

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

BRUNA PINHEIRO FALHEIRO

EDU: PLATAFORMA E-LEARNING DE ESTRUTURA DE DADOS
UMA ABORDAGEM INTERATIVA PARA APRENDIZADO ASSÍNCRONO

RIO DE JANEIRO
2025

BRUNA PINHEIRO FALHEIRO

EDU: PLATAFORMA E-LEARNING DE ESTRUTURA DE DADOS
UMA ABORDAGEM INTERATIVA PARA APRENDIZADO ASSÍNCRONO

Trabalho de conclusão de curso de graduação
apresentado ao Instituto de Computação da
Universidade Federal do Rio de Janeiro como
parte dos requisitos para obtenção do grau de
Bacharel em Ciência da Computação.

Orientador: Adriana Santarosa Vivacqua

RIO DE JANEIRO

2025

CIP - Catalogação na Publicação

F188e Falheiro, Bruna Pinheiro
EDU: PLATAFORMA E-LEARNING DE ESTRUTURA DE
DADOS. UMA ABORDAGEM INTERATIVA PARA APRENDIZADO
ASSÍNCRONO / Bruna Pinheiro Falheiro. -- Rio de
Janeiro, 2025.
96 f.

Orientadora: Adriana Santarosa Vivacqua.
Trabalho de conclusão de curso (graduação) -
Universidade Federal do Rio de Janeiro, Instituto
de Computação, Bacharel em Ciência da Computação,
2025.

1. Desenvolvimento de Software. 2. Estrutura de
Dados. 3. Usabilidade. 4. Design. 5. Ciência da
computação. I. Vivacqua, Adriana Santarosa, orient.
II. Título.


BRUNA PINHEIRO FALHEIRO

EDU: PLATAFORMA E-LEARNING DE ESTRUTURA DE DADOS
UMA ABORDAGEM INTERATIVA PARA APRENDIZADO ASSÍNCRONO


Trabalho de conclusão de curso de graduação
apresentado ao Instituto de Computação da
Universidade Federal do Rio de Janeiro como
parte dos requisitos para obtenção do grau de
Bacharel em Ciência da Computação.

Aprovado em 11 de agosto de 2025


BANCA EXAMINADORA:

Documento assinado digitalmente
 **ADRIANA SANTAROSA VIVACQUA**
Data: 19/08/2025 21:22:19-0300
Verifique em <https://validar.iti.gov.br>

Profa. Adriana Santarosa Vivacqua
D.Sc (IC/UFRJ)

Documento assinado digitalmente
 **SILVANA ROSSETTO**
Data: 20/08/2025 12:46:54-0300
Verifique em <https://validar.iti.gov.br>

Profa. Silvana Rossetto
D.Sc (IC/UFRJ)

Documento assinado digitalmente
 **JULIANA BAPTISTA DOS SANTOS FRANÇA**
Data: 20/08/2025 16:30:31-0300
Verifique em <https://validar.iti.gov.br>

Profa. Juliana Baptista dos Santos França
D.Sc (IC/UFRJ)

AGRADECIMENTOS

Minha trajetória acadêmica não começou no primeiro dia de aula na UFRJ. Para chegar até aqui, contei com pessoas que foram fundamentais para que eu seguisse firme nesse caminho.

Agradeço profundamente à minha família: aos meus pais, Rose e Mazinho, e ao meu irmão, Rennan. Em especial, agradeço aos meus pais, que estiveram ao meu lado em cada decisão, grande ou pequena, que me trouxe até este momento. Desde muito jovem, ainda antes de entrar no ensino médio, eu já sonhava em estudar na UFRJ. Esse sonho exigiu muitas escolhas difíceis e, em cada dúvida e certeza, encontrei neles o apoio emocional de que precisava. Eles confiaram no meu julgamento, respeitaram o meu caminho e me deram liberdade para trilhá-lo à minha maneira. Se hoje tenho orgulho do que construí, é porque eles se orgulharam de mim primeiro.

Sou grata também à minha tia Jacqueline, que foi a primeira pessoa a me mostrar o valor dos estudos e o poder transformador do esforço. Agradeço por todo o apoio financeiro e emocional desde o princípio.

Agradeço à minha sobrinha, Mariana, que, mesmo sem saber, se tornou um dos meus maiores motivos para seguir. Espero que, um dia, eu possa ser o mesmo exemplo que tantas pessoas foram para mim.

Essa caminhada também foi marcada por amizades incríveis que tornaram tudo mais leve. Ao meu grupo de amigos "Tropa de Choque", agradeço por cada momento de descontração, cada sessão de estudo coletivo antes das provas e cada trabalho feito em parceria. Estendo minha gratidão ao Rodrigo, Yuri, Cássio e Gustavo, nosso Primeiro Grupo. Obrigada por terem sido meu suporte e minha família dentro da faculdade.

Agradeço especialmente ao Gustavo, meu companheiro. Se eu fosse agradecer tudo o que ele fez e faz por mim, certamente haveria muito mais páginas do que este próprio trabalho. Então, vou me resumir a agradecê-lo profundamente por todas as vezes em que ele acreditou em mim, mesmo quando nem eu era capaz disso. Ter cruzado com ele nesse caminho foi uma das surpresas mais felizes dessa jornada. Sem ele, eu não teria conseguido.

Por fim, expresso minha gratidão a todos os professores que, de alguma forma, contribuíram para minha formação. Agradeço especialmente à minha orientadora, Adriana, pela dedicação, paciência e por caminhar comigo nesse último e tão importante trecho.

A todos vocês, meu mais sincero obrigado. Este trabalho carrega um pedacinho de cada um.

“Se eu vi mais longe, foi por estar sobre ombros de gigantes.”

Isaac Newton

RESUMO

O aprendizado de estruturas de dados representa um desafio significativo para estudantes de Ciência da Computação, devido à sua natureza abstrata e complexa. A dificuldade em visualizar e manipular essas estruturas impacta diretamente na compreensão e aplicação prática dos conceitos. Este trabalho apresenta o **EDu**, uma aplicação desenvolvida para centralizar ferramentas de estudo de estruturas de dados e incorporar interatividade e visualização dinâmica, com o objetivo de facilitar o aprendizado assíncrono e engajar os estudantes. O sistema foi implementado utilizando protótipos interativos e fundamentado em uma pesquisa com alunos, além de *benchmarking* de ferramentas similares. Durante a fase de testes de usabilidade, o **EDu** demonstrou eficácia em apoiar o processo de aprendizagem, recebendo feedback positivo quanto à clareza das operações, facilidade de navegação e utilidade dos recursos interativos. Como contribuição, o **EDu** se destaca por integrar simuladores, exercícios e cursos em uma única plataforma centrada no usuário. Como trabalhos futuros, propõe-se a ampliação do escopo da aplicação, com a inclusão de novos simuladores para outras estruturas de dados, além da implementação de recursos de personalização do ambiente de aprendizagem.

Palavras-chave: estruturas de dados; ensino interativo; ciência da computação; aprendizado visual;

ABSTRACT

The learning of data structures represents a significant challenge for Computer Science students due to its abstract and complex nature. The difficulty in visualizing and manipulating these structures directly impacts the understanding and practical application of the concepts. This work presents **EDu**, an application developed to centralize data structures study tools and incorporate interactivity and dynamic visualization, aiming to facilitate asynchronous learning and engage students. The system was implemented using interactive prototypes and based on research with students, in addition to benchmarking of similar tools. During the usability testing phase, **EDu** demonstrated effectiveness in supporting the learning process, receiving positive feedback regarding the clarity of operations, ease of navigation, and usefulness of interactive features. As a contribution, **EDu** stands out by integrating simulators, exercises, and courses into a single user-centered platform. As future work, it is proposed to expand the scope of the application by including new simulators for other data structures, as well as implementing customization features for the learning environment.

Keywords: data structure; interactive learning; computer science; animated learning;

LISTA DE ILUSTRAÇÕES

Figura 1 – Design Thinking ^[1]	17
Figura 2 – Respostas Pesquisa de Demanda: Conceitos mais desafiadores	23
Figura 3 – Respostas Pesquisa de Demanda: Dificuldades ao estudar	23
Figura 4 – Respostas Pesquisa de Demanda: Preferência de métodos de aprendizado	24
Figura 5 – Mapa de Empatia (Autoria Própria)	25
Figura 6 – Persona (Autoria Própria)	27
Figura 7 – Benchmarking (Autoria Própria)	30
Figura 8 – Diagrama de Casos de Uso (Autoria Própria)	35
Figura 9 – <i>Wireframe</i> e Design final - Tela do curso (Autoria Própria)	38
Figura 10 – Estrutura do projeto (Autoria Própria)	40
Figura 11 – Diagrama dos cursos (Autoria Própria)	41
Figura 12 – Diagrama dos exercícios (Autoria Própria)	41
Figura 13 – Respostas Pesquisa de Demanda: Há quanto tempo estudou estrutura de dados	53
Figura 14 – Respostas Pesquisa de Demanda: Nível de compreensão	53
Figura 15 – Respostas Pesquisa de Demanda: Conceitos mais desafiadores	54
Figura 16 – Respostas Pesquisa de Demanda: Dificuldades ao estudar	54
Figura 17 – Respostas Pesquisa de Demanda: Preferência de métodos de aprendizado	55
Figura 18 – Respostas Pesquisa de Demanda: Preferência de métodos de revisão . .	55
Figura 19 – Respostas Pesquisa de Demanda: Funcionalidades esperadas	56
Figura 20 – Wireframe: Página Inicial	66
Figura 21 – Wireframe: Página de Informações Sobre o Curso	67
Figura 22 – Wireframe: Página do Curso	68
Figura 23 – Wireframe: Página do Curso Finalizado	69
Figura 24 – Wireframe: Página de Exercícios	70
Figura 25 – Wireframe: Página do Simulador de Árvores Binárias	71
Figura 26 – Wireframe: Página do Simulador de Tabelas <i>Hash</i>	72
Figura 27 – Paleta de cores (Autoria Própria)	75
Figura 28 – Tipografia (Autoria Própria)	75
Figura 29 – Página inicial (Autoria Própria)	76
Figura 30 – Tela de informações sobre o curso (Autoria Própria)	77
Figura 31 – Tela de progressão do curso (Autoria Própria)	78
Figura 32 – Tela de exercícios (Autoria Própria)	79
Figura 33 – Simulador de Árvores Binárias - <i>Empty State</i> (Autoria Própria)	80
Figura 34 – Modal de Explicação - Simulador de Árvores Binárias (Autoria Própria)	81
Figura 35 – Simulador de Árvores Binárias (Autoria Própria)	82

Figura 36 – Simulador de Tabelas <i>Hash - Empty State</i> (Autoria Própria)	83
Figura 37 – Modal de Explicação - Simulador de Tabelas <i>Hash</i> (Autoria Própria) .	84
Figura 38 – Simulador de Tabelas <i>Hash</i> (Autoria Própria)	85
Figura 39 – Respostas Pesquisa de Usabilidade: Como você avalia a clareza da interface para realizar operações (inserção, remoção, busca, gerar árvore aleatória)	88
Figura 40 – Respostas Pesquisa de Usabilidade: Como você avalia o feedback visual fornecido após cada operação?	88
Figura 41 – Respostas Pesquisa de Usabilidade: Como você avalia a facilidade de manipular os elementos na árvore?	89
Figura 42 – Respostas Pesquisa de Usabilidade: Como você avalia a organização visual da representação gráfica da árvore?	89
Figura 43 – Respostas Pesquisa de Usabilidade: Como você avalia a clareza dos comandos para criar uma Tabela Hash?	89
Figura 44 – Respostas Pesquisa de Usabilidade: Como você avalia a clareza dos comandos para realizar operações (inserção, remoção, busca)?	90
Figura 45 – Respostas Pesquisa de Usabilidade: Como você avalia a legibilidade da tabela exibida?	90
Figura 46 – Respostas Pesquisa de Usabilidade: Como você avalia a rapidez e clareza do feedback após cada operação?	90
Figura 47 – Respostas Pesquisa de Usabilidade: Como você avalia a facilidade para localizar um módulo ou aula específica?	91
Figura 48 – Respostas Pesquisa de Usabilidade: Como você avalia a facilidade para navegar entre as aulas (avançar, voltar, pular)?	91
Figura 49 – Respostas Pesquisa de Usabilidade: Como você avalia o suporte para localizar e retomar o ponto onde parou?	91
Figura 50 – Respostas Pesquisa de Usabilidade: Como você avalia o suporte para acompanhar seu progresso ao longo do curso?	92
Figura 51 – Respostas Pesquisa de Usabilidade: Como você avalia a facilidade de uso da ferramenta de exercícios?	92
Figura 52 – Respostas Pesquisa de Usabilidade: Como você avalia o esforço necessário para concluir um exercício?	92
Figura 53 – Respostas Pesquisa de Usabilidade: Como você avalia o grau de controle que teve sobre o andamento do exercício (por exemplo, voltar, avançar)?	93
Figura 54 – Respostas Pesquisa de Usabilidade: Como você avalia a facilidade de uso da plataforma como um todo?	93
Figura 55 – Respostas Pesquisa de Usabilidade: Como você avalia a eficiência com que conseguiu realizar as tarefas desejadas?	93

Figura 56 – Respostas Pesquisa de Usabilidade: Como você avalia o esforço neces- sário para aprender a usar a plataforma?	94
--	----

SUMÁRIO

1	INTRODUÇÃO	12
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	ALGORITMOS E COMPLEXIDADE	14
2.2	PENSAMENTO COMPUTACIONAL	14
2.3	ESTRUTURA DE DADOS	14
2.4	<i>DESIGN THINKING</i>	16
2.5	<i>BENCHMARKING</i>	17
2.6	<i>MAPA DE EMPATIA</i>	17
2.7	<i>PERSONAS</i>	18
2.8	UI: INTERFACE DO USUÁRIO	18
2.9	UX: EXPERIÊNCIA DO USUÁRIO	19
2.10	<i>WIREFRAMES</i>	19
2.11	<i>ENGENHARIA DE SOFTWARE</i>	19
2.11.1	Levantamento de Requisitos	20
2.11.2	Diagrama de Casos de Uso	20
3	EMPATIA E IDEAÇÃO	22
3.1	PESQUISA DE DEMANDA	22
3.2	MAPA DE EMPATIA	25
3.3	PERSONA	27
3.4	<i>BENCHMARKING</i>	28
3.5	REQUISITOS FUNCIONAIS	31
3.6	REQUISITOS NÃO FUNCIONAIS	33
3.7	RESTRIÇÕES	34
3.8	DIAGRAMA DE CASOS DE USO	34
3.8.1	Especificação dos Casos de Uso	36
4	PROTOTIPAÇÃO	38
5	DESENVOLVIMENTO	39
5.1	<i>INFRAESTRUTURA</i>	39
5.2	<i>BACKEND</i>	40
5.3	<i>FRONTEND</i>	41
6	AVALIACÃO DO SISTEMA	43

7	CONCLUSÃO	46
7.1	RESUMO DO PROJETO	46
7.2	LIMITADORES DURANTE A EXECUÇÃO	46
7.3	TRABALHOS FUTUROS	47
	 REFERÊNCIAS	 49
	 APÊNDICE A – FORMULÁRIO DE PESQUISA DE DEMANDA	 51
	 APÊNDICE B – RESULTADOS DA PESQUISA DE DEMANDA	 53
	 APÊNDICE C – ESPECIFICAÇÃO DOS CASOS DE USO	 57
	 APÊNDICE D – WIREFRAMES	 66
	 APÊNDICE E – JSON SCHEMAS	 73
	 APÊNDICE F – DESIGN FINAL E TELAS DESENVOLVIDAS .	 75
	 APÊNDICE G – FORMULÁRIO DE PESQUISA DE USABILI- DADE	 86
	 APÊNDICE H – RESULTADOS DA PESQUISA DE USABILIDADE	 88

1 INTRODUÇÃO

Estruturas de Dados constituem um alicerce fundamental na Ciência da Computação, sendo indispensáveis para a organização e o processamento eficiente de informações em diferentes contextos tecnológicos (CORMEN et al., 2001). Essas estruturas permeiam soluções que vão desde softwares de uso cotidiano até aplicações complexas, como o cálculo de rotas em sistemas de navegação GPS, o gerenciamento de interações em redes sociais e a geração de sugestões em mecanismos de busca. Seu domínio é crucial para programadores de todos os níveis, uma vez que representa a base para o desenvolvimento de sistemas otimizados e algoritmos mais eficientes, garantindo uma manipulação de dados mais estruturada e precisa.

No entanto, o aprendizado dessas estruturas pode representar um desafio significativo para muitos estudantes, sobretudo em virtude de sua complexidade e abstração conceitual. Para que os conceitos sejam plenamente assimilados, torna-se indispensável não apenas a compreensão teórica, mas também a aplicação prática, permitindo que o estudante transite entre a abstração dos modelos e a resolução de problemas reais. Essa necessidade de conciliar teoria e prática revela-se um obstáculo frequente no processo de aprendizagem, exigindo metodologias de ensino que facilitem essa transição de maneira fluida e integrada (ROBINS; ROUNTREE; ROUNTREE, 2003).

No contexto deste estudo, foi realizada uma pesquisa com alunos do curso de Ciência da Computação da UFRJ, visando analisar os padrões de estudo adotados durante o aprendizado de Estruturas de Dados. Os resultados revelaram uma tendência significativa no uso de múltiplos recursos de aprendizagem, incluindo ferramentas online, livros e materiais didáticos diversos. Adicionalmente, identificou-se que uma parcela expressiva dos estudantes recorre a plataformas de aprendizado com animações interativas para compreender tópicos mais complexos da disciplina, evidenciando a busca por métodos visuais e dinâmicos que facilitem a assimilação dos conceitos abstratos.

A partir de um *benchmarking* acadêmico realizado, identificou-se também que grande parte das plataformas educacionais de Estrutura de Dados analisadas apresenta fragilização na interface, não sendo sempre intuitivas ou esteticamente agradáveis. Além disso, foi constatada uma ausência de integração entre conteúdos teóricos e práticos, hipótese que pode explicar o cenário evidenciado pela pesquisa, em que os estudantes recorrem a múltiplos materiais descentralizados. Essa dispersão de recursos, além de comprometer a linearidade do estudo, fragiliza a consolidação do conhecimento, dificultando a criação de conexões significativas entre os conceitos abordados e promovendo um aprendizado fragmentado e menos eficaz.

Diante desse contexto, surgiu a seguinte questão: **como consolidar as ferramentas e métodos de ensino de Estruturas de Dados em um ambiente centralizado**

que promova um aprendizado mais eficiente e integrado? Para responder a essa demanda, foi desenvolvida a aplicação EDu, projetada para unificar os principais recursos utilizados pelos estudantes em uma interface interativa e intuitiva. A plataforma visa otimizar o processo de aprendizagem ao combinar visualização de algoritmos, exercícios práticos e integração teórico-prática, permitindo uma compreensão mais profunda dos conceitos explorados. Foi utilizada a metodologia de *Design Thinking* para a realização do projeto, contando com etapas de pesquisa de demanda, mapa de empatia, personas e análise de requisitos. Por fim, foi realizada uma pesquisa de usabilidade, onde os resultados deste trabalho foram considerados satisfatórios e positivos para os alunos.

Nas seções seguintes, serão explorados os detalhes do planejamento e desenvolvimento do EDu, destacando as ferramentas utilizadas e os conceitos teóricos que fundamentaram a implementação da solução.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção abordará a base conceitual que fundamenta a implementação prática do EDu, explorando os principais conceitos e teorias que sustentam a construção do sistema. Essa exploração teórica proporcionará a compreensão necessária para contextualizar as estratégias e decisões tomadas ao longo do ciclo de vida do projeto, fundamentando e justificando as abordagens adotadas.

2.1 ALGORITMOS E COMPLEXIDADE

Um algoritmo é qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como entrada e produz algum valor ou conjunto de valores como saída (CORMEN et al., 2001).

A complexidade de um algoritmo determina o número de operações necessárias para sua execução em função do tamanho da entrada. Para analisar essa eficiência, utiliza-se a Notação $O(O)$, que representa um limite superior assintótico, simplificando expressões ao desprezar constantes e termos de menor ordem. Essa definição permite identificar algoritmos ótimos, que possuem a menor complexidade possível para um determinado problema. A análise de complexidade é essencial para comparar algoritmos e escolher soluções mais eficientes, especialmente em cenários com grandes volumes de dados (SZWARCFITER; MARKEZON, 1994).

2.2 PENSAMENTO COMPUTACIONAL

O pensamento computacional é compreendido como a análise e resolução de problemas por meio da construção de algoritmos (MARTINS; ELOY, 2019). Esse conceito, originalmente difundido por Jeannette M. Wing, tornou-se central na educação em ciência da computação e, mais recentemente, expandiu-se para outras áreas do conhecimento (WING, 2006).

Segundo (WING, 2006), pensar computacionalmente não se resume ao ato de programar, mas abrange uma série de habilidades cognitivas, como decomposição, reconhecimento de padrões, abstração e elaboração de algoritmos. Essas habilidades permitem que o indivíduo compreenda e modele problemas complexos de forma eficiente, utilizando princípios da computação como ferramenta de raciocínio.

2.3 ESTRUTURA DE DADOS

Uma Estrutura de Dados é um modo de armazenar e organizar dados com o objetivo de facilitar o acesso e modificações (CORMEN et al., 2001). A escolha da estrutura adequada

impacta diretamente na eficiência de algoritmos, influenciando o tempo de execução e o consumo de memória durante o processamento de informações. Dessa forma, o estudo de estruturas de dados permite aplicá-las de forma estratégica na resolução de problemas computacionais. O domínio desses conceitos é essencial para a construção de soluções computacionais mais eficientes, justificando sua centralidade na formação de profissionais em Ciência da Computação.

O referencial teórico sobre este tema constitui uma das partes fundamentais deste trabalho, uma vez que servirá de base para o desenvolvimento do conteúdo pedagógico do artefato proposto, fornecendo o suporte necessário para as explicações e exemplos apresentados. Para tanto, serão utilizados os livros *Estruturas de Dados e Seus Algoritmos* (SZWARCFITER; MARKEZON, 1994) e *Introduction to Algorithms* (CORMEN et al., 2001). Ambas as obras foram escolhidas com base na bibliografia recomendada pela disciplina de Estruturas de Dados da UFRJ. Neste trabalho, serão abordadas duas das Estruturas de Dados mais relevantes: Árvore Binárias e Tabelas *Hash*, conforme detalhado a seguir. A escolha dessas estruturas foi motivada pela sua recorrência na Pesquisa de Demanda apresentada no Capítulo 3, na qual foram apontadas como os temas de maior dificuldade de entendimento por parte dos alunos.

- **Árvores e Árvores Binárias:**

Uma Árvore é um tipo de estrutura de dados que representa informações de forma hierárquica, com um conjunto de nós conectados por arestas. Cada árvore possui um nó especial chamado raiz, a partir do qual todos os outros nós descendem. Cada nó pode ter zero ou mais filhos, e cada nó, exceto a raiz, possui exatamente um pai (CORMEN et al., 2001).

Já uma Árvore Binária é uma estrutura de dados do tipo Árvore em que cada nó pode possuir no máximo dois filhos, geralmente chamados de filho esquerdo e filho direito (CORMEN et al., 2001). Essa restrição confere à árvore binária propriedades específicas que facilitam a implementação de algoritmos especiais, como o Algoritmo de Busca Binária, que serão abordados neste trabalho.

