

# Sabotadores U/X

Juliano Tesser, Nicolas Almeida, Victor Coelho

## 1 Resumo

Essa atividade envolveu o desenvolvimento de um protótipo de site de compras com o objetivo de analisar e evidenciar o impacto do mau uso da experiência do usuário (UX) em interfaces digitais. O fluxo construído contemplou as principais etapas de um e-commerce — login, página inicial, busca, exibição de dados do produto, carrinho e finalização da compra. No entanto, ao contrário de um site funcional e otimizado, serão adicionados vários defeitos propositais para que torne a experiência do usuário o mais desagradável possível.

O método utilizado baseou-se na simulação do fluxo de compra tradicional, seguido da implementação de barreiras e elementos que atrapalham a usabilidade, como caminhos redundantes, sobrecarga de informações, botões confusos e posicionamento pouco intuitivo.

Como resultado, em vez de oferecer praticidade, ele ilustra de maneira prática como más decisões de design podem comprometer a interação, afastar usuários e diminuir a eficiência de um sistema.

Esse projeto tem a intenção de destacar o uso de uma UX bem planejada em projetos digitais, mostrando que não basta apenas ter um layout esteticamente agradável; é fundamental projetar interfaces que facilitem e orientem a navegação, garantindo a satisfação e o engajamento do usuário.

## 2 Tema do projeto

Esta atividade consistiu na criação de um protótipo de site de compras com o objetivo de analisar e demonstrar os efeitos do mau uso da experiência do usuário (UX) em interfaces digitais. O fluxo desenvolvido abrangeu as principais etapas de um e-commerce — login, página inicial, pesquisa, exibição de produtos, carrinho e finalização da compra. Contudo, diferentemente de um site funcional e bem projetado, foram inseridos erros de forma intencional para tornar a experiência do usuário o mais desagradável possível.

## 3 Situação atual do projeto

Nesta entrega, o projeto está finalizado e funcional dentro de sua proposta. O usuário já pode realizar todo o fluxo de compra simulado, e um novo componente foi implementado para indicar qual é a tarefa proposta em cada página. Essa tarefa foi definida de modo a incentivar o usuário a explorar ao máximo a proposta do site, e o design desse componente foi pensado para que essas instruções pareçam deslocadas, evitando que sejam confundidas com elementos naturais da página de e-commerce.

A página de finalização da compra, que não estava pronta na entrega anterior, foi construída. Também aprimoramos o JSON de produtos em estoque e organizamos imagens para garantir diferenciação visual entre os itens, além de manter uma quantidade considerável de produtos para dar utilidade real à funcionalidade de filtro.

A manipulação de produtos no carrinho foi implementada, bem como as mensagens de feedback ao adicionar itens ou alterar suas quantidades.

### 3.1 Sabotagens implementadas

Para controlar a mudança de comportamento dos componentes, optamos por utilizar um valor armazenado no localStorage. Quando o usuário escolhe qual fluxo deseja seguir, esse valor é definido. As sabotagens foram implementadas de modo a quebrar convenções de design — como remover o feedback ao adicionar um produto ao carrinho — ou a dificultar a atividade proposta, baseando-se em situações conhecidas pelos membros da equipe, como o filtro com comportamento atípico e o pop-up.

Figura 1: Componente indicando a tarefa do usuários. Pode ser escondido. Versão preliminar



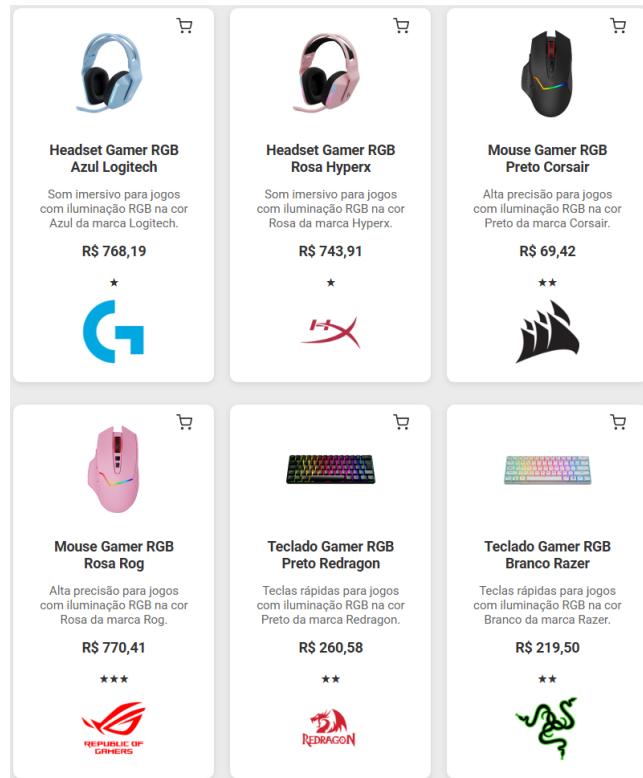
Fonte: Autoria própria.

