

# *Rebranding da Plataforma Skoob Aplicando User Experience, User Interface e Design Centrado no Usuário*

**Naiury C. da Silva, André C. da Silva**

Grupo de Pesquisa Mobilidade e Novas Tecnologias de Interação  
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas  
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP)  
Campus Hortolândia – SP – Brasil

chaves.naiury@aluno.ifsp.edu.br, andre.constantino@ifsp.edu.br

**Abstract.** *Skoob is an online platform dedicated to the world of books, where users can track their readings, write reviews, interact with other readers, and discover new titles. The rebranding project for Skoob was initiated with the goal of rethinking the user experience on a platform criticized for its confusing and outdated layout. Focused on usability concepts and user-centered design, the project proposes improvements such as reducing clicks, increasing clarity in actions, and optimizing navigation. To achieve this, knowledge of Human-Computer Interaction and Web Development was applied, along with UX/UI research to gather data. During the process, frameworks like NextJS were used to create the web app interfaces. One of the key features of the new version is the inclusion of the floating "Quick Actions" button and action status notifications, offering a more efficient and satisfying navigation experience.*

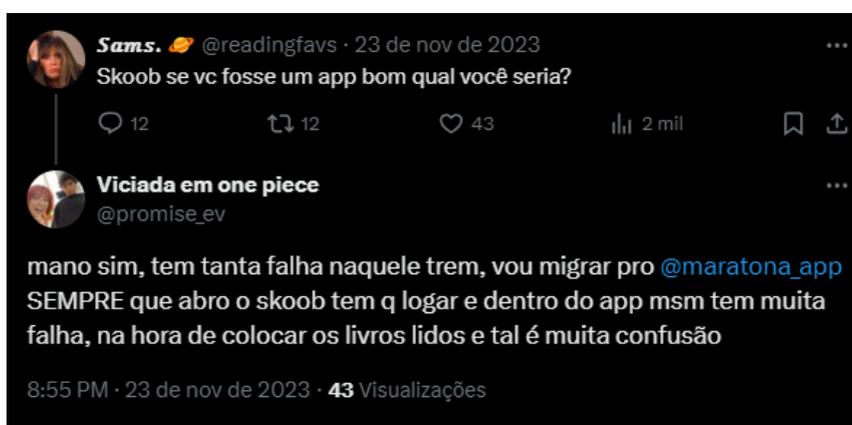
**Resumo.** *O Skoob é uma plataforma online dedicada ao universo dos livros, onde os usuários podem registrar suas leituras, fazer resenhas, interagir com outros leitores e descobrir novas obras. O projeto de rebranding do Skoob surgiu com o objetivo de repensar a experiência do usuário em uma plataforma criticada por seu layout confuso e desatualizado. Focado em conceitos de usabilidade e design centrado no usuário, o projeto propõe melhorias como a redução de cliques, maior clareza nas ações e otimização da navegação. Para isso, foram aplicados conhecimentos de Interação Humano-Computador e Desenvolvimento Web, além do uso de UX/UI na parte de pesquisa para levantamento de dados. Durante o processo, frameworks como NextJS foram utilizados para criar interfaces do web app. Um dos diferenciais da nova versão é a inclusão do botão flutuante "Quick Actions" e notificações de status em ações, oferecendo uma navegação mais eficiente e satisfatória.*

## **1. Introdução**

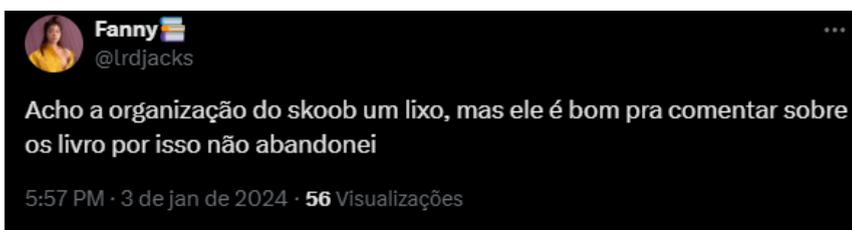
*Rebranding* é um termo em inglês que significa, em tradução livre, “reformulação” e pode ser definido, segundo Muzellec (2003) como “a prática de reconstruir um nome representativo de uma posição diferenciada na mente dos *stakeholders* e uma identidade distintiva dos concorrentes”. Neste contexto, o objetivo deste trabalho é realizar um *rebranding* na plataforma Skoob apresentando-lhe uma nova interface de usuário. A plataforma Skoob, fundada em 2009, atua como uma rede social para leitores e como um assistente literário para mais de 9 milhões de pessoas (Skoob, 2023). Dentro da plataforma é possível adicionar livros à sua estante, classificando-os como “lidos”, “lendo”, “quero ler”, entre outros divisores. Também é possível publicar resenhas e *feedbacks* avaliativos sobre cada livro lido. No entanto, em relatos nas redes sociais, é possível perceber que a plataforma não está cumprindo com as expectativas dos usuários, dessa forma, perdendo mercado para a concorrência -

aplicativos como “Maratona” com mais de 50 mil downloads em bibliotecas de apps (Maratona, 2024) e “Goodreads” com mais de 75 milhões (Goodreads, 2024).

Há diversas razões para implementar um *rebranding* de uma marca, mas é possível correlacionar com o propósito deste projeto alguns motivos dentre os definidos por Duncan (*apud* Goi; Goi, 2011), com a finalidade de acompanhar o tempo e acompanhar mudanças nas necessidades dos consumidores (por exemplo, serviços, acessibilidade, conveniência, escolha, moda e tecnologia); porque (i) uma marca ficou antiquada e está em perigo de estagnação ou já está em estado de erosão; (ii) como forma de bloquear ou manobrar concorrentes; ou (iii) uma forma de lidar com o aumento de competitividade. É de grande notoriedade que esses propósitos se aplicam ao caso do Skoob quando pesquisa-se por seu nome em redes sociais como o Twitter, por exemplo, as postagens expostas nas Figuras 1 e 2 nos anos de 2023 e 2024:



**Figura 1. Captura de tela de um tweet com críticas ao Skoob**



**Figura 2. Captura de tela de um tweet com crítica ao Skoob**

Para investigar a fundo quais os pontos de melhoria da plataforma, pretende-se realizar uma entrevista com os usuários, que irá permitir compreender qual a interação dos mesmos com o produto. Por meio dessa interação, o próximo passo é avaliar até que ponto a experiência é agradável e quais medidas devem ser tomadas para trazer maior benefício ao público-alvo.

Depois de realizada as entrevistas e coletados os dados, será desenvolvido um projeto visual com as melhorias identificadas e, após, um estudo de experiência dos usuários. Esse desenvolvimento se baseará nos conceitos de uma boa *User Experience/User Interface* (de sigla UX/UI, traduzido como Experiência de Usuário e Interface de Usuário) e a abordagem *User Centered Design* (de sigla UCD, traduzido como *Design* Centrado no Usuário).

O presente artigo é organizado da seguinte maneira: Na Seção 2 são apresentados os conceitos de usabilidade, navegabilidade, experiência de usuário, interface do usuário, *Design Centrado no Usuário*, desenvolvimento PWA e API. Na Seção 3, são apresentadas as metodologias escolhidas para o desenvolvimento do projeto. Na Seção 4, é apresentado, dividido em etapas, cada fase da construção desse trabalho. O artigo é encerrado na Seção 5 com a apresentação da conclusão.

## **2. Fundamentação Teórica**

### **2.1. Usabilidade**

Segundo Nielsen (2012), usabilidade é um atributo de qualidade que avalia a facilidade de uso das interfaces de usuário. Esse termo técnico possui 5 principais pilares, que podem ser identificados a partir das perguntas:

- **Aprendizagem:** Para os usuários, o quanto é fácil realizar tarefas simples no primeiro contato com o produto?
- **Eficiência:** Uma vez aprendido sobre o produto, o quanto rápida as tarefas podem ser executadas?
- **Memorabilidade:** Depois de um período sem utilizar o produto, quanto fácil é restabelecer a proficiência?
- **Erros:** Quantos erros são cometidos durante o uso? Qual a gravidade de cada um desses erros? O quanto fácil é se recuperar desses erros?
- **Satisfação:** Quanto agradável é usar o *design*?

Um *design* ou produto com boa usabilidade é capaz de prover, baseado nessas características acima, um fácil uso - o que colabora para uma boa experiência. É notório identificar a importância do mesmo no exemplo a seguir, proposto pelo Nielsen: Se um site for difícil de usar, as pessoas vão embora. Se a página inicial não indicar claramente o que uma empresa oferece e o que os usuários podem fazer no site, as pessoas vão embora. Se os usuários se perdem em um site, eles saem.

### **2.2. Navegabilidade**

A navegabilidade refere-se à facilidade com que um usuário pode se deslocar por um sistema ou interface digital para alcançar seus objetivos. Segundo Nielsen em seu livro "*Usability Engineering*" (1994), a navegabilidade é um dos aspectos cruciais da usabilidade, sendo definida como "a capacidade de um sistema ou interface permitir que o usuário encontre o que procura de forma intuitiva e eficiente".

### **2.3. Experiência do Usuário**

Segundo Hassenzahl (2003) no livro "*The Thing and I: Understanding the Relationship Between User and Product*", a experiência do usuário emerge da interação do indivíduo com um produto ou sistema, mediada pelo seu contexto de uso. Essa interação vai além da usabilidade, abrangendo fatores emocionais e subjetivos.