- **Tabelas *Hash*:**

Tabelas *Hash* são estruturas de dados utilizadas para implementar dicionários, suportando operações básicas como inserção, busca e remoção de elementos de forma eficiente e rápida mesmo em grande quantidade de dados. A principal vantagem das Tabelas *Hash* é sua complexidade, que em média possui tempo constante $O(1)$ (CORMEN et al., 2001).

2.4 DESIGN THINKING

O *Design Thinking* é uma abordagem centrada no ser humano para a resolução de problemas complexos, que valoriza a empatia, a experimentação e a colaboração interdisciplinar. De acordo com (BROWN et al., 2008), trata-se de um processo não-linear e iterativo que envolve etapas como imersão, ideação e prototipagem, permitindo que soluções inovadoras sejam desenvolvidas a partir de uma compreensão profunda das necessidades dos usuários.

Essa abordagem foi utilizada durante todo o desenvolvimento deste trabalho, o que permitiu uma análise mais sensível e detalhada das dificuldades enfrentadas pelos estudantes, além da concepção de uma solução mais pertinente, alinhada às reais demandas desse público.

Para orientar esse processo, foi adotado o modelo proposto pelo Nielsen Norman Group (NN/g) (Norman Nielsen Group, 2025), uma das principais referências mundiais em pesquisa e consultoria na área de usabilidade e experiência do usuário. O modelo é estruturado em um ciclo contínuo de seis fases principais:

- ***Empathize***: busca compreender profundamente o público por meio da escuta ativa e da observação, permitindo identificar seus sentimentos, motivações e desafios.
- ***Define***: consiste em sintetizar as descobertas realizadas na etapa de *Empathize* em problemas centrais claramente definidos.
- ***Ideate***: foca na geração de soluções variadas para o problema levantado.
- ***Prototype***: as ideias geradas são transformadas em representações tangíveis e iterativas que possam ser exploradas e aprimoradas.
- ***Test***: envolve a validação dessas soluções com os usuários reais, permitindo ajustes e refinamentos com base em suas reações e sugestões.
- ***Implement***: consiste em transformar a solução prototipada e testada em um produto ou serviço real, disponibilizando-o para uso e validando seu impacto no contexto prático.

A representação visual desse processo pode ser observada na Figura 1, contendo o ciclo de desenvolvimento baseado no *Design Thinking*.

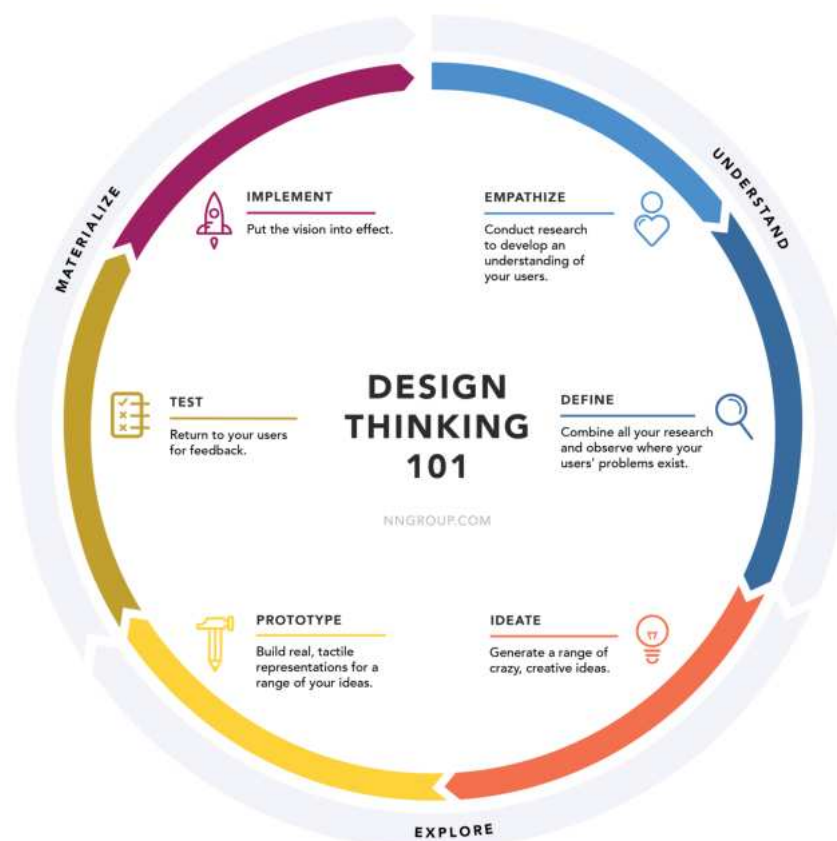


Figura 1 – Design Thinking ^[1]

2.5 BENCHMARKING

O *Benchmarking* é uma ferramenta padrão para a avaliação competitiva e comparação de sistemas ou componentes com base em características específicas, como desempenho, confiabilidade ou segurança. Esse processo permite identificar melhores práticas e otimizar o desempenho em relação a critérios de qualidade bem definidos (KISTOWSKI et al., 2015). Além disso, o *Benchmarking* possibilita uma comparação objetiva entre diferentes soluções, utilizando métricas padronizadas que garantem a reprodutibilidade dos resultados. Dessa forma, torna-se possível avaliar com precisão quais abordagens oferecem maior eficiência e usabilidade em cenários reais de aplicação.

2.6 MAPA DE EMPATIA

O Mapa de Empatia é uma ferramenta utilizada no processo de *Design Thinking* para entender de forma profunda as necessidades, emoções e comportamentos dos usuários, proporcionando uma visão mais humanizada do público-alvo (OSTERWALDER et al., 2015). De acordo com (OSTERWALDER et al., 2015), essa técnica permite reunir per-

¹ Fonte: Norman Nielsen Group. Acesso em 20 mai. 2025. Disponível em: <https://www.nngroup.com/articles/design-thinking/>

cepções sobre o que os usuários veem, pensam, sentem e fazem, facilitando a criação de soluções centradas nas reais expectativas e problemas enfrentados pelos usuários finais.

2.7 PERSONAS

Personas são representações fictícias e generalizadas dos usuários finais de um produto ou serviço, criadas com base em pesquisas e dados reais sobre os comportamentos, necessidades e motivações do público-alvo (GROUP, 2020). Essa técnica permite que equipes de desenvolvimento visualizem e compreendam melhor o perfil de seus usuários, auxiliando na tomada de decisões mais assertivas durante o processo de *design* e desenvolvimento. Ao criar personas, é possível direcionar esforços para solucionar problemas específicos e melhorar a experiência do usuário, tornando o produto mais alinhado às expectativas reais do seu público.

2.8 UI: INTERFACE DO USUÁRIO

Conhecido como UI, o conceito de interface do usuário trata-se da porção de um sistema com a qual um usuário mantém contato ao utilizá-lo, tanto ativa quanto passivamente. A interface engloba tanto *software* quanto *hardware*. Considerando a interação como um processo de comunicação, a interface pode ser vista como o sistema de comunicação utilizado neste processo.

Uma definição de interface foi proposta por Moran: "a interface de usuário deve ser entendida como sendo a parte de um sistema computacional com a qual uma pessoa entra em contato — física, perceptiva ou conceitualmente" (MORAN, 1981).

Complementando essa visão, o autor Donald Norman em seu livro "O *Design* do Dia a Dia" (NORMAN, 2002) define os seis princípios do *design* de interação:

- **Visibilidade:** Os elementos importantes da interface devem ser visíveis e claramente identificáveis;
- **Feedback:** A interface deve fornecer um retorno imediato e claro sobre as ações do usuário.
- **Restrições:** O *design* deve limitar ações incorretas e guiar os usuários para interações seguras e funcionais.
- **Mapeamento:** A interface deve estabelecer uma relação clara e lógica entre os controles e os efeitos das ações do usuário.
- **Consistência:** A interface deve manter padrões visuais e funcionais para tornar a interação previsível e confiável.

- **Affordance:** O *design* deve sugerir visualmente como o produto pode ser usado, de forma intuitiva e acessível.

Esses conceitos foram ativamente utilizados durante a concepção deste trabalho, a fim de garantir a qualidade do artefato desenvolvido, com foco na usabilidade e na eficiência da interação. Além disso, priorizar um bom *design* de interface não atende somente as necessidades funcionais, mas também promove a satisfação do usuário, fortalece a relação com o sistema e agrega valor ao produto oferecido.

2.9 UX: EXPERIÊNCIA DO USUÁRIO

O termo "Experiência do Usuário" foi cunhado por Donald Norman durante os anos 90, enquanto ainda era engenheiro na Apple, para descrever todos os aspectos possíveis das interações de uma pessoa com uma empresa e seus produtos ou serviços (NORMAN, 2002). Este termo engloba desde o primeiro contato com o produto ou serviço até sua utilização contínua, contemplando aspectos funcionais, estéticos e emocionais. Como disciplina, UX busca entender as necessidades, expectativas e comportamentos humanos, aplicando princípios de psicologia para criar interações que sejam úteis, intuitivas e prazerosas.

Este conceito foi ativamente utilizado durante a concepção deste trabalho, garantindo que a experiência do usuário seja otimizada e alinhada às necessidades do público-alvo.

2.10 WIREFRAMES

Wireframes são representações visuais de média fidelidade que delineiam a estrutura e a organização de uma interface digital, servindo como uma ferramenta essencial na fase inicial do *design* de sistemas. Eles permitem a visualização da disposição dos elementos, hierarquia da informação e funcionalidades, facilitando a comunicação entre *designers*, desenvolvedores e demais *stakeholders* do projeto (GONZATTO; COSTA, 2011). Ao focar na estrutura e funcionalidade, os *Wireframes* minimizam as influências visuais, permitindo que a equipe se concentre na experiência do usuário e na eficiência da navegação. Essa abordagem é particularmente útil para validar a arquitetura da informação e identificar possíveis melhorias antes da implementação final do *design* visual (GONZATTO; COSTA, 2011).

2.11 ENGENHARIA DE SOFTWARE

Engenharia de *Software* é uma disciplina que se preocupa com todos os aspectos da produção de *software*, desde a definição inicial dos requisitos até a manutenção contínua do sistema após sua entrega. Esta área envolve a aplicação de princípios, métodos e

ferramentas para o desenvolvimento de *software* de alta qualidade, de forma sistemática, disciplinada e mensurável (SOMMERVILLE, 2011). Para a construção deste projeto, foram escolhidos alguns dos elementos fundamentais da engenharia de *software* abaixo.

2.11.1 Levantamento de Requisitos

O Levantamento de Requisitos é uma etapa fundamental no processo de Engenharia de *Software*, onde são identificadas as necessidades e expectativas dos usuários e demais *stakeholders* em relação ao sistema a ser desenvolvido. Segundo Sommerville (SOMMERVILLE, 2011), essa fase envolve a análise e documentação dos requisitos, visando garantir que o *software* atenda aos objetivos do negócio e ofereça valor real aos seus usuários. O levantamento eficaz requer comunicação clara, técnicas adequadas de coleta de informações e o entendimento do domínio do problema, sendo crucial para evitar falhas que possam comprometer todo o projeto.

- **Requisitos Funcionais:** Definem o comportamento esperado do sistema, descrevendo as funcionalidades que o *software* deve oferecer para atender às necessidades do usuário. Esses requisitos especificam as ações que o sistema deve ser capaz de executar, como processar dados, responder a eventos e realizar cálculos. Eles são a base para o desenvolvimento do sistema, orientando o *design*, a implementação e os testes. Devem ser expressos de forma clara e detalhada para evitar ambiguidades (SOMMERVILLE, 2011).
- **Requisitos Não Funcionais:** Diferentemente dos Requisitos Funcionais, os Requisitos Não Funcionais descrevem as restrições e qualidades que o sistema deve possuir, como desempenho, usabilidade, confiabilidade e outros. Sommerville (SOMMERVILLE, 2011) destaca que esses requisitos são essenciais para garantir a satisfação dos usuários e o sucesso do software, influenciando diretamente na arquitetura e na implementação. Embora frequentemente sejam mais difíceis de especificar e medir, os Requisitos Não Funcionais asseguram que o sistema funcione adequadamente em seu contexto operacional e atenda às expectativas de qualidade.

2.11.2 Diagrama de Casos de Uso

O Diagrama de Casos de Uso é uma ferramenta visual da UML que representa as interações entre os atores (usuários ou sistemas externos) e o sistema, destacando os serviços e funcionalidades que o sistema deve prover. Segundo Sommerville (SOMMERVILLE, 2011), esse diagrama é uma forma eficaz de capturar e comunicar os requisitos funcionais, facilitando o entendimento compartilhado entre desenvolvedores e *stakeholders*. Ao mapear os Casos de Uso, é possível identificar claramente os objetivos do sistema e o

comportamento esperado em diferentes cenários, servindo como base para a modelagem e desenvolvimento do *software*.

3 EMPATIA E IDEAÇÃO

Este capítulo descreve o processo de investigação desenvolvido com o objetivo de compreender as necessidades dos estudantes no contexto do aprendizado de Estrutura de Dados, bem como os passos realizados para a definição dos requisitos do sistema proposto.

A investigação foi organizada em três etapas: **Pesquisa de Demanda**, **Mapa de Empatia** e **Levantamento de Requisitos**, cujos processos são detalhados a seguir.

3.1 PESQUISA DE DEMANDA

Com o objetivo de compreender melhor as necessidades e dificuldades enfrentadas pelos estudantes de Ciência da Computação, foi realizada uma pesquisa de demanda utilizando Formulários do Google. O questionário buscou investigar os principais obstáculos no processo de aprendizagem, os estilos de aprendizagem preferidos e os recursos mais valorizados em plataformas educacionais.

O formulário, apresentado no Apêndice A, foi divulgado entre os alunos do curso de Computação, restringindo-se àqueles que estavam cursando a disciplina de Estrutura de Dados no momento da aplicação ou que a haviam cursado em até dois períodos anteriores. Essa delimitação visou garantir que as respostas refletissem experiências recentes e pertinentes ao contexto investigado.

Durante o período de uma semana, foram coletadas 46 respostas, que podem ser acessadas no Apêndice B. A análise permitiu identificar aspectos relevantes sobre os principais obstáculos enfrentados pelos alunos, seus estilos de aprendizagem preferidos e os recursos mais valorizados em plataformas educacionais. Esses achados são apresentados a seguir:

- **Familiaridade Tecnológica:** Os resultados indicam que aproximadamente 65% dos participantes já possui contato com ferramentas de simulação e recursos digitais para o estudo de Estrutura de Dados.
- **Principais Dificuldades Enfrentadas:** Entre os conceitos considerados mais desafiadores pelos estudantes, destacam-se os Grafos, Tabelas *Hash* e as Árvore Binárias, representadas na Figura 2. Além disso, foram relatadas dificuldades expressivas na implementação de algoritmos (61,5%) e na conexão entre teoria e prática (38,8%), representadas na Figura 3. A falta de visualização clara do funcionamento das estruturas foi mencionada como um obstáculo recorrente.

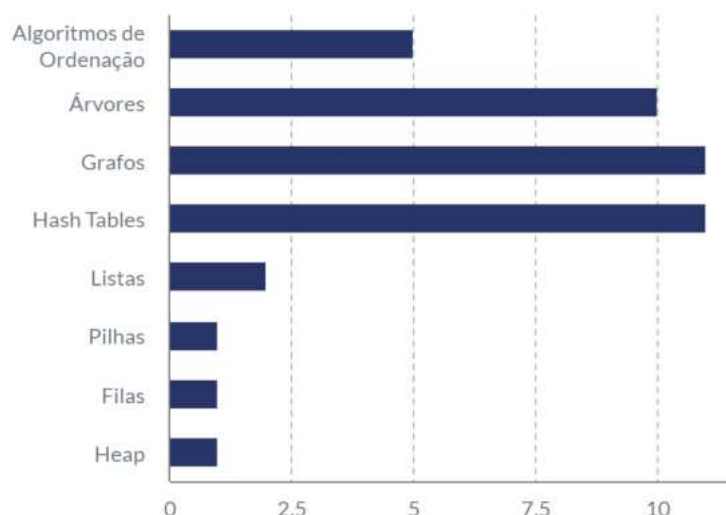


Figura 2 – Respostas Pesquisa de Demanda: Conceitos mais desafiadores

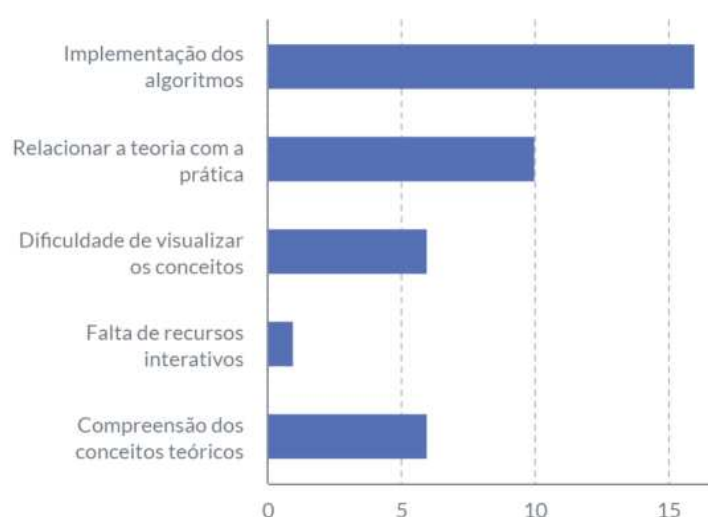


Figura 3 – Respostas Pesquisa de Demanda: Dificuldades ao estudar

- Preferências de Métodos de Estudo:** A pesquisa demonstrou uma forte preferência dos estudantes por métodos que promovam interatividade e visualização dos conceitos. Entre os recursos mais citados estão as animações interativas (65,4%), destacadas como ferramentas eficazes para entender estruturas complexas. Em seguida, foram igualmente mencionadas as videoaulas e os exercícios práticos (53,8%), ambos representados na Figura 4. Para revisão dos conteúdos, os estudantes mantiveram preferência por esses métodos mais dinâmicos, enquanto técnicas tradicionais, como resumos e provas antigas, tiveram baixa adesão.



Figura 4 – Respostas Pesquisa de Demanda: Preferência de métodos de aprendizado

- **Expectativas:** Os participantes destacaram a importância de funcionalidades que facilitem o entendimento prático dos conceitos estudados. Entre as principais demandas, foram identificadas:
 - **Simulação de algoritmos com animações**, permitindo a visualização passo a passo de estruturas complexas em tempo real;
 - **Explicações detalhadas** que auxiliem na compreensão de cada etapa dos algoritmos;
 - **Aplicação dos conceitos em problemas reais.**

Dessa forma, com base nas respostas coletadas e na análise das principais dificuldades enfrentadas pelos estudantes, foram estabelecidos os seguintes objetivos para o desenvolvimento da plataforma:

- Ser acessível, permitindo o uso em diversos sistemas operacionais, proporcionando aos estudantes flexibilidade para estudar utilizando o sistema operacional que desejarem.
- Oferecer materiais teóricos estruturados em formato de curso, cobrindo desde os conceitos fundamentais até tópicos avançados dos assuntos alinhados às dificuldades identificadas na pesquisa.
- Integrar simuladores interativos, permitindo aos estudantes visualizar o funcionamento dos algoritmos em tempo real, facilitando a compreensão prática dos conceitos.

- Disponibilizar exercícios, auxiliando os estudantes a aprimorar suas habilidades de forma autônoma e progressiva.
- Promover a aplicação dos conceitos em problemas reais, demonstrando o uso prático das estruturas de dados em cenários do mundo real, de modo a conectar o aprendizado teórico com a prática.
- Manter uma interface intuitiva e de fácil navegação, projetada para atender tanto estudantes iniciantes quanto aqueles em níveis mais avançados, garantindo uma experiência de uso fluida e eficaz.

3.2 MAPA DE EMPATIA



Figura 5 – Mapa de Empatia (Autoria Própria)

A fim de compreender de forma mais aprofundada o público-alvo do artefato, foi aplicado o Mapa de Empatia (Figura 5), uma metodologia essencial no contexto do *Design Thinking*, cujo objetivo é captar as necessidades, motivações e desafios enfrentados pelos usuários. Essa ferramenta estrutura a análise em quatro dimensões centrais (“O que vê”, “O que ouve”, “O que pensa e sente” e “O que fala e faz”), permitindo mapear a experiência do usuário em relação a uma situação específica. A utilização desse recurso serviu como base para a construção de uma Persona, apresentada no tópico subsequente, contribuindo para uma visão aprofundada dos aspectos psicológicos, emocionais e sociais dos estudantes.

O Mapa de Empatia foi elaborado na plataforma Figma ¹, organizado em seis seções que delineiam o perfil comportamental e perceptivo dos usuários. Na seção “O que vê”, observou-se um cenário marcado por elevado nível de abstração nos conteúdos relacionados a Estruturas de Dados, pouca conexão com aplicações práticas no processo de aprendizagem, além de materiais didáticos predominantemente teóricos, extensos e com baixa atratividade visual. No campo “O que ouve”, surgem relatos recorrentes de colegas que reforçam a percepção de alta complexidade da disciplina, contribuindo para a construção de um imaginário coletivo de dificuldade. Já na seção “O que pensa e sente”, destaca-se a sensação de frustração dos estudantes, que relatam dificuldades em transpor os conceitos teóricos para práticas concretas. Esse sentimento é frequentemente associado à percepção de que recursos interativos e elementos visuais poderiam atuar como facilitadores no entendimento dos algoritmos. Por fim, em “O que fala e faz”, é possível observar que os estudantes frequentemente verbalizam suas dificuldades, buscando apoio mútuo para superar os desafios.

A partir dessa análise, foram identificadas como principais dores dos usuários: o receio de não alcançar a proficiência necessária na disciplina de Estruturas de Dados e a escassez de materiais didáticos interativos que promovam um aprendizado mais efetivo. Por outro lado, os ganhos esperados incluem a expectativa de acessar recursos que proporcionem uma abordagem mais visual, dinâmica e aplicada, além do fortalecimento da compreensão dos algoritmos, tornando o processo de aprendizagem mais acessível e significativo.

¹ Figma é uma plataforma online de design colaborativo que facilita a criação de elementos visuais, diagramas, *wireframes* e *design* de interfaces. Mais informações podem ser encontradas em <https://figma.com>

3.3 PERSONA

A partir do Mapa de Empatia, foi elaborada a Persona "Mel", que representa as características comportamentais, sociais e psicológicas dos potenciais usuários. Mel é uma estudante de 21 anos, cursando o 4º semestre de Ciência da Computação, com conhecimento intermediário em programação e iniciando seus estudos em Estrutura de Dados. Entre os desafios enfrentados, destacam-se a dificuldade em compreender conceitos abstratos da disciplina, a falta de motivação diante de tópicos teóricos sem aplicação prática, e problemas para se concentrar em materiais extensos. Mel prefere métodos de aprendizado visual e prático para entender a disciplina. Seus objetivos incluem compreender melhor os conceitos de Estrutura de Dados para aplicá-los em projetos acadêmicos e pessoais, aprimorar seu domínio sobre algoritmos para se destacar em processos seletivos de estágio, e aprender de forma mais eficiente.



