

## FEEDBACK - Simulação de cliente: Iago

- ☒ ~~Avaliação da criação do ativador~~
- ☒ ~~Código ativador, semelhante a validação de Array porém como Objeto~~
- ☒ ~~Responsividade em diferentes telas~~



### Pontos de Atenção no CSS

#### 1. Cores fora do padrão da loja

- Foram criadas **novas variáveis de cores**, mas não houve reaproveitamento das já existentes no sistema da loja.
- ☒ Para essa configuração em específico, **já existem variáveis adequadas**, como:
  - `cor de botões`
  - `cor de texto dos botões`
  - `cor primária/secundária`
- ☒ Evitar criar novas variáveis quando o sistema já oferece as opções necessárias melhora a **padronização, consistência visual e facilidade de manutenção futura**.



### Pontos de Atenção no JavaScript

#### 1. Código ativador

- A configuração foi feita e está funcional, **mas não segue exatamente o padrão estabelecido para outros ativadores** do sistema.
- ☒ Pontos a corrigir:
  - Não é necessário utilizar o parâmetro `active`**. Nossa proposta é facilitar o uso por lojistas que nem sempre têm familiaridade com conceitos booleanos (`true/false`), e isso pode gerar confusão.
  - O nome do ativador precisa estar em português**, para manter a coerência com os demais e facilitar o entendimento do cliente final.
  - Não se deve expor IDs como parâmetro no ativador**. O controle de elementos internos deve ser feito diretamente no código, evitando tornar a configuração mais complexa do que o necessário.

#### 2. Criação de variações por vezes desnecessárias

- Foram criadas algumas variáveis que **não possuem reutilização ou necessidade de manipulação** posterior.
- ☒ Isso acaba deixando o código mais extenso e menos objetivo, sem trazer um ganho real.
- ☒ Em situações onde **não há necessidade de reaproveitamento ou iteração**, a criação direta no local de uso é mais prática e legível.

```
const landingPage = $('<div></div>');
landingPage.attr("id", landingPageActivator.id);

const configs = landingPageActivator.configurations;

// injeção dinâmica do html
landingPage.html(`
  <section>
    <div class="landingPageContainer">
      <div class="imageStore">
        <a href="${configs.linkButton}"></a>
      </div>

      <div class="textStore">
        <h1>${configs.title}</h1>
        <h2>${configs.subtitle}</h2>
        <p>${configs.description}</p>
        <a href="${configs.linkButton}"><button>${configs.buttonText}</button></a>
      </div>
    </div>
  </section>
`);

// lugar de injeção (before id #box-banners)
const banners = $("#box-banners");

if (banners.length > 0) {
  landingPage.insertBefore(banners);
}
```

- Poderia ter sido realizado de uma forma mais prática/simples eliminando a criação de algumas variáveis, afinal ela não terá reutilização ou manipulação por meio de Loop.
- **O código abaixo é uma representação mais otimizada do código criado anteriormente.**

```
if($("#box-banners").length) {

  $("#box-banners").before(`
    <div id="${landingPageActivator.id}">
      <section>
        <div class="landingPageContainer">
          <div class="imageStore">
            <a href="${configs.linkButton}"></a>
          </div>

          <div class="textStore">
            <h1>${configs.title}</h1>
            <h2>${configs.subtitle}</h2>
            <p>${configs.description}</p>
            <a href="${configs.linkButton}"><button>${configs.buttonText}</button></a>
          </div>
        </div>
      </section>
    </div>
  `);
}
```

👏 **No geral, as configurações atenderam às expectativas propostas.**

Apesar dos pontos levantados neste documento, que têm como objetivo aprimorar o padrão do código, é perceptível uma boa evolução no uso de **JavaScript** e **jQuery**, demonstrando progresso consistente e atenção às boas práticas.