### **2.4. Interface de Usuário**

Segundo o livro "*What Is User Interface Design?*" (Harris, 2017), o *design* da interface do usuário (UI) foca na interação do usuário com os elementos visuais apresentados na tela. Um

bom *design* de UI melhora a experiência do usuário ao fornecer elementos de *design* que são fáceis de entender e usar.

Os *designers* de UI se concentram nos elementos visuais que os usuários veem ao utilizar um software ou site para atingir seus objetivos. Eles se preocupam com a aparência de um site ou programa e com a forma como os usuários interagem com ele.

## 2.5. Design Centrado no Usuário

O *Design Centrado no Usuário* (Em inglês, *CDU*) é uma abordagem cujo propósito é criar soluções que sejam úteis, desejáveis e acessíveis para os usuários. Assim, é essencial compreender as dores, desejos e comportamentos dos usuários em relação ao produto, adaptando-o para que atenda às necessidades do público-alvo, como pode-se notar a partir da seguinte fala:

É uma abordagem para o desenvolvimento de soluções que têm como propósito promover a criação de produtos que sejam mais úteis para os usuários, atendam suas necessidades e exigências, estejam adaptados às suas características e sejam fáceis de usar. A ideia é que a solução seja adaptada aos usuários, e não o contrário. ((ISO 9241 Part 210))

Os princípios-chave do UCD são fundamentais para garantir que soluções de *design* atendam às reais necessidades dos usuários, proporcionando experiências significativas e funcionais. Esses princípios podem ser descritos da seguinte forma:

- O *design* da interface é criado com base em uma compreensão sobre quem são os usuários, quais tarefas precisam realizar e em quais contextos utilizam o sistema;
- Os usuários participam ativamente de todo o processo de *design* e desenvolvimento, garantindo que suas perspectivas e *feedback* sejam considerados desde o início;
- O projeto é guiado e continuamente refinado por meio de testes e avaliações que priorizam a experiência e as necessidades dos usuários;
- O *design* é um processo dinâmico, onde mudanças e melhorias são realizadas em ciclos contínuos de iteração;
- O objetivo principal é criar uma experiência integrada e satisfatória para o usuário, levando em conta todos os aspectos da interação com o sistema
- A diversidade de habilidades é essencial para o sucesso do projeto, com equipes compostas por profissionais de diferentes áreas colaborando ativamente.

## 2.6. Desenvolvimento PWA (*Progressive Web APP*)

Um *Web App* utiliza tecnologias da web para proporcionar uma experiência de usuário semelhante à de um aplicativo nativo, incluindo a capacidade de funcionar *offline*, enviar notificações *push* e ser adicionadas à tela inicial do dispositivo. Segundo o livro “*Progressive Web Apps*” (ATER, 2017), um *Progressive Web App* (PWA) é a maneira padrão de oferecer uma experiência moderna na *web* para dispositivos móveis. Os PWAs podem fornecer soluções rápidas e jornadas atraentes semelhantes ao que pode ser alcançado com um aplicativo nativo, mas são também detectáveis e acessíveis a todos da *web*.

## 2.7. API (*Application Programming Interface*)

APIs, ou Interfaces de Programação de Aplicações (*Application Programming Interfaces*), são conjuntos de definições e protocolos que permitem que diferentes softwares se comuniquem

entre si. Segundo o livro “*REST API Design Rulebook*” (MASSÉ, 2011), os programas clientes usam interfaces de programação de aplicativos (APIs) para se comunicar com serviços da *web*. De modo geral, uma API expõe um conjunto de dados e funções para facilitar as interações entre programas de computador e permitir-lhes trocar informação.

### 3. Metodologia

Por se tratar de um *rebranding* de uma plataforma que recebe críticas dos usuários, o ponto inicial foi entender as “dores” e necessidades não atendidas dos mesmos a partir de entrevistas com um grupo pequeno, com diferentes tipos de usuários que estão dentro do público alvo. Em seguida, elaborar um quadro resumindo as principais dificuldades encontradas, que posteriormente se tornarão os principais pontos a serem trabalhados no *rebranding*. Após mapeados as funcionalidades a serem trabalhadas, elabora-se um protótipo que consome informações de uma API, e que é refinado em reuniões com o mesmo grupo de pesquisa e melhorado conforme apontamentos dos mesmos, definindo um *design* final para o usuário.

Para construir as telas, o HTML foi empregado na construção da estrutura fundamental do site, incluindo títulos, textos e inclusão de imagens. Por outro lado, o CSS organizou esses elementos, estabelecendo cores, tamanhos e posições na tela. O Next.js foi o framework escolhido para desenvolver o *web app*, possibilitando a utilização de renderização no servidor (SSR) e rotas dinâmicas, além de simplificar a administração de componentes reutilizáveis, assegurando que o projeto atenda ao padrão definido nos wireframes.

Para melhorar a experiência do usuário, a biblioteca Next PWA foi escolhida para converter o site em um aplicativo web progressivo (PWA). Isso permite o funcionamento offline e o armazenamento automático de páginas e recursos, melhorando a disponibilidade geral do site, como, por exemplo, em conexões instáveis. Com a *Internet*, a utilização da Next PWA também possibilita a instalação do site em aparelhos móveis e desktops, oferecendo uma experiência mais envolvente e de fácil acesso, emulando um verdadeiro *app*.

Para realizar a gestão do código selecionou-se o sistema de versionamento de código Git, que permite o envio dos arquivos para um repositório online. Além disso, o Git oferece a possibilidade de acompanhar todas as versões do código, armazenando o histórico completo das alterações. Para gerenciar esse versionamento, foi utilizada a plataforma GitHub, que fornece uma interface web para o *upload* e gerenciamento dos arquivos.

### 4. Desenvolvimento

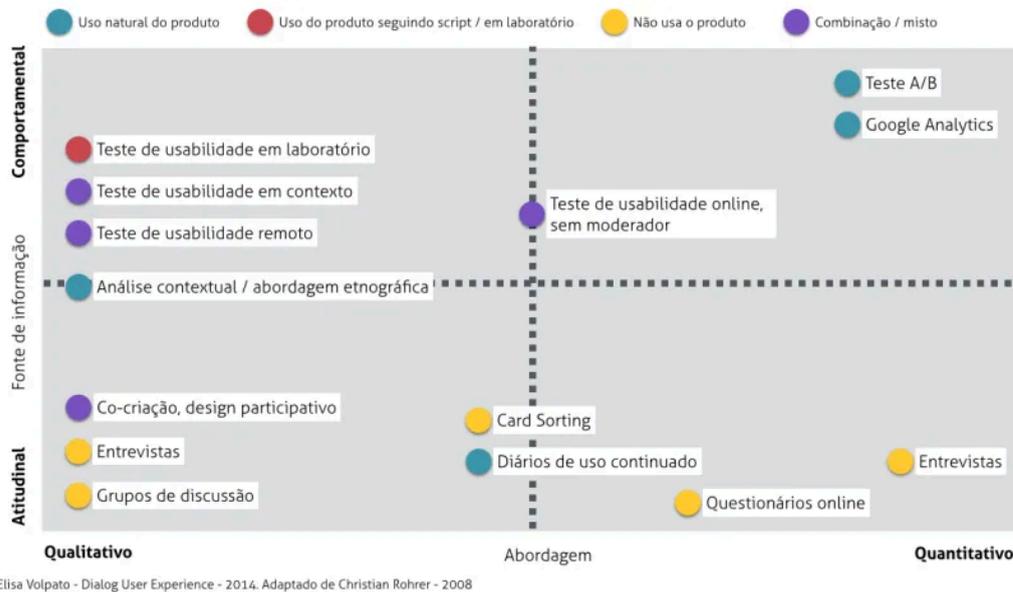
Considerando a metodologia descrita, apresentamos a execução detalhada de cada atividade nas subseções seguintes.

#### 4.1. Entrevistas

Para iniciar o desenvolvimento na perspectiva do DCU, é necessário compreender os usuários. Para escolher a melhor técnica usa-se as orientações de Christian Rohrer (2024 *apud* Volpato, 2014), que criou um diagrama (Figura 3) que ajuda a entender qual metodologia se aplica melhor às necessidades de um projeto. Nesse diagrama, os eixos que se encaixam melhor com o propósito dessa pesquisa são: qualitativo e atitudinal.

Os métodos qualitativos são mais adequados para responder perguntas sobre porquê ou como resolver um problema, e os métodos atitudinais para compreender ou medir o que

elas pensam ou sentem sobre determinado cenário. Com essas informações, ficam disponíveis cinco métodos de pesquisa, entre eles as entrevistas.



**Figura 3. Diagrama de Elisa Volpato para ajudar a escolher uma abordagem com base no objetivo de um projeto. Fonte: (Volpato, 2014)**

Para uma conversa mais dinâmica, foi escolhida a entrevista semiestruturada (com um roteiro, mas livre para perguntas que surgem durante o momento). O roteiro de entrevista foi constituído por perguntas direcionadas ao problema e exploração de tópicos relacionados à experiência do usuário durante a execução de diversas funcionalidades dentro do app com o tempo de duração de 20 a 35 minutos por pessoa, que foi o suficiente para entender como o usuário utiliza o Skoob e para mapear possíveis dores e oportunidades de melhoria para o *rebranding*. O diagrama abaixo ilustra o porquê da escolha do método de entrevista, ao avaliar

O grupo de pesquisa foi composto por:

1. Pessoas de 18 à 24 anos;
2. Mulheres (3) e homens (3);
3. Pessoas que já usaram o Skoob e pararam - 16,6% - Chamaremos de “ex-usuário”;
4. Pessoas que ainda usam Skoob - 16,6% - Chamaremos de “regular”;
5. Pessoas que nunca usaram o Skoob- 66,6% - Chamaremos de “novo”.

