UNIVERSIDADE FEDERAL DE SANTA CATARINA

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Fábio Henrique Antunes Coelho

JOGO DIGITAL PARA AUXÍLIO À APRENDIZAGEM DE ESTRUTURAS DE DADOS E ALGORITMOS DE ORDENAÇÃO

Florianópolis

2025

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

JOGO DIGITAL PARA AUXÍLIO À APRENDIZAGEM DE ESTRUTURAS DE DADOS E ALGORITMOS DE ORDENAÇÃO

Fábio Henrique Antunes Coelho

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Ciências da Computação pela Universidade Federal de Santa Catarina.

Orientador: Prof. Dr. José Eduardo de Lucca

Florianópolis - SC

2025 / 2

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC. Dados inseridos pelo próprio autor

Coelho, Fábio Henrique Antunes
JOGO DIGITAL PARA AUXÍLIO À APRENDIZAGEM DE ESTRUTURAS DE DADOS E ALGORITMOS DE ORDENAÇÃO / Fábio Henrique Antunes Coelho ; orientador, José Eduardo de Lucca, 2025.
82 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Ciências da Computação, Florianópolis, 2025.

Inclui referências.

1. Ciências da Computação. 2. Jogo Educativo. 3. Algoritmos de Ordenação. 4. Estrutura de Dados. 5. Gamificação. I. Lucca, José Eduardo de. II. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. III. Título.

Fábio Henrique Antunes Coelho

JOGO DIGITAL PARA AUXÍLIO À APRENDIZAGEM DE ESTRUTURAS DE DADOS E ALGORITMOS DE ORDENAÇÃO

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Ciências da Computação.

Orientador(a):

_____

Prof. Dr. José Eduardo de Lucca

Banca examinadora:

_____

Prof. Dr. Alexandre Gonçalves Silva

_____

Prof. Dr. Jean Carlo Rossa Hauck

AGRADECIMENTOS

Agradeço primeiramente à minha família: meus pais, Nelson e Cláudia, e meu irmão, Alisson, pelo incentivo, suporte emocional e por todo o apoio fundamental durante a minha trajetória acadêmica.

À Ana, minha companheira, por todo o apoio durante o curso e por me ajudar nos momentos de ansiedade desse trabalho. Foi graças à isso que cheguei até aqui sem desistir, apesar dos frequentes pensamentos.

Ao orientador deste trabalho, Prof. Dr. José Eduardo de Lucca, pela paciência e por todo o conhecimento compartilhado que guiou o desenvolvimento deste trabalho.

Aos membros da banca examinadora, Prof. Jean Carlo Rossa Hauck e Prof. Alexandre Gonçalves Silva, pela disponibilidade e contribuições para a melhoria deste projeto.

A todos os meus amigos, por terem feito essa jornada ser mais leve.

E por fim, agradeço aos colegas de curso, professores e comunidade UFSC no geral.

Sumário

1 Introdução **11**

1.1 Objetivo Geral 13

1.2 Objetivos específicos 13

2 Fundamentação Teórica **14**

2.1 Estruturas de Dados 14

2.2 A Evolução dos Jogos e Seus Impactos na Sociedade Contemporânea . . 15

2.3 Gamificação 16

2.3.1 Meio de Aplicação da Gamificação 18

2.3.2 Mecânicas, dinâmicas e estéticas 19

2.4 Serious games 20

2.4.1 Jogos para educação 21

2.5 Esquema de Classificação para Gamificação em Ciências da Computação 23

2.5.1 Gêneros Identificados de Gamificação em Estruturas de Dados . . 23

2.5.2 Novos Gêneros Abstratos Propostos 24

2.6 Documentação de Design de Jogos 26

2.6.1 Short Game Design Document 29

3 Trabalhos Correlatos **32**

3.1 Jogos Educacionais Para Computação 32

3.2 Jogos Educacionais Para Estruturas Dados 34

4 Proposta **40**

4.1 Objetivo Pedagógico e Público-Alvo 40

4.2 Mecânicas e Regras 41

4.3 Análise e Escolha das Tecnologias 41
4.4 Short Game Design Document 42
4.5 Prototipação de Telas 44

5 SORTilege 49

5.1 Introdução 49
5.2 Desenvolvimento do Software 53
5.3 Estrutura Utilizada 54
5.4 Problemas e Soluções Encontradas 55

6 AVALIAÇÃO DO JOGO 57

6.1 O Modelo MEEGA+ 57
6.2 Execução 58
6.3 Resultados 59

7 CONSIDERAÇÕES FINAIS 63

REFERÊNCIAS 65

Lista de Figuras

1 Trecho do jogo Stack Game mostrando a avaliação de uma expressão pós-fixa 35
2 Captura de tela do jogo Space Traveler 37
3 Lista de atividades a serem desenvolvidas 44
4 Protótipo do Menu Principal 45
5 Protótipo da tela de combate 46
6 Protótipo Da tela de obstáculo - Torre de Hanoi 47
7 Protótipo da tela final 48
8 Menu principal 49
9 Cena de combate 50
10 Cena da Torre de Hanoi 51
11 Cena da Lista 51
12 Cena Desafio da Árvore 52
13 Tela final 53
14 Gráfico de frequência de jogo 60
15 Gráfico de Usabilidade 60
16 Gráfico de Experiência do Jogador 62

RESUMO

O ensino de algoritmos e estruturas de dados é um pilar fundamental nos cursos de Ciência da Computação, porém, frequentemente apresenta dificuldades significativas para os alunos devido à natureza abstrata e complexa dos conceitos abordados. Para enfrentar esse desafio, tecnologias educacionais, como os jogos educativos, têm sido cada vez mais exploradas como ferramentas para facilitar o aprendizado. O presente trabalho propõe o desenvolvimento de um jogo digital educativo, intitulado SORTilege, como ferramenta de apoio à aprendizagem de algoritmos de ordenação e estruturas de dados. A metodologia consistiu na transposição das regras de ordenação para mecânicas de jogo, permitindo que o aluno visualize passo a passo o comportamento dos dados durante a execução dos algoritmos. O resultado é uma aplicação que visa aumentar o engajamento dos estudantes e facilitar a compreensão dos métodos de ordenação através da visualização gráfica e do *feedback* imediato, servindo como um complemento às metodologias tradicionais de ensino.

Palavras-chave: Jogo Educativo. Algoritmos de Ordenação. Estrutura de Dados. Gamificação. Jogos sérios.

ABSTRACT

The teaching of algorithms and data structures is a fundamental pillar in Computer Science courses; however, it often presents significant difficulties for students due to the abstract and complex nature of the concepts addressed. To address this challenge, educational technologies, such as educational games, have been increasingly explored as tools to facilitate learning. This work proposes the development of an educational digital game, titled SORTilege, as a support tool for learning sorting algorithms and data structures. The methodology consisted of transposing sorting rules into game mechanics, allowing the student to visualize step-by-step the behavior of data during the execution of the algorithms. The result is an application aimed at increasing student engagement and facilitating the understanding of sorting methods through graphical visualization and immediate feedback, serving as a complement to traditional teaching methodologies.

Keywords: Educational Game. Sorting Algorithms. Data Structures. Gamification. Serious Games.

1 Introdução

Nos últimos anos, o uso de dispositivos móveis, como *tablets* e, principalmente, *smartphones*, tem crescido muito. No Brasil, atualmente, a média de dispositivos móveis é de 1,8 por habitante, quando é considerado apenas a quantidade de *smartphones* a média é de 1,2 dispositivos por brasileiro (Fundação Getúlio Vargas, 2024). Como o uso desses aparelhos está difundido por toda a sociedade e usá-los já é algo cotidiano, é interessante utilizar estes aparelhos como ferramenta educacional, dando outra perspectiva sobre o assunto visto em sala de aula, podendo levar mais diversão na aprendizagem, desviando para a direção oposta da ansiedade e improdutividade (Okada; Sheehy, 2020).

A disciplina de Estruturas de Dados é um pilar na formação em Ciência da Computação, crucial para o desenvolvimento de habilidades de programação e resolução de problemas. No entanto, a metodologia de ensino predominante, que se baseia em aulas expositivas e exercícios práticos, tem se mantido inalterada por décadas, negligenciando o potencial de dispositivos tecnológicos modernos, como os *smartphones* (Carvalho *et al*., 2018). Essa abordagem tradicional, aliada à natureza abstrata e complexa dos conceitos, e ao papel de transição da disciplina entre conteúdos básicos e avançados (Chinn; Prins; Tenenberg, 2003), frequentemente resulta em dificuldades significativas para os alunos, impactando seu desempenho acadêmico.

Para enfrentar esse desafio, as tecnologias educacionais, como os jogos educativos, têm sido cada vez mais exploradas como ferramentas eficazes para engajar e facilitar o aprendizado, afinal, elas podem trazer diversão, junto com a aprendizagem e aumentar a retenção do conteúdo ensinado (Roland *et al*., 2004). Os jogos podem criar ambientes interativos e imersivos, nos quais os alunos experimentam, aplicam e aprendem conceitos de forma prática, criando um vínculo afetivo, o que pode não acontecer com os métodos tradicionais, já que quando estão em aula, os alunos podem não alcançar um resultado satisfatório, criando um bloqueio no seu processo de aprendizagem, o que os afasta desse vínculo (Roland *et al*., 2004).

A sociedade contemporânea testemunha uma interação cada vez mais dinâmica com os avanços tecnológicos, que têm moldado e sido moldados por diversas práticas culturais, incluindo os jogos. A era digital, iniciada por volta da década de 1970, impulsionou uma explosão de jogos eletrônicos, que hoje integram múltiplas formas de arte e tecnologias gráficas avançadas, oferecendo uma vasta gama de experiências culturais (Rodrigues, 2009 *apud* Freire; Guerrini, 2016). Essa evolução transformou a indústria de jogos eletrônicos em um gigante econômico global, superando o faturamento da indústria cinematográfica (Bauman, 2013; Reis; Cavichiolli, 2008; Carvalho *et al*., 2014 *apud* Freire; Guerrini, 2016). No Brasil, a Pesquisa Game Brasil 2024 (PGB 2024) revelou que o hábito de consumir jogos digitais atingiu 73,9% da população, evidenciando a consistência e o alcance massivo desse mercado no país.

