

No caso desse projeto, que foi desenvolvido no **Webflow**, que é uma plataforma **no-code**, a abordagem tradicional de codificação, escrever código front-end e back-end manualmente, é substituída pelo uso das ferramentas visuais da plataforma.

## **Relatório de Codificação do Front-End, Back-End e Aplicação Web**

### **1. Abordagem No-Code Utilizada**

O desenvolvimento do sistema foi realizado utilizando o **Webflow**, uma plataforma no-code que permite criar, gerenciar e publicar sites dinâmicos sem a necessidade de codificação tradicional. Essa abordagem substitui o desenvolvimento manual de código por uma interface visual e lógica pré-configurada, garantindo a entrega do projeto com eficiência e qualidade.

### **2. Codificação do Front-End**

No Webflow, o **front-end** é construído por meio de um editor visual que gera código HTML, CSS e JavaScript responsivo automaticamente. As principais etapas foram:

- **Estruturação das Páginas:** Utilizando layouts de grade e caixas (flexbox e grid) para criar a hierarquia visual.
- **Design Responsivo:** Configuração para diferentes dispositivos (desktop, tablet e mobile).
- **Animações e Interações:** Implementação de transições e animações utilizando a funcionalidade de interatividade nativa do Webflow.
- **Estilos Personalizados:** Configuração de cores, tipografia e outros elementos visuais diretamente no editor visual.

### **Artefatos Gerados:**

- Código HTML, CSS e JavaScript otimizado, exportável diretamente do Webflow.
- Páginas dinâmicas vinculadas ao CMS para exibir conteúdo personalizado.

### 3. Codificação do Back-End

Embora o Webflow não ofereça suporte a um back-end tradicional, ele utiliza o **CMS interno** para gerenciar dados dinâmicos e exibi-los no front-end. As funcionalidades configuradas incluem:

- **Coleções do CMS:** Modelagem e estruturação de dados, como "Produtos", "Pedidos" e "Clientes".
- **Filtros e Ordenações:** Implementados para apresentar apenas dados relevantes.
- **API do Webflow:** Disponível para integrações externas, permitindo a expansão da funcionalidade da aplicação com ferramentas como Zapier ou Make.

#### Artefatos Gerados:

- Estruturas de dados no CMS, configuradas para refletir os diagramas de classes e de persistência.
- Configurações de API para futuras integrações, como sistemas de pagamento ou notificações.

### 4. Manual de Uso

Um manual foi elaborado para explicar o funcionamento do sistema, incluindo:

- Navegação entre as páginas do site.
- Como adicionar ou gerenciar dados no CMS (ex.: novos produtos ou alterações nos pedidos).
- Configuração e uso das interações e animações visuais.

### 5. Análise da Abordagem

Embora o Webflow não exija codificação manual, ele gera automaticamente um código bem estruturado e otimizado, equivalente ao que seria produzido por desenvolvedores front-end.

## 6. Limitações e Possíveis Melhorias

- O CMS é adequado para sites de pequeno a médio porte, mas para projetos maiores pode ser necessário migrar para uma solução mais escalável.

### Relatório de Testes: Verificação e Validação

Os testes de verificação e validação foram realizados no site desenvolvido utilizando o **Webflow**, com o objetivo de garantir que o comportamento do sistema corresponde às especificações funcionais e não funcionais definidas no planejamento do projeto.

#### Objetivos dos Testes

- Garantir que todas as funcionalidades implementadas atendem aos requisitos do sistema, como navegação, exibição de produtos e conclusão de pedidos.
- Identificar e corrigir possíveis erros de funcionalidade, design ou usabilidade antes da entrega final.
- Verificar a responsividade e compatibilidade do site em diferentes dispositivos e navegadores.

#### Testes de Verificação

Os testes de verificação visaram garantir que o sistema foi implementado corretamente, conforme o planejado. As seguintes verificações foram realizadas:

- **Estruturas de páginas:** Verificar se as páginas e links estão configurados corretamente.
- **Funcionalidades do CMS:** Garantir que as coleções do CMS exibem e gerenciam os dados dinamicamente.
- **Design Responsivo:** Testar a aparência e usabilidade do site em dispositivos desktop, tablet e mobile.
- **Interações e Animações:** Validar que as animações e transições configuradas funcionam conforme o esperado.

#### Testes de Validação

Os testes de validação focaram em garantir que o sistema cumpre os objetivos e requisitos definidos pelo cliente e pelos usuários. As etapas realizadas foram:

**1. Teste Funcional:**

- Navegar pelo site como um cliente e simular o fluxo de compra: navegar pela vitrine, selecionar cupcakes, adicionar ao carrinho e finalizar a compra.

**2. Teste de Usabilidade:**

- Avaliar a clareza e facilidade de navegação.

**3. Teste de Compatibilidade:**

- Testar o site em diferentes navegadores (Google Chrome, Safari, Firefox) e dispositivos para verificar compatibilidade.

**4. Teste de Fluxo do Usuário:**

- Simular o fluxo de um pedido.

### **Evidências Coletadas**

Durante os testes, evidências foram coletadas para documentar o comportamento do sistema, como:

- Vídeos mostrando o fluxo de navegação e funcionamento das interações.

### **Resultados**

Os testes confirmaram que:

- As funcionalidades planejadas foram implementadas e funcionam corretamente.
- O site atende aos requisitos de responsividade e usabilidade.
- Não foram identificados erros críticos durante a navegação e uso das funcionalidades.

### **Limitações e Melhorias**

- **Limitações:** A validação de alguns formulários depende do pagamento de funcionalidades mais complexas.

**Melhorias:** Sugere-se a implementação de ferramentas de monitoramento de desempenho e