O objetivo com o grupo de pesquisa dividido em 3 tipos de usuários foi para: (i) capturar, com os usuários regulares, dicas de melhoria; (ii) descobrir os motivos que os ex-usuários pararam de utilizar o aplicativo; e (iii) entender as dificuldades dos novos usuários. Também preocupou-se em entender até que ponto o aplicativo tem uma boa memorabilidade - conceito que faz parte da usabilidade, e boa navegabilidade a partir do primeiro uso.

#### 4.1.1. Planejamento e execução

Para atender melhor cada tipo de usuário, dois roteiros distintos foram escritos: (i) focado na experiência que o usuário já habituado com o Skoob tem e (ii) focado no primeiro uso das principais funcionalidades. Porém, em ambos os roteiros os usuário precisam utilizar o

aplicativo. O Apêndice III apresenta as perguntas elaboradas. Do primeiro conjunto, destaque-se as perguntas:

1. Qual era sua motivação para usá-lo?
2. Você comentou desses divisores/marcadores, você já chegou a utilizar todos?
3. Quanto tempo você levou para aprender as principais funcionalidades?
4. E sobre a aba de mensagens? Você já utilizou? O que acha dela?

Do segundo conjunto, destacamos:

1. Vamos começar adicionando um livro. Por favor, faça o percurso que você acha que é para essa funcionalidade. Você também pode desistir se cansar de procurar.
2. Descreva e avalie de 0 a 10 sua experiência com essa funcionalidade
3. Olhando esses marcadores, você acha que todos eles teriam utilidade para você?
4. Qual foi o momento que você mais teve dificuldade no aplicativo?

O primeiro roteiro foi adaptado para buscar melhor entendimento da experiência dos usuários no primeiro contato com o aplicativo, resultando no segundo roteiro. Por sua vez, o segundo, avaliou a navegabilidade, usabilidade e experiência de usuário durante o primeiro uso.

Todos os participantes assinaram o termo de consentimento antes da entrevista ser marcada.

#### **4.1.2. Análise das entrevistas**

As entrevistas foram feitas via Google Meets (serviço de comunicação por vídeo desenvolvido pelo Google). Nela, os usuários (os 3 tipos - regular, ex-usuário, novo) com aplicativo em mãos acessaram as funcionalidades pedidas, como adicionar livros à estante virtual, mudar marcadores, adicionar resenhas etc. Após concluir cada ação referente à funcionalidade, os usuários definiram uma nota de 0 a 10 e uma descrição para experiência. Também foi marcado o tempo de execução de cada uma.

Após transcrever todas as entrevistas, ocorreu a sumarização dos dados, que foi um passo importante para transformar os dados “brutos” e dispersos das entrevistas em percepções que poderiam guiar o desenvolvimento do produto final de uma maneira mais alinhada com as reais necessidades dos usuários. As respostas dos usuários foram encaixadas em padrões/subgrupos para uma melhor identificação de padrões.

Dessa forma, foi levantado: tempo de execução, nota qualitativa e nota quantitativa. Três parâmetros muito importantes para serem avaliadas posteriormente o que seria trabalhado no *rebranding*. Por exemplo: Na Figura 4, para adicionar um livro, o tempo médio empregado foi de 57 segundos e a nota geral foi 3,5 de 10, já como nota qualitativa pode-se mencionar: “Não concordo de estar aí”, “Deveria ser na aba home ou na de livros”, “Escondido, tem que adivinhar”.

Pergunta	Entrevistado A	Entrevistado B	Entrevistado C	Entrevistado D	Entrevistado E	Entrevistado F
Tempo de uso	6 meses	Primeira vez	2 anos	Primeira vez	Primeira vez	Primeira vez
Como você faz para adicionar uma resenha sobre um livro?	Tempo 2 minutos e 4 segundos	Tempo 3 minutos e 37 segundos	25 segundos	3 minutos e 47 segundos	4 minutos e 01 segundo	2 minutos e 56 segundo
Qual sua opinião sobre como publicar uma resenha?	Muito confuso, nota 3	Escondido, nota 0	5, tem que procurar bastante - aprendeu com o tempo	Escondido e no local errado, nota 3	Acha que deve ser na aba de resenhas, nota 0	Escondido mas se procurar bastante acha, 3
Como você faz pra adicionar feedback de uma leitura em andamento?	Desistiu com 1 minuto e 5 segundos	Concluiu em 4 minutos e 22 segundos	27 segundos	3 minutos	3 minutos e 10 segundos	2 minutos 58 segundos
Qual sua opinião sobre como publicar um feedback de leitura?	Não conseguiu publicar 0	O usuário não vai achar 2	Tem que procurar bastante 2	Acha que está no local errado 0	Deveria ser na aba de resenhas também 0	Deveria ser mais destacado no aplicativo 3
O que você acha de ter essas funcionalidades na versão web? Por que?	Muito interessante	Útil	Tem, mas é bagunçado	Legal	Útil	Útil
Como você faz para adicionar um livro?	Concluiu em 30 segundos	Concluiu com 1 minuto e 30 segundos	Tempo 20 segundos	Tempo 1 minuto e 15 segundos	Tempo 1 minutos e 20 segundos	Tempo 47 segundos
Experiência adicionando livros	Escondido, tem que adivinhar, nota 3	Acha que a funcionalidade está no local errado, nota 3	Acho OK 5	Não concordo de estar ai 2	Deveria ser ou na home, ou na aba de livros 1	ok ser ai no busca 7

**Figura 4. Sumarização dos dados dos entrevistados**

Através das entrevistas, foi percebido que são consideradas as principais funcionalidades do aplicativo: Adicionar livro, resenha e feedback de leitura. Porém, nota-se pelo conteúdo da Figura 4, que o tempo e esforço empregados para encontrá-las foi grande ao considerarmos que parte dos usuários passaram mais de 3 minutos tentando concluir uma tarefa. O que viola os princípios de navegabilidade, que defende que a navegação deve ser intuitiva, que exija pouco esforço mental, facilite a ação e diminua a chance de erros.

Além da navegabilidade, o aplicativo também não está de acordo com os princípios da experiência de usuário, que defende que o objetivo com o uso do *app* seja atingido de maneira rápida, fácil e eficiente. Desse modo poderia trazer uma boa experiência, no entanto, o grupo de entrevistados teve grande dificuldade, além de relatarem uma experiência de confusão durante o uso.

Por fim, a plataforma também não está de acordo com os termos da usabilidade, visto que não foi fácil realizar tarefas simples no primeiro contato, violando o pilar aprendizagem.

Com os ex-usuários, percebe-se que a eficiência e a memorabilidade não tiveram um bom desempenho. Com os ex usuários e usuários novos, foram cometidos muitos erros até concluir uma ação e a satisfação geral não foi boa. Tudo isso contribui para que a dificuldade de uso do aplicativo aumente, e se o aplicativo for difícil, as pessoas não o usarão. Após reunir os dados na tabela, um mural foi criado dividido em 3 partes:

1. Caminho real para realização das funcionalidades como adicionar livros, mover marcador, adicionar resenha, adicionar histórico (Figura 1 do Apêndice I);
2. Caminho que a maioria dos usuários percorre (Figura 2 do Apêndice I);
3. Caminho proposto pelo *design* final do Skoob (Figura 3 do Apêndice I).

#### 4.2. Elaboração de *wireframes*

Para criar os protótipos de *design* foram criados *wireframes* que, segundo Becker (2019), são como desenhos de rascunhos do *design* do sistema. Para criação foi utilizada a ferramenta Figma, que atua como um editor gráfico de vetor e prototipagem de projetos de *design*.

A parte criativa do *design* foi elaborada analisando as críticas e sugestões dos usuários nas entrevistas realizadas. A principal intenção deste *rebranding* é fazer com que a plataforma seja fácil de reconhecer e utilizar. Com isso, algumas medidas foram tomadas para fazer com que o uso seja facilitado em relação ao *app* atual, os resultados serão discutidos na Seção “Discussão de resultados”.

Ao inserir as necessidades e evitar causar confusão ou complicação de uso dos usuários dentro das diferentes telas do sistema, a maioria das telas corresponderam a uma ou mais funcionalidades. A Figura 5 expõe os *wireframes* criados e avaliados pelo orientador deste projeto, André Constantino da Silva, na qual:

- **Frame B:** Uma estante virtual, cujo na versão atual, os possíveis marcadores de livros foram reduzidos a 3 principais estados (lido, lendo, quero ler);
- **Frame C:** Página de perfil, os *posts* agora são separados por “Progresso” e “Resenha”, ao invés de todos juntos;
- **Frame D:** Ainda no perfil, a aba de livros substitui a de vídeos e troca de ícone por nome da aba;
- **Frame F:** Botão *Quick Actions* para agilizar e facilitar principais funcionalidades, presente em várias telas;
- **Frame H:** Página de busca por livro, antes o botão “+” era menor e perto de outras áreas clicáveis. No *rebranding*, o botão “+” possui tamanho aumentado e é longe de outra área de clique;
- **Frame I:** Página de informação sobre um livro, na proposta deste trabalho a ação “Adicionar livro” está escrita e no centro ao invés de do lado do livro e em ícone;
- **Frame J:** Ainda na página de informação sobre um livro, a ação de “Adicionar livro” possui um *dropdown* para selecionar qual status do livro (lido, lendo, quero ler) ao invés de esperar um modal abrir;
- **Frame K:** Na página de informação sobre um livro, após clicar em “Adicionar livro” vem um atalho para adicionar resenha ou progresso assim que adicionar um livro à estante;
- **Frame L e M:** A página L é um formulário para postar um progresso e a M para postar uma resenha. Neste protótipo, a funcionalidade foi reduzida a uma só página;
- **Frame N:** Retorno de status da atividade após adicionar um livro ou uma publicação.