Mel
21 / Feminino

Ocupação: Estudante de Ciência da Computação, 4º semestre

Conhecimento: Intermediário em programação, iniciando em Estrutura de Dados

Desafios

- Dificuldade em compreender conceitos abstratos de Estrutura de Dados.
- Falta de motivação para estudar tópicos teóricos sem aplicações práticas.
- Problemas para manter o foco em leituras longas e materiais teóricos extensos.
- Falta de recursos didáticos que unam teoria e prática de forma dinâmica.

Comportamento

1. Estuda programação de forma independente, complementando as aulas da faculdade com cursos online e projetos pessoais.
2. Costuma aprender melhor de forma visual e prática, valorizando animações e representações gráficas para entender algoritmos.
3. Participa de grupos de estudos online, compartilhando dúvidas e soluções sobre assuntos da computação.

Objetivos:

- Compreender melhor os conceitos de Estrutura de Dados para aplicá-los em projetos acadêmicos e pessoais.
- Aumentar seu domínio sobre algoritmos para se destacar em processos seletivos de estágio.
- Aprender de forma mais eficiente e visual, reduzindo o tempo gasto tentando entender conceitos apenas pela teoria.

Figura 6 – Persona (Autoria Própria)

A construção da Persona (Figura 6) possibilitou uma compreensão aprofundada das

necessidades, dificuldades e expectativas dos usuários em potencial, servindo como base para orientar as decisões de projeto da plataforma. A partir desse perfil, foi possível, nas próximas seções, delinear funcionalidades que dialogam diretamente com os desafios enfrentados pelo público-alvo, garantindo maior relevância da solução proposta.

Em seguida, foi conduzido um processo de *Benchmarking* acadêmico, com uma análise de ferramentas e plataformas que operam em contextos semelhantes ao artefato proposto. Essa etapa teve como objetivo identificar boas práticas, estratégias e oportunidades que pudessem ser incorporadas ao desenvolvimento, contribuindo para o aprimoramento do projeto.

3.4 BENCHMARKING

O *Benchmarking* representa uma etapa importante no processo de ideação da solução, contribuindo diretamente para a definição das funcionalidades e da usabilidade da plataforma. A partir da análise comparativa de ferramentas já existentes, é possível identificar boas práticas, recursos positivos e lacunas que podem ser exploradas para oferecer uma proposta mais alinhada às necessidades dos estudantes.

Com base nisso, foram selecionadas plataformas que compartilham objetivos semelhantes aos do **EDu**, com foco no ensino de Estrutura de Dados, para a realização do *Benchmarking*. A seguir, as plataformas escolhidas foram:

- USFCA *Data Structure Visualizations* ²
- DS *Visualizer* ³
- Visualgo ⁴
- CSVistool ⁵

Para conduzir o *Benchmarking*, foram adotados cinco critérios de avaliação para comparar as plataformas disponíveis. Esses critérios buscaram abranger aspectos essenciais tanto do conteúdo quanto da experiência do usuário. Os critérios utilizados e suas descrições foram:

- **Quantidade de estruturas de dados:** Considera a quantidade e diversidade de Estruturas de Dados cobertas pela plataforma, como Listas, Árvores, Filas, Pilhas, Grafos, Tabelas *Hash*, entre outros. A avaliação dos critérios foram:

² USFCA - *Data Structure Visualizations*. Disponível em: <https://www.cs.usfca.edu/~galles/visualization/Algorithms>. Acesso em: 5 maio 2025.

³ DS Visualizer. Disponível em: <https://dsa-visualizer-delta.vercel.app>. Acesso em: 5 maio 2025.

⁴ VISUALGO. Disponível em: <https://visualgo.net/en>. Acesso em: 5 maio 2025.

⁵ CSVISTOOL. Disponível em: <https://csvistool.com>. Acesso em: 5 maio 2025.

- **Baixa** - até 3.
 - **Moderada** - de 4 a 10.
 - **Grande** - de 10 a 20.
 - **Muito alta** - mais de 20.
- **Formatos de conteúdo:** Avalia os diferentes tipos de recursos oferecidos, como explicações teóricas, exercícios práticos, animações interativas, quizzes, entre outros. A avaliação dos critérios foram:
 - **Pouco diversificado** - Apenas um tipo de conteúdo.
 - **Razoavelmente diversificado** - Dois tipos de conteúdo.
 - **Bem diversificado** - Três tipos de conteúdo.
 - **Extremamente diversificado** - Quatro ou mais tipos de conteúdo bem integrados.
 - **Qualidade da interface:** Examina a facilidade de navegação e a qualidade do *design* da interface. A avaliação dos critérios foram:
 - **Ruim** - Layout confuso, elementos desalinhados e pouca atenção ao *design*.
 - **Regular** - Interface funcional, mas com falhas visuais ou pouca atenção estética.
 - **Bom** - Interface bem organizada mas com pequenas falhas visuais.
 - **Excelente** - Interface visualmente agradável, coerente e altamente funcional.
 - **Usabilidade dos simuladores interativos:** Verifica a usabilidade dos simuladores interativos que permitam a visualização e manipulação dinâmica das Estruturas de Dados. A avaliação dos critérios foram:
 - **Ruim** - Difícil de usar.
 - **Ok** - Funcional, mas com problemas.
 - **Bom** - Poucos ajustes necessários.
 - **Excelente** - Não há considerações.
 - **Facilidade de uso:** Analisa o grau de complexidade de uso da plataforma. A avaliação dos critérios foram:
 - **Complexa** - Difícil, pouco intuitiva.
 - **Moderada** - Requer algum esforço para começar.
 - **Simples** - Fácil de usar após breve exploração.
 - **Muito simples** - Extremamente simples e acessível.

	Quantidade de estrutura de dados	Formato de conteúdos	Qualidade da interface	Usabilidade dos simuladores interativos	Facilidade de uso
USFCA	Muito alta - 51	Pouco diversificado - Apenas simuladores	Regular	Bom. Pontos positivos: Conta com controles de velocidade e play/pause das animações. Execução auto-explicativa com as animações. Pontos negativos: Não é possível utilizar os controles das animações em todas as animações. Alguns simuladores possuem interação limitada.	Simples
DS Visualizer	Moderada - 6	Pouco diversificado - Apenas simuladores	Excelente	Ok. Pontos positivos: Interface agradável e altamente intuitiva. Conta com uma seção de detalhes sobre a estrutura de dados gerada. Pontos negativos: Exibição de algumas estruturas de dados não funcionam bem.	Muito simples
Visualgo	Muito alta - 26	Bem diversificado - Simuladores, material de leitura e exercícios	Bom	Bom. Pontos positivos: Explicação passo a passo sobre os simuladores e sobre a execução do código. Conta com controles de velocidade e play/pause das animações. Pontos negativos: Interface um pouco confusa, necessitando de uma exploração prévia até entender o seu funcionamento.	Simples
CS Vistool	Muito alta - 44	Pouco diversificado - Apenas simuladores	Bom	Excelente: Conta com controles de velocidade e play/pause das animações.	Muito simples

Figura 7 – Benchmarking (Autoria Própria)

De modo geral, a análise das plataformas revelou pontos positivos, mas também limitações importantes. Embora algumas ofereçam diversas estruturas, apenas uma disponibiliza mais de um formato de conteúdo, o que evidencia um foco prático e pode limitar a compreensão teórica das Estruturas de Dados por parte dos alunos. Além disso, as interfaces, em sua maioria, são simples e funcionais, mas nem sempre intuitivas ou esteticamente agradáveis. A usabilidade dos simuladores é satisfatória e todas contam com controles de velocidade e ajustes das animações. Por fim, nenhuma das plataformas analisadas possui sistema de login e senha, aspecto não incluído no *benchmarking*.

Com base nos dados levantados pelo desenvolvimento da Persona e do *Benchmarking*, foi possível dar início ao Levantamento dos Requisitos do sistema. Esse processo envolveu uma análise de todas as pesquisas construídas ao longo deste trabalho até essa etapa. Os requisitos obtidos foram, então, classificados em duas categorias distintas: Requisitos Funcionais e Não Funcionais.

3.5 REQUISITOS FUNCIONAIS

Os Requisitos Funcionais representam as funcionalidades essenciais que o sistema deve oferecer para atender às necessidades dos usuários (SOMMERVILLE, 2011). Eles foram definidos com base nos resultados da pesquisa exploratória e nas dificuldades relatadas pelos estudantes no aprendizado de Estrutura de Dados. A estruturação desses requisitos foi organizada em duas categorias principais: cursos e exercícios, e simuladores. A seguir, são detalhados os requisitos funcionais definidos para cada uma dessas categorias:

Cursos e Exercícios: Esta categoria abrange funcionalidades relacionadas ao acesso, navegação e interação com os conteúdos teóricos e práticos dos cursos, garantindo que os estudantes possam explorar os tópicos de forma clara e estruturada. Na plataforma, o termo “curso” é utilizado para se referir a um conjunto de conteúdos e exercícios voltados a um único tópico específico e independente.

- **RQF1. Filtrar Cursos:** O sistema deve permitir ao usuário filtrar cursos de acordo com os seletores disponíveis (por todos os cursos, cursos em andamento e cursos concluídos).
- **RQF2. Visualizar Informações do Curso:** O sistema deve exibir detalhes do curso, como ementa, carga horária e conteúdo programático ao usuário.
- **RQF3. Iniciar Curso:** O sistema deve permitir ao usuário iniciar um curso selecionado e disponibilizar seu conteúdo.
- **RQF4. Trocar Tópico do Curso:** O sistema deve permitir ao usuário alternar para qualquer tópico desejado dentro do curso em andamento.

- **RQF5. Finalizar Curso:** O sistema deve permitir ao usuário concluir um curso, registrando o término na plataforma.
- **RQF6. Acessar Exercícios:** O sistema deve permitir ao usuário acessar os exercícios relacionados ao conteúdo do curso.
- **RQF7. Responder Exercícios:** O sistema deve permitir ao usuário responder questões e atividades propostas nos exercícios.
- **RQF8. Trocar Exercício:** O sistema deve permitir ao usuário selecionar e alternar para qualquer exercício disponível dentro da lista.
- **RQF9. Reportar Problema:** O sistema deve permitir ao usuário reportar problemas encontrados durante o uso da plataforma, sejam em cursos, simuladores ou exercícios.

Simuladores: Esta categoria foca nas operações práticas realizadas nos simuladores de Árvores Binárias e Tabelas *Hash*, permitindo aos usuários manipular essas estruturas em um ambiente interativo e visual.

- **RQF10. Acessar Simuladores:** O sistema deve permitir ao usuário acessar os simuladores disponíveis para praticar conceitos estudados.
- **RQF11. Visualizar Informações do Simulador:** O sistema deve exibir informações sobre o funcionamento e os detalhes do simulador selecionado.
- **Árvores Binárias:**
 - **RQF12. Criar Árvore Binária:** O sistema deve permitir ao usuário criar uma árvore binária no simulador.
 - **RQF13. Inserir Nó:** O sistema deve permitir ao usuário adicionar um novo nó em uma estrutura de árvore binária.
 - **RQF14. Remover Nó:** O sistema deve permitir ao usuário remover um nó existente em uma estrutura de árvore binária.
 - **RQF15. Buscar Nó:** O sistema deve permitir ao usuário realizar uma busca por um nó específico na estrutura de árvore binária.
 - **RQF16. Gerar Árvore Aleatória:** O sistema deve permitir ao usuário criar uma árvore binária com valores dos nós gerados de forma aleatória.
 - **RQF17. Limpar Árvore:** O sistema deve permitir ao usuário remover todos os nós da árvore binária, deixando-a vazia.
 - **RQF18. Ver Pseudocódigo:** O sistema deve permitir ao usuário visualizar o pseudocódigo correspondente à operação que está sendo realizada no simulador.

- **Tabelas *Hash***

- **RQF19. Criar Tabela *Hash*:** O sistema deve permitir ao usuário gerar uma Tabela *Hash* vazia.
- **RQF20. Definir Tamanho:** O sistema deve permitir ao usuário especificar o tamanho da Tabela *Hash* durante sua criação.
- **RQF21. Definir Função *Hash*:** O sistema deve permitir ao usuário definir a função hash utilizada para o mapeamento dos valores.
- **RQF22. Definir Método de Colisão:** O sistema deve permitir ao usuário definir o método de tratamento de colisão para a Tabela *Hash*.
- **RQF23. Inserir Valor:** O sistema deve permitir ao usuário inserir um valor na Tabela *Hash* criada.
- **RQF24. Buscar Valor:** O sistema deve permitir ao usuário realizar a busca de um valor na Tabela *Hash*.
- **RQF25. Limpar Tabela:** O sistema deve permitir ao usuário remover todos os valores presentes na Tabela *Hash*.

3.6 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais especificam atributos de qualidade e restrições necessárias para o bom funcionamento da plataforma (SOMMERVILLE, 2011). Eles foram definidos com base nas necessidades identificadas durante a pesquisa, visando garantir uma experiência de uso fluida e positiva para os usuários.

A seguir, são apresentados os Requisitos Não Funcionais definidos para a aplicação:

- **RQNF1. Usabilidade:** A interface deve ser intuitiva, facilitando o acesso e a interação do usuário com os cursos, exercícios e simuladores.
- **RQNF2. Desempenho:** As operações de busca, filtragem e acesso aos simuladores devem ser realizadas em tempo hábil, com resposta rápida às interações do usuário.
- **RQNF4. Manutenibilidade:** O sistema deve ser projetado de forma modular para facilitar futuras atualizações e correções.
- **RQNF5. Confiabilidade:** O sistema deve garantir que operações realizadas (como inserção de nós em uma árvore ou inserção de valores em uma Tabela *Hash*) sejam persistidas corretamente e recuperadas sem erros.
- **RQNF6. Portabilidade:** O sistema deve ser capaz de ser executado em diferentes plataformas operacionais (Windows, Linux, MacOS) sem a necessidade de modificações significativas.

3.7 RESTRIÇÕES

Durante o Levantamento de Requisitos, foram identificadas algumas restrições que orientaram as decisões de *design* e implementação do sistema. Essas limitações definem os parâmetros técnicos e operacionais que precisam ser considerados no desenvolvimento do projeto, garantindo que os objetivos sejam alcançados de maneira otimizada e dentro dos requisitos estabelecidos. As restrições do sistema estão descritas abaixo:

- **Escopo de Estruturas de Dados:** A plataforma terá como foco exclusivo Árvore Binárias e Tabelas *Hash* no desenvolvimento dos simuladores interativos e no conteúdo programático. Essa escolha foi guiada pelos resultados da pesquisa, que apontaram essas duas estruturas como umas das que mais geram dificuldades entre os estudantes. A estrutura de Grafos foi retirada do escopo se justifica pela complexidade técnica envolvida na implementação dessa estrutura, o que tornaria o projeto excessivamente amplo.
- **Ambiente *Desktop-First*:** O desenvolvimento da aplicação será voltado, prioritariamente, para o ambiente *desktop*, com suporte a múltiplas plataformas, incluindo Windows, macOS e Linux. A decisão se apoia na natureza interativa dos simuladores, que exigem uma área de tela mais ampla para exibir de forma clara as Estruturas de Dados. Ainda assim, a plataforma será projetada com responsividade, assegurando uma experiência funcional e adaptada em diferentes tamanhos de tela.
- **Ausência de Dependências Externas:** A aplicação será instalada localmente no dispositivo do usuário, sem depender de servidores *web* ou da hospedagem de serviços na nuvem para funcionar. Essa abordagem simplifica a infraestrutura técnica, elimina a necessidade de manter serviços externos ativos e permite que a ferramenta seja utilizada mesmo sem conexão com a Internet. Além de proporcionar maior autonomia ao usuário, essa decisão também reduz significativamente os custos operacionais e de manutenção, tornando o projeto mais viável.

3.8 DIAGRAMA DE CASOS DE USO

Com base nos Requisitos Funcionais e nas Restrições definidos, foi desenvolvido um Diagrama de Casos de Uso (Figura 8) para representar as principais interações entre os usuários e a aplicação. Essa ferramenta foi escolhida por sua capacidade de ilustrar, de forma clara e objetiva, os serviços oferecidos pelo sistema a partir da perspectiva do usuário. Cada requisito funcional identificado no levantamento de requisitos foi mapeado como um caso de uso específico, permitindo uma visualização estruturada das funcionalidades

esperadas e das interações previstas, garantindo que todas as funcionalidades essenciais estejam devidamente contempladas no modelo⁶.

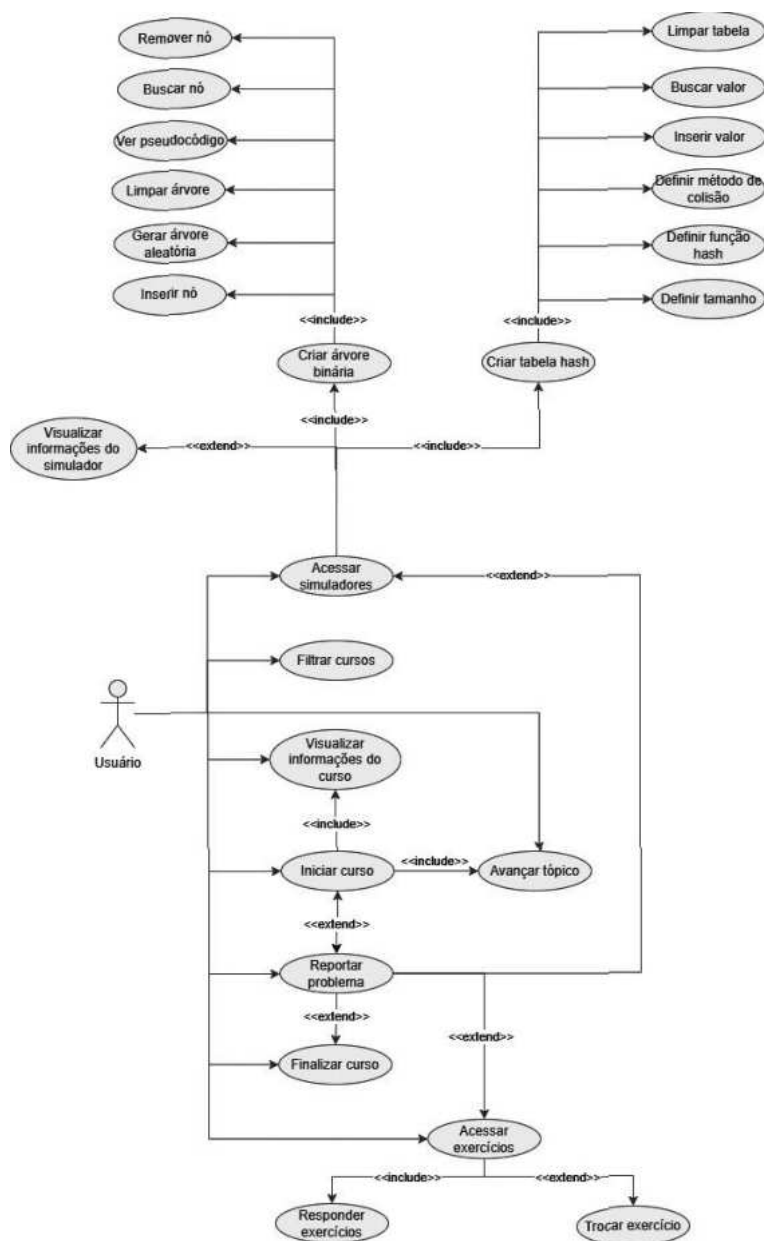


Figura 8 – Diagrama de Casos de Uso (Autoria Própria)

⁶ Nota da autora: No diagrama de casos de uso, "include" representa um caso obrigatório dentro de outro, enquanto "extend" define um caso opcional, que pode ser adicionado sob certas condições.

3.8.1 Especificação dos Casos de Uso

Nesta etapa, foram selecionados os dez principais casos de uso que representam de forma mais significativa as funcionalidades essenciais do sistema. A definição desses casos foi orientada por critérios de relevância e abrangência, priorizando aqueles que melhor refletem os processos centrais da aplicação e possuem maior riqueza de detalhes nas especificações.

Como exemplo, apresenta-se a seguir o caso de uso "Responder exercício", por se tratar de uma funcionalidade central da plataforma e conter uma especificação mais completa. Os demais quadros com a descrição dos outros casos de uso podem ser consultados no Apêndice C.

Caso de uso	Acessar exercícios
Descrição	Disponibiliza o acesso aos exercícios relacionados ao conteúdo do curso
Atores	Usuário
Fluxo de eventos	<ul style="list-style-type: none"> • Fluxo básico: <ul style="list-style-type: none"> – Usuário acessa a página principal; – Usuário acessa a página de informações do curso; – Usuário acessa a página de exercícios. • Fluxo alternativo: <ul style="list-style-type: none"> – Usuário finaliza um curso; – Página de curso finalizado é exibida; – Usuário acessa a página de exercícios.

Pontos de extensão	Exibir exercícios para o usuário durante o andamento do curso.
Pós-condições	Todos os exercícios do assunto correspondente são exibidos. Um menu é exibido para auxiliar na navegação entre exercícios.

4 PROTOTIPAÇÃO

Com os objetivos e os requisitos da plataforma devidamente definidos, deu-se início à etapa de prototipação, uma fase crucial para a visualização e validação das funcionalidades antes da implementação final. A criação dos protótipos possibilitou identificar problemas de usabilidade, ajustar fluxos de navegação e reduzir retrabalhos nas fases subsequentes do desenvolvimento. Além disso, permitiu-se o planejamento da identidade visual e da organização das interfaces, garantindo maior assertividade nas decisões de *design*.

No primeiro momento, foram visualmente estruturadas as principais interfaces e funcionalidades previstas nos requisitos do sistema. Para tal, foram elaborados *wireframes*, que representaram, de forma simplificada, a organização dos elementos nas telas. Esses esquemas facilitaram a validação preliminar dos fluxos de navegação e da disposição das informações. Os *wireframes* desenvolvidos podem ser consultados no Apêndice D.

Posteriormente, foi concebido o *design* final da aplicação, cuja estética seguiu a linha do neo-brutalismo, um estilo visual caracterizado por formas sólidas, cores contrastantes e elementos gráficos diretos. Essa escolha busca transmitir autenticidade e um apelo mais jovial e despojado, alinhando-se à proposta da plataforma (NN. Group, 2020). Os resultados dessa etapa podem ser visualizados no Apêndice E.

Na figura abaixo encontra-se um exemplo de um *wireframe* desenvolvido para a tela de exercícios em comparação com a tela final desenvolvida.

