

O objetivo deste projeto é desenvolver, utilizando a abordagem no-code no Webflow, um sistema para uma loja virtual de cupcakes, com foco na vitrine online, no processo de pedidos, no pagamento eletrônico e na entrega. O desafio consiste em criar uma plataforma que permita aos clientes visualizar, selecionar e comprar cupcakes de maneira conveniente e segura, garantindo, ao mesmo tempo, um processo eficiente de pagamento e entrega dos produtos.

O sistema foi desenvolvido usando o Webflow, uma ferramenta no-code para criação de sites, foi utilizado conceitos de modelagem UML para estruturar e entender o sistema antes de implementá-lo. O objetivo da verificação e adequação é para garantir que a implementação no Webflow reflita as modelagens propostas e as eventuais melhorias ou alterações na modelagem que leve a uma maior precisão.

### Diagrama de Classes

As coleções do CMS do Webflow correspondem às classes no diagrama.

- **Verificação:** confirmar que todas as entidades necessárias foram criadas como coleções. Por exemplo:
  - Classe Produto foi representada pela coleção "Produtos".
  - Relacionamentos foram mapeados usando os campos de referência e múltiplas referências.
- **Adequação:** Melhorias, adicionar campos e ajustar relacionamentos no CMS. Por exemplo:
  - Adicionar uma relação direta entre os Pedidos e os itens.

### Diagrama de Casos de Uso

Conectar os **casos de uso** às funcionalidades criadas no Webflow. Por exemplo:

- **Verificação:** Confirmar que os casos de uso descritos foram implementados. Por exemplo:
  - Usuário seleciona cupcakes e adiciona ao carrinho está implementado pela página de vitrine e pela funcionalidade de adicionar itens ao carrinho.
- **Adequação:** Fazer melhorias, como adicionar mensagens de erro mais claras. Por exemplo:
  - Melhorar a exibição de mensagens de erro quando um campo obrigatório não é preenchido no checkout.

## Diagrama de Sequência

Relacionar o diagrama de sequência às ações configuradas no Webflow e às automações externas. Por exemplo:

- **Verificação:** Listar as ações e verificar se elas seguem a sequência planejada. Por exemplo:
  - Usuário seleciona um produto, adiciona ao carrinho e confirma a compra segue a sequência configurada no Webflow.
- **Adequação:** Propor melhorias, como detalhar melhor os passos. Por exemplo:
  - Dividir o fluxo de checkout em mais etapas para melhorar a usabilidade.

## Diagrama de Atividades

Verificar como o fluxo das atividades foi implementado. Por exemplo:

- **Verificação:** Comparar o diagrama de atividades com a navegação e interações configuradas no Webflow. Por exemplo:
  - Navegar pela vitrine, selecionar produtos, finalizar compra está representado pelas páginas e interações.
- **Adequação:** Melhorias, otimizar etapas. Por Exemplo:
  - Consolidar telas para reduzir cliques desnecessários.

## Mensagens de Erro

Verificar as mensagens de erro. Por exemplo:

- **Verificação:** Verificar se mensagens de erro informam claramente o problema ao usuário (ex.: "Campo obrigatório não preenchido").
- **Adequação:** Fazer melhorias na clareza ou consistência das mensagens.

## Revisão do Protótipo

Incrementar o protótipo:

- Ajustar os elementos visuais para melhor clareza e usabilidade.

## Conclusão

As verificações e adequações dos elementos de modelagens da solução impactaram positivamente na modelagem e no sistema final. Garantindo maior precisão entre o modelo teórico e a implementação no Webflow.

A IHC se refere ao design das interações entre usuários e o sistema, incluindo a disposição visual das telas, a funcionalidade das interfaces e a clareza das mensagens apresentadas. Tem como objetivo garantir uma experiência de usuário intuitiva, eficiente e agradável.

## **Elementos Revisados na IHC**

Aspectos principais revisados.

### **a. Estrutura das Telas**

- **Descrição:** As telas do site foram projetadas para refletir os fluxos de navegação.
- **Ação:** Houve melhoria na navegação com base na revisão.
- **Exemplo:**
  - Antes: As informações de produtos estavam desorganizadas.
  - Depois: Reorganizei as informações com imagens maiores, descrições detalhadas e botões de ação mais claros.

### **b. Mapas Conceituais**

- **Descrição:** Mencione se você criou ou revisou um mapa conceitual que descreve a relação entre as telas, os dados e as funcionalidades.
- **Exemplo:** A organização das coleções de produtos, clientes e pedidos no CMS foi mapeada e conectada às telas de forma lógica.

### **c. Mensagens de Erro**

- **Descrição:** Busquei melhorar as mensagens de erro, caso o usuário encontre problemas.
- **Exemplo:**
  - Implementação de mensagens claras e específicas.

### **d. Protótipo e Navegação**

- **Descrição:** No protótipo inicial busuei melhorar a experiência do usuário.
- **Ação:** Adição de botões com informações mais claras.

## **Ferramentas Utilizadas**

- Devido a sua capacidade de criar um protótipo funcional, facilitando a integração de elementos de IHC como responsividade e personalização, o Webflow foi utilizado para o desenvolvimento do site.

## Processo de Verificação e Adequação

- **Método de Avaliação:** Utilizei critérios como:
  - **Usabilidade:** Verificar se as telas são fáceis de usar.
  - **Clareza:** Verificar se as mensagens são compreensíveis.
  - **Consistência:** Se o design é coerente em todas as páginas.
  - **Acessibilidade:** Se o site é acessível para diferentes dispositivos.
- **Testes Realizados:** Testes de navegação clicando nos botões para verificar a funcionalidade.