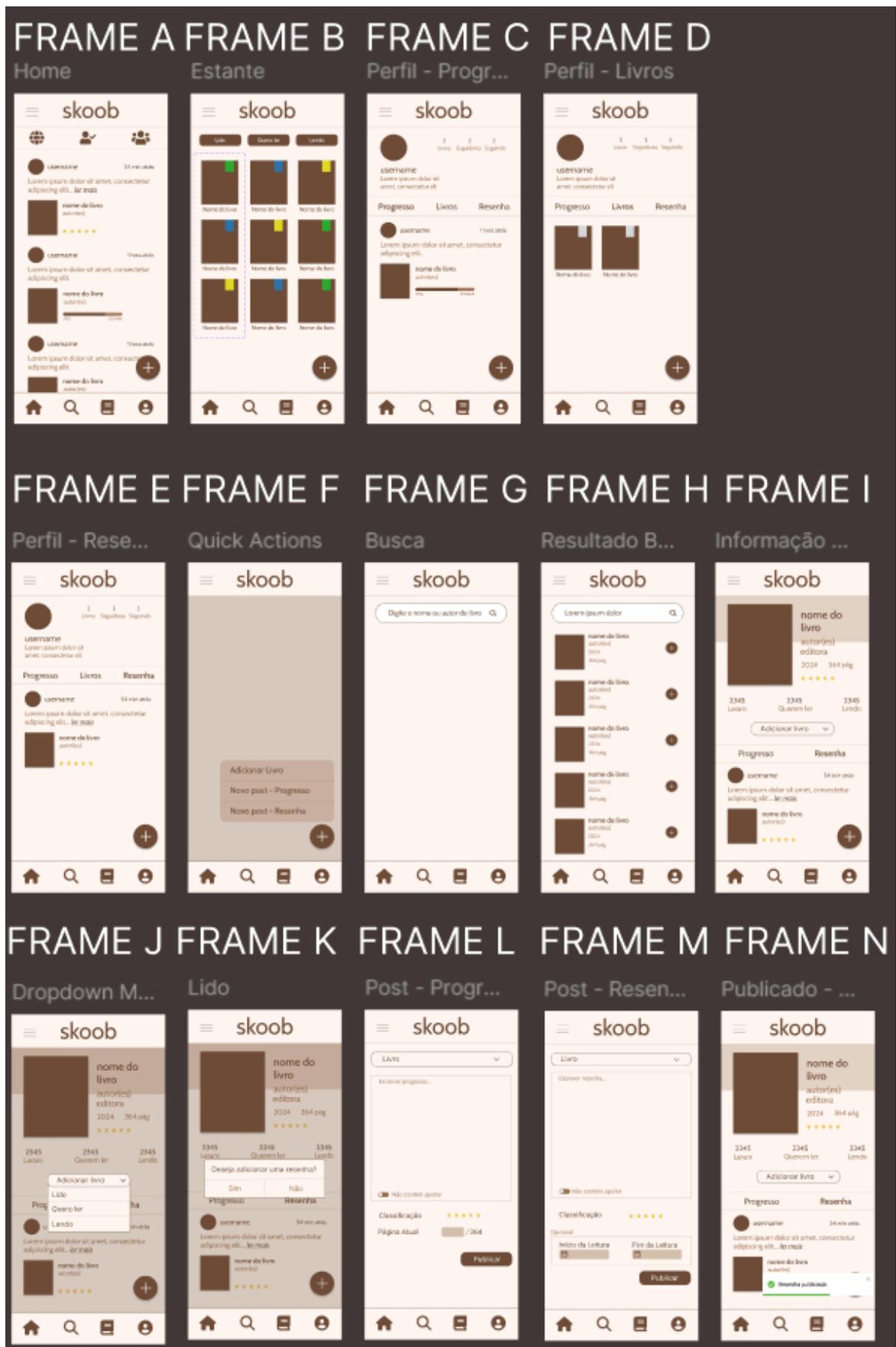


Figura 5. Wireframes da primeira versão do design das interfaces do sistema

### 4.3. API do Skoob

O Skoob possui uma API, porém não o disponibiliza para o público, dessa forma, foi necessário procurar em sites de repositório de código (Como o GitHub) APIs que pudessem proporcionar as informações necessárias (como dados de um livro). Durante a pesquisa, foi encontrada uma API, ainda em desenvolvimento, que tem o objetivo de facilitar a criação de aplicativos que utilizam dados do Skoob. Ela combina informações da API do Skoob com técnicas de *web scraping* no site da Amazon para obter detalhes adicionais sobre os livros, que foi utilizado para que o protótipo tenha informações de livros reais, assim, não necessitando a criação manual de livros no aplicativo.

### 4.4. Codificação do protótipo

Depois do projeto dos *wireframes*, foi iniciada a codificação do *WebApp*, convertendo o protótipo (Figura 5) de um esboço para o site através do uso de JavaScript, HTML5 e CSS3, todos integrados ao *framework* Next.js. Foi codificado apenas a parte em que se refere ao *front-end*, pois o projeto consumiu uma API externa para usar dados dos usuários do Skoob e livros cadastrados no mesmo. A Figura 6 apresenta a organização das pastas que organiza o código criado.



Figura 6. Organização de pastas do projeto com foco na pasta Services

Para melhor organização do projeto, o mesmo foi dividido em camadas (componentes, páginas e serviços), sendo a camada de serviços responsável pela conexão com a API. No arquivo `api.js` (Figura 7), foi utilizada a biblioteca `Axios`, que simplifica a realização de requisições HTTP (linha 1). Na linha 3, foi utilizado o `“axios.create()”`, função que cria uma instância do `Axios` com configurações customizadas, assim as requisições feitas com esta instância vão ter as mesmas configurações básicas, como o `baseURL` e os `header`. Uma das configurações do `Axios` é a `“baseURL: 'http://localhost:3000/api'”`, nela definimos a URL que iremos usar para fazer as requisições. A outra configuração é `“Content-Type: 'application/json’”`, aqui o cabeçalho `Content-Type` é configurado como `application/json`, o que indica que o corpo das requisições e respostas será no formato JSON.

O arquivo `bookService.js`, na pasta `services`, realiza algumas chamadas ao `endpoint` da API (Figura 8). Por exemplo, para buscar um livro pelo seu ID, é chamada a função definida na linha 3. Essa linha define-se uma função assíncrona chamada `getBookById`, que recebe um parâmetro de entrada. Na linha 5, é realizada a requisição de um `endpoint` da API do Skoob, e a palavra reservada `“await”` é usada para indicar que deve-se esperar a resposta da requisição HTTP antes de executar as instruções seguintes da função. O trecho `“api.get(/books/${bookId})”` realiza uma requisição GET para o `endpoint`

“/books/{bookId}” da API, onde “{bookId}” é substituído pelo valor do parâmetro fornecido para a função quando chamada. Após a requisição ser concluída, se sucesso, a função retorna os dados do livro (linha 6). Em caso de erro durante a requisição (se a API estiver indisponível, o ID não for encontrado, ou houver qualquer outro problema), o código dentro do *catch* será executado, exibindo no *console* uma mensagem de erro (linha 8).

```
services > JS api.js > ...
1  import axios from 'axios';
2
3  const api = axios.create({
4    baseUrl: 'http://localhost:3000/api',
5    headers: {
6      'Content-Type': 'application/json',
7    },
8  });
9
10 export default api;
```

Figura 7. Instância do Axios para realizar requisições a API

```
services > JS bookService.js > ...
1  import api from './api';
2
3  export const getBookById = async (bookId) => {
4    try {
5      const response = await api.get(`/books/${bookId}`);
6      return response.data;
7    } catch (error) {
8      console.error('Não foi possível carregar os dados do livro', error);
9      throw error;
10   }
11 };
12
```

Figura 8. Chamada ao *endpoint* que retorna informações de um livro pelo ID

Com as funções codificadas, foi iniciada a implementação dos *wireframes*.

#### 4.5. Iteração 1: codificando a tela inicial, o perfil pessoal e a estante de livros

Na primeira iteração de codificação do protótipo foram criados os *frames* relacionados às abas principais da plataforma (com exceção à de busca, que foi implementada na iteração 2). Com isso, foram criados os frames A, B, C, D e E da Figura 5. Os *posts* de progresso e resenha são componentes que foram reutilizados várias vezes durante o projeto, organizados na pasta Post e Resenha, de modo que fossem usados nas outras páginas. Também foram criados os componentes *Bookshelf*, *Footer*, *Livro*, *Navbar*, *Perfil*, *QuickActions* e *SearchBar* (organização da pasta de código é exibida na Figura 9). Cada componente é necessário dentro do contexto de sua página. Por exemplo: não há como realizar pesquisas na página de busca sem o componente *SearchBar*.