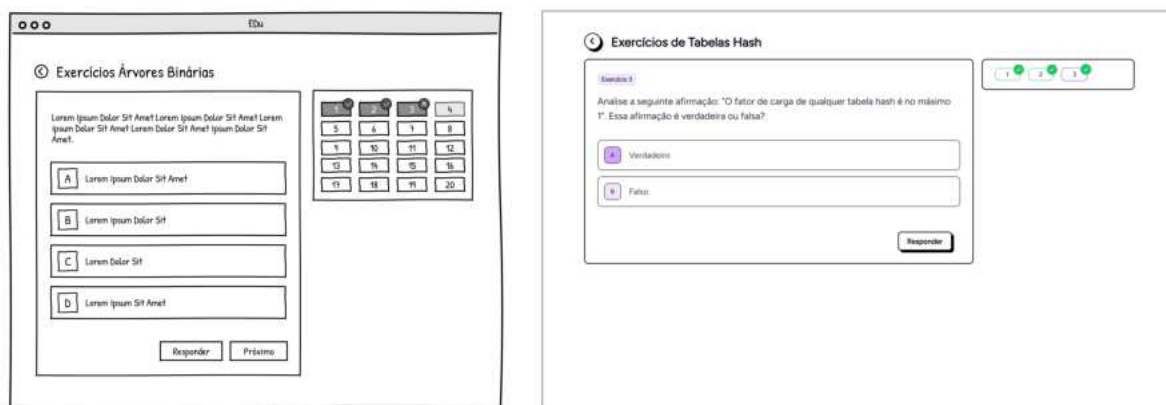


Figura 9 – *Wireframe* e Design final - Tela do curso (Autoria Própria)

5 DESENVOLVIMENTO

Neste capítulo serão abordadas as estratégias tomadas no desenvolvimento da aplicação, que foi estruturada em duas principais camadas: *Backend* e *Frontend*.

Todo o material desenvolvido para o EDu pode ser acessado através do GitHub ¹: <https://github.com/brunafalheiro/edu>. A seguir, cada componente é descrito em detalhes.

5.1 INFRAESTRUTURA

Como definido no Requisito do Sistema, a aplicação foi limitada a um *software desktop* multiplataforma. Para isso, foi utilizado o *framework* **Electron** (Electron, 2025), responsável por empacotar tecnologias web HTML, CSS e JavaScript em uma aplicação nativa. Além disso, foi utilizado o **Vite** como *bundler* ², a fim de acelerar o tempo de desenvolvimento.

Na Figura 10, é exibida a estrutura de diretórios do projeto, organizando os arquivos de configuração, banco de dados e componentes da interface.

¹ O GitHub é uma plataforma baseada em nuvem em que é possível armazenar, compartilhar e trabalhar com outras pessoas para escrever códigos. (GITHUB, 2025).

² Um *bundler* é uma ferramenta que reúne múltiplos arquivos e módulos de um projeto em um ou mais arquivos finais otimizados para melhorar o desempenho e simplificar o carregamento da aplicação (CONTRIBUTORS, 2025).

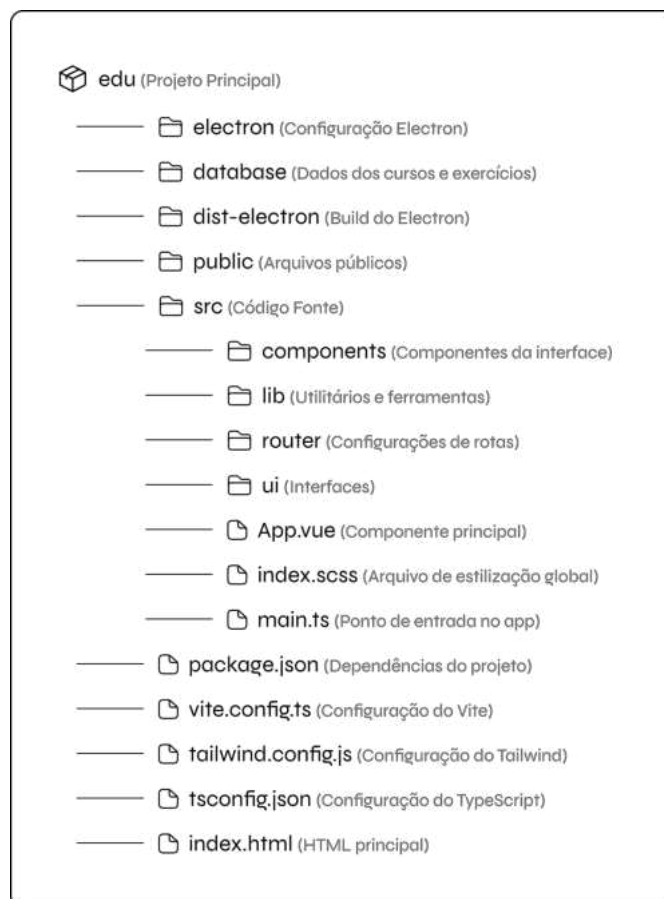


Figura 10 – Estrutura do projeto (Autoria Própria)

5.2 BACKEND

O desenvolvimento do **EDu** não contou com um sistema de banco de dados, uma vez que não inclui funcionalidades complexas, como autenticação e *login*. No lugar, foi utilizado exclusivamente arquivos JSON ³ para o armazenamento estruturado dos dados, permitindo otimizar o tempo de desenvolvimento sem comprometer a organização e a escalabilidade do sistema. Os arquivos JSON foram armazenados dentro do diretório *database*, situado na raiz do projeto.

Os dados da aplicação foram separados em dois arquivos JSON principais:

- **Courses:** Contém a estrutura hierárquica dos cursos, organizados em um *array* onde cada objeto representa um curso com identificador, nome, descrição e um conjunto de aulas. Cada aula, por sua vez, é composta por tópicos, os quais armazenam conteúdos classificados por tipo, como textos ou imagens.
- **Exercises:** Agrupa os exercícios por curso, permitindo a inclusão de questões do

³ JSON (*JavaScript Object Notation*) é um formato de estruturação de dados baseado em texto, que utiliza uma estrutura composta por pares de chave e valor (Ecma International, 2017).

tipo *quiz*, com múltiplas opções e indicação da resposta correta, ou questões discursivas, com respostas compostas por textos ou imagens.

A especificação dos dados de cursos e exercícios é detalhada nos JSON Schemas ⁴ presentes no Apêndice E. Na Figura 11, é apresentado o diagrama dos cursos, contando com informações sobre as aulas, tópicos e conteúdos, incluindo os tipos de dados associados a cada entidade. Já a Figura 12 representa o diagrama dos exercícios, com detalhes sobre as opções de resposta, alternativas corretas e a estrutura das perguntas.

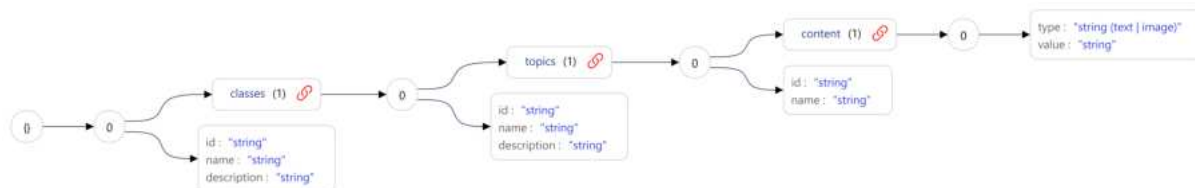


Figura 11 – Diagrama dos cursos (Autoria Própria)

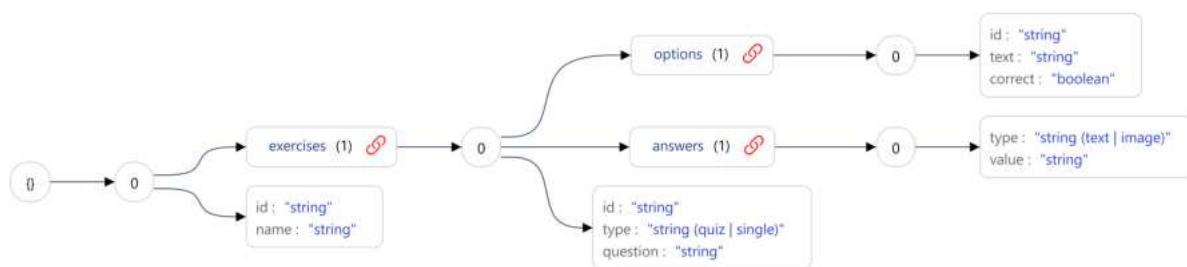


Figura 12 – Diagrama dos exercícios (Autoria Própria)

Os dados de progresso do curso e dos exercícios realizados foram armazenados em um *store*, um mecanismo de armazenamento persistente disponibilizado pelo pacote *electron-store*. Dessa forma, o usuário pode fechar e reabrir a aplicação sem perder os dados, que permanecerão salvos. Para adicionar novos conteúdos à plataforma, basta adicionar ao JSON correspondente, seguindo a estrutura do Schema fornecido.

5.3 FRONTEND

O desenvolvimento do *frontend* foi conduzido utilizando uma combinação de **Vue.js** (YOU, 2025b), **Vite** (YOU, 2025a) e **TailwindCSS** (LABS, 2025), integrados a um *boilerplate* pré-configurado para o **Electron**. Essa escolha de tecnologias visou otimizar a criação de interfaces reativas, modulares e de fácil manutenção.

A seleção do **Vue.js** ocorreu devido à sua capacidade de construir interfaces altamente reativas e estruturadas em componentes isolados. Esse modelo de desenvolvimento foi

⁴ JSON Schema é uma especificação que define a estrutura, o conteúdo e as restrições de dados em documentos JSON, permitindo validação automática e padronização de dados entre sistemas (JSON Schema, 2020).

essencial para telas mais complexas, como a de navegação do curso, que integra elementos como menu de tópicos, progressão do aprendizado e porcentagem de conclusão. Com o uso de componentes independentes, foi possível realizar ajustes pontuais e permitir expansões futuras e manutenções de forma ágil, sem comprometer outras partes do sistema.

Para a construção visual, optou-se pelo uso do **TailwindCSS** em conjunto com a biblioteca de componentes **Shadcn for Vue** (UNOVUE, 2025). Essa combinação permitiu a criação de interfaces de alto padrão, facilitando a personalização dos elementos de acordo com a identidade visual da plataforma. Essa agilidade foi especialmente relevante em telas que demandam maior detalhamento visual, garantindo qualidade estética e alta usabilidade sem comprometer a produtividade.

A partir dessas escolhas e dos *wireframes* levantados, foram implementadas as telas em conjunto com a identidade visual do sistema, apresentadas no Apêndice F.

6 AVALIAÇÃO DO SISTEMA

A avaliação do sistema foi realizada por meio de uma pesquisa aplicada a estudantes do curso de Ciência da Computação, com o objetivo de examinar a usabilidade, a clareza da interface e a eficácia das funcionalidades implementadas na plataforma. A pesquisa adotou uma abordagem mista, composta por perguntas objetivas, em escala de 1 a 5, e uma pergunta aberta, com o intuito de captar *feedbacks* mais detalhados por parte dos usuários.

Ao todo, foram coletadas 21 respostas ao longo de uma semana. As perguntas objetivas permitiram mensurar, de forma direta, o grau de satisfação dos estudantes com diferentes aspectos do sistema, enquanto as perguntas abertas proporcionaram um espaço para sugestões, elogios e críticas espontâneas. A estrutura completa do questionário pode ser consultada no Apêndice G, e os dados brutos das respostas estão disponíveis no Apêndice H.

A pesquisa avaliou os seguintes critérios, agrupados de acordo com os componentes principais da plataforma:

- Clareza da interface e dos comandos para realizar operações nos simuladores de Árvores Binárias e Tabelas *Hash*;
- Qualidade do *feedback* visual fornecido após as interações com os simuladores;
- Facilidade de manipulação dos elementos gráficos e legibilidade das representações visuais;
- Navegação entre módulos e aulas, bem como o suporte para localização e retomada de progresso nos cursos;
- Usabilidade da ferramenta de exercícios, incluindo esforço necessário para completá-los e controle sobre o andamento das atividades;
- Avaliação geral sobre facilidade de uso, eficiência na realização de tarefas e curva de aprendizado da plataforma.

De modo geral, os resultados revelam uma recepção bastante positiva por parte dos participantes. A análise das respostas objetivas indica um bom nível de satisfação, tendo a maioria das métricas avaliadas apresentado médias superiores a 4,5. Entre as mais bem avaliadas, destacam-se:

- Organização visual da árvore binária (4,86);
- Facilidade para localizar módulos e aulas (4,81);

- Eficiência na execução de tarefas (4,76);
- Clareza dos comandos para criação e manipulação de estruturas (4,57);
- *Feedback* visual das operações realizadas (entre 4,57 e 4,76);
- Baixo esforço de aprendizado (4,71).

Além disso, as respostas livres também indicam uma boa satisfação com o sistema. Muitos estudantes elogiaram a didática da proposta e o visual simples e agradável. A organização dos cursos, especialmente pela oferta de uma visão geral prévia do conteúdo, foi amplamente destacada. As simulações de estruturas de dados também receberam comentários positivos, sendo descritas como claras, instrutivas e visualmente eficazes para a compreensão dos conceitos. Tais resultados evidenciam que tanto o design da interface quanto a arquitetura da informação contribuíram para uma experiência fluida e intuitiva, facilitando o processo de aprendizagem.

Contudo, apesar da avaliação majoritariamente favorável, alguns pontos de atenção foram identificados a partir das respostas dos alunos. Dois critérios obtiveram médias relativamente inferiores:

- Manipulação de elementos na árvore binária (4,24);
- Esforço exigido para concluir os exercícios (4,10).

Nos comentários abertos, foram mencionadas oportunidades específicas de melhoria, entre as quais se destacam:

- Ausência de funcionalidades como remoção de elementos e visualização de colisões nas simulações de Tabelas *Hash*;
- Falta de pseudocódigo nas simulações de Tabelas *Hash*, em contraste com o que já está presente nas simulações de árvores binárias;
- Sugestão de tornar a área de pseudocódigo redimensionável e com formatação colorida, semelhante a editores de código;
- Ausência de um botão “voltar” na interface dos exercícios;
- Desejo por uma funcionalidade que gere estruturas aleatórias nas simulações de Tabelas *Hash*.

A combinação entre altos índices de satisfação e a riqueza de *feedbacks* qualitativos demonstrou que a plataforma apresenta um desempenho sólido em termos de usabilidade, clareza e eficiência. Ao mesmo tempo, indica um potencial de evolução que, se atendido,

poderá tornar a experiência ainda mais completa e personalizada às necessidades dos estudantes. Embora não medido, entende-se também que a plataforma por ser de valia para os docentes, uma vez que pode atuar como ferramenta de auxílio durante e após as aulas.

7 CONCLUSÃO

7.1 RESUMO DO PROJETO

O presente trabalho teve como objetivo o desenvolvimento de um software educacional para o apoio ao estudo de Estruturas de Dados, com foco em uma abordagem dinâmica e interativa. A solução proposta visa facilitar o aprendizado de conceitos fundamentais, utilizando animações, visualizações gráficas e pseudocódigo para ilustrar o funcionamento interno das estruturas, promovendo uma compreensão mais intuitiva e prática. O projeto foi idealizado para atender principalmente estudantes com conhecimentos introdutórios em programação, buscando suprir lacunas identificadas em métodos tradicionais de ensino.

7.2 LIMITADORES DURANTE A EXECUÇÃO

Durante o desenvolvimento, alguns limitadores impactaram a implementação de certas funcionalidades. Entre os principais desafios, destacam-se:

- Restrição de tempo para o desenvolvimento de todas as funcionalidades abrangidas pelas outras plataformas analisadas no *benchmarking*;
- Limitações técnicas na integração de algumas animações mais complexas;
- Restrição de tempo para a abrangência de mais estruturas de dados.

Além disso, na fase final de desenvolvimento do projeto, surgiu um desafio relevante relacionado ao processo de *build*¹ do *software*, onde não estava sendo possível gerar o arquivo executável do **EDu**. Foram realizadas diversas tentativas para contornar os erros e inconsistências que apareciam nesse processo.

Essa dificuldade despertou uma reflexão mais ampla sobre a viabilidade do modelo inicialmente adotado. Como o sistema havia sido projetado para funcionar sem um banco de dados, a distribuição por meio de executáveis gerava um problema estrutural importante: a atualização de conteúdo. A cada mudança para adicionar módulos, corrigir exercícios ou implementar melhorias, os usuários teriam que baixar novamente a versão atualizada do *software*. Em um contexto educacional, isso representaria um obstáculo considerável para a escalabilidade da ferramenta. No possível cenário de um estudante estar utilizando uma versão desatualizada, o mesmo perderia acesso aos conteúdos pedagógicos mais recentes até que realizasse a atualização do sistema.

Diante desse cenário, foi tomada a decisão de alterar a arquitetura do projeto, removendo o Electron e realizando uma migração para uma aplicação *web* hospedada. A

¹ O processo de *build*, no contexto deste projeto, refere-se a geração do arquivo executável necessário para a distribuição da aplicação desenvolvida com Electron.

plataforma passou a ser publicada por meio da Vercel ², que oferece *deploy* automático, sem a necessidade de grandes configurações. Essa escolha abriu maior espaço para o desenvolvimento colaborativo, uma vez que qualquer pessoa pode sugerir melhorias e, assim que aprovadas, essas contribuições são incorporadas ao código-fonte e automaticamente integradas ao site. O código alterado pode ser encontrado diretamente através do repositório no GitHub: <https://github.com/brunafalheiro/edu-web>. O acesso à plataforma hospedada pode ser feito através do link: <https://edu-estruturas-de-dados.vercel.app>.

A remoção do Electron do projeto não gerou grandes efeitos colaterais no código, uma vez que sua função principal era atuar como um empacotador responsável por transformar uma aplicação naturalmente web em um software executável para desktop. Portanto, a lógica e estrutura da aplicação web permaneceram essencialmente as mesmas, independentemente da presença do Electron. A única modificação significativa foi a substituição da biblioteca *electron-store*, utilizada para armazenamento persistente no desktop, pelo *localStorage*, API nativa dos navegadores para salvar dados no cliente.

Embora essa decisão tenha promovido uma complexidade adicional ao final do desenvolvimento, exigindo uma reestruturação do sistema, o ganho de atualização contínua compensou a limitação inicial.

A principal desvantagem dessa abordagem é a dependência de conexão com a Internet. Como o sistema passou a ser acessado via navegador, é necessário que o dispositivo do usuário tenha acesso à Internet. Ainda assim, a escolha foi justificada devido à necessidade de garantir maior abertura à colaboração no desenvolvimento da ferramenta.

7.3 TRABALHOS FUTUROS

Com a finalização deste projeto, surgem diversas possibilidades de aprimoramento e expansão que podem torná-lo ainda mais robusto e interativo. Além das sugestões de melhoria que podem ser extraídos dos *feedbacks* dos alunos, algumas direções para trabalhos futuros incluem:

- **Adição de novas Estruturas de Dados:** A introdução de novas estruturas de dados mais complexas pode enriquecer a arquitetura do sistema. Isso possibilitaria otimizações no processamento de informações, maior eficiência na busca por conteúdos e uma expansão das funcionalidades, adequando o sistema para demandas mais sofisticadas.
- **Integração com banco de dados:** A utilização de um banco relacional é uma evolução natural do projeto. A conversão do modelo atual baseado em JSON para uma base de dados seria relativamente simples e viável.

² Vercel é uma plataforma de hospedagem de aplicações *web*, permitindo *deploys* rápidos e escaláveis, com integração nativa a diversos *frameworks* e ferramentas de desenvolvimento front-end.

- **Inclusão de simuladores nos cursos e exercícios:** Incorporar simuladores interativos diretamente nos cursos e exercícios para aumentar ainda mais a interatividade do sistema.
- **Exercícios com *feedback*:** Integrar explicação de solução nos exercícios realizados.
- **Integração entre os alunos:** Adicionar funcionalidades que permitam a interação entre os alunos, como fóruns de discussão, troca de mensagens ou atividades colaborativas.

REFERÊNCIAS

BROWN, T. et al. Design thinking. **Harvard business review**, v. 86, n. 6, p. 84, 2008.

CONTRIBUTORS, V. **Vite Documentation**. 2025. Accessed: 2025-05-28. Disponível em: <https://vitejs.dev/guide/>.

CORMEN, T. H. et al. **Introduction to Algorithms**. 2nd. ed. The MIT Press, 2001. ISBN 0262032937. Disponível em: <http://www.amazon.com/Introduction-Algorithms-Thomas-H-Cormen/dp/0262032937%3FSubscriptionId%3D13CT5CVB80YFWJEPWS02%26tag%3Dws%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0262032937>.

Ecma International. **The JSON Data Interchange Syntax**. 2. ed. Geneva, 2017. Disponível em: <https://www.ecma-international.org/publications-and-standards/standards/ecma-404/>. Acesso em: 5 maio 2025.

Electron. **Electron**. 2025. <https://www.electronjs.org/pt/>. Acesso em 16 ago. 2025.

GITHUB. **Sobre o GitHub e o Git**. 2025. Disponível em: <https://docs.github.com/pt/get-started/start-your-journey/about-github-and-git>. Acesso em 11 junho 2025. Disponível em: <https://docs.github.com/pt/get-started/start-your-journey/about-github-and-git>.

GONZATTO, R. F.; COSTA, K. d. C. Arquitetura de informação sem wireframe. **Perspectivas em Gestão e Conhecimento**, v. 1, p. 160–181, out. 2011. Disponível em: <https://periodicos.ufpb.br/ojs2/index.php/pgc/article/view/10793>.

GROUP, N. N. **Personas vs. Archetypes**. 2020. Disponível em: <https://www.nngroup.com/articles/personas-archetypes/?lm=personas-study-guidept=article>. Acesso em: 5 maio 2025.

JSON Schema. **Understanding JSON Schema**. 2020. <https://json-schema.org/understanding-json-schema/>. Disponível em: <https://json-schema.org/understanding-json-schema/>. Acesso em: 05 maio 2025.

KISTOWSKI, J. v. et al. How to build a benchmark. In: **Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering**. New York, NY, USA: Association for Computing Machinery, 2015. (ICPE '15), p. 333–336. ISBN 9781450332484. Disponível em: <https://doi.org/10.1145/2668930.2688819>.

LABS, T. **Tailwind CSS**. 2025. <https://tailwindcss.com/>. Acesso em 16 ago. 2025.

MARTINS, A.; ELOY, A. Educação integral por meio do pensamento computacional-letramento em programação: Relatos de experiências e artigos científicos (1. a edição). **Editora Appris Ltda**, 2019.

MORAN, T. P. The command language grammar: a representation for the user interface of interactive computer systems. **International Journal of Man-Machine Studies**, v. 15, n. 1, p. 3–50, 1981. ISSN 0020-7373. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020737381800223>.

NN. Group. **Neobrutalism: Definition and Best Practices**. 2020. <https://www.nngroup.com/articles/neobrutalism/>. Disponível em: <https://www.nngroup.com/articles/neobrutalism/>. Acesso em: 05 maio 2025.

NORMAN, D. A. **The Design of Everyday Things**. USA: Basic Books, Inc., 2002. ISBN 9780465067107.

Norman Nielsen Group. **Design Thinking**. 2025. <https://www.nngroup.com/articles/design-thinking/>. Acesso em 20 mai. 2025.

OSTERWALDER, A. et al. **Value proposition design: How to create products and services customers want**. [S.l.]: John Wiley & Sons, 2015.

ROBINS, A.; ROUNTREE, J.; ROUNTREE, N. Learning and teaching programming: A review and discussion. **Computer Science Education**, v. 13, p. 137–, 06 2003.

SOMMERVILLE, I. **Software Engineering, 9/E**. [S.l.]: Pearson Education India, 2011.

SZWARCFITER, J. L.; MARKEZON, L. **Estruturas de Dados e Algoritmos**. [S.l.]: Paperback, 1994.

UNOVUE. **Shadcn-Vue**. 2025. <https://www.shadcn-vue.com/>. Acesso em 16 ago. 2025.