Nesse contexto, a gamificação surge como uma abordagem promissora. Ela não se define como a criação de um jogo completo, mas sim como a aplicação de elementos de jogos, como pontos, regras, *feedback* e participação voluntária, em contextos não relacionados a jogos, com o propósito de tornar atividades cotidianas mais envolventes e divertidas (Kim, 2015), como discutido na subseção 2.3. Outro conceito que surge neste contexto é o de jogos sérios e, mais especificamente, os jogos educacionais (Becker, 2021), categoria proeminente dentro dos *serious games*, como discutido na subseção 2.4.1.

A literatura acadêmica corrobora os múltiplos benefícios associados ao emprego de jogos na educação. Eles são capazes de aumentar a eficácia da aprendizagem, o interesse, a motivação e a persistência dos alunos. Por serem intrinsecamente divertidos, os jogos facilitam o aprendizado e elevam a capacidade de retenção do conteúdo, funcionando como uma “ginástica mental” (Roland *et al*., 2004). Um dos aspectos mais relevantes é a promoção da aprendizagem ativa, na qual os alunos se engajam em atividades como diálogo, resolução de problemas e pensamento crítico, em vez de serem meros receptores passivos de informação. Essa abordagem alinha-se diretamente com as teorias

construtivistas da educação, que valorizam a participação e a experimentação do sujeito na construção de seu próprio conhecimento. Especificamente, jogos digitais oferecem vantagens distintivas na simulação de sistemas complexos, na automação de regras e no fornecimento de *feedback* imediato e individualizado, tornando-os particularmente adequados para o ensino de conceitos abstratos (Roland *et al*., 2004).

1.1 Objetivo Geral

O presente trabalho tem como objetivo geral desenvolver um jogo digital educativo, alinhado ao gênero de Visualização de Ideias Abstratas (VAI) (Spanier; Harms; Hastings, 2021), para auxiliar na fixação de conceitos fundamentais da disciplina de Estruturas de Dados, como algoritmos de ordenação e estruturas vistos na disciplina, por meio de visualização interativa e aplicação prática em um ambiente lúdico e engajador.

1.2 Objetivos específicos

O1. Realizar um levantamento bibliográfico sobre a evolução dos jogos digitais, os conceitos de gamificação e *serious games*, e os modelos de documentação de design de jogos, com foco nas aplicações para o ensino de Ciência da Computação;

O2. Analisar trabalhos correlatos de jogos educacionais para o ensino de Estruturas de Dados, identificando suas abordagens, mecânicas e resultados;

O3. Conceber a arquitetura e as mecânicas de jogo;

O4. Implementar o jogo educacional;

O5. Avaliar a potencial eficácia do *serious game* no auxílio à compreensão e fixação dos conceitos de Estruturas de Dados;

2 Fundamentação Teórica

2.1 Estruturas de Dados

O aprendizado de conceitos fundamentais da Ciência da Computação, como os vistos na disciplina de Estruturas de Dados, é essencial para o desenvolvimento de habilidades de programação e resolução de problemas. O ensino dessa disciplina é realizado por meio de métodos expositivos, comumente baseados em aulas teóricas e exercícios práticos, sem atualizações há décadas e fazendo pouco ou nenhum aproveitamento de dispositivos tecnológicos atuais, como os *smartphones* (Carvalho et al., 2018).

O escopo da disciplina abrange desde conceitos de baixo nível, como a alocação dinâmica de memória e o uso de ponteiros, até a análise de algoritmos mais complexos. O conteúdo programático avança sobre a organização de dados em estruturas lineares, como listas, pilhas e filas, e explora também estruturas não lineares, como árvores (binárias, AVL, Rubro-Negra, B) e Tabelas de Espalhamento (*hash tables*). Além da implementação e manipulação dessas estruturas, a disciplina aborda tópicos cruciais como a análise de complexidade de algoritmos e o estudo de diferentes métodos de ordenação, que vão desde os de complexidade quadrática (Inserção, Seleção, Bolha) até os mais eficientes (Quicksort, Heapsort) (Von Wangenheim; Silva, 2025).

Muitos alunos enfrentam dificuldades para compreender plenamente esses conceitos devido à abstração e à complexidade envolvidas, além de a disciplina ser considerada um ponto de transição entre os conteúdos iniciais e os mais avançados em computação (Chinn; Prins; Tenenberg, 2003), o que pode afetar negativamente seu desempenho acadêmico.

2.2 A Evolução dos Jogos e Seus Impactos na Sociedade Contemporânea

A sociedade contemporânea é marcada por uma interação dinâmica com os avanços tecnológicos, que moldam e são moldados por diversas práticas culturais, incluindo os jogos. A tecnologia transformou o universo lúdico, desde os espaços de jogo e relações humanas até a produção cultural e a dinâmica dos jogos tradicionais (Freire; Guerrini, 2016).

A era digital, iniciada por volta da década de 1970, trouxe uma explosão de jogos eletrônicos, que integram múltiplas formas de arte (cinema, quadrinhos, música) e tecnologias gráficas avançadas, oferecendo uma diversidade de experiências culturais que transcendem os jogos populares tradicionais (Rodrigues, 2009 apud Freire; Guerrini, 2016).

Houve uma migração dos espaços públicos físicos (ruas, praças), antes centrais para os jogos populares, para o ciberespaço. Plataformas digitais e a internet globalizaram o ato de jogar, permitindo interações virtuais mediadas por avatares. Além do espaço, a interação também foi alterada, antes “homem-homem” evoluiu para “homem-máquina” e posteriormente, impulsionado pela popularização da internet, “homem-máquina-homem” (Primo, 2000; Rodrigues, 2009 *apud* Freire; Guerrini, 2016).

Atualmente a indústria de jogos eletrônicos tornou-se um gigante econômico global, superando o faturamento da indústria cinematográfica, impulsionada por avanços gráficos, jogabilidade diversificada e uma cultura de consumo constante de novidades (Bauman, 2013; Reis; Cavichiolli, 2008; Carvalho *et al.*, 2014 *apud* Freire; Guerrini, 2016). De acordo com a Pesquisa Game Brasil 2024 (PGB 2024), o hábito de consumir jogos digitais no Brasil atingiu 73,9% da população, um crescimento de 3,8 pontos percentuais em relação ao ano anterior, o que demonstra a consistência do mercado no país.

2.3 Gamificação

Segundo a autora Kim (2015), na revista *Understanding gamification*, gamificação não é a criação de um jogo completo, mas a aplicação de elementos de jogos (como pontos, regras, *feedback* e participação voluntária) em contextos não relacionados a jogos. O objetivo é tornar atividades cotidianas mais envolventes e divertidas.

Zichermann e Cunningham (2011), conforme citado por Kim (2015), definem gamificação como "o processo de pensamento de jogo e mecânicas de jogo para engajar usuários e resolver problemas". Esta definição é funcional, concentrando-se no propósito da gamificação: o engajamento do usuário e a solução de problemas práticos.

Uma perspectiva ligeiramente diferente é oferecida por Deterding *et al*. (2011), como destacado por Kim (2015), para quem a gamificação é "o uso de elementos de design de jogo característicos de jogos em contextos não-jogo". O diferencial aqui é a ênfase no "contexto não-jogo". A autora considera esta definição mais precisa para distinguir gamificação de jogos, pois "a natureza do problema que a gamificação tenta resolver não é ficcional, mas real". Para que uma aplicação seja considerada gamificação sob esta ótica, "seu objetivo deve ser resolver um problema do mundo real".

É importante diferenciar a gamificação de conceitos correlatos. Deterding *et al*. (2011) a situam como o uso de elementos de design de jogos em contextos não lúdicos. Isso a distingue dos "jogos propriamente ditos" (que são um todo em si) e do "*playful design*" (design divertido), que, embora busque o lúdico, não possui necessariamente regras explícitas e metas competitivas. Um exemplo clássico de "*playful design*" é a antiga imagem da "Fail Whale" do Twitter, que mitigava a frustração dos usuários durante quedas do sistema.

Existem, contudo, definições mais amplas. Kapp (2012), por exemplo, como observado por Kim (2015), define a gamificação no contexto da aprendizagem como a entrega de conteúdo com um propósito que não seja o puro entretenimento, utilizando o pensamento e a mecânica dos jogos. Nessa visão, não há uma distinção nítida entre gamifica-

ção, “*playful design*” e “*serious games*” (jogos sérios), desde que o objetivo principal não seja apenas a diversão.

A empresa de pesquisa Gartner (2014), conforme discutido por Kim (2015), por sua vez, refinou sua definição para “o uso de mecânicas de jogo e design de experiência para engajar e motivar digitalmente as pessoas a alcançarem seus objetivos”. Essa definição enfatiza o meio digital, sendo o formato de aplicativo móvel particularmente comum para a gamificação. No entanto, limitar a gamificação ao domínio digital é considerado restritivo por alguns pesquisadores, que argumentam que o design de jogos é uma categoria transmidiática. O fundamental na gamificação é o engajamento e o auxílio para que as pessoas atinjam seus objetivos na vida real por meio de elementos de jogos apropriados.

De forma ainda mais restritiva, Domínguez *et al*. (2013), como observado por Kim (2015), sugerem que a gamificação poderia ser definida como “incorporar elementos de jogo em uma aplicação de software não-jogo para aumentar a experiência e o engajamento do usuário”. Esta visão limita a gamificação estritamente ao domínio de aplicações de software.