Exemplificando a criação desses componentes, tem-se o componente *post* de progresso (Figura 10a), que é utilizado para dois fins: comentar algo durante a leitura do livro ou apenas marcar a página em que parou. O objetivo do componente é mostrar, em formato de "cartão", detalhes como:

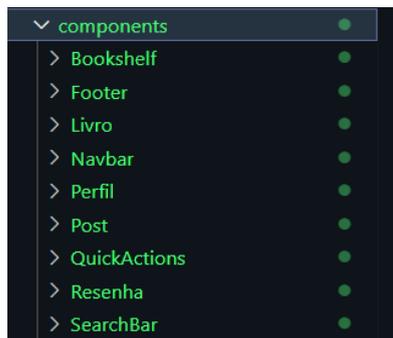


Figura 9. Componentes criados e organizados dentro da pasta *components*



Figura 10. Informações sobre um livro (a) visualizado pelo usuário e (b) dados emitidos pelo *back-end*.

- O nome do usuário que fez a publicação e o tempo decorrido desde a postagem;
- Uma descrição ou comentário sobre o livro, sendo essa opcional;
- A capa do livro, título, autor(es) e o progresso de leitura representado por uma barra visual e informações numéricas, como a porcentagem concluída e o número de páginas lidas em relação ao total.

Há informações para cada post de um livro específico. Dessa forma, a Figura 10b exemplifica os dados em formato JSON. Para o componente de *post* de progresso, os dados são definidos dentro do próprio código, pois a API não disponibiliza *endpoint* para essa funcionalidade. Já o *card* de resenha consome os dados da API, sua interface pode ser visualizada na Figura 10a e parte de seu código na Figura 10b.

Os *cards* podem ser visualizados na Figura 5, *frames A e C*. Cada componente é estruturado e estilizado com uma combinação de HTML e CSS, exposta na Figura 11, para criar uma apresentação atrativa e funcional. Além disso, o componente aproveita recursos do Next.js para otimizar a exibição de imagens. Esse código importa os estilos CSS em forma de módulos, por isso cada atributo que é estilizado possui o formato `{styles.nomeDaClasseCSS}`. Já para a parte de informações, é usado o *map* para renderização dinâmica, que itera sobre o *array* “books” (onde as informações estão definidas), e cada elemento da lista representa os dados de um livro. Para cada livro, o código cria um novo cartão; então se tivéssemos dois livros, ou seja, dois elementos na lista, seriam criados dois cartões com o mesmo estilo, porém informações diferentes.

```

const BookResenha: React.FC = () => {
  const [books, setBooks] = useState<Book[]>([]);
  const username = 'lelebr12'; // a api nao traz essa informação
  const [expanded, setExpanded] = useState<boolean[]>([]);

  useEffect(() => {
    const fetchBookshelf = async () => {
      try {
        const userBooks = await getUserBookshelf(10706889);
        setBooks(userBooks);
        setExpanded(new Array(userBooks.length).fill(false));
      } catch (error) {
        console.error('Erro ao buscar a estante de livros do usuário:', error);
      }
    };

    fetchBookshelf();
  }, []);

  > const toggleReadMore = (index: number) => { ...
  };

  return (
    <>
      {books.map((book, index) => (
        <div key={index} className={styles.card}>
          <div className={styles.header}>
            <div className={styles.avatar}></div>
            <div className={styles.info}>
              <div className={styles.userRow}>
                <p className={styles.username}>{username}</p>
                <p className={styles.timeAgo}>
                  {book.review_date ? new Date(book.review_date).toLocaleString() : '1 hora atrás'}
                </p>
              </div>
            </div>
          </div>
        </div>
      )
    )}
  );
}

```

Figura 11. Código que cria cards de posts de resenha

Na Figura 11, no componente *BookResenha* é estabelecido dois estados: *books*, que armazena os livros - informação que vem da API, e *expanded*, que controla a exibição dos textos expandidos. O *username* é definido com um valor fixo ('lelebr12'), pois a API não fornece esta informação. A função *fetchBookshelf* realiza uma requisição para obter os dados da estante de livros do usuário, a partir de um dos *endpoints* da API, atualizando os estados *books* e *expanded*. A função *toggleReadMore* permite alternar entre a exibição completa e truncada do texto de resenha para o livro no índice fornecido, caso o texto seja muito longo, para que não ocupe muito espaço na tela. Durante a renderização, o componente mapeia os livros retornados, exibindo informações como nome do usuário, data da resenha e texto truncado ou expandido conforme o estado *expanded*.

Na Figura 12, expõe-se o código que realiza a chamada dos componentes para a página *index*. Os componentes criados anteriormente estão sendo importados na linha 1 e 2 e utilizados na linha 10 e 11. O mesmo ocorre para o outro elemento *FloatingActionButton*.

#### 4.6. Iteração 2: Botão *quick actions*, tela de busca e informações do livro

Nesta etapa foi integrado as chamadas de API para utilizar os dados reais nas páginas e componentes. Com a inserção de dados reais da API o *design* sofreu leves alterações para

conter os dados como, por exemplo, tamanhos de fonte e reposicionamento de elementos na tela. O código da Figura 13 exemplifica o uso de dados reais chamando a API.

```
pages > index.tsx > ...
1 | import BookPost from '@components/Post/bookPost'
2 | import BookResenha from '@components/Resenha/Resenha'
3 | import Page from '@components/page'
4 | import FloatingActionButton from '@components/QuickActions/quickActionsBtn'
5 | import Section from '@components/section'
6 |
7 | const index = () => (
8 |   <Page>
9 |     <Section>
10 |       <BookPost/>
11 |       <BookResenha/>
12 |       <FloatingActionButton/>
13 |     </Section>
14 |   </Page>
15 | )
16 |
17 | export default index
```

Figura 12. Chamada do componente *BookPost* na página index

```
9 | const SearchBooksPage = () => {
10 |   const [books, setBooks] = useState([]);
11 |   const [isLoading, setIsLoading] = useState(false);
12 |   const [errorMessage, setErrorMessage] = useState('');
13 |
14 |   const handleSearch = async (query) => {
15 |     if (!query.trim()) {
16 |       setBooks([]);
17 |       return;
18 |     }
19 |     setIsLoading(true);
20 |     setErrorMessage('');
21 |
22 |     try {
23 |       const results = await searchBooksByTitle(query);
24 |       console.log('Resultados da busca:', results);
25 |       setBooks(results);
26 |     } catch (error) {
27 |       console.error('Erro ao buscar livros:', error);
28 |       setErrorMessage('Não foi possível buscar livros. Tente novamente.');
```

Figura 13. Código implementado com a API para realizar buscas

A função *handleSearch* (definida na linha 14) é responsável por processar a busca de livros com base no termo fornecido pelo usuário (parâmetro *query* na linha 14). Na linha 15 verifica-se que o termo de busca não está vazio ou apenas com espaços em branco. Se estiver, ela limpa a lista de livros exibida (linha 16) e encerra a função (linha 17). Caso contrário, a função ativa o estado de carregamento (linha 19) e limpa qualquer mensagem de erro anterior (linha 20). Em seguida, tenta realizar a busca chamando a função *searchBooksByTitle*, que

consulta a API que foi definida nas Figuras 7 e 8. Se os resultados forem retornados com sucesso, eles são armazenados na variável *books* (linha 25).

Se ocorrer um erro durante a busca, a função captura a exceção, exibe uma mensagem de erro para o usuário (linha 28) e registra o problema no console para depuração (linha 27). Após a execução, independente do sucesso ou falha, o estado de carregamento é desativado (linha 30), garantindo que a interface seja atualizada corretamente. Essa abordagem garante uma experiência de busca dinâmica, responsiva e com tratamento adequado de erros.

O código exposto na Figura 13 organiza a interface da página de busca de livros. Ele exibe a barra de busca (*SearchBar*), que é um componente que aciona a função *handleSearch* ao pesquisar. Há o uso de mensagens dinâmicas para indicar erro ou carregamento, assim o usuário recebe retornos em relação ao estado do que está acontecendo. Já a lista de livros (*bookList*) mostra os resultados usando o componente *BookCard* (Apêndice II, Figura 3, que recebe detalhes do livro e uma função para navegar para a página de informações do livro caso haja clique no componente. Caso não haja resultados e a busca não esteja carregando, uma mensagem de "nenhum livro encontrado" é exibida.