WING, J. M. Computational thinking. **Communications of the ACM**, ACM, v. 49, n. 3, p. 33–35, 2006.

YOU, E. **Vite**. 2025. <https://vite.dev/>. Acesso em 16 ago. 2025.

YOU, E. **Vue.js**. 2025. <https://vuejs.org/>. Acesso em 16 ago. 2025.

APÊNDICE A – FORMULÁRIO DE PESQUISA DE DEMANDA

O formulário a seguir foi divulgado para obter informações sobre o nível de compreensão e preferências dos estudantes em relação à disciplina de estrutura de dados.

Você está cursando ou já cursou a disciplina de Estrutura de Dados?

- ☐ Estou cursando
- ☐ Cursei nos últimos 6 meses
- ☐ Cursei há mais de 6 meses

Como você avalia seu nível de compreensão sobre Estrutura de Dados?

- ☐ Iniciante
- ☐ Intermediário
- ☐ Avançado

Quais dos seguintes conceitos de Estrutura de Dados você considera mais difíceis de serem aprendidos?

- ☐ Algoritmos de Busca
- ☐ Algoritmos de Ordenação
- ☐ Árvores
- ☐ Grafos
- ☐ Hash Tables
- ☐ Listas
- ☐ Pilhas
- ☐ Filas
- ☐ Heap

Qual é a maior dificuldade que você encontra ao estudar Estrutura de Dados?

- ☐ Compreensão dos conceitos teóricos
- ☐ Implementação dos algoritmos
- ☐ Relacionar teoria com a prática
- ☐ Dificuldade de visualizar os conceitos
- ☐ Falta de recursos interativos

Quais métodos de ensino você prefere utilizar quando está estudando Estrutura de Dados?

- ☐ Animações interativas
- ☐ Videoaulas
- ☐ Exercícios práticos ou de código
- ☐ Material de leitura
- ☐ Plataforma de cursos online (ex.: Coursera, Udemmy ou outros)

Quais métodos de ensino você prefere utilizar quando está revisando Estrutura de Dados?

- ☐ Animações interativas
- ☐ Videoaulas
- ☐ Exercícios práticos ou de código
- ☐ Material de leitura
- ☐ Plataforma de cursos online (ex.: Coursera, Udemmy ou outros)
- ☐ Resumos

Quais funcionalidades você gostaria de ver em um software de aprendizado de Estrutura de Dados?

- ☐ Aplicação dos conceitos em problemas reais
- ☐ Comparação de diferentes algoritmos
- ☐ Exercícios práticos com feedback
- ☐ Simulação de algoritmos com animações
- ☐ Explicações passo a passo de algoritmos

APÊNDICE B – RESULTADOS DA PESQUISA DE DEMANDA

1. Você está cursando ou já cursou a disciplina de Estrutura de Dados?



Figura 13 – Respostas Pesquisa de Demanda: Há quanto tempo estudou estrutura de dados

2. Como você avalia seu nível de compreensão sobre Estrutura de Dados?

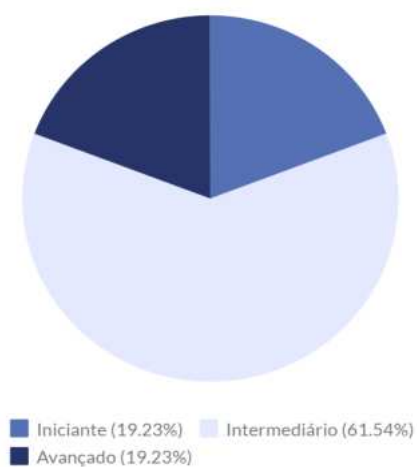


Figura 14 – Respostas Pesquisa de Demanda: Nível de compreensão

3. Quais dos seguintes conceitos de Estrutura de Dados você considera mais difíceis de serem aprendidos?

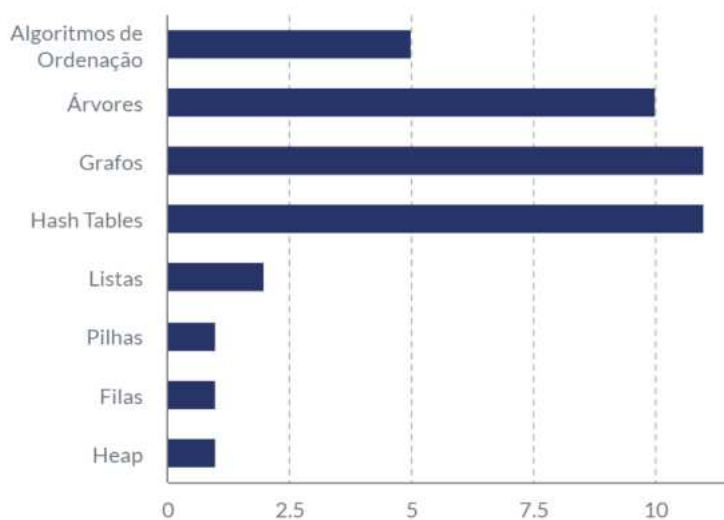


Figura 15 – Respostas Pesquisa de Demanda: Conceitos mais desafiadores

4. Qual é a maior dificuldade que você encontra ao estudar Estrutura de Dados?

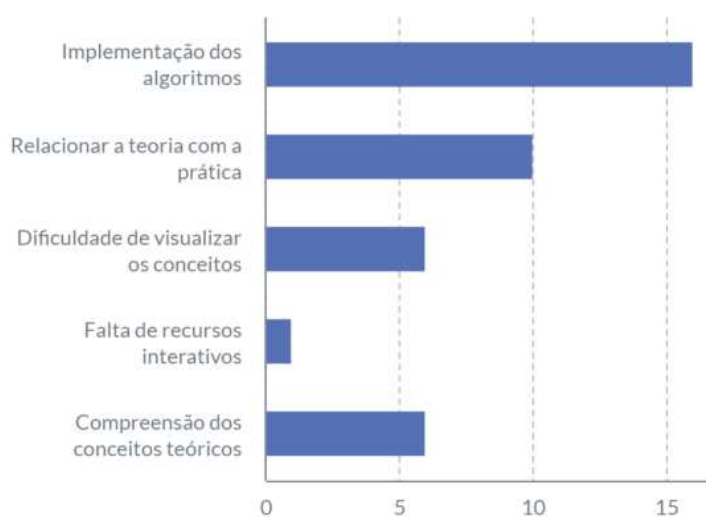


Figura 16 – Respostas Pesquisa de Demanda: Dificuldades ao estudar

5. Quais métodos de ensino você prefere utilizar quando está estudando Estrutura de Dados?



Figura 17 – Respostas Pesquisa de Demanda: Preferência de métodos de aprendizado

6. Quais métodos de ensino você prefere utilizar quando está revisando Estrutura de Dados?



Figura 18 – Respostas Pesquisa de Demanda: Preferência de métodos de revisão

7. Quais funcionalidades você gostaria de ver em um software de aprendizado de Estrutura de Dados?

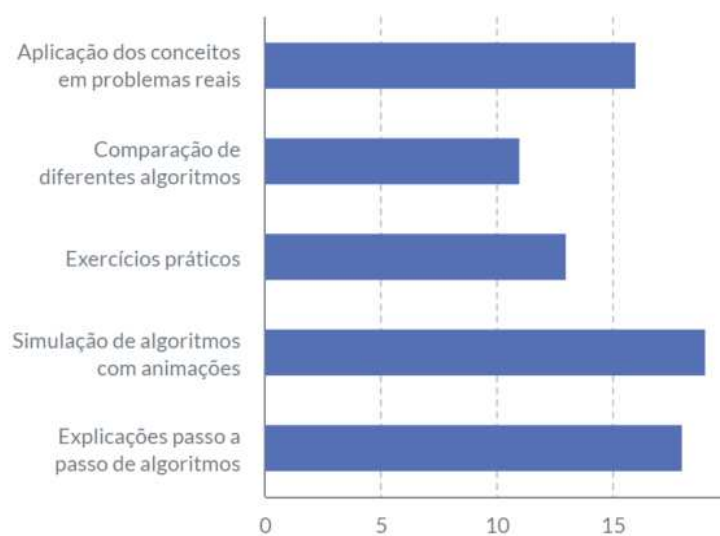


Figura 19 – Respostas Pesquisa de Demanda: Funcionalidades esperadas

APÊNDICE C – ESPECIFICAÇÃO DOS CASOS DE USO

Caso de uso	Filtrar cursos
Descrição	Permite ao usuário filtrar os cursos em andamento, finalizados e todos
Atores	Usuário
Fluxo de eventos	<ul style="list-style-type: none">• O usuário acessa a página inicial;• O usuário seleciona os filtros desejados.
Pontos de extensão	Adicionar mais filtros conforme aumentar a quantidade de cursos
Pós-condições	Cursos filtrados são exibidos na interface
Mini estrutura de informações	<ul style="list-style-type: none">• Seletores.

Caso de uso	Visualizar informações do curso
Descrição	Exibe detalhes sobre o curso, como ementa, carga horária e conteúdo programático
Atores	Usuário
Fluxo de eventos	<ul style="list-style-type: none"> • O usuário acessa a página inicial; • O usuário seleciona o curso desejado.
Pontos de extensão	Exibir informações de de avaliação e de quantos alunos já finalizaram o curso
Mini estrutura de informações	<ul style="list-style-type: none"> • Tempo de duração; • Quantidade de aulas; • Quantidade de tópicos; • Informações das aulas; • Informações dos tópicos.

Caso de uso	Iniciar curso
Descrição	Inicia o curso selecionado e disponibiliza seu conteúdo
Atores	Usuário
Pré-Condições	O curso deve estar com status de não iniciado.
Fluxo de eventos	<ul style="list-style-type: none"> • Usuário acessa a página de informações do curso; • Usuário inicia o curso.
Pós-condições	Usuário é redirecionado para a página do curso.
Mini estrutura de informações	<ul style="list-style-type: none"> • Identificador do curso.

Caso de uso	Finalizar curso
Descrição	Conclui o curso em andamento, registrando o término na plataforma
Atores	Usuário
Pré-Condições	Todos os tópicos de todas as aulas do curso devem estar com status completado.
Fluxo de eventos	<ul style="list-style-type: none"> • Fluxo basico: <ul style="list-style-type: none"> – Usuário estuda os tópicos de maneira sequencial; – Usuário finaliza o curso. • Fluxo alternativo: <ul style="list-style-type: none"> – Usuário estuda os tópicos de maneira aleatória; – Usuário finaliza o curso.
Pós-condições	Tela de curso completado é exibida.
Mini estrutura de informações	<ul style="list-style-type: none"> • Identificador do curso.

Caso de uso	Acessar exercícios
Descrição	Disponibiliza o acesso aos exercícios relacionados ao conteúdo do curso
Atores	Usuário
Fluxo de eventos	<ul style="list-style-type: none"> • Fluxo básico: <ul style="list-style-type: none"> – Usuário acessa a página principal; – Usuário acessa a página de informações do curso; – Usuário acessa a página de exercícios. • Fluxo alternativo: <ul style="list-style-type: none"> – Usuário finaliza um curso; – Página de curso finalizado é exibida; – Usuário acessa a página de exercícios.
Pontos de extensão	Exibir exercícios para o usuário durante o andamento do curso.
Pós-condições	Todos os exercícios do assunto correspondente são exibidos. Um menu é exibido para auxiliar na navegação entre exercícios.

Caso de uso	Responder exercício
Descrição	Permite ao usuário responder questões e atividades propostas nos exercícios
Atores	Usuário
Pré-Condições	Ter selecionado uma resposta
Fluxo de eventos	<ul style="list-style-type: none"> • Usuário seleciona uma resposta; • Usuário clica em "Responder".
Pontos de extensão	<ul style="list-style-type: none"> • Feedback - Exibir explicação da resposta caso o usuário erre a resposta.
Pós-condições	<ul style="list-style-type: none"> • Usuário errou a resposta: Questão correspondente no menu de questões é marcada como errada. • Usuário acertou a resposta: Questão correspondente no menu de questões é marcada como acertada.
Mini estrutura de informações	<ul style="list-style-type: none"> • Identificador da questão; • Identificador da resposta selecionada.

Caso de uso	Inserir nó
Descrição	Adiciona um novo nó em uma estrutura de árvore binária no simulador
Atores	Usuário
Fluxo de eventos	<ul style="list-style-type: none"> • Usuário seleciona o valor a ser inserido • Usuario clica em inserir nó
Pós-condições	<ul style="list-style-type: none"> • Árvore está vazia: Nó inserido se torna raiz. • Caso contrário: O nó torna-se uma folha e é alocado através do algoritmo de inserção.
Mini estrutura de informações	<ul style="list-style-type: none"> • Valor do nó; • Árvore.

Caso de uso	Gerar árvore aleatória
Descrição	Cria uma árvore binária com valores dos nós gerados de forma aleatória
Atores	Usuário
Pré-Condições	<ul style="list-style-type: none"> • Ter definido a quantidade de nós da árvore.
Fluxo de eventos	<ul style="list-style-type: none"> • Usuário acessa o Simulador de Árvores Binárias; • Usuário seleciona a quantidade de nós da árvore; • Usuário clica em "Gerar árvore aleatória".
Pós-condições	Uma árvore com a quantidade de nós selecionada e valores aleatórios é exibida na interface.
Mini estrutura de informações	<ul style="list-style-type: none"> • Quantidade de nós.

Caso de uso	Criar Tabela <i>Hash</i>
Descrição	Cria uma Tabela <i>Hash</i> vazia
Atores	Usuário
Pré-Condições	Ter definido o tamanho da tabela, a função <i>Hash</i> e o método de colisão a ser utilizado.
Fluxo de eventos	<ul style="list-style-type: none"> • Usuário acessa o Simulador de Tabelas <i>Hash</i>; • Usuário define o tamanho da tabela; • Usuário seleciona a função <i>Hash</i> a ser utilizada; • Usuário seleciona o método de colisão a ser utilizado.
Pós-condições	Uma Tabela <i>Hash</i> a quantidade de <i>slots</i> definido previamente é exibida na interface. A função <i>Hash</i> e método de colisão definem o comportamento da tabela ao adicionar um novo valor.
Mini estrutura de informações	<ul style="list-style-type: none"> • Tamanho da tabela; • Função <i>Hash</i>; • Método de colisão.

APÊNDICE D – WIREFRAMES

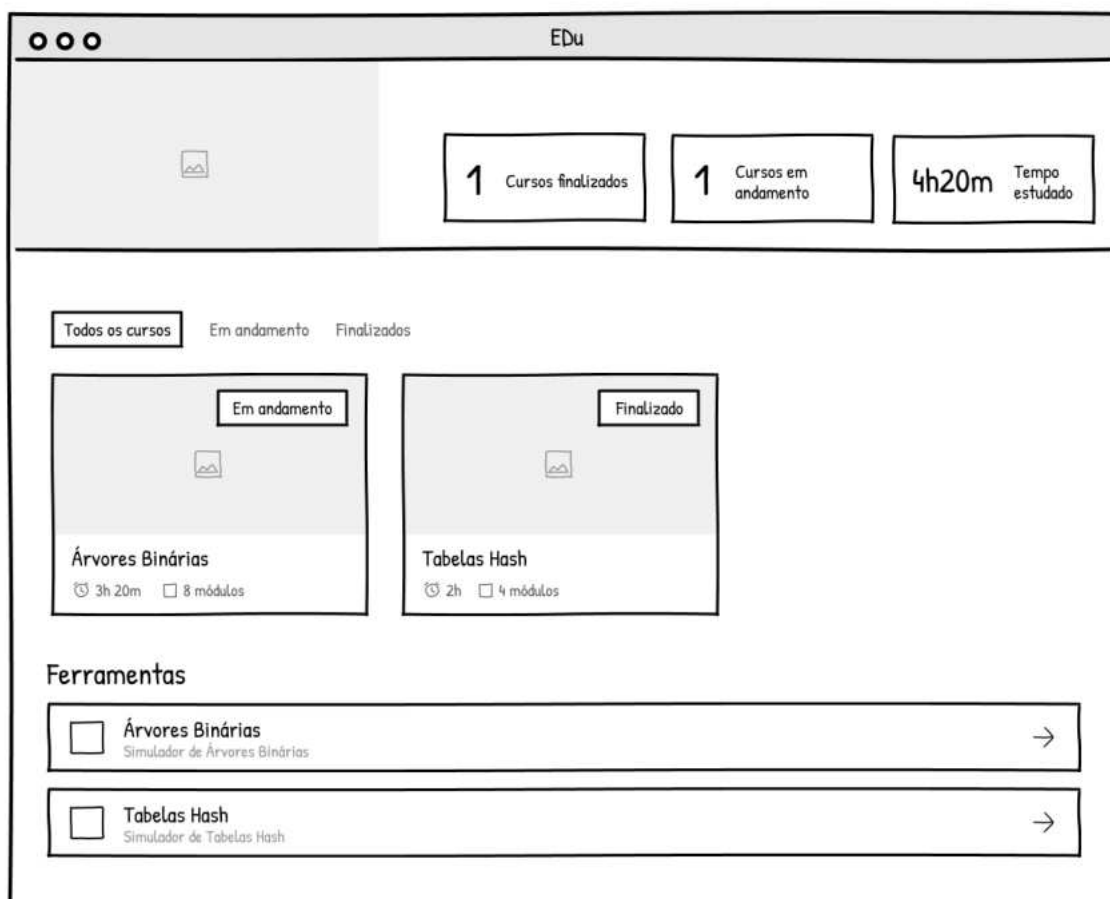


Figura 20 – Wireframe: Página Inicial

A página inicial é o ponto de entrada do usuário na plataforma. Nela, são exibidos os principais módulos de navegação e destaques dos cursos disponíveis. O design é pensado para oferecer uma experiência clara e intuitiva, facilitando o acesso rápido às funcionalidades principais.

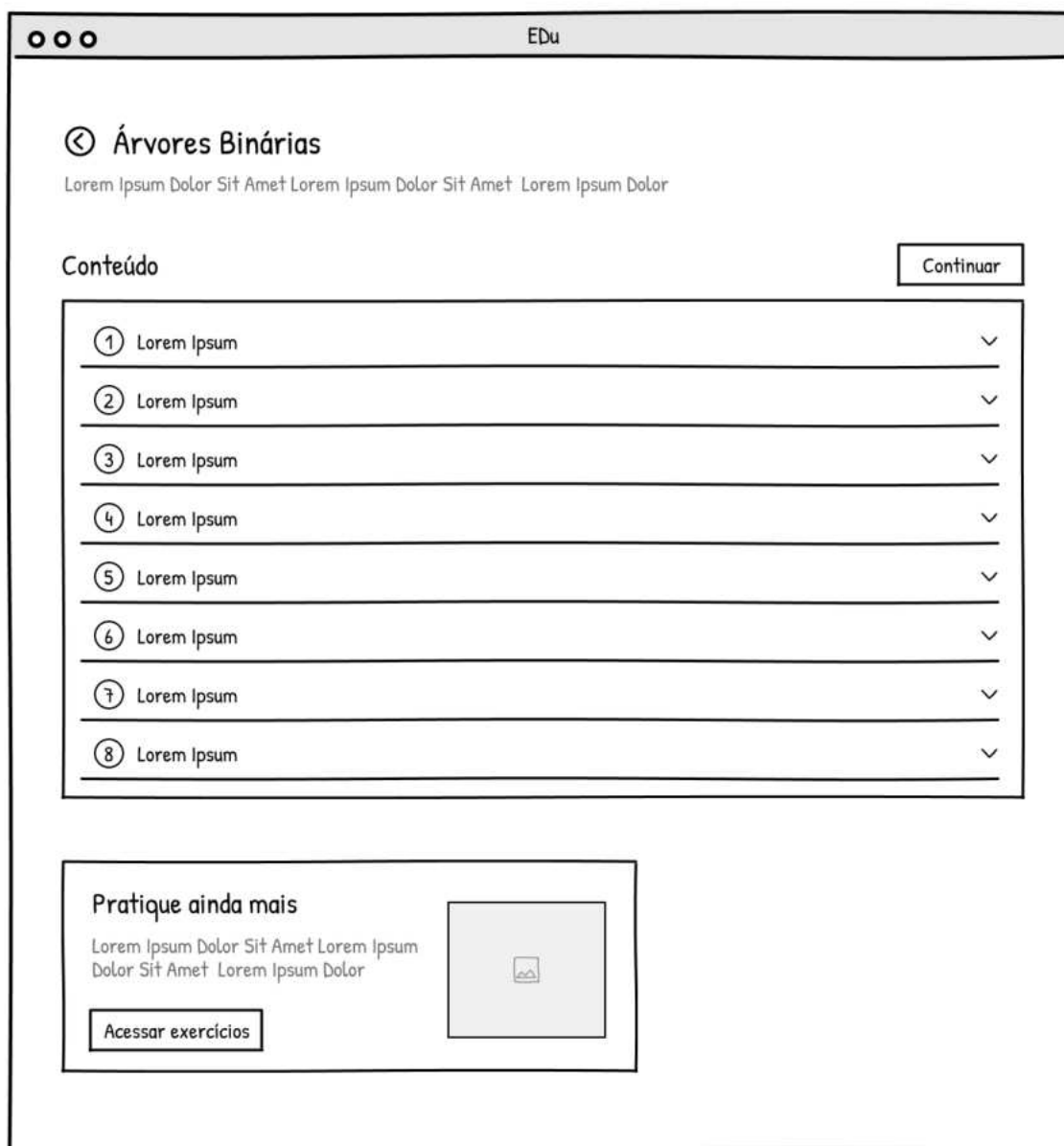


Figura 21 – Wireframe: Página de Informações Sobre o Curso

Nesta tela, o usuário encontra todas as informações detalhadas sobre o curso selecionado. São exibidos o conteúdo programático e descrição dos módulos.

A página do curso é dedicada ao acompanhamento das aulas e do progresso do usuário. Cada módulo é apresentado de forma sequencial, permitindo fácil navegação entre as etapas do aprendizado. Imagens e explicações textuais são acessíveis de maneira direta. O layout prioriza clareza e foco no conteúdo, reduzindo distrações durante o estudo.

