

# TONON - E-commerce de Moda Premium

## Relatório Técnico de Desenvolvimento Front-End

**Data:** 29 de Novembro de 2025

### Integrantes do Grupo:

- Eduardo Tonon
- Daniel Miranda
- Ester
- Isabella Cardoso
- Gabriel Charles

## 1. Introdução ao Produto

O projeto **TONON** consiste no desenvolvimento de um site completo de comércio eletrônico focado no Front-End, simulando uma loja de roupas e acessórios de alto padrão. O conceito central da marca é unir elegância e conforto através de um design minimalista, proporcionando uma experiência de usuário (UX) fluida e sofisticada.

O site atua como uma vitrine virtual e plataforma de vendas, dividida em três linhas principais:

- **Moda Feminina:** Curadoria de peças sofisticadas (vestidos de seda, alfaiataria, tricôs).
- **Moda Masculina:** Do casual urbano a ternos de corte italiano.
- **Acessórios:** Itens de luxo acessível como relógios, bolsas e joias.

## 2. Público-Alvo

Com base na proposta de valor da TONON ("Moda Premium" e "Alto Padrão"), o público-alvo foi identificado como:

- **Perfil Demográfico:** Homens e mulheres, majoritariamente entre 25 e 50 anos, com poder aquisitivo médio-alto a alto.
- **Comportamento:** Consumidores que valorizam a qualidade dos tecidos, o corte das peças e a exclusividade. São usuários que buscam praticidade na

compra online, mas exigem uma apresentação visual impecável e detalhada dos produtos.

- **Interesses:** Moda atemporal, minimalismo, tendências de luxo e consumo consciente de peças duráveis.

### 3. Tecnologia e Framework

O desenvolvimento seguiu práticas modernas de desenvolvimento web, priorizando a performance e a componentização mesmo em um ambiente sem build complexo.

#### Stack Tecnológico

- **HTML5 Semântico:** Estruturação do conteúdo visando acessibilidade e SEO.
- **CSS3 + Bootstrap 5:** Utilização do sistema de Grid para responsividade total e componentes pré-estilizados, com personalização via CSS para a identidade visual da marca (fontes e paleta de cores).
- **JavaScript (ES6+):** Manipulação do DOM e lógica imperativa.
- **Vue.js 3 (CDN):** O principal motor de interatividade do projeto. Utilizado para criar a reatividade do sistema, gerenciando o estado da aplicação (lista de produtos e carrinho) sem a necessidade de recarregar a página.
- **SweetAlert2:** Biblioteca para feedback visual aprimorado (modais de sucesso ou erro).
- **LocalStorage:** Tecnologia de persistência de dados no navegador, garantindo que o carrinho do usuário não se perca ao fechar a aba.

### 4. Estrutura e Funcionalidades

O projeto foi estruturado em arquivos separados para facilitar a manutenção e organização do código.

#### Arquitetura de Arquivos

A estrutura de arquivos local segue o padrão:

- `index.html`: Página inicial (Home) com destaques.
- `produtos.html`: Catálogo completo com renderização dinâmica.
- `detalhe.html`: Visualização específica de um produto.
- `checkout.html`: Resumo do pedido e simulação de pagamento.
- `assets/`: Pasta contendo `style.css` e `script.js`.

## Principais Funcionalidades

1. **Renderização Dinâmica:** O Vue.js renderiza automaticamente a lista de 50 produtos a partir de um array de objetos JSON.
2. **Carrinho Persistente:** Adição e remoção de itens com atualização em tempo real do valor total. O estado é salvo no `localStorage`.
3. **Filtros de Categoria:** Capacidade de filtrar produtos por gênero ou tipo de acessório.
4. **Feedback ao Usuário:** Uso do SweetAlert2 para confirmar ações críticas, como "Adicionar ao Carrinho" ou "Finalizar Compra".

## 5. Instruções de Instalação e Execução

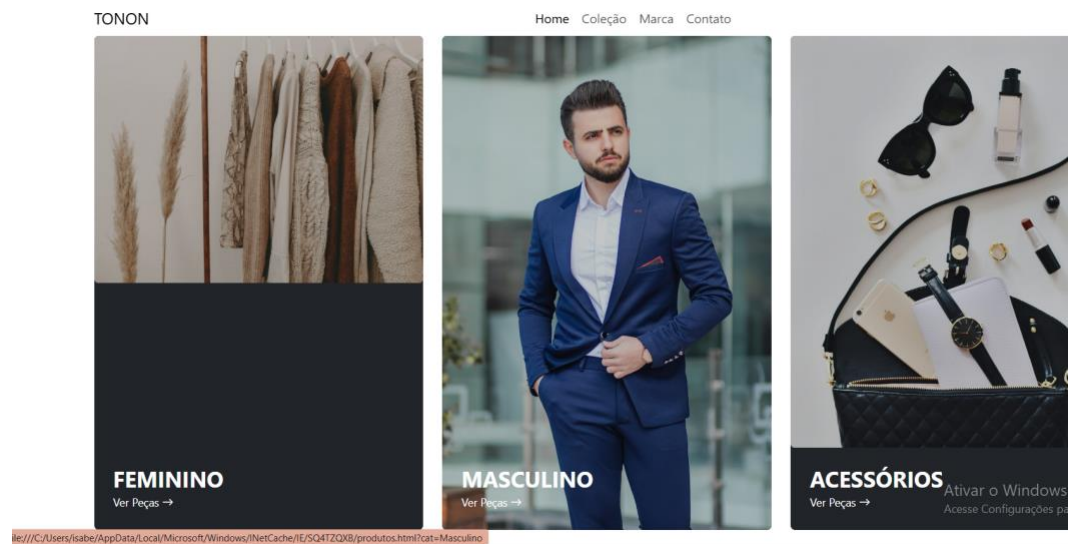
Para testar o projeto localmente, foram definidas as seguintes diretrizes de ambiente:

1. Baixe todos os arquivos do repositório e mantenha-os na mesma estrutura de pastas.
2. Certifique-se de que todos os arquivos HTML (`index.html`, `produtos.html`, etc.) e a pasta `assets` estejam presentes.
3. **Método Recomendado:** Abra a pasta do projeto no VS Code, instale a extensão "Live Server", clique com o botão direito no arquivo `index.html` e escolha a opção "Open with Live Server". Isso simula um ambiente de servidor real, evitando bloqueios de CORS.
4. **Método Simples:** Dê um duplo clique no arquivo `index.html` para abrir diretamente no navegador padrão (algumas funcionalidades do `LocalStorage` podem ter comportamento variável dependendo das configurações de segurança do navegador neste modo).

## 6. Fluxo de Navegação e Telas

Abaixo são descritas as etapas visuais que compõem o fluxo de compra do usuário na plataforma.

## Página Inicial e Catálogo



A vitrine principal onde o usuário tem o primeiro contato com a identidade minimalista da marca, visualizando a grid de produtos e suas informações básicas. O layout prioriza imagens de alta qualidade e uma navegação intuitiva pelas categorias.

## Carrinho e Checkout


### DADOS DE ENTREGA


**Nome Completo \***

**Email \***

**Endereço Completo \***

### PAGAMENTO

☒ **Cartão de Crédito** 

☐ **PIX** 

**Finalizar Pedido**

O momento de decisão de compra, onde o usuário revisa os itens selecionados. Nesta etapa, é apresentada uma tabela detalhada com os produtos, quantidades e valores parciais. O sistema calcula automaticamente o total e permite a simulação segura da finalização do pedido.

## 7. Desafios e Soluções

Durante o desenvolvimento do projeto TONON, a equipe enfrentou desafios técnicos e de design.

### Gerenciamento de Estado sem Vuex/Pinia

**Desafio:** Como o projeto utiliza Vue.js via CDN sem ferramentas de build complexas (como Webpack ou Vite), gerenciar o estado global (carrinho de compras) entre diferentes páginas HTML foi um obstáculo. O Vue.js, por padrão, reinicia sua instância a cada recarregamento de página.

**Solução:** Implementamos um sistema híbrido utilizando o `localStorage` do navegador como a "fonte da verdade". Ao iniciar a instância do Vue em cada página, um *hook* do ciclo de vida (`mounted`) verifica se há dados salvos localmente e popula o estado da aplicação, garantindo a continuidade da navegação.

### Responsividade do Grid de Produtos

**Desafio:** Manter a estética "Premium" e minimalista em dispositivos móveis, garantindo que as imagens dos produtos não ficassem distorcidas ou pequenas demais.

**Solução:** Utilizamos profundamente o sistema de Grid do Bootstrap 5 (`col-12 col-md-4 col-lg-3`), ajustando as margens e paddings manualmente via CSS para garantir que, no mobile, os produtos ocupassem uma área confortável para o toque (touch), melhorando a usabilidade.

### Interatividade do Usuário

**Desafio:** O uso de `alert()` nativo do navegador quebrava a imersão de luxo proposta pelo design da marca.

**Solução:** Integração da biblioteca **SweetAlert2**. Isso permitiu a criação de pop-ups estilizados e animados que seguem a paleta de cores do site, oferecendo um feedback visual profissional ao adicionar itens ao carrinho ou finalizar a compra.

## 8. Conclusão

O projeto TONON cumpriu com êxito o objetivo de criar uma experiência de e-commerce premium focada no Front-End. A utilização do Vue.js, mesmo em sua forma mais simples via CDN, provou-se eficaz para garantir a reatividade necessária, enquanto o Bootstrap 5 assegurou a consistência visual e a responsividade.

As soluções aplicadas para a persistência de dados e a interatividade resultaram em uma navegação fluida, simulando com precisão o funcionamento de uma loja virtual real e preparando a base para futuras integrações com Back-End.