A multiplicidade de definições evidencia um campo conceitual em maturação, buscando um consenso. As formulações variam em escopo (de mais amplas a mais restritas) e em foco (propósito, elementos, contexto, meio). Essas divergências indicam que o conceito de gamificação não é monolítico, mas um construto dinâmico. Apesar das variações, um fio condutor na maioria das definições, especialmente nas de Deterding *et al*. (2011) e Kapp (2012), é a noção de que a gamificação se aplica a contextos que não são jogos em sua essência e/ou serve a um propósito que transcende o puro entretenimento. Zichermann e Cunningham (2011) também focam na “resolução de problemas”, e Gartner (2014) em “alcançar objetivos”. Isso reforça a ideia de que a gamificação é, fundamentalmente, sobre a aplicação da lógica e dos elementos de jogos para atingir fins extrínsecos ao jogo em si. O entretenimento, nesse contexto, atua como um facilitador do engajamento em direção a esses fins.

2.3.1 Meio de Aplicação da Gamificação

A discussão sobre a natureza da gamificação estende-se ao seu meio de aplicação. Enquanto alguns autores, como Domínguez et al. (2013) e a definição dada por Gartner (2014), sugerem que a gamificação está restrita ao ambiente digital, outros, como Deterding et al. (2011), contrapõem-se à visão restritiva, argumentando que jogos e design de jogos são, em sua essência, "categorias transmidiais" (Kim, 2015). Adicionalmente, fenômenos como a "convergência de mídias e computação ubíqua estão cada vez mais borrando a distinção entre digital e não-digital". Com base nisso, Kim (2015, p. 16) conclui que "limitar a gamificação ao reino digital ou aplicação de software é excessivamente restritivo". A gamificação, portanto, pode manifestar-se tanto na forma de "um jogo de papel e lápis ou de um aplicativo móvel". A inclinação de algumas definições para o digital reflete, em parte, a proeminência da tecnologia – especialmente aplicativos móveis e software – como veículo para a implementação da gamificação na prática contemporânea. Kim (2015, p. 15) reconhece que "muita gamificação assume a forma de mídia digital" e que "o aplicativo móvel parece ser uma forma particularmente adequada". No entanto, a insistência na natureza transmidiática sugere que o princípio da gamificação – usar elementos de jogo para engajar e motivar – é mais fundamental do que a plataforma de sua implementação. O crescimento da gamificação está diretamente ligado à popularização de smartphones, da web móvel e das mídias sociais, além da maior penetração que os jogos digitais têm na sociedade atual, no Brasil, hoje, o hábito de jogar videogames já está presente na vida de 73,9% da população (PGB 2024). Esses fatores permitiram que os elementos de jogo fossem integrados de forma contínua e social à vida cotidiana. Aplicativos como Nike+, que gamificam atividades como corrida, só são possíveis graças à mobilidade e conectividade oferecidas pelos smartphones (Kim, 2015).

A gamificação representa uma mudança cultural, em que atividades reais ganham valor adicional por meio de recompensas virtuais e interações sociais. Isso aproxima o comportamento online e offline, criando novas formas de motivação, aprendizado e enga-

jamento. Embora compartilhe características com os jogos, como a presença de regras, a busca por diversão e, por vezes, o uso de tecnologia, a gamificação se distingue por seu propósito fundamental de resolver questões práticas. (Kim, 2015).

Independentemente do meio escolhido, o critério fundamental para o sucesso e a validade de uma aplicação gamificada reside em sua eficácia. Kim (2015) enfatiza que “o que é importante na gamificação é que ela engaje e ajude as pessoas a alcançar seus objetivos da vida real usando elementos e dinâmicas de jogo apropriados”. A capacidade de gerar engajamento e auxiliar no alcance de objetivos reais supera considerações sobre a pureza digital da aplicação, o conjunto específico de mecânicas empregadas ou a adesão estrita a uma determinada definição teórica.

2.3.2 Mecânicas, dinâmicas e estéticas

Embora muitos textos e plataformas sobre gamificação se limitem a listar elementos como pontos, emblemas, níveis e rankings, essa abordagem tende a ser superficial e ignora as diferentes camadas de abstração que compõem a experiência de jogo (Kim, 2015). Para superar essa limitação, apresenta-se o modelo MDA (*Mechanics, Dynamics and Aesthetics*) como uma estrutura teórica capaz de organizar e explicar a complexidade do design de jogos e sua aplicação à gamificação.

As mecânicas correspondem às regras e ações possíveis no sistema — por exemplo, mover peças, coletar pontos, ou completar tarefas. Já as dinâmicas referem-se aos comportamentos e interações que emergem dessas mecânicas, como competição, cooperação, estratégia ou descoberta coletiva. Por fim, as estéticas são os objetivos subjetivos do jogo: sensações e emoções que ele busca gerar, como diversão, desafio, imersão ou expressão pessoal.

Os elementos estéticos podem ser classificados em categorias como: sensação (jogo como prazer sensorial), fantasia (jogo como faz-de-conta), narrativa (jogo como drama), desafio (jogo como obstáculo), confraternização (jogo como estrutura social), descoberta

(jogo como exploração), expressão (jogo como autoexpressão) e submissão (jogo como passatempo). As dinâmicas são projetadas para criar essas experiências estéticas — por exemplo, a pressão do tempo e a presença de oponentes favorecem o sentimento de desafio; sistemas de customização e criação de personagens promovem a expressão.

Reduzir a gamificação a mecânicas visíveis, como pontos e prêmios, pode comprometer seu potencial pedagógico ou motivacional, especialmente em contextos como a educação (Kim, 2015). A eficácia de uma experiência gamificada depende da interação harmoniosa entre mecânicas bem projetadas, dinâmicas significativas e estéticas alinhadas aos objetivos e ao público-alvo. O modelo MDA, portanto, serve como uma ferramenta valiosa para educadores e designers compreenderem como os elementos de jogo operam em diferentes níveis, permitindo a criação de experiências mais envolventes, motivadoras e intencionais.

2.4 Serious games

A evolução dos jogos eletrônicos, conforme discutido anteriormente, transcendeu a esfera do entretenimento, consolidando-os como uma força cultural e econômica com profundo impacto social (Freire; Guerrini, 2016). Um jogo, em sua definição mais fundamental, é uma atividade interativa, regida por regras, com um ou mais objetivos e uma forma quantificável de medir o progresso ou o sucesso (Becker, 2021). É essa estrutura de regras, metas e desafios que permite a adaptação dos jogos para contextos que transcendem o entretenimento.

Paralelamente à expansão cultural, emergiu um campo de aplicação que utiliza as mecânicas e a capacidade de engajamento dos jogos para fins que vão além da pura diversão (Sawyer, 2003 *apud* Becker, 2021). Este campo é o dos serious games (jogos sérios), que são definidos como jogos cujo propósito primário não é a pura diversão, mas sim a consecução de um objetivo específico, como a aquisição de conhecimento, o desenvolvimento de uma habilidade ou a promoção de uma mudança de comportamento ou

atitude (Dempsey; Rasmussen; Lucassen, 1996 *apud* Battistella, 2016).O termo, popularizado por Clark Abt em 1970, abrange um vasto leque de aplicações, incluindo (Becker, 2021) :

- **Jogos para a saúde:** Utilizados para treinamento de procedimentos médicos, gestão da dor e adesão a tratamentos.
- **Jogos para mudança social:** Focados em conscientização sobre questões políticas e de justiça social.
- **Advergames:** Desenvolvidos com o propósito primário de publicidade e marketing.
- **Jogos para treinamento:** Aplicados em contextos militares, corporativos e industriais.
- **Jogos para Aprendizagem:** Categoria que engloba os jogos educacionais.

É crucial diferenciar os *serious games* de conceitos correlatos como a gamificação e a aprendizagem baseada em jogos (*game-based learning* - GBL). Conforme discutido na seção 2.3, a gamificação consiste no uso de elementos de design de jogos em contextos não lúdicos, com o objetivo de aumentar o engajamento em uma atividade preexistente (Deterding *et al*., 2011, *apud* Kim, 2015). Em contrapartida, um serious game é um jogo completo, uma experiência autônoma projetada em torno de um objetivo não lúdico (Becker, 2021). A GBL, por sua vez, não é um produto, mas uma abordagem pedagógica, o processo de aprender utilizando jogos, que podem ser tanto comerciais quanto educacionais (Becker, 2021)

2.4.1 Jogos para educação

Os jogos educacionais são, por definição, jogos especificamente projetados para ensinar um determinado assunto, expandir conceitos, reforçar o desenvolvimento de uma

habilidade ou buscar uma mudança de atitude no jogador (Dempsey; Rasmussen; Lucassen, 1996 *apud* Battistella, 2016). O desafio central em seu design reside no equilíbrio entre o conteúdo instrucional e a jogabilidade (*gameplay*). Um jogo que prioriza excessivamente a pedagogia em detrimento da diversão torna-se um exercício enfadonho, enquanto um jogo que foca apenas na jogabilidade, com conteúdo educacional superficial, falha em seu propósito instrucional.

A literatura acadêmica aponta uma gama de benefícios associados ao uso de jogos na educação. Eles podem aumentar a eficácia da aprendizagem, o interesse, a motivação e a persistência dos alunos. Ao serem divertidos, facilitam o aprendizado e aumentam a capacidade de retenção do conteúdo, funcionando como uma “ginástica mental” (Roland *et al*., 2004). Um dos benefícios mais relevantes é a promoção da aprendizagem ativa, na qual os alunos se engajam em atividades como diálogo, resolução de problemas e pensamento crítico, em vez de serem meros receptores de informação. Essa abordagem está alinhada com as teorias construtivistas da educação, que valorizam a participação e a experimentação do sujeito na construção de seu próprio conhecimento (Roland *et al*., 2004).