Figura 2: Pop-up que atrapalha o uso e redireciona para outra página



Fonte: Autoria própria.

Figura 3: Produtos diferenciados por cor e logo da marca



Fonte: Autoria própria.

Figura 4: Carrinho reseta ao sair da página de checkout



Fonte: Autoria própria.

Figura 5: Filtros intencionalmente sabotados, introduzindo atrasos na resposta e ocultação temporária de valores exibidos ao usuário.

The figure shows a user interface for filtering search results. It includes sections for 'Categoría' (Mouse, Teclado, Fone), 'Marca' (Logitech, Redragon, Razer, Corsair, Hyperx, Rog), 'RGB' (Sim, Não), 'Cor' (Preto, Branco, Cinza, Azul, Verde, Rosa), 'Avaliação' (\*\*\*\*\* to \*\*\*), and a 'Preço' slider set to 'Até R\$ 10001'. The interface is designed to demonstrate intentional sabotage by introducing delays and temporary value hiding.

Fonte: Autoria própria.

## 4 Organização da Equipe e Projeto

Nesta etapa do projeto, a organização do grupo evoluiu em relação à estrutura inicial adotada. No início, os papéis dos integrantes não eram fixos, e todas as atividades — desde o levantamento de requisitos até os testes — eram realizadas de forma colaborativa e pouco estruturada. Com o avanço do desenvolvimento, tornou-se necessário adotar um modelo de trabalho mais organizado e alinhado às práticas ágeis.

Atualmente, a equipe mantém a colaboração entre os membros, mas segue um processo mais estruturado, que além de usar XP inicialmente, foram adotadas práticas de Scrum e Kanban, como reuniões para planejamento de sprint e revisão das entregas, definição de backlog com prioridades e responsáveis, e acompanhamento das tarefas por meio de status (A fazer, Em andamento, Concluído).

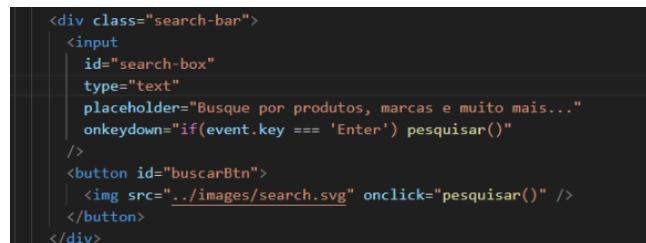
Mesmo com essa maior organização, o grupo preserva princípios do XP, como comunicação constante, busca por soluções simples e melhoria contínua. A prática de pair programming foi substituída por revisões de código entre os integrantes, assegurando a qualidade do desenvolvimento.

Uma das páginas planejadas foi removida (página de produto), pois a equipe julgou que a interação com esta página consumiria tempo sem oferecer nada substancial à ideia do projeto, que é implementar falhas de design ou inconveniências propositais, ao menos, não da forma que consideramos mais efetiva. Outra página foi adicionada ao planejamento, mostrando o resultado (em tempo) para a conclusão da tarefa e redirecionando o usuário para que a repita no fluxo alternativo (sabotado ou normal).

## 5 Exemplos de Código

### 5.1 HTML

Figura 6: Imagem: código html antigo. Código javascript inline foi removido posteriormente seguindo recomendações



```
<div class="search-bar">
  <input
    id="search-box"
    type="text"
    placeholder="Busque por produtos, marcas e muito mais..."
    onkeydown="if(event.key === 'Enter') pesquisar()"
  />
  <button id="buscarBtn">
    
  </button>
</div>
```

Fonte: Autoria própria.

Essa função valida o login do usuário de forma propositalmente limitada. Ela representa uma mecânica de interação frustrante, típica do site sabotado, exibindo uma mensagem confusa caso o login não seja exatamente "admin/admin".