## 6. Discussão dos Resultados

Com base nas entrevistas, o projeto de *rebranding* focou em conceitos fundamentais de usabilidade, *design* centrado no usuário (*User-Centered Design*) e boas práticas de UX/UI. Essa abordagem não apenas resolve o *layout* datado da versão anterior, mas também proporciona um uso mais flexível e satisfatório, reforçando a conexão do usuário com a plataforma. A seguir, é detalhada as principais mudanças implementadas, considerando heurísticas e pilares dos fundamentos de CDU e UX/UI:

- Visibilidade de status do sistema: No Skoob, ao adicionar uma resenha, um livro ou um *post* de progresso, não é emitida nenhuma mensagem ou notificação na tela de que a ação foi bem sucedida, fazendo com que o usuário tenha que sair da tela atual e entrar no perfil para ver se foi bem executado. Já na versão desse projeto, todas as ações de adição ou exclusão são notificadas ao usuário;
- Tamanho e espaçamento apropriados para o toque: No Skoob, temos um exemplo de botão que fica próximo a uma imagem clicável e ambos levam a locais diferentes dentro do *app*, o que pode causar confusão caso clique um pouco mais para cima/baixo ou caso possua uma digital larga. Já na versão que esse projeto propõe, esses dois elementos estão no extremo oposto um do outro, impossibilitando acioná-los por engano;
- Uso de texto para descrever ações, ao invés de ícones: Em uma das telas do Skoob, há um botão de “+” ao lado da imagem do livro na página de informação de um livro que, muitas vezes, passava despercebido pelos usuários. Quando era um livro já adicionado na estante, no local do botão de “+” é apresentado um marca-página, cuja inconstância causava confusão aos usuários, muito pelo fato de que o botão dos livros já adicionados não aparentava ser um botão. Já nesse projeto, esse botão foi substituído por um com a escrita “Adicionar”, que deixa claro desde o princípio que é um botão de adicionar;
- Flexibilidade e eficiência: Uma das propostas desse *rebranding*, é a inclusão do botão flutuante “*quick actions*” que está disponível nas principais telas servindo como um atalho às funcionalidades de adicionar livro, resenha ou progresso;
- Redução de cliques e navegação entre telas: As telas de progresso e resenha foram otimizadas na página de informações do livro. Na versão original do Skoob, era

necessário adicionar o livro à sua estante e depois clicar no livro novamente para poder adicionar uma resenha ou progresso (alguns entrevistados não imaginavam e ficaram cerca de 3 minutos e 30 segundos procurando essa funcionalidade). Na versão desse projeto, assim que adicionado um livro à estante, o usuário já tem a opção de criar uma resenha ou progresso para ele.

## 6.1 Resultados Gerais e Impacto

Após a construção do protótipo, os entrevistados da primeira rodada foram novamente convidados, mas dessa vez, para experimentar a nova interface. Para cada usuário foi perguntado uma nota e uma opinião, assim como tempo de realização das ações foram cronometrados, nas duas rodadas de entrevista. A fim de facilitar a visualização, foram criadas tabelas com os dados quantitativos e qualitativos obtidos no primeiro e segundo *round* de entrevistas, sendo a Tabela 1 referente à primeira rodada (versão original do Skoob) e a Tabela 2 referente a versão construída nesse projeto.

**Tabela 1. Dados obtidos na primeira rodada de entrevistas, utilizando o Skoob**

Adicionar Resenha			
Usuário	Tempo	Nota	Opinião
1	2min 4seg	3	Muito confuso
2	3min 37seg	0	Escondido
3	25seg	5	Tem que procurar bastante
4	3min 47seg	3	Escondido e no local errado
5	4min 1seg	0	Deve ser na aba de resenha
6	2min 56seg	3	Escondido mas se procurar bastante acha
média	3min 20seg	2,33333	

**Tabela 2. Dados obtidos na segunda rodada de entrevista, utilizando o protótipo**

Usuário	Tempo/s	Nota	Opinião
1	2seg	10	Bem prático esse botão direto na página inicial
2	2seg	9,5	Rápido de identificar, manter só 1 jeito
3	2seg	10	Melhorou bastante minha experiência
4	2seg	10	Praticidade do botão, agilizou muito minha navegação
5	2seg	10	Botão inicial facilitou muito o uso
6	2seg	10	Botão inicial ajudou bastante por ser um atalho
média	2seg	9,91667	

Analisando as tabelas, podemos concluir que:

- **Notas médias:** A nova versão recebeu uma nota média de 9,91, demonstrando a alta satisfação dos usuários em relação às funcionalidades propostas;
- **Tempo médio de execução:** As tarefas foram realizadas em apenas 2 segundos, mostrando que a interface está alinhada com os princípios de eficiência.

Em resumo, as melhorias propostas, como visibilidade do status, ajustes nos botões e redução de cliques, proporcionaram uma experiência mais intuitiva e ágil. No entanto, há

oportunidades para refinar ainda mais algumas funcionalidades, como explorar soluções adicionais para inclusão de usuários com necessidades específicas.

## 7. Conclusões

O objetivo deste projeto foi propor uma nova interface de usuário para o Skoob com melhor usabilidade e propor uma melhor experiência do usuário, com redução de cliques, clareza nas ações e maior eficiência nas interações. Dados de possíveis pontos de melhorias foram obtidos com entrevistas com usuários ativos, ex-usuários e novos usuários. Com base nesses dados e em conceitos de *design* aprendidos nas matérias de Interação Humano-Computador e Desenvolvimento Web, um protótipo foi desenvolvido e avaliado com o auxílio do orientador deste trabalho, validando e sugerindo mudanças nos *wireframes*, além de acompanhar o desenvolvimento de cada incremento, analisando e sugerindo modificações para os 3 incrementos que compõem a primeira versão do *rebranding* do Skoob. Ao final, foi realizada uma avaliação da proposta com voluntários.

É possível afirmar, a partir de dados obtidos com uma segunda rodada de entrevistas e testes, que o objetivo do projeto foi alcançado com êxito. Com base nessa segunda rodada de investigação a partir de entrevistas, a nova versão apresentou uma melhoria na satisfação dos usuários em relação à versão original, destacando-se pela inclusão de notificações de status, otimização de navegação e introdução do botão “Quick Actions”, além da otimização de tempo, cujo foi reduzido em 97,12%.

Entretanto, algumas funcionalidades não puderam ser implementadas devido a limitações na API atual, como a impossibilidade de visualizar *posts* de progresso diretamente na aplicação. Como solução, sugere-se a investigação de outras APIs que permitam o registro e acesso completo a essas informações. A API utilizada atualmente está em desenvolvimento, o que também limita sua integração total com algumas funcionalidades propostas. Esses pontos são oportunidades claras para melhorias futuras, favorecendo ainda mais uma melhor experiência do usuário.

Neste projeto, foi possível aprender a desenvolver um sistema voltado para *web* com foco em usabilidade e melhorias na experiência do usuário, utilizando princípios fundamentais de *design* centrado no usuário e metodologias ágeis. Por meio de entregas incrementais, o processo foi estruturado com o aluno atuando como desenvolvedor e o professor orientador desempenhando o papel de cliente, validando e sugerindo ajustes a cada iteração. Foram aplicados e integrados conhecimentos adquiridos nas disciplinas de Interação Humano-Computador, Desenvolvimento Web, Metodologias Ágeis, Arquitetura de Software. Essa abordagem permitiu criar um protótipo funcional que alia aprendizado técnico e boas práticas de UX/UI, culminando em uma solução mais intuitiva, eficiente e alinhada às necessidades dos usuários. Após finalizar, foi disponibilizado o código-fonte do projeto no seguinte link: <https://github.com/nychavs/skoob-pwa.git>.

## Referências

DUNCAN, R. Brands and branding: Why re-brand? Part 2, n.d. Em: GOI, C.; GOI, M. Review on Models and Reasons of Rebranding. IACSIT Press, Singapore. 2011 International Conference on Social Science and Humanity. IPEDR, Vol. 5. pp 445-449. 2011.

- HUME, D. Foreword. In: PROGRESSIVE Web Apps. Simon and Schuster: Manning Publications, 2017. Disponível em: <https://books.google.com.br/books?id=ozozEAAAQB AJ&printsec=frontcover&hl=pt-BR#v=onepage&q&f=false>. Acesso em: 2 jun. 2024.
- MASSÉ, M. Introduction. In: REST Api design rulebook: Designing Consistent RESTful Web Service Interfaces. [S. l.]: O'Reilly Media, Inc., 2011. Disponível em: <https://books.google.com.br/books?id=eABpzyTcJNIC&printsec=frontcover#v=onepage &q&f=false>. Acesso em: 1 jun. 2024.
- MUZELLEC, L; DOOGAN, K; LAMBKIN, M; “Corporate Rebranding – An Exploratory Review,” Irish Marketing Review, vol. 16, no. 2, 2003, pp. 31-40.
- NIELSEN, J. Usability 101: Introduction to Usability. UX Basics: Study Guide, Nielsen Norman Group, 2012. Disponível em: <https://www.nngroup.com/articles/usability-101-in-troduction-to-usability/>. Acesso em: 5 maio 2024.
- NORMAN, D.; NIELSEN, J. The Definition of User Experience (UX). UX Basics: Study Guide, Nielsen Norman Group, 1998. Disponível em: <https://www.nngroup.com/articles/definition-user-experience/>. Acesso em: 5 maio 2024.
- PATRICIA, H. Introduction. In: WHAT Is User Interface Design?. [S. l.]: Rosen Publishing Group, 2017. Disponível em: [https://www.google.com.br/books/edition/What\\_Is\\_U ser\\_Interface\\_Design/3EBgDwAAQBAJ?hl=pt-BR&gbpv=1&dq=user+interface&printsec=frontcover](https://www.google.com.br/books/edition/What_Is_U ser_Interface_Design/3EBgDwAAQBAJ?hl=pt-BR&gbpv=1&dq=user+interface&printsec=frontcover). Acesso em: 2 jun. 2024.
- SILVINO, A.; ABRAHÃO, J. Navegabilidade e inclusão digital: usabilidade e competência. Ciência e Tecnologia, RAE electron, 2003. Disponível em: <https://doi.org/10.1590/S1676-56482003000200002>. Acesso em: 1 jun. 2024.
- SKOOB API. <https://github.com/fsdearruda/skoob-api>, 2022. Disponível em: <https://github.com/fsdearruda/skoob-api>. Acesso em: 2 jan. 2024.
- SOUZA, C.; SAVI, Rafael. DESIGN CENTRADO NO USUÁRIO E O PROJETO DE SOLUÇÕES EDUCACIONAIS. Especial Design, E-Tech: Tecnologias para Competitividade Industrial, Florianópolis, jan. 2015. Disponível em: <https://etech.sc.senai.br/revista-cientifica/article/download/615/386>. Acesso em: 1 jun. 2024.
- TEIXEIRA, F. Introdução e boas práticas em UX Design. 1 ed. São Paulo: Casa do Código, 2014
- UNGER, R; CHANDLER, C. A Project Guide to UX Design: For user experience designers in the field or in the making, 2º Edição, New Riders, 2012.
- VOLPATO, E. Pesquisa com usuários: como escolher a técnica certa?. UX Collective, [s. l.], 14 set. 2014. Disponível em: <https://brasil.uxdesign.cc/pesquisa-com-usuarios-como-escolher-a-tecnica-certa-bdd09ee0f302>. Acesso em: 11 jul. 2024.