A escolha da plataforma — digital ou não-digital — também possui implicações pedagógicas. Uma revisão da literatura realizada por Battistella (2016) identificou uma predominância de jogos digitais, mas também um número considerável de jogos não-digitais (tabuleiro, cartas, etc.). Jogos não-digitais podem ser superiores para o desenvolvimento de competências socioemocionais, como comunicação e trabalho em equipe, por exigirem interação face a face. Por outro lado, jogos digitais oferecem vantagens na simulação de sistemas complexos, na automação de regras e no fornecimento de *feedback* imediato e individualizado.

2.5 Esquema de Classificação para Gamificação em Ciências da Computação

A gamificação tem se mostrado uma abordagem promissora para aumentar o engajamento, a motivação e o aprendizado em diversas áreas, incluindo a Ciência da Computação. No entanto, a falta de um estudo formalizado e uma análise abrangente dificulta a compreensão do que torna uma gamificação eficaz nesse contexto (Spanier; Harms; Hastings, 2021). Visando preencher essa lacuna, especialmente no âmbito do ensino de Estruturas de Dados (ED) e Algoritmos, Spanier, Harms e Hastings (2021) propuseram um sistema de classificação para implementações de gamificação, identificando gêneros existentes e potenciais.

A pesquisa dos autores focou inicialmente em aplicações de gamificação em cursos de ED, devido à complexidade inerente e à abstração dos conceitos fundamentais dessa área, que representam um desafio educacional e, ao mesmo tempo, uma oportunidade para abordagens inovadoras como a gamificação. A partir de uma revisão da literatura de aplicações de gamificação em ED, os autores categorizaram as implementações e analisaram seus prós e contras.

2.5.1 Gêneros Identificados de Gamificação em Estruturas de Dados

Com base na análise de aplicações existentes, foi identificado pelos mesmos autores três padrões distintos que levaram à definição dos seguintes gêneros de gamificação em ED:

- **Visualização de Ideias Abstratas (VAI – *Visualization of Abstract Ideas*)**: Este gênero agrupa aplicações que utilizam a visualização para descrever conceitos abstratos. Dada a natureza complexa de algoritmos de busca, árvores de grafos, entre outras estruturas, a capacidade de visualizar efetivamente esses processos permite aos alunos compreender ideias que, de outra forma, seriam de difícil assimilação. A

gamificação digital, como benefício inerente, permite que instrutores demonstrem de forma flexível o passo a passo de algoritmos e processos, possibilitando aos alunos interagir diretamente com a visualização para melhor compreender as operações discretas das estruturas e algoritmos.

- **Exame Aprimorado (EE – *Enhanced Examination*)**: Este gênero refere-se a aplicações que adicionam uma interface gamificada a programas de questionários ou exercícios. Na tentativa de engajar melhor os alunos com o conteúdo de uma avaliação, um apelo estético e uma interface mais interativa são adicionados. Neste gênero, busca-se transformar avaliações tradicionais em experiências mais atraentes e menos tediosas, utilizando elementos de jogos para motivar a participação e o engajamento dos alunos com o material avaliado.

- **Engajamento Social e Colaborativo (SCE – *Social and Collaborative Engagement*)**: Este gênero engloba aplicações que utilizam a interação social e colaborativa para facilitar o aprendizado e a motivação dos alunos. Aplicações de gamificação podem ser desenvolvidas para que os alunos interajam mais facilmente entre si, de maneiras que aumentem a motivação e o engajamento. Este gênero foca no potencial da gamificação para criar ambientes onde os alunos podem colaborar, competir de forma saudável ou interagir socialmente em torno dos objetivos de aprendizagem.

Spanier, Harms e Hastings (2021) observam que a maioria das aplicações revisadas se enquadra no gênero VAI, indicando uma certa homogeneidade e um campo ainda subdesenvolvido para a gamificação em ED, necessitando de mais pesquisa e diversificação.

2.5.2 Novos Gêneros Abstratos Propostos

Além dos gêneros observados nas aplicações existentes, Spanier, Harms e Hastings (2021) identificaram a ausência de certas abordagens, inferindo a existência de gêneros

ainda não explorados ou subdesenvolvidos na gamificação em ED. Esses “gêneros ausentes” são aqueles que não surgem intuitivamente e requerem uma tentativa intencional de identificação. Os autores propuseram dois novos gêneros abstratos:

- **Gamificação Dinâmica (DG – *Dynamic Gamification*)**: Refere-se a todas as aplicações gamificadas que mudam dinamicamente de acordo com a entrada do aluno ao longo de seu ciclo de vida gamificado. Diferentemente de implementações mais estáticas, a DG utilizaria o desenvolvimento de software liderado pelo aluno, usando estruturas de dados e algoritmos existentes e novos, ao mesmo tempo em que proporciona oportunidades para a criatividade dos alunos. A DG ainda exibiria mecânicas de jogo comuns (como placares de líderes, avatares, emblemas), mas adicionaria uma camada de desenvolvimento de jogo liderado pelo aluno. As inovações lideradas pelos alunos dentro de uma estrutura de jogo dada mudariam dinamicamente a aparência, a sensação, as mecânicas de jogo e o conjunto geral de características exibidas pela aplicação, atendendo ao que os alunos sentem que seria mais benéfico.

- **Desenvolvimento Colaborativo de Gamificação (CGD – *Collaborative Gamification Development*)**: Pertence a todas as aplicações que utilizam o envolvimento colaborativo dos alunos na formação da própria estrutura de gamificação. Em vez de serem passivamente expostos a uma aplicação gamificada, os alunos participam da conceituação e desenvolvimento da estrutura pela qual a aplicação é implementada. O CGD é particularmente útil, pois mantém a instância de uma determinada aplicação gamificada atualizada e diretamente formulada para as necessidades de um determinado grupo de alunos. Este gênero se diferencia da DG pelo fato de que a estrutura da aplicação é criada através de um processo colaborativo liderado pelos alunos.

A introdução desses novos gêneros abstratos, DG e CGD, poderia adicionar funciona-

lidade e benefícios além do que as aplicações de gamificação existentes em ED fornecem. Permitir que os alunos sejam parte integrante da escolha das mecânicas gamificadas que mais lhes agradam pode potencialmente aumentar ainda mais os níveis de motivação e engajamento, além de implementar projetos liderados por alunos, que demonstram aumentar a motivação intrínseca e a retenção (Spanier; Harms; Hastings, 2021).

A classificação proposta por Spanier, Harms e Hastings (2021) oferece uma base para entender melhor o estado atual da gamificação em cursos de ED e direciona futuras pesquisas para explorar gêneros ainda incipientes, com o objetivo de tornar a prática da gamificação mais eficiente e eficaz no ensino de Ciência da Computação.

2.6 Documentação de Design de Jogos

O desenvolvimento de um jogo digital é um processo multifacetado que abrange diversas etapas e exige a colaboração sinérgica de uma equipe multidisciplinar. Neste contexto, a clareza na comunicação e o alinhamento conceitual são de extrema importância para o êxito de qualquer projeto. Uma das ferramentas mais vitais para consolidar essa fundação é o Documento de Design de Jogos, conhecido pela sigla GDD (*Game Design Document*). Contudo, o GDD não é um artefato monolítico; sua forma e função evoluíram significativamente, espelhando as transformações nas próprias metodologias de desenvolvimento de software.

O Documento de Design de Jogos é universalmente reconhecido como o documento central que serve como um projeto detalhado para a construção de um jogo. Ele articula e detalha todas as características de um jogo, incluindo sua história, conceitos, personagens, cenários, mecânicas, elementos de arte, sons e interface do usuário (Hira *et al*., 2016). Essencialmente, o GDD funciona como um manual de referência para todos os membros da equipe de desenvolvimento, garantindo que a visão do designer seja comunicada de forma clara e consistente para programadores, artistas, músicos e gerentes de projeto.

A literatura especializada, conforme sintetizado por Hira *et al*. (2016), atribui ao GDD três funções primordiais que justificam sua centralidade no processo de desenvolvimento. A primeira é a de Registro, que visa suprir as limitações da memória humana, servindo como um repositório formal para todas as decisões, regras e definições relacionadas ao jogo. A segunda função é a de Comunicação, atuando como uma base de referência consultiva que alinha a equipe e facilita a interação entre as diversas áreas, especialmente entre o designer e os demais desenvolvedores. A terceira, e talvez mais abstrata, é a Concretização de um conceito, que busca permitir que o leitor, seja ele um membro da equipe ou um investidor, compreenda o funcionamento e a experiência do jogo antes mesmo que uma linha de código seja escrita, possibilitando um “jogo mental” da proposta (Adams, 2007; Schell, 2008, *apud* Hira *et al*., 2016)

A ausência de uma documentação estruturada pode acarretar consequências severas para um projeto. Problemas como o desvio do escopo original (*scope creep*), alocação inadequada de recursos, inconsistências no design e uma acentuada dificuldade para corrigir falhas e incidentes são frequentemente associados à falta de um GDD robusto (Hira *et al*., 2016). Essa percepção é corroborada por análises de *post-mortems* da indústria, onde a falta de documentação ou a documentação inadequada são frequentemente citadas como fatores que contribuem para atrasos e falhas em projetos de jogos (Politowski et al., 2018)

A evolução do GDD resultou em uma variedade de modelos, cada um refletindo uma filosofia de documentação distinta e adequada a diferentes contextos de produção. A análise desses modelos revela um espectro que vai de abordagens holísticas e detalhadas, ideais para grandes produções, a formatos modulares e pragmáticos, focados em agilidade e viabilidade.

