn', 'true');
document.getElementById('modal-auth').style.display = 'none';
} else {
  alert('Por favor, preencha todos os campos de registro!');
}
});
```

Login

Email:

Senha:

Entrar

Não tem uma conta? [Registrar-se](#)

5. Ao clicar no produto ele deveria falar mais sobre ele, para evitar problemas com alergia ou outras situações ruins. Eu criei um botão:

```
<button class="know-more-btn">Saiba mais</button>
```

E coloquei esse evento:

```
document.querySelectorAll(".menu .box .know-more-btn").forEach(button
=> {
  button.addEventListener("click", (event) => {
    event.preventDefault(); // Impede que qualquer outra ação aconteça
    ao clicar

    const cupcake = button.closest(".box"); // Encontra o cupcake clicado
    const name = cupcake.querySelector("h3").textContent;
    const description = cupcake.getAttribute("data-description");
    const allergens = cupcake.getAttribute("data-allergens");

    // Exibe o modal com as informações
    document.getElementById("modal-title").textContent = name;
    document.getElementById("modal-description").textContent = descrip-
tion;
```

```
document.getElementById("modal-allergens").textContent = "Alergê-  
cos: " + allergens;  
  
// Exibe o modal  
document.getElementById("modal-more").style.display = "block";  
});  
});  
  
// Evento para fechar o modal quando clicar no "X"  
document.querySelector(".modal-more .close").addEventListener("click", ()  
=> {  
    document.getElementById("modal-more").style.display = "none";  
});  
  
// Fecha o modal se o usuário clicar fora da área do conteúdo  
window.addEventListener("click", (event) => {  
    if (event.target === document.getElementById("modal-more")) {  
        document.getElementById("modal-more").style.display = "none";  
    }  
});  
});
```

Após clicar:



Cupcake Amora

R\$10,99

Adicione ao Carrinho

Saber mais →



Cupcake Chocolate Branco

R\$15,99 ~~R\$20,99~~

Adicione ao Carrinho

Saber mais →



Cupcake de Nozes

R\$17,99 ~~R\$20,99~~

Adicione ao Carrinho

Saber mais →

Cupcake Chocolate Branco

Delicioso cupcake de Chocolate Branco com cobertura cremosa e uma decoração elegante. Ideal para quem quer impressionar!

Alergênicos: Sem restrições

Vídeo da Solução atualizada

Após levantar os *feedbacks* e executar as correções necessárias e pertinentes, grave um vídeo de até 5 minutos apresentando as modificações realizadas no sistema.

Link para o vídeo	
--------------------------	--