Figura 7: Imagem: código html.



```
<div id="overlay_login">
  <div id="loginModal">
    <h2>Login</h2>
    <input type="text" id="usuario" placeholder="Usuário" autofocus />
    <input type="password" id="senha" placeholder="Senha" />
    <button id="entrarBtn">Entrar</button>
    <div id="msg"></div>
  </div>
</div>
```

Fonte: Autoria própria.

Responsável por capturar o texto digitado e redirecionar o usuário para a página de resultados. É um exemplo importante de manipulação de DOM e redirecionamento de páginas, mostrando como o JavaScript conecta a interface com a navegação real.

### 5.2 CSS

O modelo de layout CSS utilizado baseia-se no display flex, que permite organizar e alinhar elementos de forma eficiente e responsiva. Com o flexbox, os componentes podem ser distribuídos horizontal ou verticalmente, mantendo o alinhamento mesmo com diferentes tamanhos de tela. Essa abordagem foi usada, por exemplo, para centralizar elementos na tela e alinhar itens dentro de contêineres, garantindo um layout limpo e adaptável a diferentes dispositivos.

Figura 8: Imagem: código css.

```
.banner {  
    text-align: center;  
    background-color: var(--bg-color);  
    padding: 20px 0;  
}  
  
.banner img {  
    border-radius: 8px;  
    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);  
}
```

Fonte: Autoria própria.

Define a aparência da área do banner principal. O border-radius arredonda as bordas e o box-shadow dá profundidade à imagem. Essas regras deixam o banner visualmente destacado e moderno, contribuindo para a identidade visual do site.

Figura 9: Imagem: código css.

```
/* ===== Login Modal ===== */  
#overlay_login {  
    position: fixed;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
    background-color: rgba(0,0,0,0.5);  
    z-index: 3;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
}
```

Fonte: Autoria própria.

Esse CSS cria uma sobreposição translúcida sobre toda a tela. O uso de position: fixed e z-index garante que o modal apareça sempre acima de todos os elementos. É um exemplo claro de uso de posicionamento e hierarquia visual no layout.

### 5.3 Javascript

Figura 10: Imagem: código javascript.

```
function tentarLogin() {
    const usuario = usuarioInput.value.trim();
    const senha = senhaInput.value.trim();

    if (usuario === "admin" && senha === "admin") {
        overlay.style.display = "none";
    } else {
        msg.textContent = "Tente aquele login padrão do seu roteador";
    }
}
```

Fonte: Autoria própria.

Essa função valida o login do usuário de forma propositalmente limitada. Ela representa uma mecânica de interação frustrante, típica do site sabotado, exibindo uma mensagem confusa caso o login não seja exatamente "admin/admin".

Figura 11: Imagem: código javascript.

```
function pesquisar() {
    const termo =
document.getElementById("search-box").value.trim();

    if (termo) {
        window.location.href =
`./search/index.html?busca=${encodeURIComponent(termo)}`;
    }
}
```

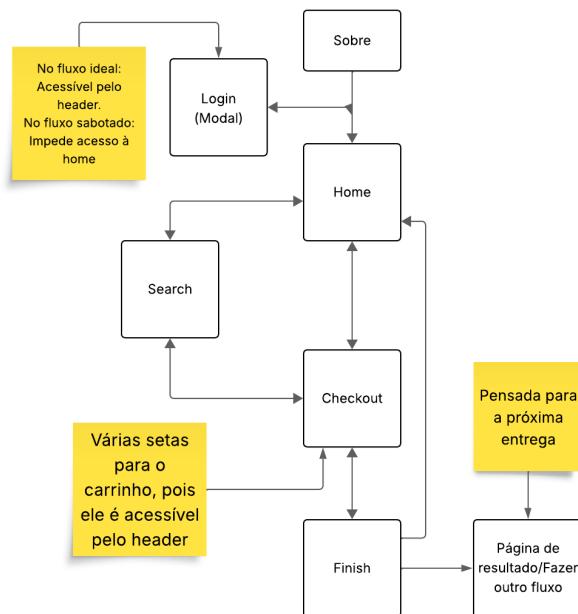
Fonte: Autoria própria.

Responsável por capturar o texto digitado e redirecionar o usuário para a página de resultados. É um exemplo importante de manipulação de DOM e redirecionamento de páginas, mostrando como o JavaScript conecta a interface com a navegação real.