Os modelos mais tradicionais de GDD caracterizam-se pela sua estrutura abrangente, que busca documentar exaustivamente todos os aspectos do jogo. O modelo proposto por Rouse III (2001), como apontado por Hira *et al*.(2016) é um exemplo clássico dessa

abordagem. Ele enfatiza a comunicação efetiva e organiza o documento em seções detalhadas, como: Sumário, Introdução/Visão Geral, Mecânica do Jogo, Inteligência Artificial, Elementos do Jogo (divididos em Personagens, Itens e Objetos/Mecanismos), Visão Geral da História, Progressão do Jogo e Menus do Sistema (Hira *et al*., 2016). A força desse modelo reside em sua capacidade de fornecer uma visão completa e unificada, servindo como uma fonte de verdade para equipes grandes e multidisciplinares, embora sua rigidez possa ser um desafio em ambientes de desenvolvimento rápido.

De forma semelhante, o modelo de Schuytema (2008), segundo Hira *et al*.(2016) propõe uma estrutura essencialista e hierárquica. Ele organiza o conteúdo em seções que vão do geral ao específico, começando com uma “Visão geral essencial” (que inclui um resumo, aspectos fundamentais e os “*golden nuggets*” — os diferenciais do jogo) e se aprofundando em “Contexto do game”, “Objetos essenciais”, “Conflitos e Soluções”, entre outros. Assim como o modelo de Rouse III, sua natureza é compreensiva, mas com uma filosofia organizacional ligeiramente diferente, permitindo que certas seções sejam expandidas ou suprimidas conforme o gênero do jogo, como em um puzzle ou em um jogo sem narrativa (Hira *et al*., 2016).

Com a maturação da indústria, surgiram modelos que reconhecem a necessidade de diferentes níveis de detalhe para diferentes públicos e fases do projeto. O modelo de Rogers (2012) é um marco nesse sentido, pois propõe não um único documento, mas uma “pipeline” de documentação escalável, composta por três artefatos distintos (Hira *et al*., 2016).

- ***One-Sheet*:** Um resumo de uma única página que serve como a “ideia” do jogo. Crucialmente, ele inclui não apenas elementos de design, mas também informações de negócio, como público-alvo, classificação etária pretendida (ESRB), análise de concorrentes e sistema de vendas. Isso o torna uma ferramenta de *pitching* ideal para apresentar o projeto a investidores e editoras.

- ***Ten-Pager*:** Um documento mais detalhado, de aproximadamente dez páginas, que

expande a visão do *One-Sheet* para o alinhamento interno da equipe. Ele aprofunda a mecânica, a história, o fluxo do jogo e o design dos personagens, respondendo não apenas “o que” é o jogo, mas “como” ele funciona.

- ***Beat Chart*:** O documento mais granular, focado na produção. Organizado por níveis ou fases, ele detalha a experiência momento a momento do jogador, especificando inimigos, *puzzles*, *power-ups*, elementos de história e novas mecânicas introduzidas em cada estágio. É um guia prático para *level designers* e artistas.

O modelo de Fullerton, Swain e Hoffman (2008) se destaca por sua seção de “Viabilidade Técnica”. Além de cobrir os aspectos tradicionais como gameplay, história e personagens, este modelo exige uma análise explícita de como o jogo será codificado, as tecnologias necessárias, os custos de produção e os requisitos de hardware e software. Ao forçar a equipe a confrontar as restrições técnicas e financeiras desde o início, este modelo promove um design mais realista e fundamentado, sendo extremamente valioso para equipes independentes ou projetos com orçamentos limitados (Hira *et al*., 2016).

Por sua vez, o modelo de Ryan (1999), relatado por Hira *et al*.(2016) é notável por destacar as mecânicas no jogo no inicio do documento. Ao estruturar o documento para começar com a “Mecânica do jogo” e a “Interface de usuário”, deixando o “Enredo” para depois, ele apoia inerentemente o desenvolvimento de jogos onde o *gameplay* é o elemento central e a narrativa pode ser secundária ou emergente. Essa abordagem é eficiente para projetos cujo sucesso depende primariamente da qualidade do seu loop de jogabilidade (Hira et al., 2016).

2.6.1 Short Game Design Document

No extremo do espectro da agilidade e do pragmatismo, encontra-se o *Short Game Design Document* (SGDD), proposto por Motta e Trigueiro Jr. (2013). Este modelo não é apenas uma versão abreviada de um GDD tradicional, mas uma ferramenta com uma

filosofia fundamentalmente diferente, concebida especificamente para os cenários de desenvolvimento mais críticos em termos de tempo, como game jams, prototipagem rápida e a criação de advergames. O SGDD representa a aplicação prática e direta da mentalidade ágil à documentação, priorizando a ação imediata e a clareza funcional sobre a especificação exaustiva.

A inovação central do SGDD reside em sua metodologia de criação, que se afasta da estrutura de seções isoladas para adotar um formato narrativo e linear. O processo, conforme detalhado por Motta e Trigueiro Jr. (2013), consiste nas seguintes etapas:

- **Descrição Sintética do Enredo:** Uma breve introdução que estabelece o conceito geral do jogo.
- **Descrição em "Texto Corrido":** Esta é a essência do SGDD. O designer descreve toda a experiência do jogador, da tela inicial ao final do jogo, em um texto contínuo. O objetivo é narrar um *playthrough* completo, permitindo que a equipe realize um "jogo mental" e visualize a experiência de forma coesa e cronológica.
- **Marcação de Elementos:** Diretamente no texto corrido, o designer marca, utilizando cores ou formatação, todos os elementos de produção necessários. Por exemplo, podem ser destacadas as necessidades de arte e interface, as mecânicas de programação e os efeitos sonoros e músicas.
- **Criação de Listas:** Os elementos marcados são então extraídos do texto e organizados em listas de tarefas acionáveis (*assets* de arte, rotinas de programação, trilhas sonoras), que funcionam como um *backlog* de produção inicial.
- **Level Design (Opcional):** Caso necessário, o documento pode ser complementado com desenhos, fluxogramas ou esboços que detalhem o design dos níveis.

A eficácia do SGDD transcende sua brevidade. Sua principal força reside em sua função como uma ferramenta de tradução e empoderamento para a equipe de desenvolvi-

mento. Enquanto os GDDs tradicionais frequentemente isolam a informação em arquivos ou seções distintas, exigindo que os membros da equipe consultem múltiplas fontes e sintetizem as informações por conta própria, o SGDD adota a abordagem oposta.

O “texto corrido” fornece o contexto unificado; todos os elementos de produção são apresentados em sua ordem de aparição e em relação direta uns com os outros. A subsequente criação de listas é um ato de tradução direta dessa visão contextual em um *backlog* de produção. Um artista que lê a tarefa “Animação: Personagem pedalando” sabe exatamente qual é a sua função, onde ela se encaixa no fluxo do jogo e qual a sua importância para a experiência central, sem a necessidade de cruzar referências em múltiplos documentos.

Essa metodologia resulta em uma velocidade notável. O SGDD de *Rockergirl Bikeway*, dado como exemplo no artigo elaborado pelos autores, foi elaborado em aproximadamente duas horas, permitindo que o desenvolvimento começasse quase que imediatamente (Motta; Trigueiro Jr., 2013). Além da eficiência, o SGDD promove o que os autores chamam de “liberdade criativa”. Ao comunicar claramente o “o quê” (uma animação de pulo) e o “porquê” (para desviar de obstáculos em um jogo de corrida infinita), o documento deixa o “como” (o estilo específico, a fluidez e o carisma da animação) a cargo da expertise do artista ou do programador. Isso fomenta um senso de propriedade e autonomia criativa, características essenciais de equipes ágeis e de alto desempenho.

Em suma, o *Short Game Design Document* não é meramente um GDD mais curto, mas sim uma abordagem fundamentalmente distinta. Ele serve como uma ponte eficiente entre a visão conceitual e a execução prática, tornando-se uma ferramenta de documentação ideal para projetos que exigem agilidade, colaboração e uma tradução clara da ideia para um produto tangível com máxima eficiência e alinhamento de equipe.

3 Trabalhos Correlatos

Foram selecionados, como trabalhos relacionados, jogos voltados para o ensino de conceitos da área de computação, com ênfase especial em conteúdos da disciplina de Estruturas de Dados. A busca inicial utilizou termos mais amplos como “Jogos Educativos para Computação” e “*Games for Learning*”, sendo posteriormente refinada com expressões específicas relacionadas a estruturas de dados, como “Jogos para ensino de estruturas de dados” ou “Jogos para ensino de algoritmos de ordenação”. As pesquisas foram conduzidas em bases acadêmicas como Google Scholar, IEEE Xplore, Biblioteca Digital da Sociedade Brasileira de Computação, além de repositórios de Trabalhos de Conclusão de Curso de universidades brasileiras, especialmente o repositório da Universidade Federal de Santa Catarina. Foram descartados estudos que não abordam jogos diretamente ligados à área da Computação.

Tabela 1: Tabela Resumo dos Trabalhos Correlatos

Título do Jogo/Trabalho	**Referência**	**Classificação**
Gidget	(Lee *et al*., 2014)	Computação (Geral)
LEARN	(Sousa e Marques, 2020)	Computação (Geral)
Stack Game	(Dicheva; Hodge, 2018)	Estruturas de Dados
Space Traveler	(Zhang *et al*., 2015)	Estruturas de Dados
Sort It Out	(Glatz, 2023)	Estruturas de Dados
Plataforma Interativa	(Toda *et al*., 2013)	Estruturas de Dados

3.1 Jogos Educacionais Para Computação

Nesta seção, são apresentados jogos que abordam o ensino de temas ligados à Computação de forma geral, mas que não tratam diretamente dos conteúdos específicos da disciplina de Estruturas de Dados. Como primeiro exemplo, destaca-se o Gidget, um jogo educacional desenvolvido por Lee *et al*. (2014). O jogo é focado no ensino de programação por meio da depuração de código (*debugging*). Na história, o robô Gidget foi enviado para limpar uma área contaminada por um vazamento químico que ameaça a vida animal.

