

## Ecommerce

### Nome dos integrantes

Francisco Junior - Shaw

### Objetivo da aplicação

Desenvolver um ecommerce funcional que atenda aos requisitos da disciplina, demonstrando habilidades na criação de uma aplicação web interativa com HTML, CSS e JavaScript, incluindo manipulação do DOM e consumo de dados de uma API.

### Funcionalidades implementadas

Analisando o repositório no GitHub e o site publicado, o projeto implementa com sucesso as seguintes funcionalidades exigidas:

**Estrutura de 4 Páginas:** O site é composto pelas páginas Inicial (apresentação), Carros (produtos), Sobre (institucional) e Contato (formulário).

**Catálogo de Produtos Dinâmico:** A página "Carros" carrega os veículos dinamicamente a partir de uma fonte de dados (API), exibindo imagem, nome e preço de cada um.

#### Interatividade com JavaScript:

**Filtro de Produtos:** Na página de carros, foi implementado um filtro que permite ao usuário selecionar uma marca e visualizar apenas os veículos correspondentes de forma interativa.

**Carrossel de Imagens:** A página inicial possui um slider de destaque, totalmente funcional, para exibir os principais produtos.

**Formulário com Validação:** A página "Contato" contém um formulário funcional que utiliza JavaScript para validar os campos (nome, email, mensagem), impedindo o envio de dados em branco e exibindo mensagens de erro.

**Layout Responsivo:** Todo o site se adapta de forma fluida a diferentes tamanhos de tela, desde desktops a dispositivos móveis, garantindo uma boa experiência de navegação.

### Explicação técnica do código (principais arquivos)

**index.html, carros.html, sobre.html, contato.html:** Arquivos que estruturam o conteúdo de cada página, utilizando tags de HTML semântico (<header>, <main>, <section>, <footer>) para garantir a organização e acessibilidade do código.

**assets/js/carros.js:** É o principal arquivo de lógica para a página de produtos. Ele é responsável por:

- Realizar a requisição (fetch) para a API para buscar os dados dos veículos.

- Renderizar dinamicamente os cards de cada carro na tela.

- Implementar a funcionalidade de filtro por marca.

assets/js/contato.js: Contém o script dedicado à validação do formulário de contato, garantindo que os dados inseridos pelo usuário sejam válidos antes do envio.

assets/css/: Pasta que armazena as folhas de estilo. O style.css contém os estilos globais e de responsividade, enquanto os demais arquivos (carros.css, contato.css, etc.) aplicam estilos específicos para cada página.

## API Utilizada e Instruções de Execução

O projeto utiliza um arquivo db.json como uma API própria simulada para fornecer os dados dos produtos. Para que o site funcione corretamente em um ambiente de desenvolvimento, a API precisa ser iniciada.

Para rodar o programa localmente, siga os passos:

Na pasta da API:

Abra o terminal e execute `npm install` para instalar as dependências.

Em seguida, execute `npm run dev` para iniciar o servidor da API.

Na pasta do Projeto (frontend):

Abra um novo terminal.

Execute `npm run dev` para iniciar o servidor de desenvolvimento que servirá a aplicação.