## Apêndice I - Diagramas de Fluxo da Interação do Usuário no Skoob



Figura 1. Fluxo ideal para funcionalidades do Skoob

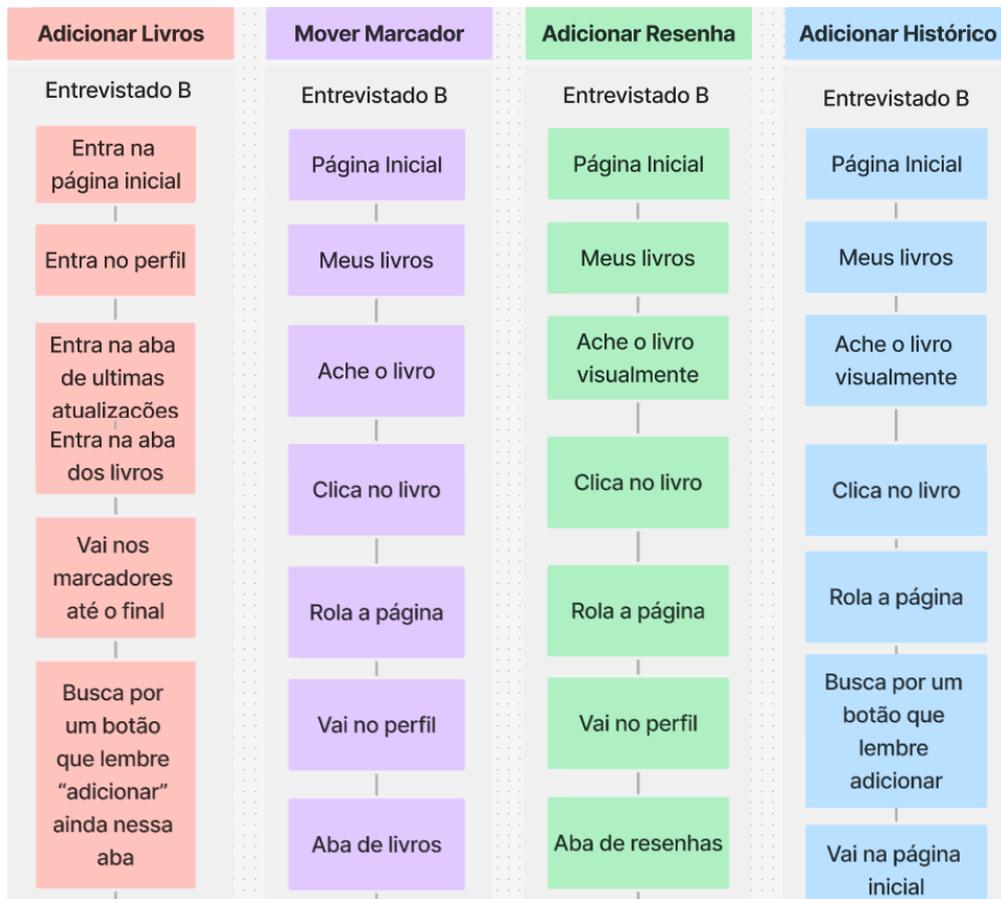
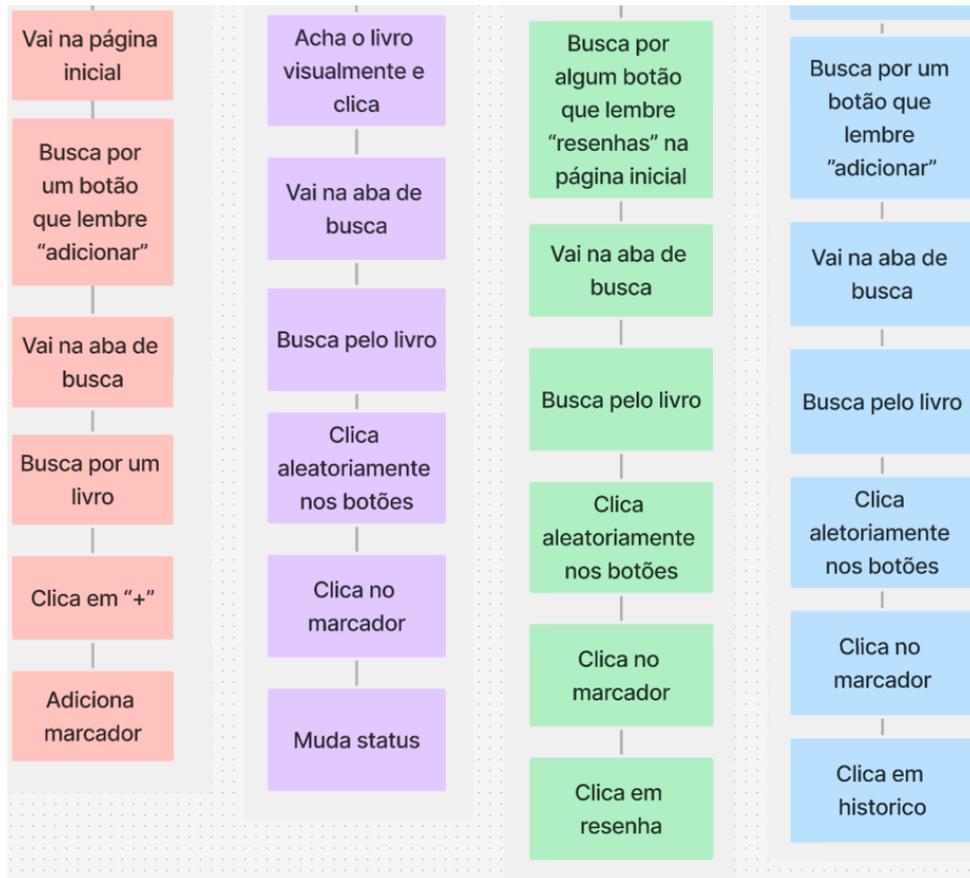


Figura 2. Parte 1- Fluxo atual percorrido pela maioria dos entrevistados

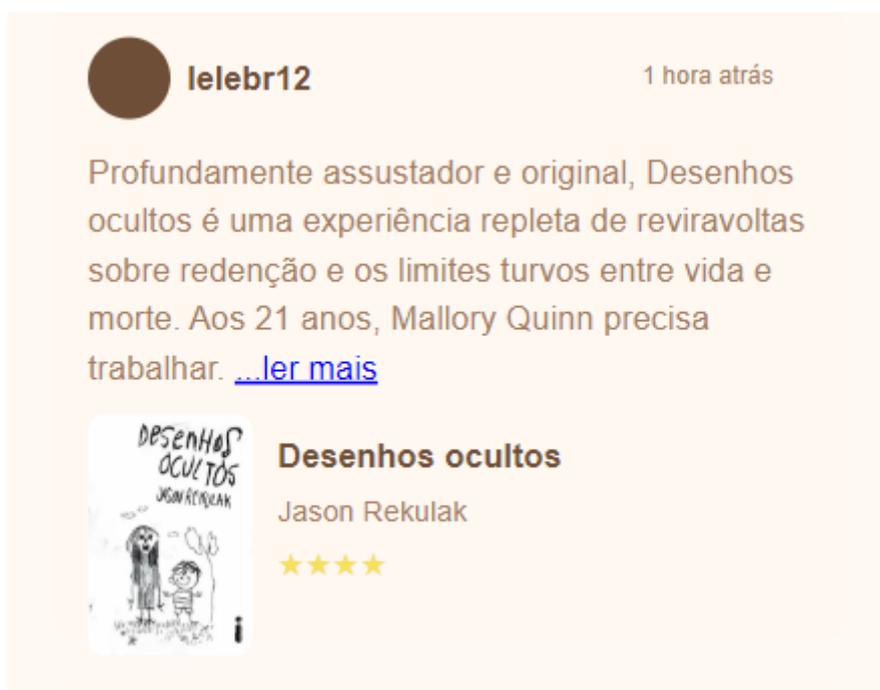


**Figura 2. Parte 2- Fluxo atual percorrido pela maioria dos entrevistados**



**Figura 3. Fluxo proposto pelo rebranding**

## Apêndice II - Componente de Resenha, Componente de Progresso e Componente de *Card* de Livro ao fazer buscas



**lelebr12** 1 hora atrás

Profundamente assustador e original, *Desenhos ocultos* é uma experiência repleta de reviravoltas sobre redenção e os limites turvos entre vida e morte. Aos 21 anos, Mallory Quinn precisa trabalhar. [...ler mais](#)

 **Desenhos ocultos**  
Jason Rekulak  
★★★★

Figura 1. Componente de *Post* de Resenha



**nychavs** 1 hora atrás

Eu não acredito que a Édra fez isso!