No entanto, Gidget está danificado e fornece códigos com erros. Cabe ao jogador ajudar Gidget corrigindo esses códigos para atingir os objetivos de cada fase, definidos como condições (*assertions*) sobre o estado do mundo do jogo.

O jogo é estruturado em 34 fases distribuídas por 7 unidades, que cobrem tópicos fundamentais de programação, como variáveis, listas, estruturas condicionais, laços de repetição e funções. A linguagem utilizada é imperativa, com sintaxe semelhante à do Python. Durante o jogo, os alunos utilizam ferramentas semelhantes a depuradores reais para executar o código passo a passo e observar o comportamento do programa, com *feedback* detalhado sobre os efeitos de cada comando.

Além das atividades de resolução de *puzzles*, Gidget também oferece um editor de níveis, permitindo que os jogadores criem, modifiquem e compartilhem suas próprias fases. O jogo foi projetado com atenção à inclusão de gênero, evitando terminologias violentas e mecânicas competitivas, e utilizando uma narrativa socialmente relevante. Ele também oferece múltiplas formas de ajuda e tutoriais, como *feedback* contextualizado, guias de referência e sugestões interativas, visando atender diferentes estilos de aprendizado.

Foram encontrados também jogos analógicos para o ensino de áreas da computação, um dos exemplos destaca-se o LEARN de Sousa e Marques (2020), trabalho que propõe e avalia um jogo de tabuleiro não-digital para auxiliar no ensino de conceitos e padrões de Arquitetura de Software. Desenvolvido a partir da metodologia *Design Science Research*, o jogo visa dinamizar o processo de ensino-aprendizagem, abordando a dificuldade dos estudantes em compreender conceitos abstratos e decisões arquiteturais.

A pesquisa incluiu um estudo de caso com 16 estudantes de Engenharia de Software, utilizando o questionário MEEGA+ para coletar dados sobre usabilidade e experiência de jogo. Os resultados da avaliação indicaram que o jogo possui boa usabilidade, é fácil de aprender e jogar, e contribui para a interação social e o aprendizado eficiente dos estudantes.

3.2 Jogos Educacionais Para Estruturas Dados

Foram encontrados também jogos para o auxílio no ensino de conceitos apresentados na disciplina de Estruturas de Dados.

Como trabalho correlato, destaca-se o Stack Game, um jogo educacional projetado para auxiliar no aprendizado da estrutura de dados Pilha (Dicheva; Hodge, 2018). O jogo tem como objetivo abordar três aspectos cruciais do ensino de estruturas de dados: a conceituação, a aplicação e a implementação. Esses aspectos são integrados em três partes do jogo.

No Stack Game, o jogador controla um robô que precisa resolver problemas relacionados a pilhas para superar obstáculos e retornar ao seu planeta. A primeira parte do jogo foca na compreensão do conceito da estrutura, no qual o jogador resolve um quebra-cabeça similar à Torre de Hanói, movendo blocos coloridos entre pilhas para formar uma sequência específica que funciona como uma chave para destravar portas. Os níveis aumentam em dificuldade, inclusive com a ocultação do conteúdo das pilhas (exceto o topo) para aprofundar o entendimento do conceito.

A segunda parte do jogo aborda a aplicação de pilhas. O jogador deve usá-las para resolver problemas como conversão de expressões aritméticas da notação infixa para pós-fixa, avaliar expressões pós-fixas e avaliar expressões infixas, fazendo uso de ferramentas como Pilha de Operandos e Calculadora.

A terceira parte do jogo é voltada para a implementação de pilhas. O jogador precisa completar ou escrever códigos em Java para os métodos principais da classe (*pop(), push(), peek(), isEmpty()*) para ativar luzes e abrir a entrada principal de uma fortaleza. Esta seção inclui problemas de Parsons (o código é dado em fragmentos que precisam ser ordenados) e a escrita direta de trechos de código, abordando tanto implementações baseadas em *array* quanto em listas encadeadas.

Dicheva e Hodge (2018) conduziram uma avaliação do Stack Game em dois cursos consecutivos de Estruturas de Dados na *southern liberal arts college*, com resultados

mostrando ganhos de aprendizado estatisticamente significativos para os estudantes e uma forte atitude positiva em relação a essa forma de aprendizado ativo. Os autores destacam que o jogo foi projetado para oferecer uma experiência de aprendizado interativa e engajadora. O estudo sugere que o “Stack Game” é uma ferramenta de aprendizado eficaz, que ajudou os alunos a desenvolverem um modelo conceitual mais correto e sólido sobre pilhas e seu comportamento.

Figura 1: Trecho do jogo Stack Game mostrando a avaliação de uma expressão pós-fixa

Fonte:`https://www.researchgate.net/figure/The-Stack-Game-Part-2_fig2_311607952`. Acesso em: 27 maio 2025.

Outro trabalho no campo de jogos educacionais para estruturas de dados é o “Space Traveler”, desenvolvido utilizando o *GameMaker Studio* (Zhang et al., 2015). Este jogo foi desenvolvido com o objetivo principal de auxiliar os estudantes a visualizar, compreender e dominar os conceitos e as operações fundamentais das listas encadeadas (*linked lists*), uma estrutura de dados que os autores observaram ser particularmente desafiadora para os alunos.

O Space Traveler adota uma mecânica similar ao clássico jogo “*Snake*”, ou “jogo da

cobrinha” como é popularmente conhecido no Brasil. No jogo, o jogador controla uma nave espacial, que representa a lista encadeada. A nave cresce em comprimento à medida que coleta “*orbs*”, que simbolizam os nós da lista. O jogo se concentra em permitir que os estudantes pratiquem as operações essenciais em listas encadeadas, como inserção e remoção de nós. Para realizar as operações na “nave espacial” (a lista encadeada), como adicionar armas (novos nós) ou remover partes danificadas (nós existentes), o jogador é apresentado a segmentos de pseudo-código, com foco na linguagem Java. Os jogadores precisam completar as partes faltantes desses códigos para executar as operações de inserção ou remoção. Para auxiliar nesse processo, o jogo oferece menus com opções de código, guiando o aluno para a solução correta.

O jogo inclui quatro níveis de tutoriais que cobrem os controles básicos e cada uma das operações de lista encadeada suportadas. Esses tutoriais são obrigatórios e visam garantir que o aluno compreenda a mecânica antes de prosseguir para a missão principal, na qual deve ser realizado um número específico de cada operação. A visualização das alterações na estrutura da nave após cada operação ajuda a reforçar a compreensão do impacto do código.

Para avaliar seu impacto, o jogo foi utilizado em uma turma da disciplina de Estruturas de Dados na *Winston-Salem State University* (WSSU) em 2014. A avaliação incluiu a aplicação de pré-testes e pós-testes, além de uma pesquisa de opinião com os alunos. Os resultados da avaliação inicial foram promissores, mostrando melhorias significativas no desempenho dos alunos após a utilização do módulo do jogo. O *feedback* dos estudantes também foi bastante positivo, com 93% relatando uma melhor compreensão das operações em listas encadeadas após jogarem o Space Traveler.

Figura 2: Captura de tela do jogo Space Traveler

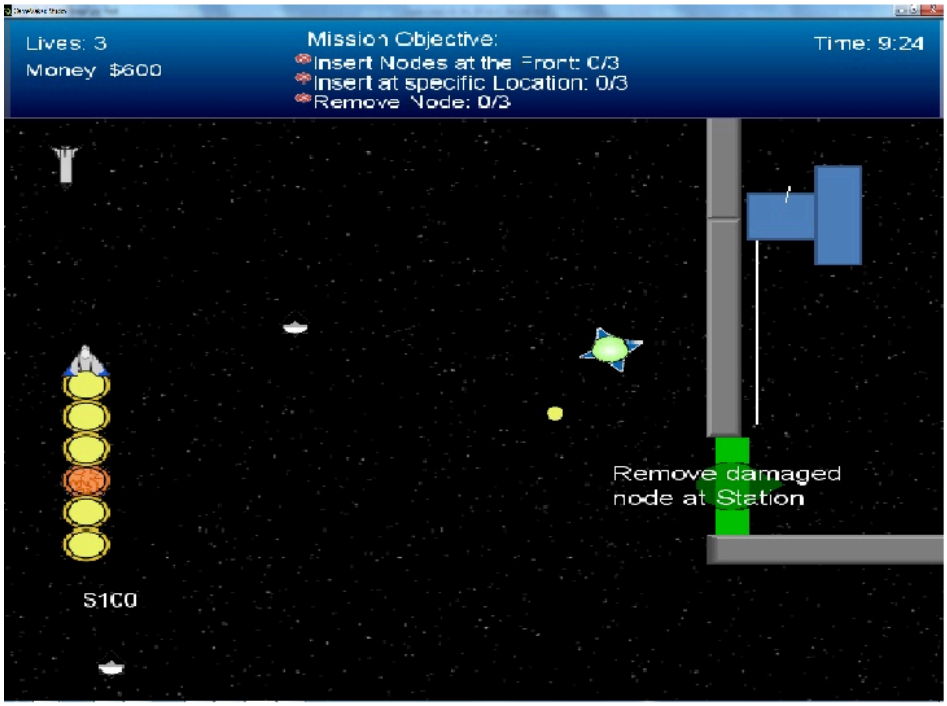


Fig 1. Screenshot of the Space Traveler Game

Fonte: `https://www.semanticscholar.org/paper/Reinforcing-student-understanding-of-linked-list-in-Zhang-Atay/3a9ba9aa08e87149893b00be65bb872024e6bd77/figure/0`. Acesso em: 27 maio 2025.

