

# Documentação do Sistema PetPal

## 1. Visão Geral

O projeto PetPal é uma interface de e-commerce completa e interativa, desenvolvida com tecnologias web front-end. A plataforma simula um ambiente real onde usuários podem navegar por diferentes seções, como adoção de animais, compra de produtos e agendamento de serviços. O sistema foi projetado para ser intuitivo e responsivo, com um forte foco na experiência do usuário e na interatividade do lado do cliente.

## 2. Funcionalidades

O sistema é dividido em várias funcionalidades principais para fornecer uma experiência de e-commerce completa.

### 2.1. Autenticação de Usuário

- **Login/Logout:** O sistema possui uma funcionalidade de autenticação. Um usuário pode fazer login usando as credenciais teste@petpal.com e ihc2023.
- **Persistência de Sessão:** O estado de login é mantido no localStorage do navegador, permitindo que o usuário permaneça conectado ao navegar entre as páginas ou ao fechar e reabrir o navegador.
- **Acesso Restrito:** Funcionalidades como "Minha Conta", "Favoritos" e "Histórico de Compras" são acessíveis apenas para usuários logados.

### 2.2. Navegação e Vitrines de Itens

- **Páginas Principais:** O site é composto por quatro seções principais: Home, Adoção, Loja e Serviços.
- **Busca e Filtragem:** Cada página de listagem (Adoção, Loja, Serviços) possui uma barra de busca e filtros específicos (ex: por espécie, categoria, preço, avaliação) para ajudar o usuário a encontrar o que deseja.
- **Modal Interativo:** Ao clicar em qualquer item, um modal é exibido com mais detalhes e um botão de ação (ex: Adicionar ao Carrinho, Adicionar aos Favoritos), evitando a necessidade de navegar para uma nova página.

### 2.3. Gestão de Compras e Serviços

- **Carrinho de Compras:** Os usuários podem adicionar e remover produtos do carrinho, bem como ajustar a quantidade de cada item. O ícone do carrinho no cabeçalho é atualizado dinamicamente com a quantidade de itens.
- **Simulação de Checkout:** A finalização da compra é simulada. Ao clicar em "Finalizar Compra", o sistema exibe um feedback de processamento e sucesso.

- **Histórico de Compras:** Após uma compra bem-sucedida, o carrinho é limpo e os itens são adicionados à página "Histórico de Compras", que pode ser visualizada pelo usuário logado.
- **Agendamento de Serviços:** O usuário pode agendar serviços, que são salvos e exibidos na página "Meus Agendamentos".

## 2.4. Área do Usuário

- **Favoritos:** Usuários podem favoritar pets na página de adoção, e estes são listados em sua página de favoritos para fácil acesso.
- **Gerenciamento de Dados:** A página "Minha Conta" permite ao usuário visualizar e simular a edição de suas informações pessoais e de endereço.

# 3. Arquitetura e Tecnologias

## 3.1. Arquitetura

A aplicação é puramente **client-side** (front-end), sem a necessidade de um back-end. Toda a lógica de negócio, interatividade e persistência de dados é manipulada no navegador. O estado da aplicação (como itens no carrinho, favoritos do usuário, etc.) é gerenciado através do localStorage do navegador, o que garante que os dados persistam mesmo após o fechamento da janela.

## 3.2. Tecnologias

- **HTML5:** Utilizado para a estrutura semântica de todas as páginas do site.
- **Tailwind CSS:** Um framework CSS utilitário que permitiu a criação de um design moderno e responsivo de forma ágil.
- **JavaScript (Vanilla):** Usado para toda a lógica e interatividade, incluindo:
  - Manipulação do DOM para renderizar dinamicamente os componentes.
  - Gerenciamento de eventos de clique e entrada do usuário.
  - Controle do estado da aplicação.
- **Local Storage:** Empregado como "banco de dados" do lado do cliente para armazenar dados como o estado de login, itens do carrinho, agendamentos e favoritos.

# 4. Estrutura dos Arquivos

O código-fonte está organizado de forma modular para facilitar a manutenção:

- **Arquivos .html:** Cada arquivo HTML representa uma página específica do site (ex: home\_page.html, products\_page.html).
- **Arquivos .js:** A lógica está separada em diferentes arquivos JavaScript, cada um com uma responsabilidade clara:
  - auth.js: Gerencia a autenticação do usuário.
  - cart.js e cart-page.js: Lidam com a lógica do carrinho.

- modal.js: Controla a exibição e o conteúdo dos modais.
- favorites.js, bookings.js, order\_history.js: Gerenciam as seções da conta do usuário.
- Arquivos de filtro (filter-logic.js, etc.): Contêm a lógica para filtrar os itens nas páginas de listagem.

## 5. Fluxos de Usuário

### Exemplo de fluxo de compra:

1. O usuário acessa a página "Shop".
2. Clica em um produto, um modal com detalhes aparece.
3. Clica em "Add to Cart"; o item é adicionado ao carrinho e o ícone do cabeçalho é atualizado.
4. O usuário navega para a página do carrinho e vê os itens.
5. Clica em "Finalizar Compra", o checkout é simulado, e o usuário é redirecionado para a página de sucesso.
6. Se logado, o pedido agora aparece no "Histórico de Compras".

## 6. Acesso ao Projeto

O projeto está hospedado no GitHub Pages e pode ser acessado em:  
[https://tung-a.github.io/EP-IHC/interface/home\\_page.html](https://tung-a.github.io/EP-IHC/interface/home_page.html)