## 6 Mapa do site

A Figura 12 apresenta uma versão base do site de e-commerce. Sobre ele serão implementados elementos que dificultem o seu uso. O fluxo representado no mapa serve para ambas as versões (normal e sabotada), que são diferenciadas pelo comportamento de certos elementos da página.

Figura 12: Diagrama: mapa do site.



Fonte: Autoria própria.

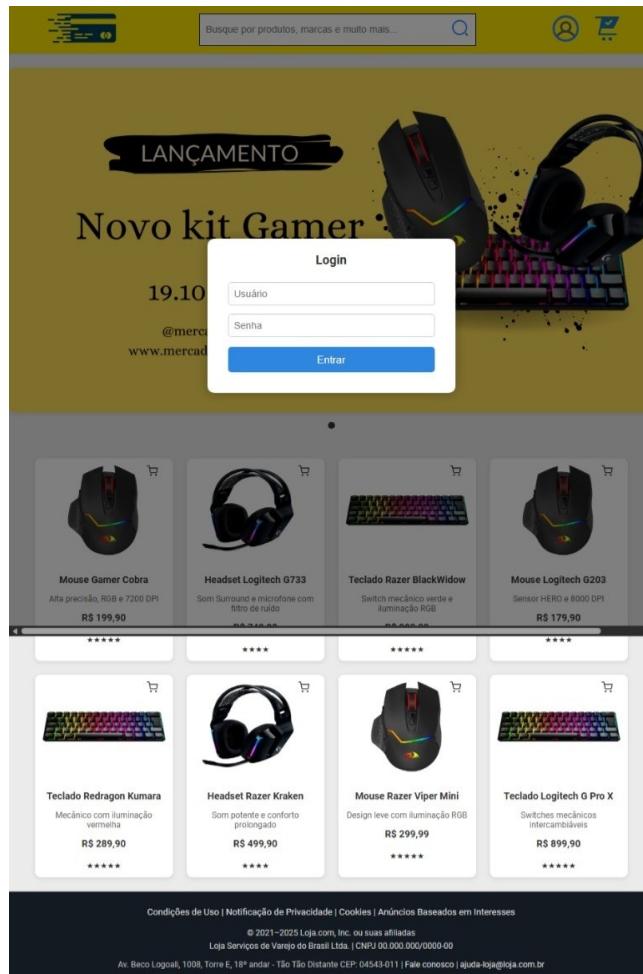
## 7 Wireframe do leiaute do projeto

Nesta seção será mostrado esboços das telas que iremos desenvolver no projeto.

### 7.1 Home

A página Home, apresentada na figura 13, será acessível via logo do site, representado pelo círculo amarelo. Possui o cabeçalho com logo, barra de busca, acesso ao login e carrinho. Tem banner promocional, indicação de produtos e um rodapé.

Figura 13: Layout: página home.

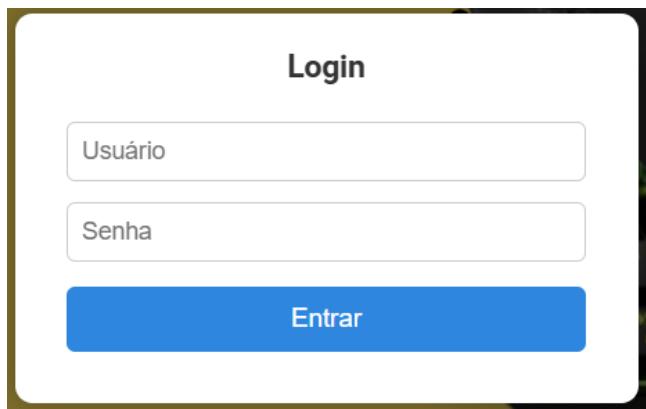


Fonte: Autoria própria.

## 7.2 Modal de login

A modal de login, apresentada na figura 14, será acessível pelo elemento representado pelo emoji com óculos presente na home page e em outras páginas.

Figura 14: Layout: modal login



Fonte: Autoria própria.

## 7.3 Página das informações do projeto

A página das informações do projeto, apresentado na figura 15, será Acessível pelo rodapé da home.

Figura 15: Layout : pagina dos dados do projeto.