Um trabalho que se destaca por ser uma ferramenta de auxilio ao ensino,e à revisão, de algoritmos de ordenação é o *Sort It Out* de Glatz (2023). O jogo permite que os estudantes executem os passos dos algoritmos diretamente, manipulando elementos de estruturas de dados conforme a lógica de cada algoritmo, promovendo um maior engajamento e compreensão prática dos conceitos vistos em sala de aula. O jogo foi pensado para ser utilizado tanto como ferramenta de apoio em aulas expositivas quanto de forma autônoma, como reforço posterior.

Inicialmente, foram desenvolvidas fases para o algoritmo *Bubble Sort*, onde o aluno seleciona pares de elementos e decide se deve trocá-los, seguindo a lógica de ordenação crescente. O sistema de pontuação premia acertos e penaliza erros ou uso de dicas, incentivando o raciocínio. O jogo possuí também fases usando o algoritmo *Merge Sort*, com dois tipos distintos de desafios: no primeiro, o aluno apenas indica a sequência cor-

reta dos passos do algoritmo, no segundo, ele preenche manualmente os elementos nas posições adequadas. A escolha por uma implementação leve, com foco em portabilidade e simplicidade gráfica, priorizou o uso de linguagens tradicionais ao invés de *engines* gráficas.

A avaliação de *Sort It Out* foi realizada em sala de aula com 15 estudantes de Sistemas de Informação, utilizando o modelo MEEGA+. Os resultados foram positivos quanto à relevância do jogo para o conteúdo da disciplina e a percepção de aprendizagem dos alunos. No entanto, foram observados pontos a serem melhorados na usabilidade, como a necessidade de explicações mais detalhadas sobre o funcionamento do aplicativo e um fluxo mais claro entre as fases, além de questões relacionadas à imersão e possível monotonia das tarefas (Glatz, 2023).

Outro trabalho relevante nesta categoria é uma plataforma interativa para aprendizagem desenvolvida no motor de jogos *Construct* 2 (Toda *et al.*, 2013). O projeto visa criar um ambiente de aprendizagem interativo que utiliza conceitos de gamificação como recurso motivacional para estudantes de cursos da área de Tecnologia da Informação, com foco nos conceitos da disciplina de Estrutura de Dados.

A plataforma busca engajar os estudantes por meio de uma série de minijogos e tarefas interativas relacionadas a algumas estruturas de dados, como *arrays* (vetores), *stacks* (pilhas), *queues* (filas), *lists* (listas) e *trees* (árvores).

O ambiente de aprendizagem propõe diversas atividades para reforçar o conhecimento. Duas tarefas gerais, aplicáveis a todas as estruturas, são:

- **Tarefa de associação**: os alunos recebem uma lista de algoritmos ou fragmentos de código e devem associá-los às suas funções corretas dentro do contexto da estrutura de dados trabalhada.

- **Tarefa de ordenação**: fragmentos de código de uma função são apresentados de forma desordenada, e o aluno deve organizá-los na sequência correta para que o algoritmo funcione.

Além dessas, cada estrutura de dados pode possuir tarefas adicionais baseadas em seus algoritmos conceituais específicos, como a Torre de Hanói para exercícios com pilhas, e o balanceamento de árvores binárias de busca.

Para motivar os estudantes, o sistema incorpora elementos de gamificação, como sistema de níveis, pontuação, conquistas e placares de líderes.

4 Proposta

Tendo em vista a importância do aprendizado dos conceitos abordados na disciplina de Estrutura de Dados, além de considerar a dificuldade encontrada por muitos alunos, como mencionado na seção 2.1 deste trabalho, é proposto o desenvolvimento de um *serious game* digital para aprendizagem.

O jogo proposto, seguindo o esquema de classificação discutida na seção 2.5, é do gênero de Visualização de Ideias Abstratas, visando auxiliar a fixação de conceitos de estruturas de dados e algoritmos de ordenação expostos na disciplina, por meio da visualização do funcionamento de métodos de ordenação e interação com estruturas de dados utilizando seus métodos.

4.1 Objetivo Pedagógico e Público-Alvo

O objetivo principal do jogo é servir como uma ferramenta complementar de aprendizado para estudantes da disciplina de Estruturas de Dados, tipicamente cursada nos semestres iniciais de graduações em Ciência da Computação e áreas afins. A proposta foca em dois tópicos centrais que frequentemente apresentam desafios de abstração aos alunos:

- **Algoritmos de Ordenação:** Permitir que o aluno visualize, passo a passo, a execução de diferentes algoritmos de ordenação, associando o conceito teórico a uma representação lúdica e interativa.

- **Estruturas de Dados Fundamentais:** Proporcionar uma experiência prática na qual o jogador manipula estruturas como Pilhas, Listas Encadeadas e Árvores por meio de suas operações básicas (ex: *push*, *pop*, inserção), resolvendo quebra-cabeças que reforçam o entendimento de seu funcionamento.

4.2 Mecânicas e Regras

O jogo é estruturado em um ciclo que alterna dois tipos de fases, conforme detalhado no Apêndice A:

- **Fases de Combate (Algoritmos de Ordenação):** O jogador enfrenta um grupo de inimigos numerados e desordenados. Para derrotá-los, ele deve escolher um "feitiço", que é uma abstração de um algoritmo de ordenação (ex: *Bubble Sort*, *Merge Sort*). Ao escolher, o jogo executa uma animação que demonstra o algoritmo ordenando os inimigos passo a passo. O desempenho do jogador é medido por um sistema de pontuação fundamentado na eficiência da execução. O cálculo baseia-se em um "custo de mana" que penaliza o consumo excessivo de recursos: o custo total é composto por uma taxa fixa de conjuração do feitiço somada a um custo variável proporcional a cada operação (troca ou comparação) realizada. Dessa forma, a pontuação final da fase é obtida subtraindo esse custo acumulado de um valor máximo predefinido, recompensando, assim, as escolhas que resolvem o problema com o menor custo de energia para o personagem.

- **Fases de Obstáculo (Estruturas de Dados):** Entre os combates, o jogador encontra um quebra-cabeça que deve ser resolvido com a manipulação de uma estrutura de dados. Por exemplo, construir uma ponte usando operações de uma lista encadeada ou organizar degraus de uma escada seguindo as regras de uma pilha (LIFO - Last-In, First-Out), como no desafio da Torre de Hanói.

O jogo termina após uma sequência predefinida de fases, e o jogador recebe uma pontuação final e uma avaliação baseada em sua performance.

4.3 Análise e Escolha das Tecnologias

A seleção das tecnologias foi guiada pelos requisitos do projeto, que incluem a necessidade de desenvolvimento para plataformas móveis, um fluxo de trabalho 2D eficiente e

a viabilidade de implementação por um único desenvolvedor dentro de um cronograma acadêmico. Foi realizada uma análise comparativa entre as principais ferramentas disponíveis no mercado.

Foram consideradas tecnologias como o motor de jogos Unity[1], o motor de código aberto Godot[2], conhecido por sua leveza e robusto suporte a 2D, e o Unreal Engine[3], reconhecido por sua alta capacidade gráfica. Contudo, o Unreal Engine, com sua curva de aprendizado mais acentuada e foco em gráficos 3D de alta fidelidade, mostrou-se superdimensionado para as necessidades deste projeto.

Diante desse cenário, a análise comparativa concentrou-se nos motores Unity e Godot. Ambos apresentaram vantagens semelhantes, como a capacidade de exportação multiplataforma, ferramentas que facilitam o desenvolvimento bidimensional e a gratuidade de uso.

Optou-se, no entanto, pela Unity como ferramenta para a execução deste projeto. O fator decisivo para esta seleção foi a experiência prévia do autor com a plataforma e sua respectiva linguagem de programação.

Para a criação dos recursos gráficos, alinhado ao requisito de simplicidade e compatibilidade, optou-se pelo estilo *pixel art*. A ferramenta escolhida foi o aplicativo Aseprite, um editor especializado que oferece grande praticidade no trabalho com este estilo visual, além de facilitar a exportação de *spritesheets* e animações para o Unity.

4.4 Short Game Design Document

Por se tratar de um jogo com escopo reduzido e equipe enxuta, optou-se por utilizar um *Short GDD* (*Game Design Document*) para a documentação do projeto, conforme discutido na seção 2.6.1, uma vez que esse formato é recomendado para projetos com essas características. Nesse documento, toda a experiência do jogador é descrita em

[1] `https://unity.com`
[2] `https://godotengine.org`
[3] `https://www.unrealengine.com`

forma de texto corrido, conforme proposto por Motta e Trigueiro Jr. (2013). O artefato desenvolvido se encontra no apêndice A deste trabalho.

Com base nessa descrição textual, foram realizadas marcações por cores para identificar os elementos de produção necessários: elementos de design foram destacados em verde, elementos de implementação em amarelo e os elementos de áudio em roxo. A partir dessas marcações, foram extraídas atividades e organizadas em listas específicas para cada área envolvida na produção do jogo.