## Melhorias e Incrementos

- Melhorar a organização de menus ou links.
- Tornar o site responsivo para dispositivos móveis.

## Conclusão

As adequações realizadas no projeto visam garantir que os objetivos do sistema fossem alcançados.

Diagramas de classes de persistência são usados para modelar o armazenamento dos dados de uma aplicação, detalhando entidades, atributos e relacionamentos com o objetivo de garantir que o armazenamento dos dados seja consistente, bem estruturado e normalizado.

O Webflow não possui uma modelagem de dados explícita como em sistemas tradicionais de desenvolvimento, mas utiliza coleções no **CMS** que correspondem às entidades e relacionamentos do projeto. O **Webflow CMS** é a base para o gerenciamento de conteúdo e dados no Webflow. Ele permite criar coleções (equivalentes a tabelas ou classes persistentes) para armazenar e organizar informações do site.

## Relacionando o Diagrama de Classes ao Webflow

- Os elementos do diagrama de classes são refletidos nas coleções do CMS da seguinte forma:
  - **Entidades principais:** Entidades do diagrama de classes: Produto, Cliente, Pedido, Item de Pedido, Pagamento, Entrega.
  - **Mapeamento no CMS:**
    - Produto → Coleção "Produtos" no Webflow.

- Cliente → Não gerido diretamente, mas pode ser representado através de um sistema externo integrado.
- Pedido → Coleção personalizada para gerenciar informações de pedidos.
- Item de Pedido → Relacionamento entre "Pedido" e "Produto".
- Pagamento e Entrega → Informações que podem ser integradas a campos personalizados ou sistemas externos (como gateways de pagamento e logística).

### Revisão do Projeto Conceitual

- **Descrição do projeto conceitual:**
  - As entidades são organizadas no **Webflow CMS da seguinte forma.**
  - **Produto:** Coleção no CMS contendo campos como Nome, Descrição, Preço e Imagem.
  - **Pedido:** Coleção com campos para associar produtos, cliente e status.
  - **Relacionamento Produto–Pedido:** Configurado por meio de **campos de referência** no Webflow.

### Revisão do Projeto Lógico Normalizado

- O CMS do Webflow permite criar relações entre coleções. Como exemplo de normalização podemos citar **Produto** e **Pedido** como coleções distintas, conectadas por um campo de referência (relacionamento 1:N).

### Melhorias e Incrementos

- Durante a revisão buscou-se a:
  - Reorganização das coleções para refletir melhor os relacionamentos do diagrama.
  - Adicionar novos campos para atender a necessidades específicas.

### Conclusão

- O uso do Webflow permitiu uma implementação prática e funcional dos dados modelados no diagrama de classes.

O projeto físico do banco de dados é a etapa em que se implementa a modelagem lógica no sistema de gerenciamento de banco de dados (SGBD), definindo tabelas, campos, relacionamentos e índices. O objetivo é garantir que os dados sejam armazenados de maneira eficiente e segura para atender às necessidades da aplicação.

O Webflow, como uma ferramenta no-code, utiliza o **CMS** para gerenciar dados e não requer um banco de dados físico tradicional. Sendo assim, o Webflow utiliza um CMS que funciona como um banco de dados simplificado, permitindo criar coleções de dados (semelhantes a tabelas) e relacionar informações entre elas. Como o Webflow não possui um SGBD tradicional, a implementação do banco físico é realizada diretamente no CMS.

### **Estrutura Física do Banco de Dados**

- **Descrição do CMS como banco de dados:**
  - O CMS do Webflow serve como base de dados e as coleções configuradas nele representam as tabelas do banco.
- **Mapeamento das coleções/tabelas:**
  - As entidades do projeto físico (Produto, Cliente, Pedido, Item de Pedido, Pagamento, Entrega) foi implementada como uma coleção no Webflow:
    - **Produto:** Coleção com campos como Nome, Descrição, Preço, Imagem.
    - **Cliente:** Simulado por campos adicionais em formulários ou coleções externas integradas.
    - **Pedido:** Coleção com campos para associar Cliente, Produto e Status do Pedido.
    - **Item de Pedido:** Pode ser implementado como uma coleção separada para gerenciar a relação entre Pedido e Produto.
    - **Pagamento:** Documentado por meio de campos em Pedido ou integrado com um sistema externo.
    - **Entrega:** Incluído como atributos em Pedido.

Como exemplo citamos a coleção para o **Produto** :

- ID: Identificador único (gerado automaticamente pelo Webflow).
- Nome: Texto curto.
- Preço: Número decimal.
- Descrição: Texto longo.
- Imagem: Arquivo.

## Relacionamentos

- Relacionamentos configurados no CMS:
  - **Produto – Pedido**: Relacionamento de muitos para muitos (representado por referência ou coleção intermediária no CMS).
  - **Pedido – Cliente**: Relacionamento de um para muitos (Pedido referência Cliente).
  - **Entrega – Pedido**: Relacionamento de um para um.

## Dicionário de Dados

- Como exemplo, de dicionário de dados detalhado para cada coleção, descrevo a coleção produto.

Campo	Tipo de Dado	Descrição	Restrições
ID	Texto (automático)	Identificador único do produto	Gerado automaticamente
Nome	Texto curto	Nome do produto	Obrigatório
Preço	Número decimal	Preço do produto em reais	Obrigatório, > 0
Descrição	Texto longo	Descrição detalhada do produto	Opcional
Imagem	Arquivo de imagem	Imagem do produto	Obrigatório

## Conclusão

O uso do Webflow CMS simplificou a implementação do banco de dados físico, enquanto manteve os princípios da modelagem lógica e conceitual. O "banco de dados" foi implementado no contexto do Webflow mantendo a relação com os conceitos tradicionais de modelagem e persistência de dados.