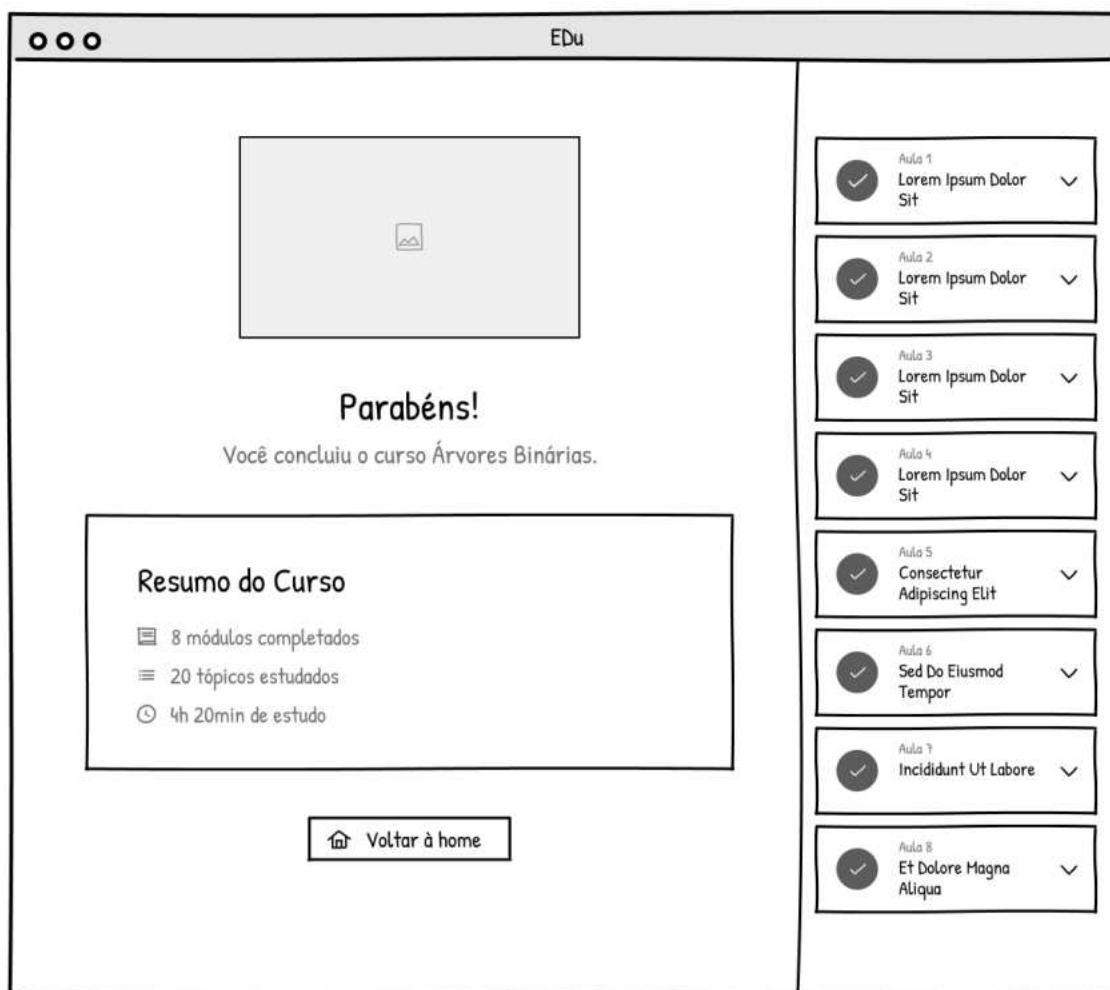


Figura 23 – Wireframe: Página do Curso Finalizado

Esta página é exibida ao finalizar todos os tópicos do curso. Nela, são apresentadas informações gerais, como a quantidade de módulos e tópicos concluídos, além do tempo total de estudo.

EDu

⏪ Exercícios Árvores Binárias

Lorem Ipsum Dolor Sit Amet Lorem Ipsum Dolor Sit Amet Lorem
Ipsum Dolor Sit Amet Lorem Dolor Sit Amet Ipsum Dolor Sit
Amet.

A

 Lorem Ipsum Dolor Sit Amet

B

 Lorem Ipsum Dolor Sit

C

 Lorem Dolor Sit

D

 Lorem Ipsum Sit Amet

Responder

Próximo

1 ✓

2 ✓

3 ✗

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

Figura 24 – Wireframe: Página de Exercícios

Esta página é projetada para a prática dos conhecimentos adquiridos durante o curso. Os exercícios são apresentados de forma interativa, permitindo que o usuário responda diretamente pela interface. Feedbacks imediatos são exibidos após cada resposta, reforçando o aprendizado.

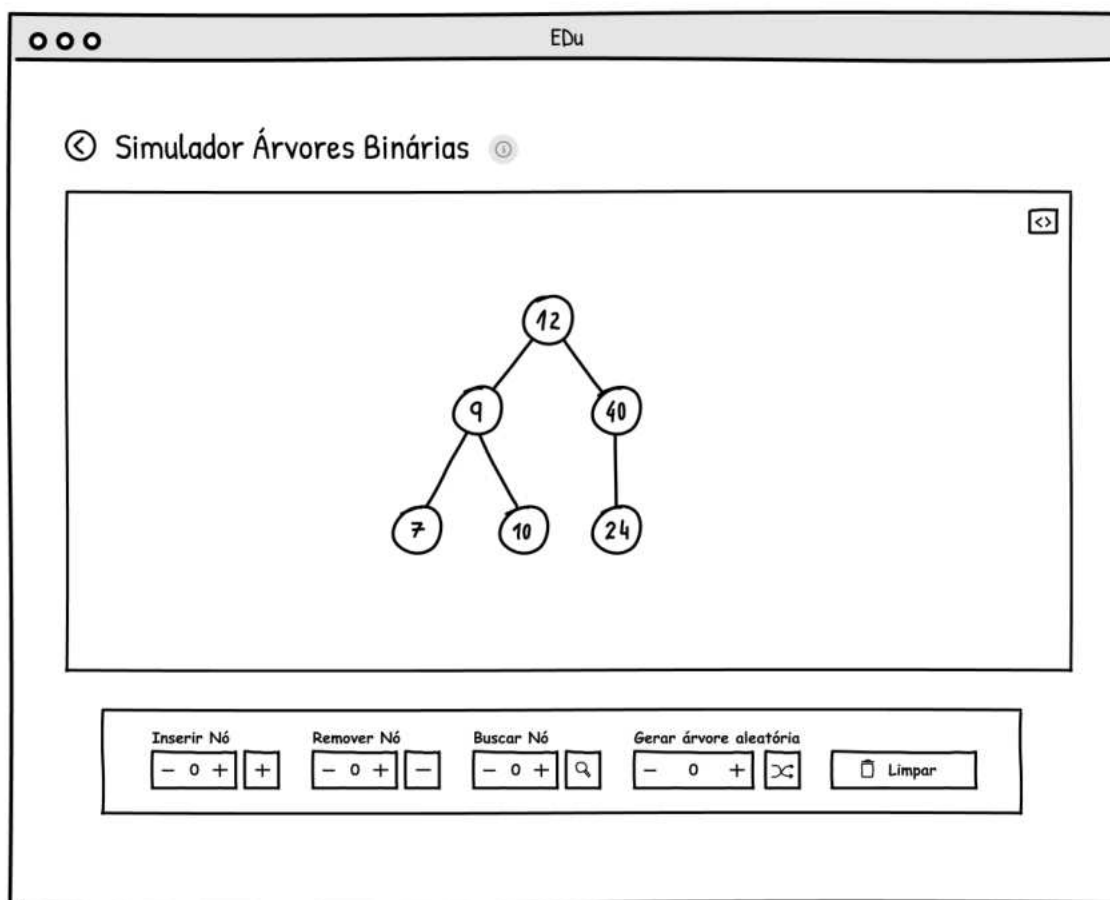


Figura 25 – Wireframe: Página do Simulador de Árvores Binárias

Esta interface busca facilitar a visualização das estruturas em tempo real, auxiliando na compreensão dos conceitos teóricos. As alterações são exibidas de maneira dinâmica, proporcionando uma experiência de aprendizado mais interativa. O usuário pode testar diferentes cenários para explorar o comportamento da árvore.

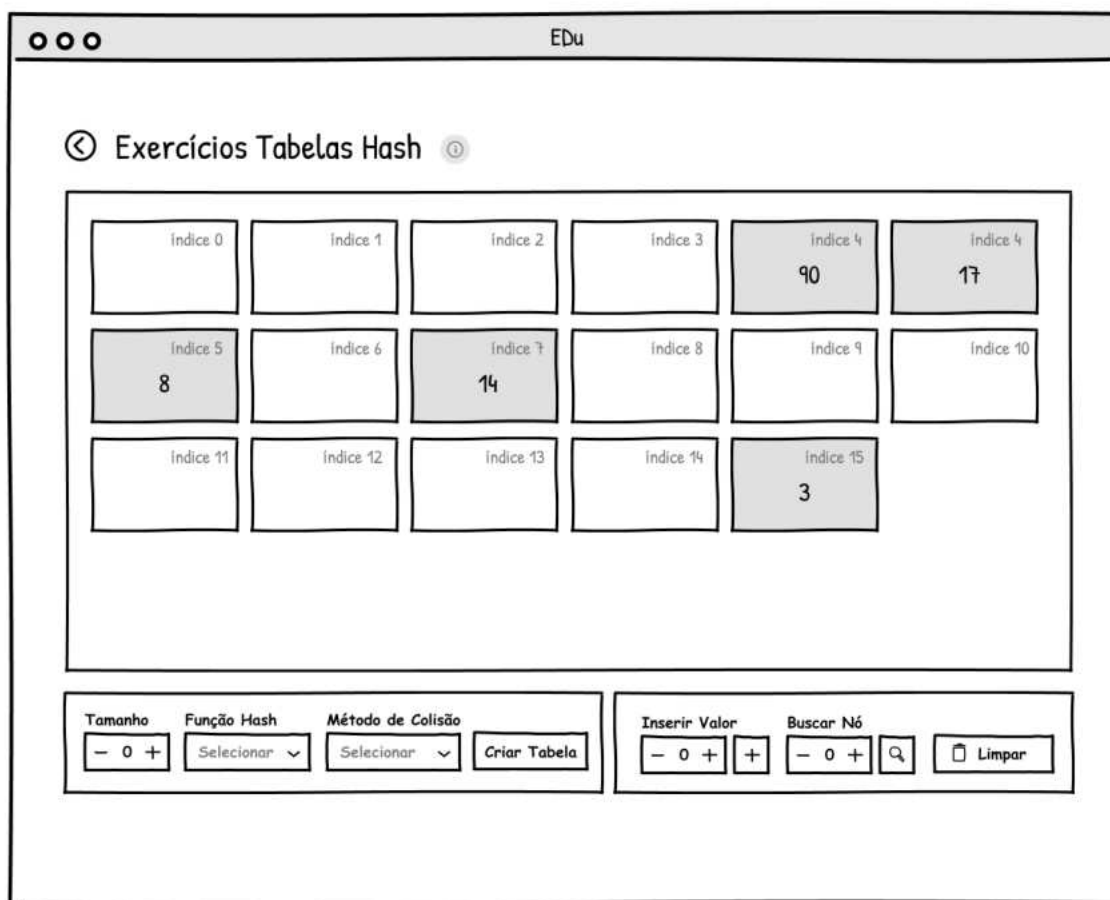


Figura 26 – Wireframe: Página do Simulador de Tabelas *Hash*

Nesta página, o usuário interage com um simulador de Tabelas *Hash* para explorar seus conceitos. A interface permite a configuração de parâmetros e a execução de operações comuns.

APÊNDICE E – JSON SCHEMAS

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": { "type": "string" },
      "name": { "type": "string" },
      "description": { "type": "string" },
      "classes": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "id": { "type": "string" },
            "name": { "type": "string" },
            "description": { "type": "string" },
            "topics": {
              "type": "array",
              "items": {
                "type": "object",
                "properties": {
                  "id": { "type": "string" },
                  "name": { "type": "string" },
                  "content": {
                    "type": "array",
                    "items": {
                      "type": "object",
                      "properties": {
                        "type": { "type": "string", "enum": ["text", "image"] },
                        "value": { "type": "string" }
                      },
                      "required": ["type", "value"]
                    }
                  }
                }
              }
            }
          },
          "required": ["id", "name", "content"]
        }
      }
    },
    "required": ["id", "name", "description", "topics"]
  },
  "required": ["id", "name", "description", "classes"]
}

```

Código 1 – *Schema* dos Cursos (Autoria Própria)

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": { "type": "string" },
      "name": { "type": "string" },
      "exercises": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "id": { "type": "string" },
            "type": { "type": "string", "enum": ["quiz", "single"] },
            "question": { "type": "string" },
            "options": {
              "type": "array",
              "items": {
                "type": "object",
                "properties": {
                  "id": { "type": "string" },
                  "text": { "type": "string" },
                  "correct": { "type": "boolean" }
                },
                "required": ["id", "text", "correct"]
              }
            }
          },
          "required": ["id", "text", "correct"]
        }
      },
      "answers": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "type": { "type": "string", "enum": ["text", "image"] },
            "value": { "type": "string" }
          },
          "required": ["type", "value"]
        }
      }
    },
    "required": ["id", "type"]
  }
},
"required": ["courseId", "exercises"]
}

```

Código 2 – *Schema* dos Exercícios (Autoria Própria)

APÊNDICE F – DESIGN FINAL E TELAS DESENVOLVIDAS

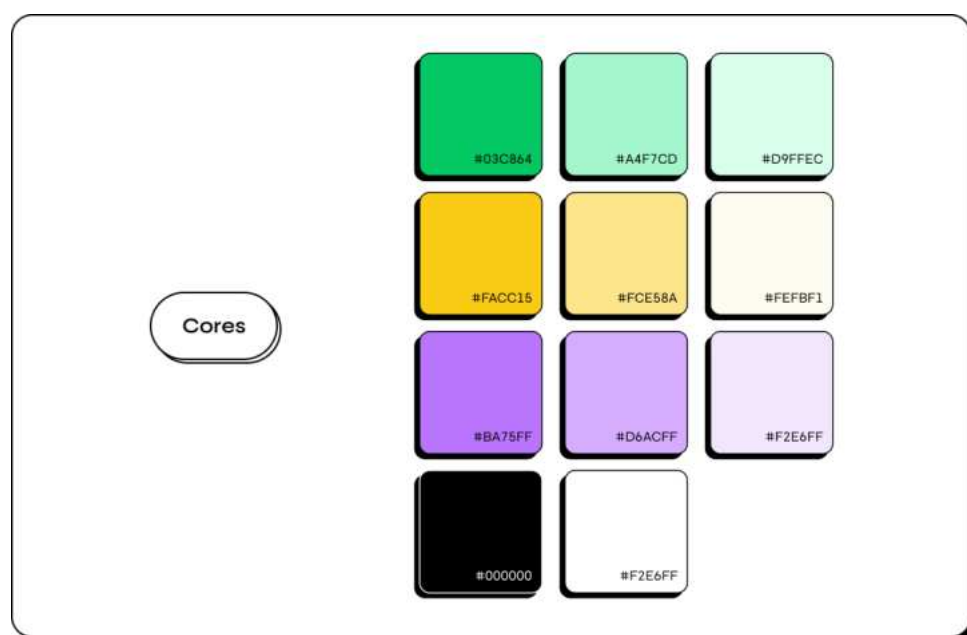


Figura 27 – Paleta de cores (Autoria Própria)

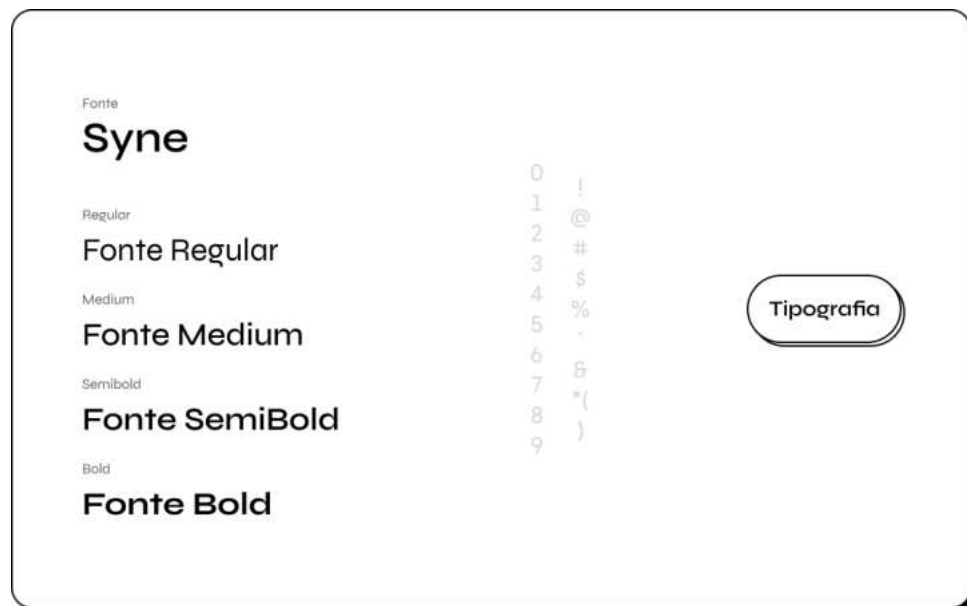


Figura 28 – Tipografia (Autoria Própria)

1Cursos finalizados

3Cursos em andamento

3h50mTempo estudado

Todos os cursos

Em andamento

Finalizados

Em andamento

```
graph TD; 26((26)) --- 12((12)); 26 --- 48((48)); 48 --- 31((31)); 48 --- 67((67));
```

Árvores Binárias

3h 20m 8 módulos

20%

Finalizado

0	Winter
1	Summer
2	Spring
3	Autumn

Tabelas Hash

2h 4 módulos

Ferramentas

Árvores Binárias

Simulador de árvores binárias


→

Tabelas Hash

Simulador de tabelas hash

→

Figura 29 – Página inicial (Autoria Própria)



Lorem Ipsum Dolor Sit

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis rhoncus hendrerit arcu sed efficitur. Integer ac efficitur est. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Cras lectus lectus, rutrum quis justo quis, pharetra pellentesque mauris.

Conteúdo

Ver curso

1

Lorem ipsum dolor sit amet consectetur

^

2

Lorem ipsum dolor sit amet consectetur

^

3

Lorem ipsum dolor sit amet consectetur

^

4

Lorem ipsum dolor sit amet consectetur

^

5

Lorem ipsum dolor sit amet consectetur

^

6

Lorem ipsum dolor sit amet consectetur

^

7

Lorem ipsum dolor sit amet consectetur

^

8

Lorem ipsum dolor sit amet consectetur

^

Pratique ainda mais

Coloque em prática o que você aprendeu com os exercícios do curso.

Acessar exercícios




Figura 30 – Tela de informações sobre o curso (Autoria Própria)

Lorem ipsum dolor sit amet, consectetur

Lorem ipsum dolor sit amet, consectetur: Lorem ipsum dolor sit amet, consectetur

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis rhoncus hendrerit arcu sed efficitur. Integer ac efficitur est. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Cras lectus lectus, rutrum quis justo quis, pharetra pellentesque mauris. Phasellus eget semper diam, vel sollicitudin libero. Aliquam sollicitudin neque elit, id accumsan leo gravida et. Nunc rhoncus arcu purus, in vehicula magna ornare non. Pellentesque sem orci, luctus ac urna ac, dictum laoreet nibh. Integer eget turpis sed ante fringilla suscipit. Duis neque dolor, commodo et tincidunt sit amet, varius at ante. Ut quis sapien sit amet nunc pellentesque accumsan. Ut urna velit, malesuada sit amet dictum quis, pellentesque et dui. Proin molestie enim sed consequat tincidunt. Donec vel tempus erat. Nunc semper laoreet finibus. Proin posuere sit amet tortor non egestas. Nulla ac dignissim erat, ultricies volutpat augue. Donec ultrices, urna in dapibus aliquet, massa lectus viverra tortor, lobortis venenatis odio tortor vitae tortor. Curabitur quis arcu quis felis placerat facilisis. Curabitur quis aliquam dui. Nulla egestas, nulla condimentum semper aliquet, felis arcu viverra lectus, a facilisis nulla quam at nisi. Nullam ut urna a sapien elementum elementum.

Praesent lectus dolor, tincidunt vitae dui ornare, tincidunt viverra elit. Duis eu porta felis. Suspendisse potenti. Phasellus aliquet urna quam, a viverra purus pulvinar tincidunt. Nulla massa elit, rutrum ac accumsan aliquam, dictum non turpis. Duis et nunc nunc. Sed mollis vel tortor at sodales. Donec eget enim lacinia, sollicitudin purus non, pulvinar augue. Donec nec tempor tortor. Etiam porta nisi ut diam condimentum laoreet. Ut mollis tincidunt magna efficitur semper. Duis a dolor ultricies, vehicula lorem non, vehicula massa. Quisque sodales purus ut lorem viverra egestas. Mauris leo risus, elementum non posuere ac, cursus a dolor. Phasellus lobortis id ipsum quis commodo. Sed sodales nisl sapien, ac malesuada sapien fringilla non, Mauris fringilla venenatis dolor a tincidunt. Sed sit amet congue lectus. Aliquam velit augue, rutrum ut magna vitae, viverra finibus elit. Morbi fermentum nisl urna, vitae eleifend felis maximus non. Donec consequat erat non magna efficitur, sit amet ultrices velit lacinia, Praesent tincidunt eu mauris quis scelerisque. Pellentesque luctus turpis eu dui sodales, ut tristique velit pellentesque. Praesent eleifend dui et felis viverra, vitae fermentum nisl fringilla. Mauris eu elit a purus egestas volutpat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

[Ir para o próximo item](#)

[Reportar problema](#)

67%

Aula 1
 Lorem ipsum dolor sit amet, consectetur

✓

Lorem ipsum dolor sit amet, consectetur

5 min

✓

Lorem ipsum dolor sit amet, consectetur

8 min

✓

Lorem ipsum dolor sit amet, consectetur

12 min

✓

Lorem ipsum dolor sit amet, consectetur

6 min

Lorem ipsum dolor sit amet, consectetur

2 min

Lorem ipsum dolor sit amet, consectetur

5 min

0%

Aula 2
 Lorem ipsum dolor sit amet, consectetur

0%

Aula 3
 Lorem ipsum dolor sit amet, consectetur

0%

Aula 4
 Lorem ipsum dolor sit amet, consectetur

0%

Aula 5
 Lorem ipsum dolor sit amet, consectetur

0%

Aula 6
 Lorem ipsum dolor sit amet, consectetur


0%

Aula 7
 Lorem ipsum dolor sit amet, consectetur

0%

Aula 8
 Lorem ipsum dolor sit amet, consectetur

Figura 31 – Tela de progressão do curso (Autoria Própria)

 **Exercícios de Tabelas Hash**

Exercício 3

Analise a seguinte afirmação: "O fator de carga de qualquer tabela hash é no máximo 1". Essa afirmação é verdadeira ou falsa?