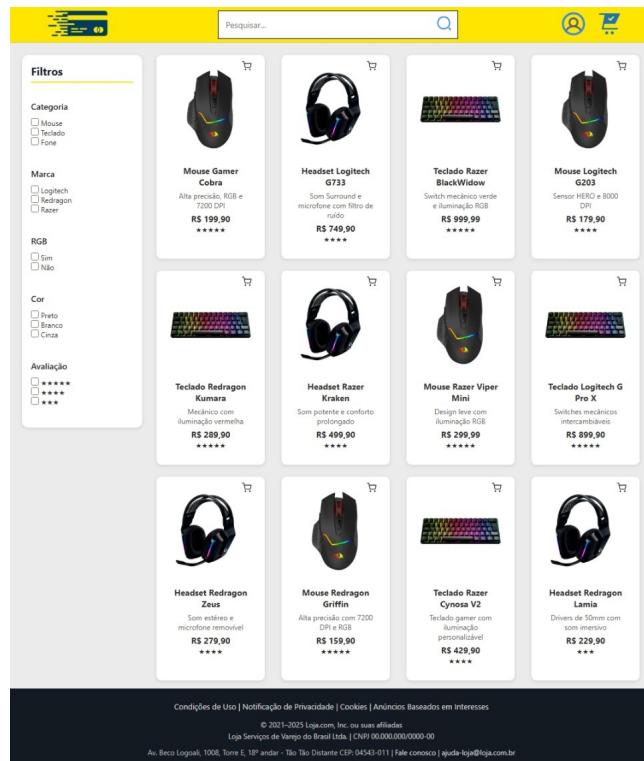
The screenshot shows the homepage of the 'Sabotadores U/X' project. At the top, there's a dark header bar with the project name in white. Below it, a main content area has a light gray background. The first section is titled 'Resumo' (Summary) which contains a brief description of the project's purpose and its findings. The second section is 'Descrição do Projeto' (Project Description), detailing the prototype's structure and the comparison between the sabotaged version and the one with good UX practices. The third section is 'Descrição do Site' (Site Description), explaining the elements of the e-commerce mockup. The bottom section is 'Iniciar teste' (Start Test), featuring a yellow callout box with a warning about the experiment's goal, followed by a 'Missão do teste' (Test Mission) list and two large buttons for selecting a platform: 'Acessar site Sabotado' (Access Sabotaged Site) in red and 'Acessar site Com boas práticas de UX' (Access Site with Good UX Practices) in blue. At the very bottom, there's a footer with small text about the project's development and license.

Fonte: Autoria própria.

## 7.4 Página de resultados da busca

A página de resultados da busca, apresentado na figura 16, será acessível a partir do elemento barra de busca. Contém o mesmo cabeçalho da home, uma área lateral para ajustar filtros de busca e uma área central com os produtos válidos para a busca.

Figura 16: Layout: página de resultados de busca

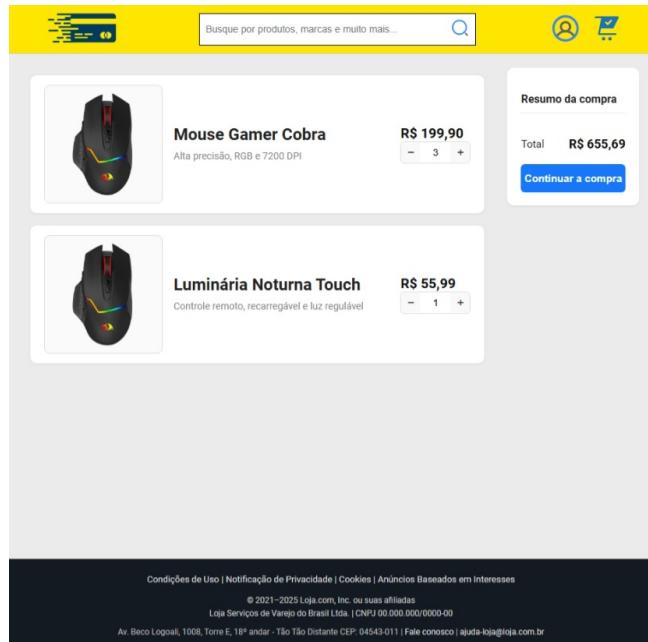


Fonte: Autoria própria.

## 7.5 Página de carrinho

A página de carrinho, apresentada na figura 17, contém o mesmo cabeçalho da home. Nela é possível alterar a quantidade de cada ítem no carrinho e excluir algum dos produtos. Exibe subtotal e dá acesso a página de finalização da compra.

