

# Relatório de Desenvolvimento: Estrutura do Projeto GoFood Delivery

## Equipe

Eros Eloi - 12723126200

Cauã Souza Conceição - 1272310702

Paullo Victor de carvalho Marques - 12723117914

Caio Sterphen Barbosa Santos - 12723115470

Anthony Clayton Barros de Jesus Carvalho - 12723110413

Anderson Silva Dantas Junior - 1272311567

Matheus Lobo de Souza Martins - 1272322529

Italo da Conceição Araújo - 12723114551

Este relatório detalha a estrutura do projeto front-end do aplicativo GoFood Delivery, seus benefícios, como ela suporta a escalabilidade e manutenção, e sua relação com aspectos das heurísticas de usabilidade de Nielsen.

## 1. Visão Geral da Estrutura

O projeto GoFood Delivery emprega uma estrutura organizada e modular, alinhada com as boas práticas de desenvolvimento front-end. A análise da organização de pastas e arquivos demonstra um planejamento voltado para otimizar a manutenção e assegurar a escalabilidade futura do aplicativo.

## 2. Benefícios da Estrutura Adotada

A estrutura implementada no projeto GoFood Delivery confere diversas vantagens:

**Organização Modular:** A separação clara de recursos (CSS, JavaScript, imagens) e sua organização por contexto (global ou específico de página) facilita a localização, modificação de arquivos e a reutilização de componentes globais, promovendo uma clara separação de responsabilidades. **Escalabilidade:** A arquitetura facilita o crescimento ordenado. Novas páginas podem ser adicionadas seguindo padrões estabelecidos, a expansão de componentes é integrada facilmente e o gerenciamento de recursos se torna mais eficiente com o aumento do projeto. **Flexibilidade:** A independência entre os componentes minimiza o impacto de alterações. A organização modular facilita a refatoração, e a estrutura pode ser adaptada para novos requisitos ou tecnologias sem comprometer a base existente.

## 3. Detalhamento da Estrutura

O diretório `src/` concentra todo o código-fonte e recursos do aplicativo:

**Arquivos HTML:** `index.html`: Página principal (produtos e categorias). `carrinho.html`: Página do carrinho de compras.

**Diretório `assets/`:** Contém recursos estáticos, organizados por tipo: `css/`: Estilos CSS.

`global/`: Estilos globais (`global.css`, `header.css`, `menuLateral.css`, `modal.css`, `normalize.css`).

pages/: Estilos por página (index/, tela-de-carrinho/, com subdiretórios parciais/ para componentes específicos). js/: Arquivos JavaScript. components/: Componentes reutilizáveis (header.js, inputRange.js, menuLateral.js, modal.js). pages/: Scripts por página (index/, tela-de-carrinho/, com arquivos para inicialização e eventos). img/: Imagens (globais e por página).

#### 4. Como a Estrutura Favorece a Escalabilidade

A escalabilidade é intrínseca à estrutura, facilitada por:

Padrão de Componentes: Permite desenvolvimento paralelo, testabilidade e manutenção simplificada de elementos da interface. Organização por Contexto: Otimiza o carregamento de recursos, previne conflitos e confere clareza ao propósito de cada arquivo. Modularização de JavaScript: Utiliza módulos ES6, separa responsabilidades e garante inicialização controlada.

#### 5. Relação com Heurísticas de Usabilidade de Nielsen

A estrutura do projeto GoFood Delivery e as práticas de desenvolvimento nela baseadas impactam positivamente a aplicação de diversas heurísticas de usabilidade de Nielsen, tanto no processo de desenvolvimento quanto, indiretamente, na experiência do usuário final.

Visibilidade do status do sistema (Visibility of system status): Embora seja uma heurística focada na interação do usuário, a estrutura do projeto facilita sua implementação. A modularidade do JavaScript (components/, pages/) e a organização do CSS permitem que os desenvolvedores adicionem facilmente feedback visual ou textual (como indicadores de carregamento, mensagens de sucesso/erro) em componentes específicos ou em fluxos de página, garantindo que o usuário seja informado sobre o que está acontecendo.

Correspondência entre o sistema e o mundo real (Match between system and the real world): A estrutura, ao organizar o código em torno de componentes e páginas que frequentemente correspondem a conceitos do mundo real (como "carrinho", "cabeçalho", "filtros", "cards de produtos"), torna a base de código mais intuitiva para a equipe de desenvolvimento. Nomes de pastas como tela-de-carrinho ou arquivos como header.js utilizam uma linguagem familiar, facilitando a localização e compreensão do código associado a elementos visuais que os usuários finais reconhecem do "mundo real" do aplicativo. Consistência e padrões (Consistency and standards): Este é um dos pontos fortes diretamente suportados pela estrutura. A divisão em estilos global/ e pages/ no CSS, o uso de componentes JavaScript reutilizáveis e a estrutura repetida para os diretórios de página (css/pages/index/, js/pages/index/, etc.) promovem uma consistência inerente ao projeto. A equipe é incentivada a seguir esses padrões estabelecidos, resultando em um código mais uniforme e, conseqüentemente, em uma interface de usuário mais consistente e previsível, o que reduz a carga cognitiva do usuário final. Reconhecimento em vez de memorização (Recognition rather than recall): Para a equipe de desenvolvimento, a estrutura clara e organizada reduz drasticamente a necessidade de memorizar caminhos de arquivos ou a localização de funcionalidades específicas. A hierarquia lógica dos diretórios (src/, assets/, css/, js/, img/, components/, pages/) permite que os desenvolvedores reconheçam onde encontrar a maior parte do código relevante com base no seu tipo ou contexto (e.g., "Preciso mudar o estilo do cabeçalho? Está em

assets/css/global/header.css"). Para o usuário final, a reutilização de componentes da interface, facilitada pela estrutura, pode levar a padrões de interação reconhecíveis em diferentes partes do aplicativo. Design estético e minimalista (Aesthetic and minimalist design): A organização modular, especialmente no CSS (global/, pages/, partials/), permite que os estilos sejam gerenciados de forma mais controlada. Isso facilita a implementação de um design limpo e focado, evitando a sobrecarga visual causada por estilos conflitantes ou mal organizados. A separação de responsabilidades no código JavaScript também ajuda a garantir que a lógica da interface seja clara e não polua a experiência do usuário com comportamentos inesperados ou redundantes.

## 6. Conclusão

A estrutura do projeto GoFood Delivery foi concebida com foco em organização, manutenção e escalabilidade. Sua abordagem modular, a clara separação de responsabilidades e o estabelecimento de padrões facilitam o crescimento sustentável e o trabalho colaborativo. Adicionalmente, esta arquitetura subjacente apoia a aplicação de princípios de usabilidade de Nielsen, tornando o processo de desenvolvimento mais eficiente e intuitivo para a equipe (flexibilidade, correspondência com o mundo real, reconhecimento, documentação intrínseca) e estabelecendo uma base sólida para a construção de uma interface de usuário consistente, compreensível e eficiente (visibilidade do status, consistência, design minimalista), contribuindo assim para uma melhor experiência geral com o aplicativo GoFood Delivery.