A

 Verdadeiro

B

 Falso

Responder

1

2

3

Figura 32 – Tela de exercícios (Autoria Própria)

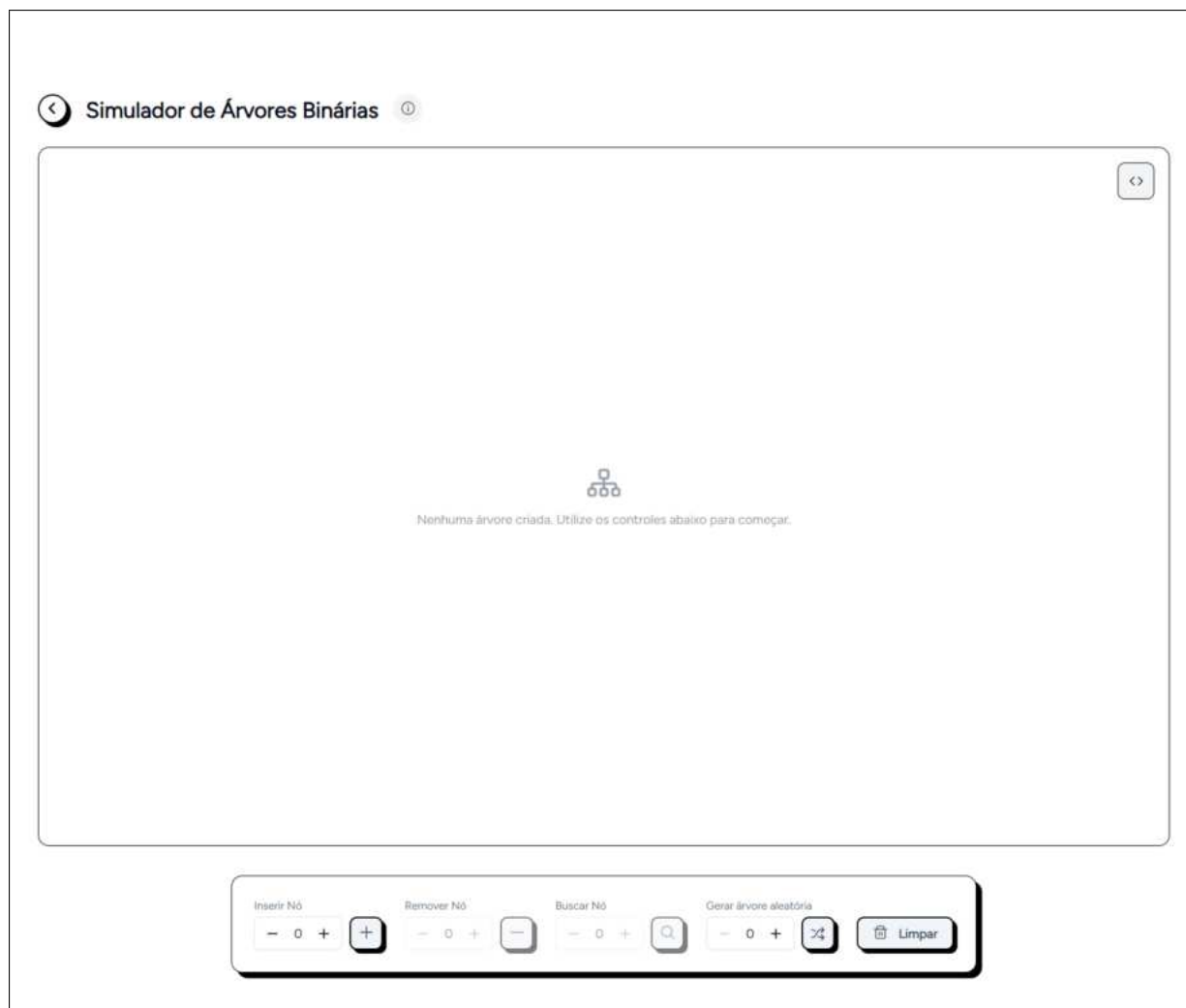


Figura 33 – Simulador de Árvores Binárias - *Empty State* (Autoria Própria)

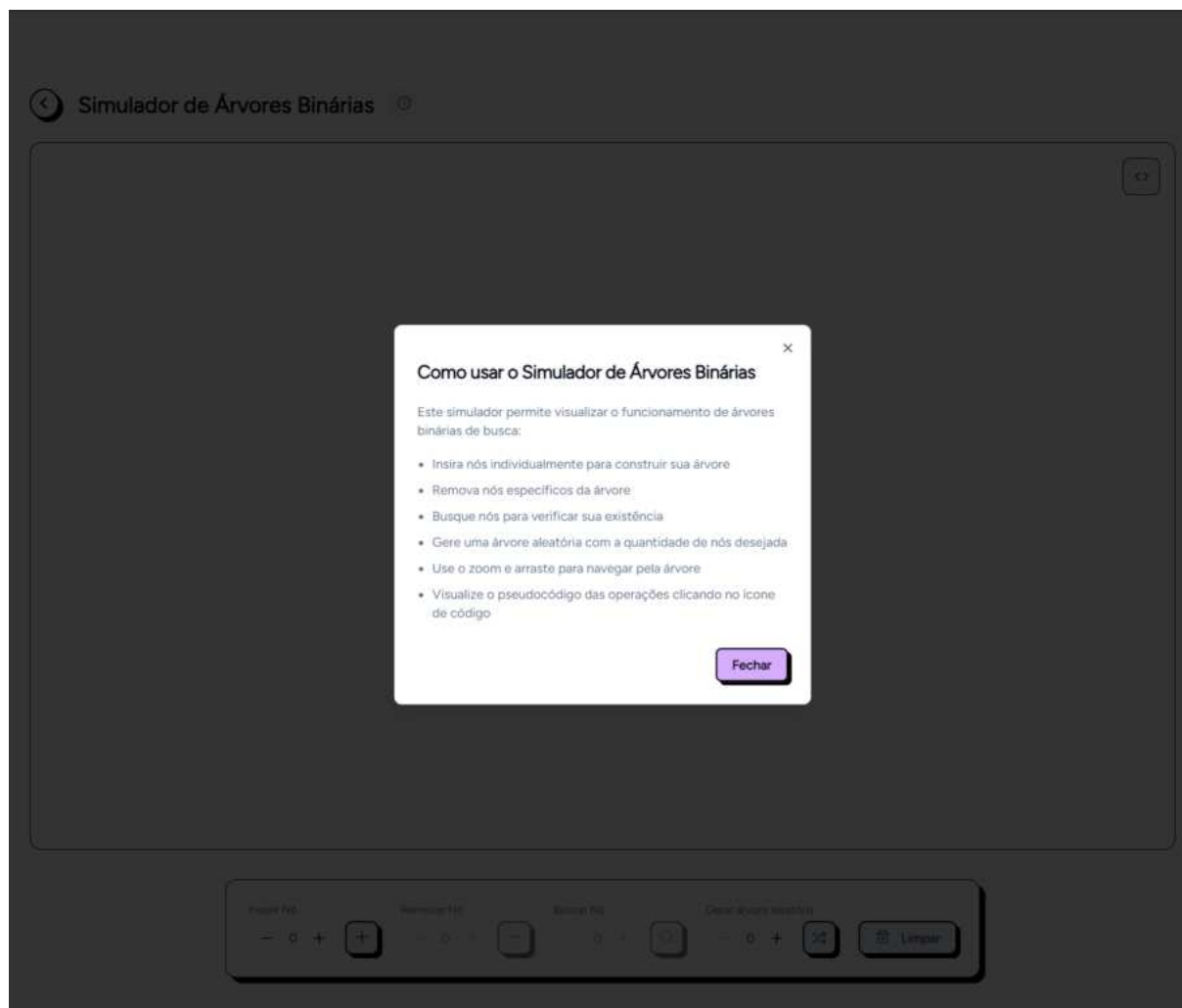
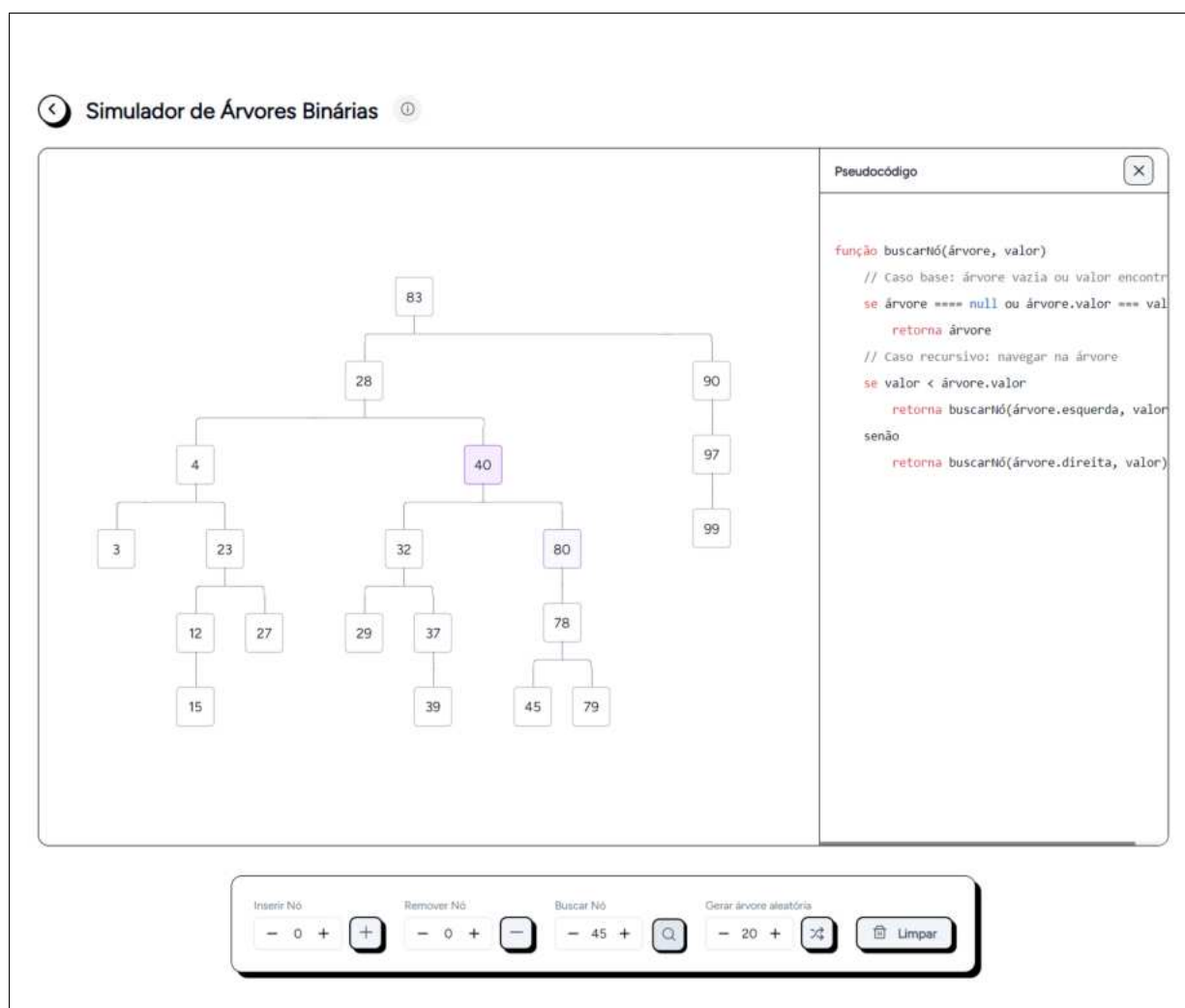


Figura 34 – Modal de Explicação - Simulador de Árvores Binárias (Autoria Própria)



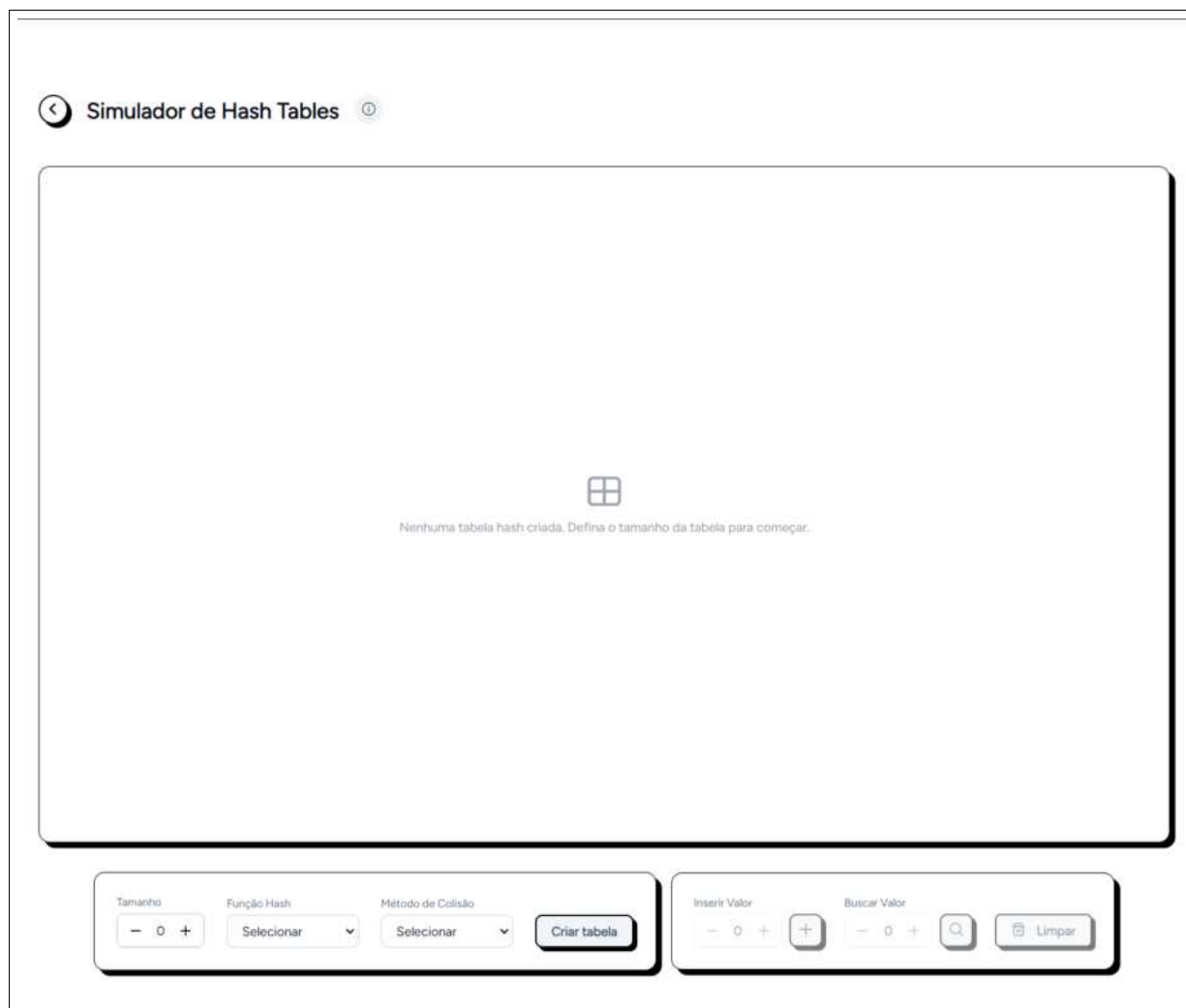


Figura 36 – Simulador de Tabelas *Hash* - *Empty State* (Autoria Própria)

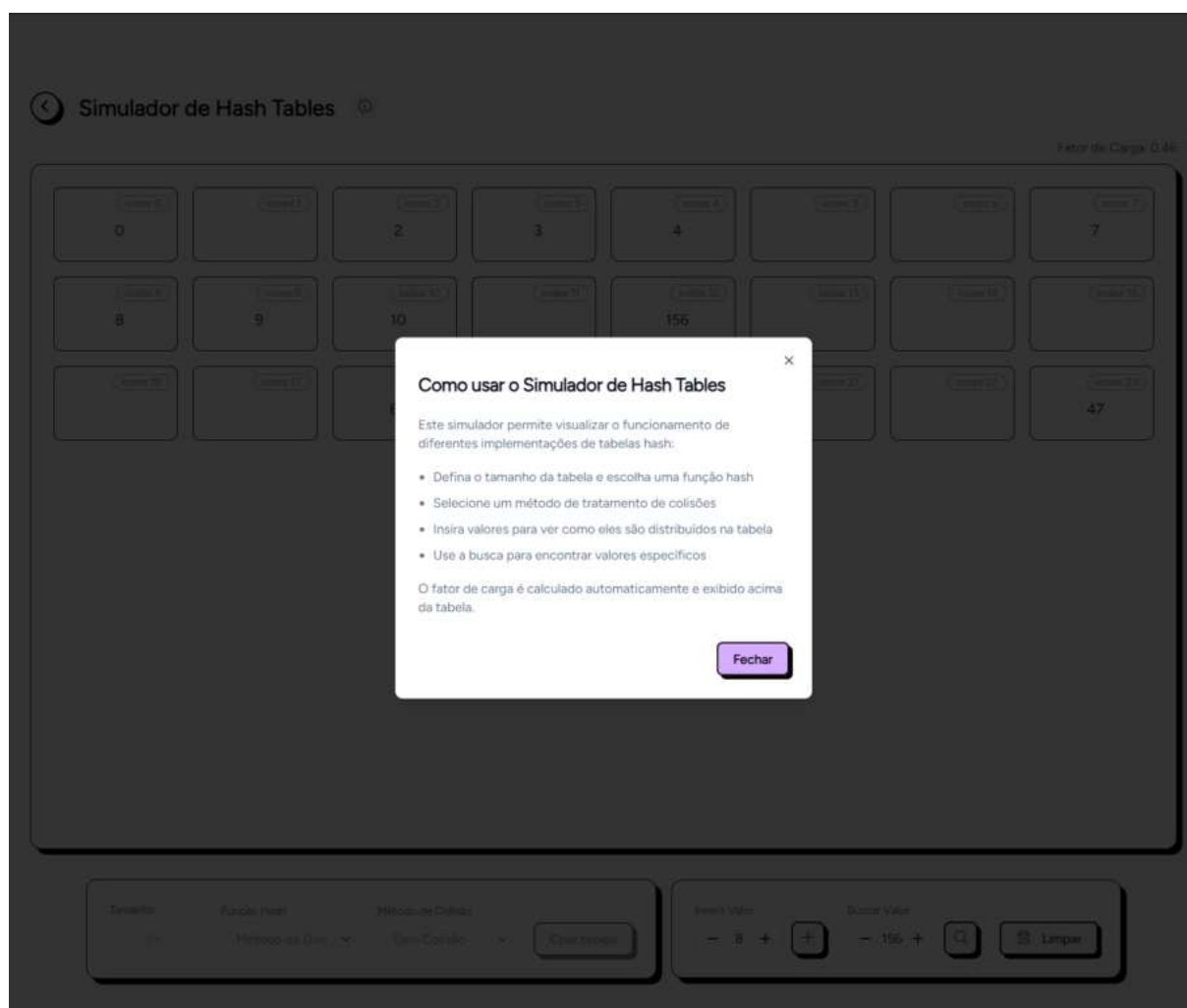




Figura 37 – Modal de Explicação - Simulador de Tabelas *Hash* (Autoria Própria)

 **Simulador de Hash Tables** 

Fator de Carga: 0.46

index 0 0	index 1	index 2 2	index 3 3	index 4 4	index 5	index 6	index 7 7
index 8 8	index 9 9	index 10 10	index 11	index 12 156	index 13	index 14	index 15
index 16	index 17	index 18 66	index 19	index 20	index 21	index 22	index 23 47

Tamanho
- 24 +

Função Hash:
Método da Divi

Método de Colisão
Sem Colisão

Criar tabela

Insertar Valor
- 8 +

Buscar Valor
- 156 +

Limpar

Figura 38 – Simulador de Tabelas *Hash* (Autoria Própria)

APÊNDICE G – FORMULÁRIO DE PESQUISA DE USABILIDADE

O formulário a seguir foi divulgado para obter avaliações sobre a usabilidade da plataforma desenvolvida, com foco nas funcionalidades de simuladores, exercícios e cursos.

Simulador de Árvores Binárias

1. Como você avalia a clareza da interface para realizar operações (inserção, remoção, busca, gerar árvore aleatória)?

Totalmente insatisfatório [1] [2] [3] [4] [5] Totalmente satisfatório

2. Como você avalia o feedback visual fornecido após cada operação?

Totalmente insatisfatório [1] [2] [3] [4] [5] Totalmente satisfatório

3. Como você avalia a facilidade de manipular os elementos na árvore?

Totalmente difícil [1] [2] [3] [4] [5] Totalmente fácil

4. Como você avalia a organização visual da representação gráfica da árvore?

Totalmente insatisfatório [1] [2] [3] [4] [5] Totalmente satisfatório

Simulador de Tabelas *Hash*

1. Como você avalia a clareza dos comandos para criar uma Tabela *Hash*?

Totalmente insatisfatório [1] [2] [3] [4] [5] Totalmente satisfatório

2. Como você avalia a clareza dos comandos para realizar operações (inserção, remoção e busca)?

Totalmente insatisfatório [1] [2] [3] [4] [5] Totalmente satisfatório

3. Como você avalia a legibilidade da tabela exibida?

Totalmente insatisfatório [1] [2] [3] [4] [5] Totalmente satisfatório

4. Como você avalia a rapidez e clareza do feedback após cada operação?

Totalmente insatisfatório [1] [2] [3] [4] [5] Totalmente satisfatório

Cursos

1. Como você avalia a facilidade para localizar um módulo ou aula específica?

Totalmente difícil [1] [2] [3] [4] [5] Totalmente fácil

2. Como você avalia a facilidade para navegar entre as aulas (avançar, voltar, pular)?

Totalmente difícil [1] [2] [3] [4] [5] Totalmente fácil

3. Como você avalia o suporte para localizar e retomar o ponto onde parou?

Totalmente insatisfatório [1] [2] [3] [4] [5] Totalmente satisfatório

4. Como você avalia o suporte para acompanhar seu progresso ao longo do curso?

Totalmente insatisfatório [1] [2] [3] [4] [5] Totalmente satisfatório

Exercícios

1. Como você avalia a facilidade de uso da ferramenta de exercícios?

Totalmente difícil [1] [2] [3] [4] [5] Totalmente fácil

2. Como você avalia o esforço necessário para concluir um exercício?

Muito alto [1] [2] [3] [4] [5] Muito baixo

3. Como você avalia o grau de controle que teve sobre o andamento do exercício (por exemplo, voltar, avançar)?

Totalmente insatisfatório [1] [2] [3] [4] [5] Totalmente satisfatório

Avaliação Geral

1. Como você avalia a facilidade de uso da plataforma como um todo?

Totalmente difícil [1] [2] [3] [4] [5] Totalmente fácil

2. Como você avalia a eficiência com que conseguiu realizar as tarefas desejadas?

Totalmente ineficiente [1] [2] [3] [4] [5] Totalmente eficiente

3. Como você avalia o esforço necessário para aprender a usar a plataforma?

Muito alto [1] [2] [3] [4] [5] Muito baixo

4. Por favor, deixe abaixo quaisquer comentários, sugestões ou críticas sobre a sua experiência de uso da plataforma:

APÊNDICE H – RESULTADOS DA PESQUISA DE USABILIDADE

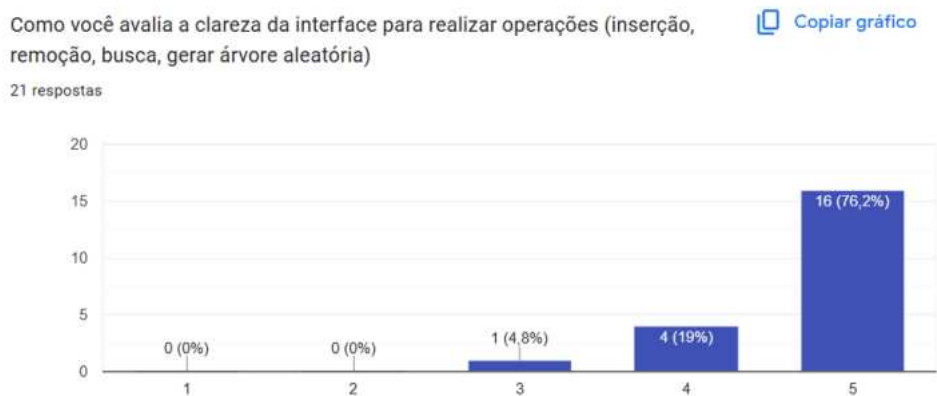


Figura 39 – Respostas Pesquisa de Usabilidade: Como você avalia a clareza da interface para realizar operações (inserção, remoção, busca, gerar árvore aleatória)

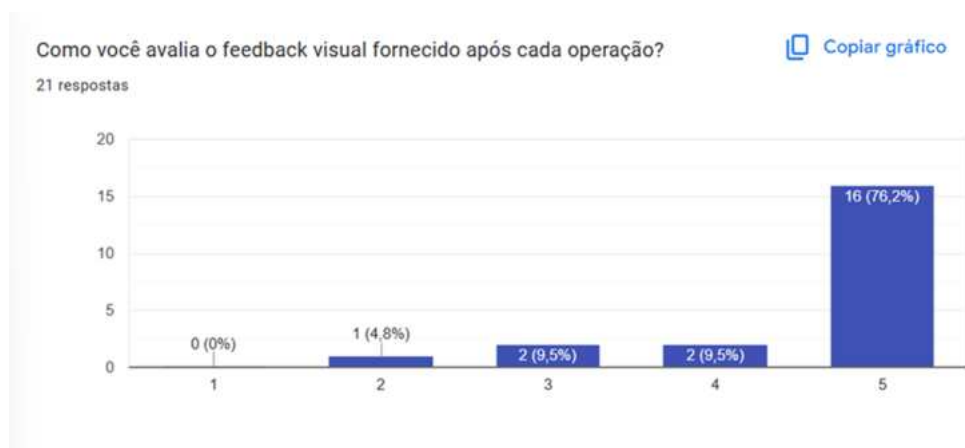


Figura 40 – Respostas Pesquisa de Usabilidade: Como você avalia o feedback visual fornecido após cada operação?