 **Coisas óbvias sobre o amor**  
Elayne Baeta

55% 355/644

Figura 2. Componente de Progresso de Leitura

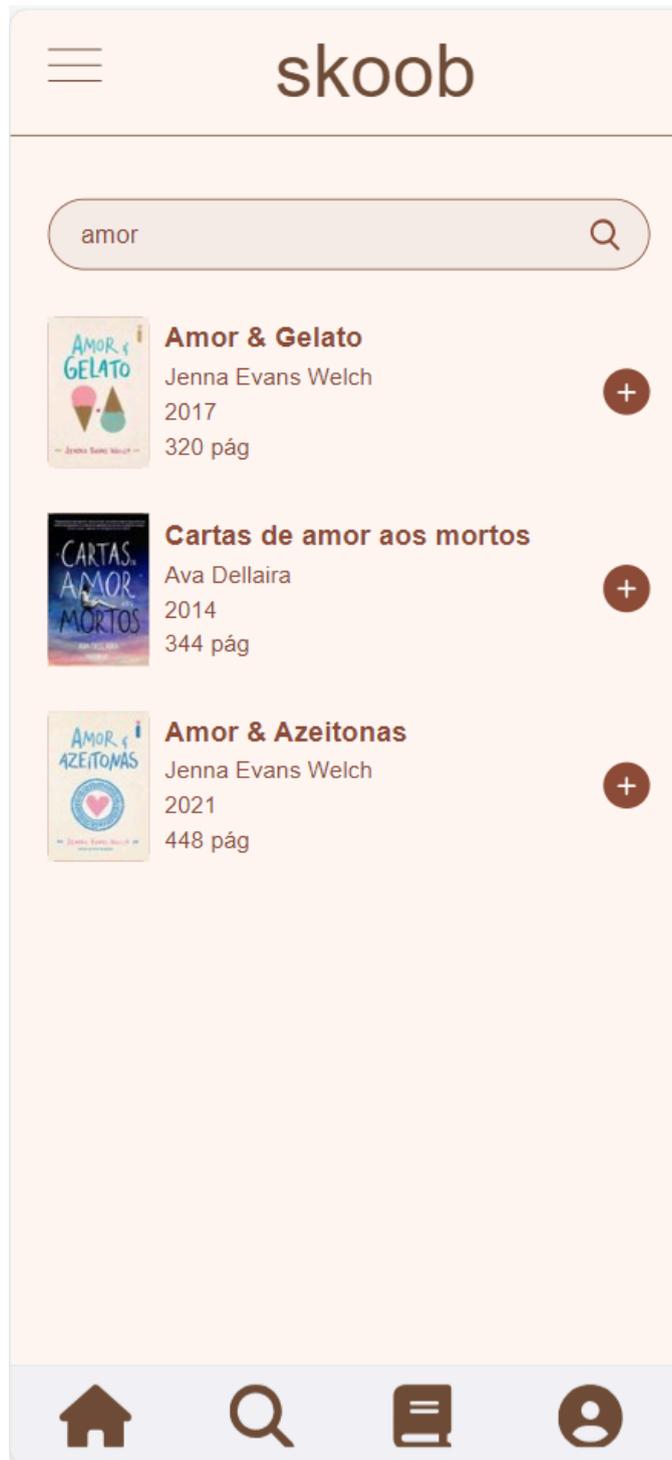


Figura 3. Componente BookCard ao realizar buscas

## Apêndice III - Lista de perguntas feitas aos usuários na entrevista

Usuário Habitado com o Skoob
Há quanto tempo você usa o skoob?
Qual é sua motivação para usá-lo?
Ele atende suas necessidades como leitor?
Você entra no aplicativo para visualizar post de outras pessoas?
Você se recorda de algum momento que teve dificuldade durante o uso do aplicativo?
E sobre a aba de mensagens? Você já utilizou? O que acha dela?
Você já chegou a utilizar todos os marcadores?
Perto dessa aba de resenha, há uma resenha chamada "vídeos" você já utilizou ela?
Você entende a proposta? (aba videos)
Você acha que seria interessante ter essas funcionalidades no SKOOB versão web? Por quê?
Você se recorda de algum momento que teve dificuldade durante o uso do aplicativo?
Como você descreveria sua experiência geral ao usar o skoob?
Há algum outro ponto que não foi mencionado por você anteriormente e você gostaria de me falar?
Você pode citar algumas funcionalidades que acha que não são tao utilizadas?
Se você fosse indicar pra alguma pessoa, você indicaria sem precisar explicar ou teria que fazer um mini tour pra pessoa entender as funcionalidades?
Você gostaria que tivesse alguma aba de ajuda?
*as perguntas dos novos usuários também foram feitas aos usuários habitados*

**Figura 1. Lista das perguntas feitas aos usuários que usam ou já usaram o SKOOB durante as entrevistas**

Novo usuário
Vamos começar adicionando um livro. Por favor, faça o percurso que você acha que é para essa funcionalidade . Você também pode desistir se cansar de procurar.
Como você avalia esse percurso? descreva em palavras
E de 0 a 10, como você avalia?
Ok, agora vamos adicionar um feedback de leitura, olha na sua página inicial, vê que tem alguns posts? tente fazer isso para um livro cuja leitura está em andamento
Descreva e avalia de 0 a 10 sua experiência com essa funcionalidade
Agora quero que você muda o status de "lendo" para "lido"
Descreva e avalia de 0 a 10 sua experiência com essa funcionalidade
Agora vamos publicar uma resenha, que é quando o livro já foi lido, tente fazer isso
Descreva e avalia de 0 a 10 sua experiência com essa funcionalidade
Olhando esses marcadores, você acha que todos eles teriam utilidade pra você?
Agora vamos na aba de mensagens, o que você acha da divisória disso?
Outra funcionalidade é de video. Você acha que isso seria util em algum momento? Você acha que é muito utilizado pelos usuários?
Como você descreveria sua experiência geral ao usar o skoob?
Qual foi o momento que voce mais teve dificuldade no aplicativo?
Você pode citar algumas funcionalidades que acha que não são tao utilizadas?
Se você fosse indicar pra alguma pessoa, você indicaria sem precisar explicar ou teria que fazer um mini tour pra pessoa entender as funcionalidades?
Você gostaria que tivesse alguma aba de ajuda?

**Figura 2. Lista das perguntas feitas aos usuários cujo primeiro uso foi durante as entrevistas**

## Apêndice IV - Resultados obtidos nas entrevistas

Adicionar Progresso			
Usuário	Tempo	Nota	Opinião
1	1min 5 seg	3	Péssimo, não consegui
2	4min 22seg	0	O usuário não vai achar
3	27seg	0	Tem que procurar bastante
4	3min	2	Acha que está no local errado
5	3min 10seg	2	Deveria ser na aba de resenhas
6	2min 58seg	0	Deveria ser mais destacado no aplicativo
média	2min 50seg	1,16667	

**Figura 15. Dados da primeira rodada de entrevistas, utilizando o Skoob, funcionalidade “Adicionar Progresso”**

Usuário	Tempo/s	Nota	Opinião
1	2seg	10	Bem prático esse botão direto na página inicial
2	2seg	9,5	Rápido de identificar, manter só 1 jeito
3	2seg	10	O atalho economiza bastante tempo
4	2seg	10	O botão inicial facilita bastante a navegação
5	2seg	10	Botão inicial facilitou muito o uso
6	2seg	10	Botão inicial ajudou bastante por ser um atalho
média	2seg	9,91667	

**Figura 16. Dados da segunda rodada de entrevistas, utilizando o protótipo, funcionalidade “Adicionar Progresso”**

Adicionar Livro			
Usuário	Tempo	Nota	Opinião
1	30seg	3	Escondido, tem que adivinhar
2	1min 30seg	3	Funcionalidade está no local errado
3	20seg	5	OK
4	1min 15seg	2	Não concordo de estar ai
5	1min 20seg	1	Deveria ser ou na home, ou na aba de livros
6	47seg	7	OK
média	57seg	3,5	

**Figura 17. Dados da segunda rodada de entrevistas, utilizando o Skoob, funcionalidade “Adicionar Livro”**

Usuário	Tempo/s	Nota	Opinião
1	8seg	10	Bem prático esse botão direto na página inicial
2	10seg	10	Melhor do que era antes
3	10seg	10	Legal ter deixado os 2 jeitos
4	9seg	10	Praticidade do botão, agilizou muito minha navegação
5	7seg	10	Botão inicial facilitou muito o uso
6	7seg	10	Botão inicial ajudou bastante por ser um atalho
média	8,5seg	10	

**Figura 18. Dados da segunda rodada de entrevistas, utilizando o protótipo, funcionalidade “Adicionar Livro”**

# Documento Digitalizado Público

## Anexo I - artigo - TCC

**Assunto:** Anexo I - artigo - TCC  
**Assinado por:** Andre Constantino  
**Tipo do Documento:** Relatório Externo  
**Situação:** Finalizado  
**Nível de Acesso:** Público  
**Tipo do Conferência:** Documento Digital

Documento assinado eletronicamente por:

- **Andre Constantino da Silva, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 26/02/2025 00:36:15.

Este documento foi armazenado no SUAP em 26/02/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

**Código Verificador:** 1951281

**Código de Autenticação:** b5d492b2eb