Figura 17: Layout: página de carrinho

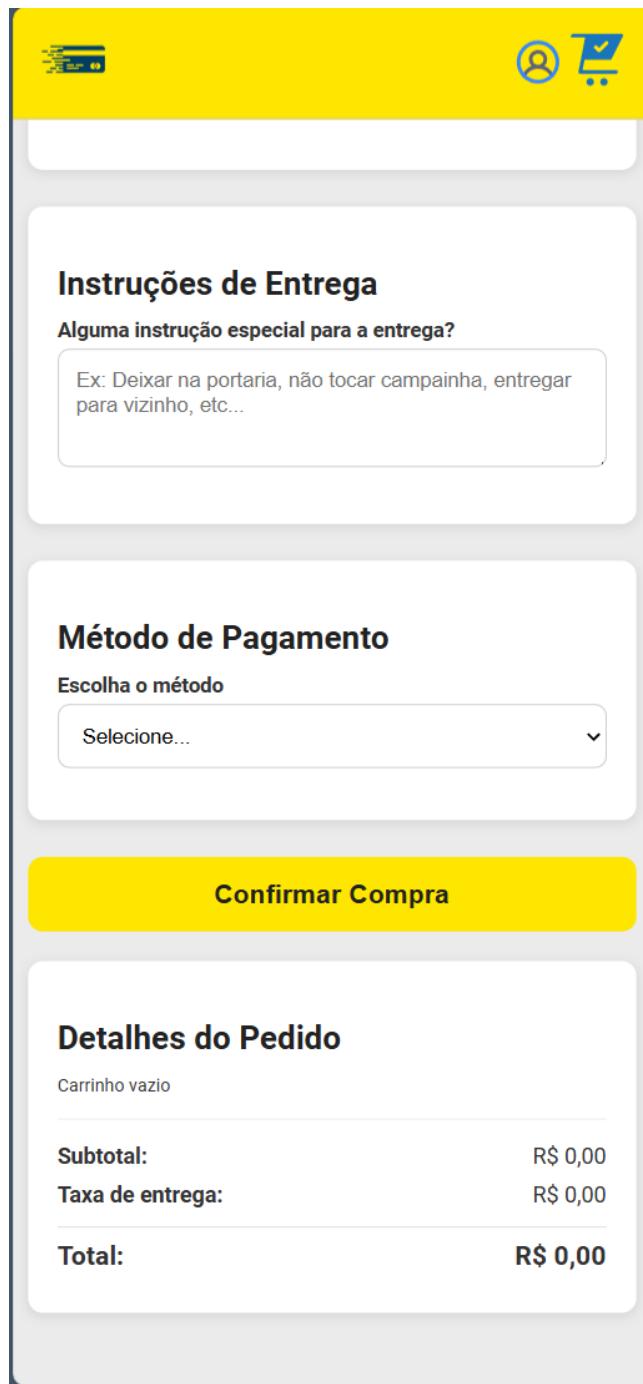


Fonte: Autoria própria.

## 7.6 Página de finalização da compra

A página de finalização da compra, apresentado na figura 18, será acessível pela página de carrinho. Nela é possível escolher endereço de entrega, modalidade de frete, meios de pagamento e confirmar a compra. O elemento representado pela seta branca em fundo azul retorna para a página de carrinho após um click.

Figura 18: Layout: página de pagamento na visualização mobile (implementada nesta última entrega)



Fonte: Autoria própria.

## 8 Ações durante a conclusão do projeto

Para a conclusão do projeto, a equipe concentrou esforços na finalização das funcionalidades restantes e na correção de eventuais falhas identificadas durante os testes. Também foram realizadas modificações na UX do projeto, para incluir uma versão com a UX "defeituoso" e uma com boas práticas de UX.

Ao definir melhor as diferenças entre o site sabotado e o normal, definimos também as tarefas de forma que essas diferenças fossem impactantes durante a navegação. Nesta etapa também foram testados mais casos diferentes para ajuste da responsividade, que consideramos um requisito importante.

O site entregue foi construído de maneira a ter exibição adequada nos casos mais comuns de uso e a funcionalidade de simular estes ambientes no modo desenvolvedor do navegador foi constantemente usada. A equipe também buscou remover as chamadas inline de javascript presentes nos arquivos html, buscamos também corrigir a inserção de imagem via javascript (algumas estavam sem o atributo alt, por exemplo)