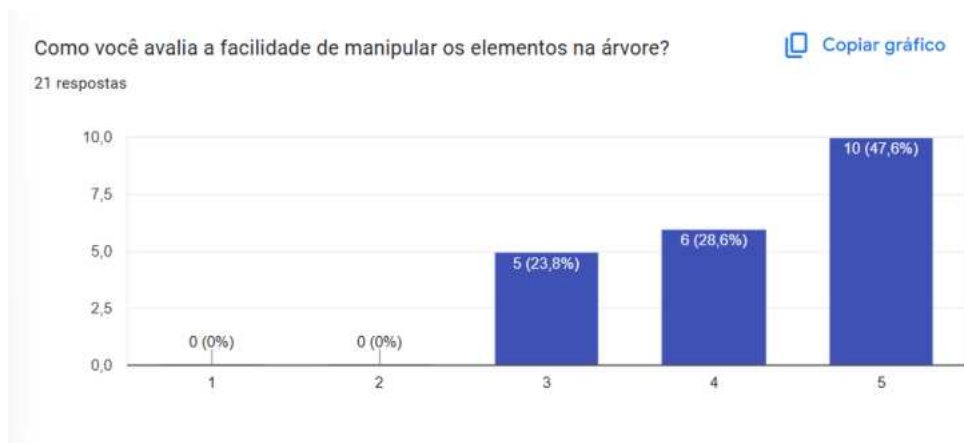


Figura 41 – Respostas Pesquisa de Usabilidade: Como você avalia a facilidade de manipular os elementos na árvore?

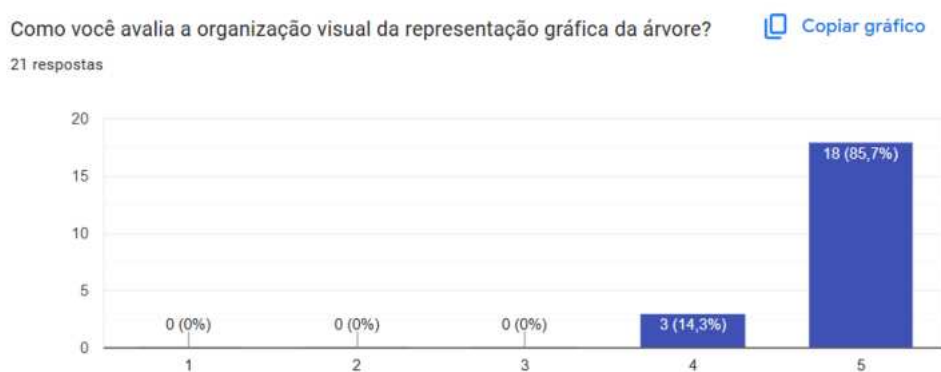


Figura 42 – Respostas Pesquisa de Usabilidade: Como você avalia a organização visual da representação gráfica da árvore?



Figura 43 – Respostas Pesquisa de Usabilidade: Como você avalia a clareza dos comandos para criar uma Tabela Hash?

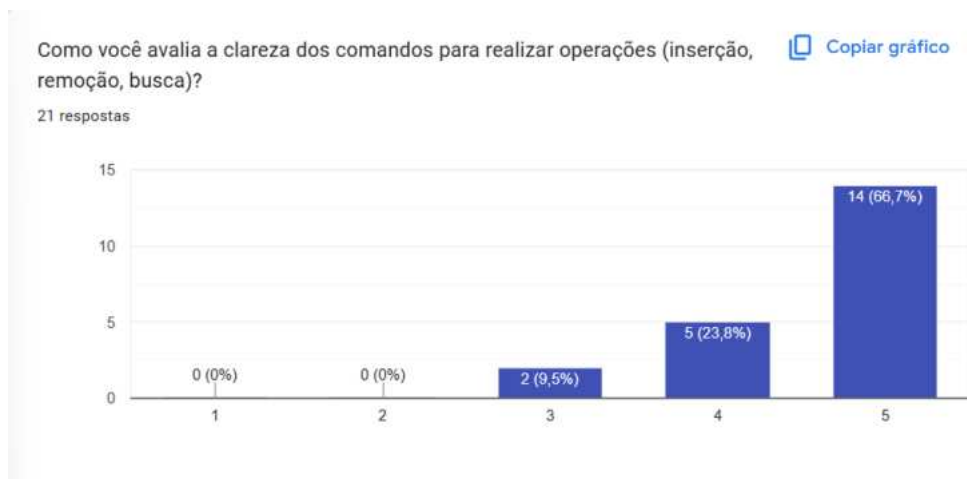


Figura 44 – Respostas Pesquisa de Usabilidade: Como você avalia a clareza dos comandos para realizar operações (inserção, remoção, busca)?

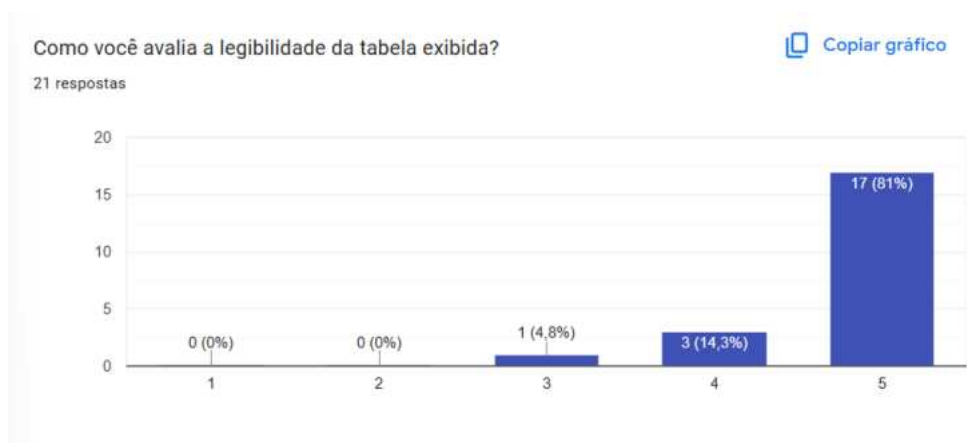


Figura 45 – Respostas Pesquisa de Usabilidade: Como você avalia a legibilidade da tabela exibida?

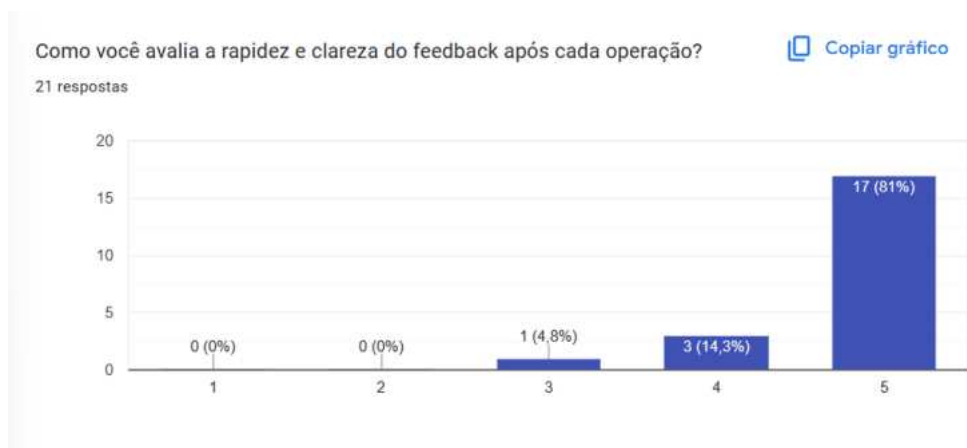


Figura 46 – Respostas Pesquisa de Usabilidade: Como você avalia a rapidez e clareza do feedback após cada operação?

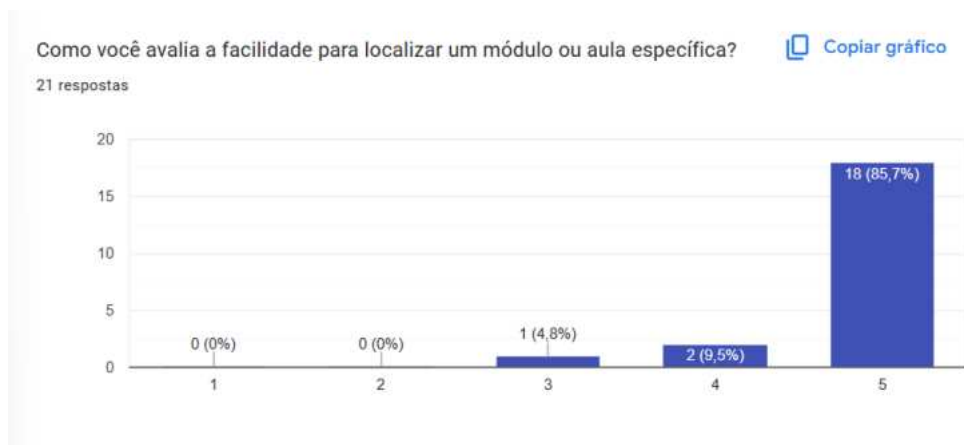


Figura 47 – Respostas Pesquisa de Usabilidade: Como você avalia a facilidade para localizar um módulo ou aula específica?

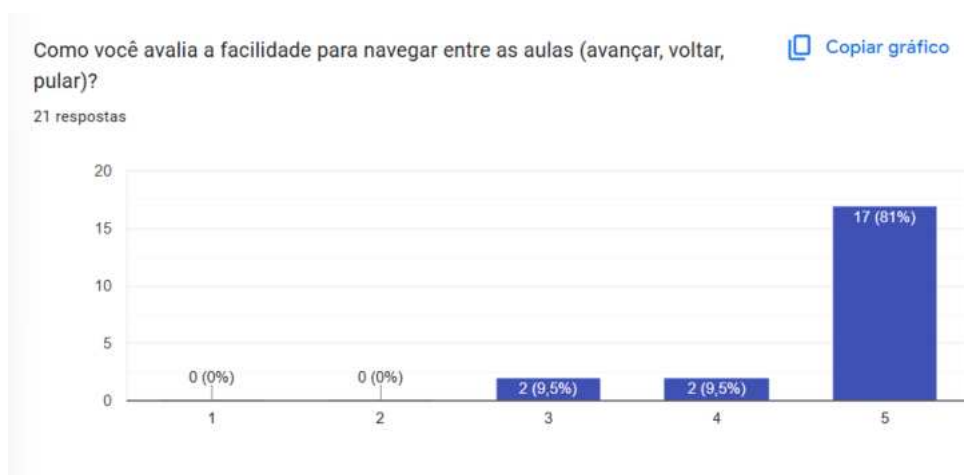


Figura 48 – Respostas Pesquisa de Usabilidade: Como você avalia a facilidade para navegar entre as aulas (avançar, voltar, pular)?

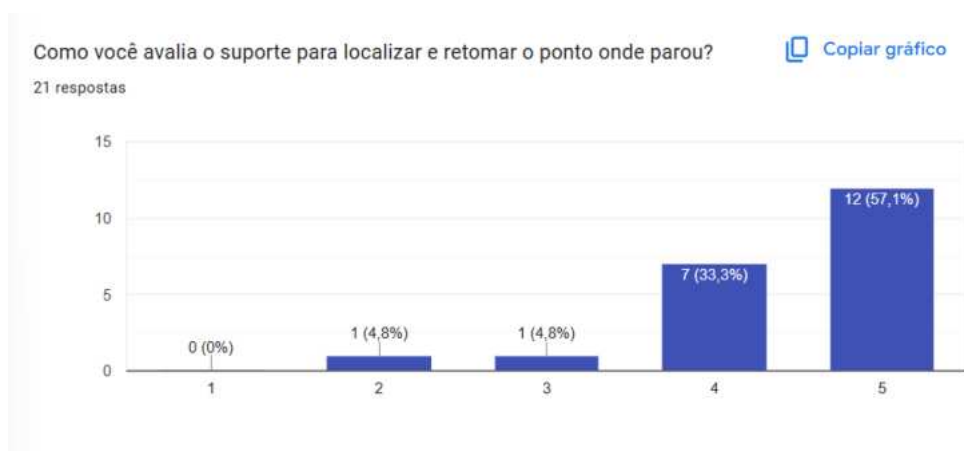


Figura 49 – Respostas Pesquisa de Usabilidade: Como você avalia o suporte para localizar e retomar o ponto onde parou?

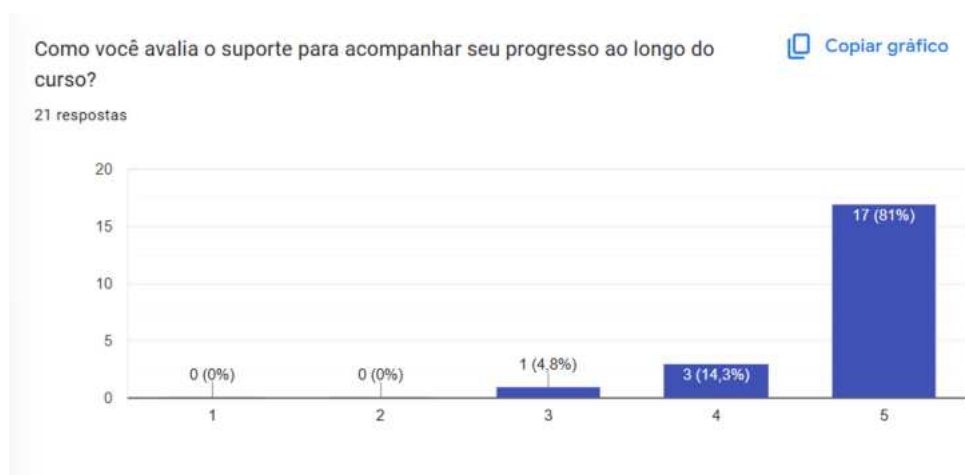


Figura 50 – Respostas Pesquisa de Usabilidade: Como você avalia o suporte para acompanhar seu progresso ao longo do curso?

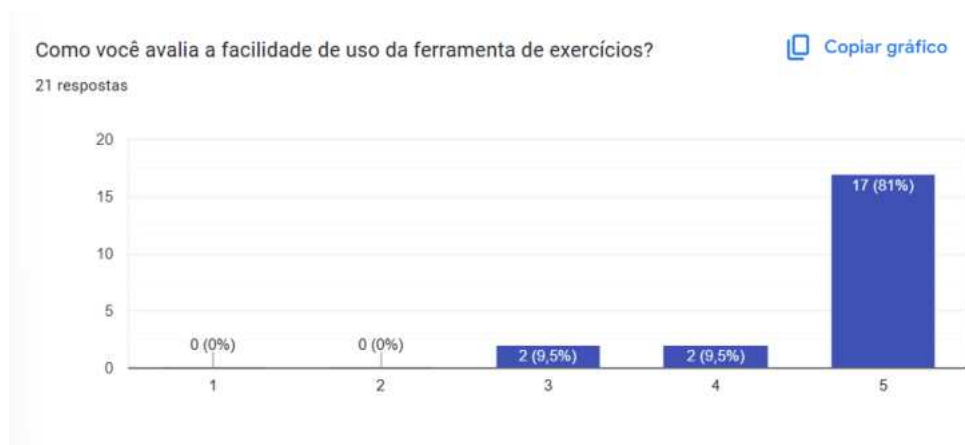


Figura 51 – Respostas Pesquisa de Usabilidade: Como você avalia a facilidade de uso da ferramenta de exercícios?

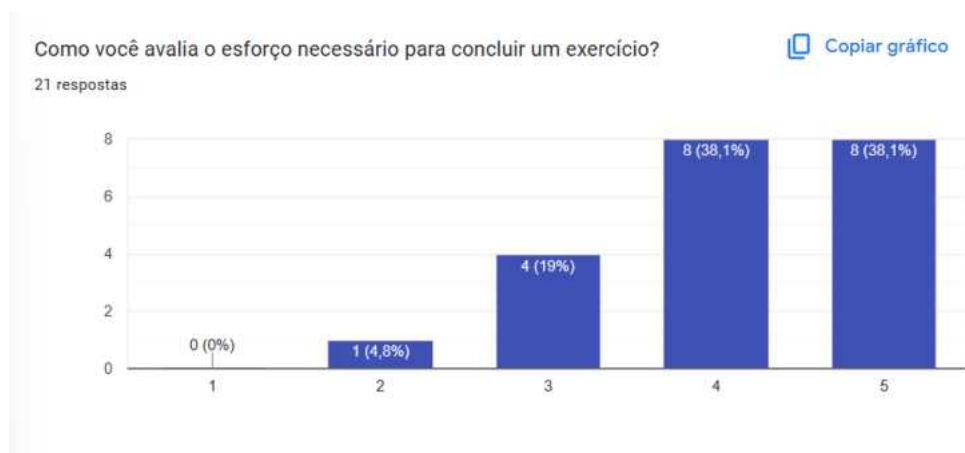


Figura 52 – Respostas Pesquisa de Usabilidade: Como você avalia o esforço necessário para concluir um exercício?

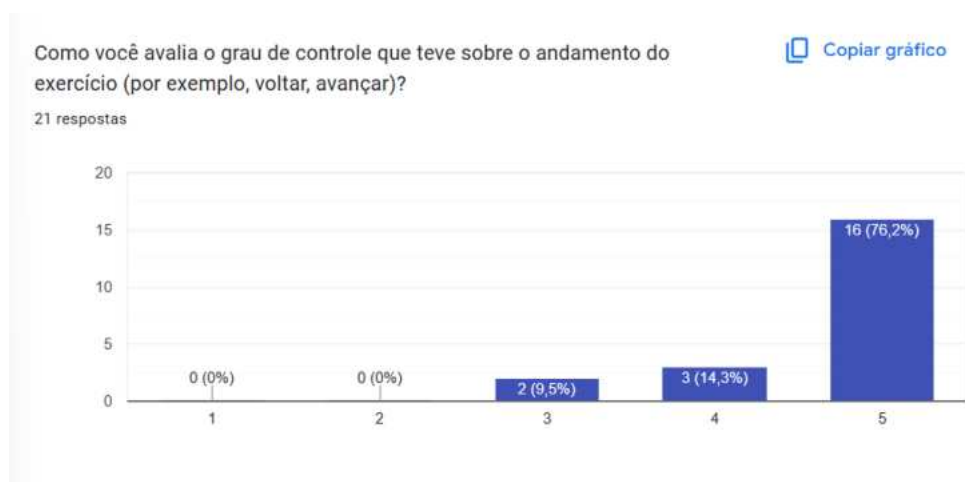


Figura 53 – Respostas Pesquisa de Usabilidade: Como você avalia o grau de controle que teve sobre o andamento do exercício (por exemplo, voltar, avançar)?

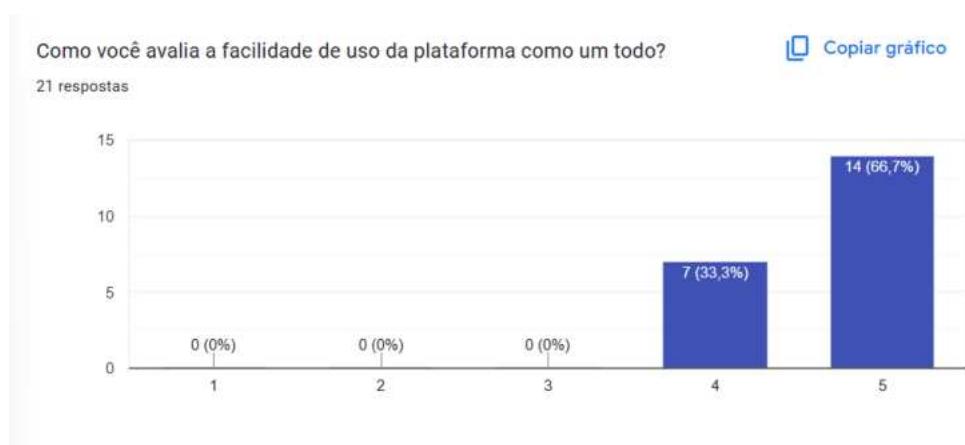


Figura 54 – Respostas Pesquisa de Usabilidade: Como você avalia a facilidade de uso da plataforma como um todo?



Figura 55 – Respostas Pesquisa de Usabilidade: Como você avalia a eficiência com que conseguiu realizar as tarefas desejadas?

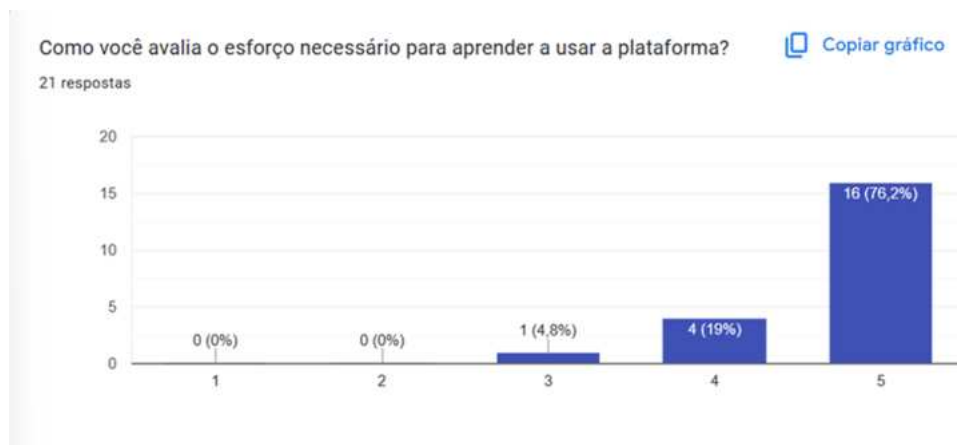


Figura 56 – Respostas Pesquisa de Usabilidade: Como você avalia o esforço necessário para aprender a usar a plataforma?