Figura 3: Lista de atividades a serem desenvolvidas

Assets de Design	Programação

• Arte de background
• botão iniciar jogo
• botões áudio e músicas
• Sprite mago
  ○ idle;
  ○ andando;
  ○ conjurando magias;
• Sprite inimigos
  ○ idle;
  ○ andando;
• Animações algoritmos de sort;
• Botão de ajuda;
• Dialog de ajuda;
• Tela obstáculos
  ○ Degraus torre de hanoi
  ■ botões pop e push
  ○ tábuas - pontes ligadas;
  ○ árvore mágica
  ■ frutos mágicos;
  ■ barreira mágica
  ○ Background desfiladeiro;
• Tela final
  ○ Troféus - bronze, prata, ouro, platina
	• Menu inicial;
	• Iniciar partida;
	• Habilitar/desabilitar sons;
	• Sorteador de encontros de combate;
	• passo a passo de ordenação
	  ○ Botões sorting;
	  ○ Insertion sort;
	  ○ Selection sort;
	  ○ Bubble sort;
	  ○ Merge Sort;
	  ○ Heapsort;
	• Dialog de ajuda;
	• Gerenciador de fases - alternadas entre combate e obstáculo;
	• Fase torre de Hanoi - Pilha;
	• Fase travessia do rio - Lista encadeada;
	• Fase árvore mágica - Árvore;
	• Fim de jogo

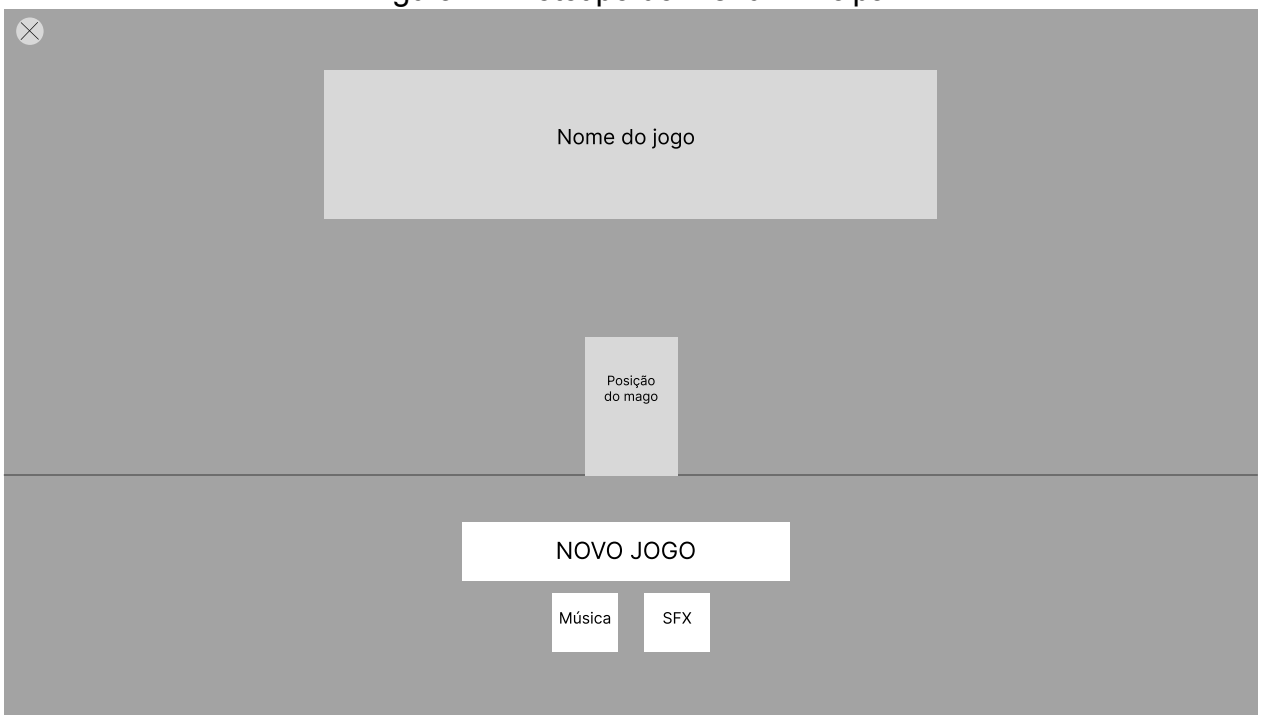
Assets áudio
• Música tema • Som de clique • som de vitória

Fonte:Autoria Própria

4.5 Prototipação de Telas

Utilizando a descrição do jogo realizada no artefato do *Short GDD*, foram criados protótipos para as telas de menu principal, tela de combate, fase de obstáculo de pilha e a tela final.

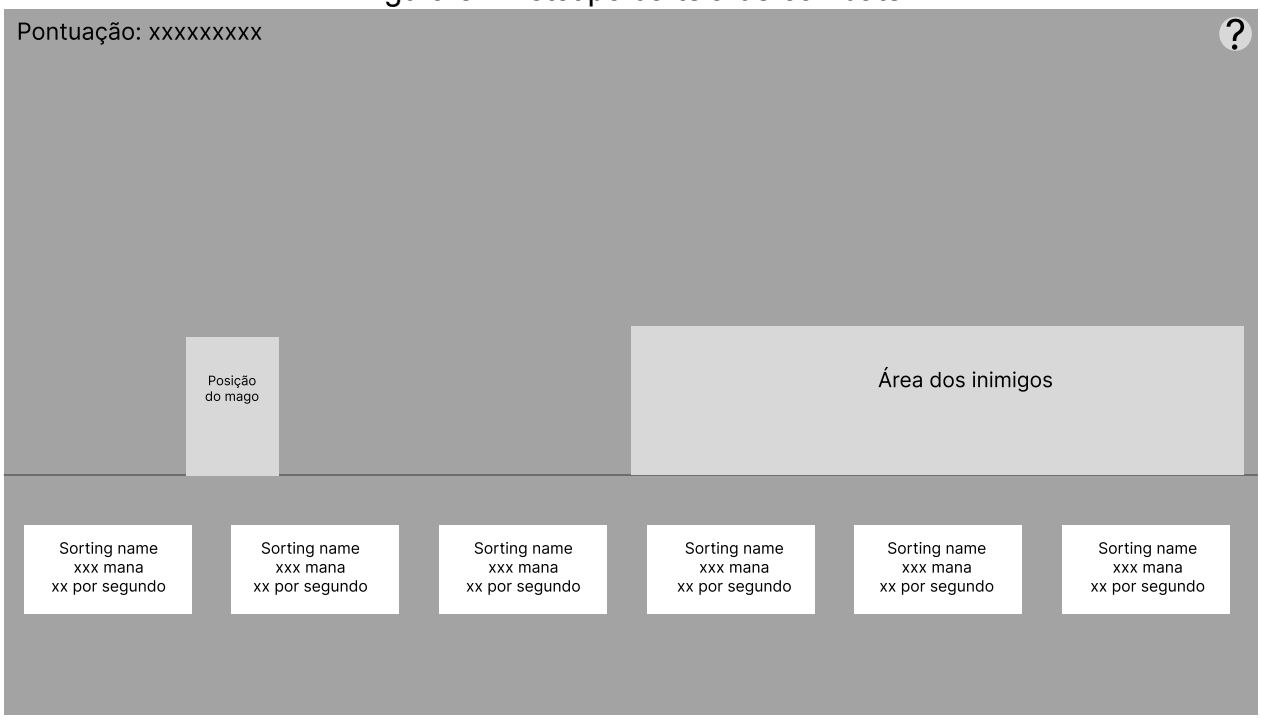
Figura 4: Protótipo do Menu Principal



Fonte:Autoria Própria

Na tela de menu inicial é mostrado o mago já ambientado no cenário do jogo, botões para desabilitar o áudio da música e dos efeitos de som e o botão para inicial o jogo. Ao iniciar o jogo, a cena transiciona para a tela de combate.

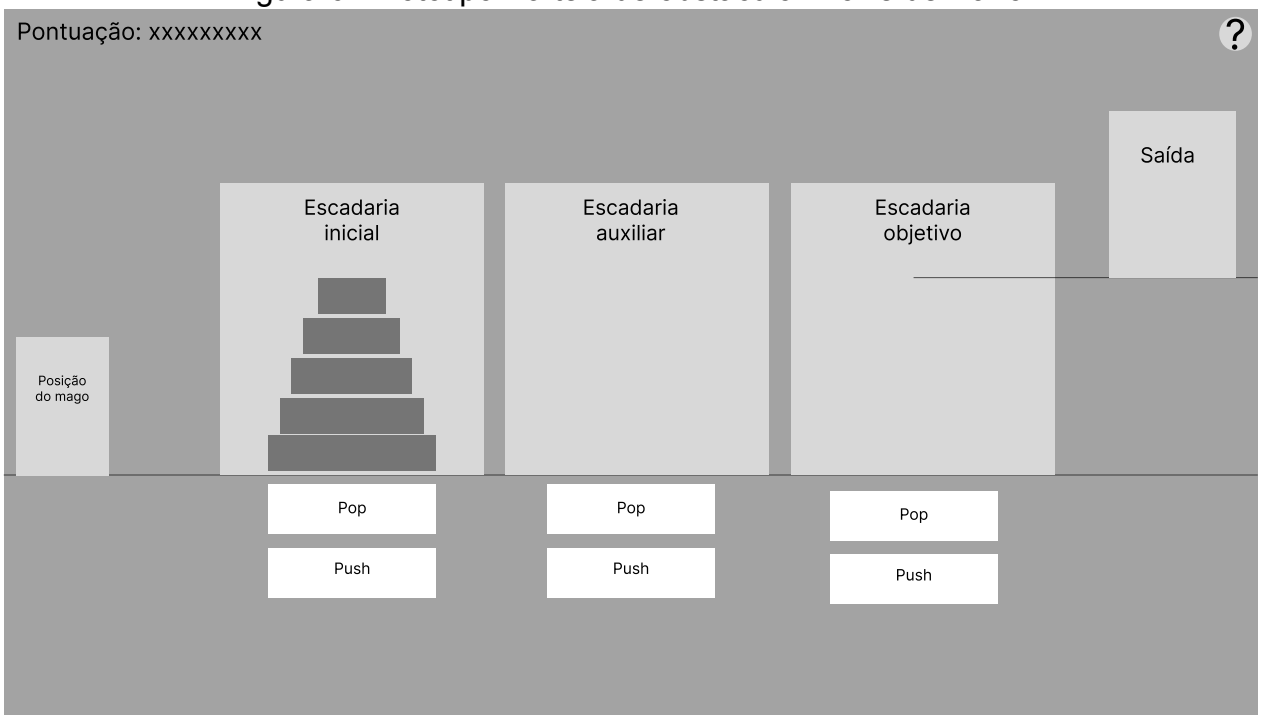
Figura 5: Protótipo da tela de combate



Fonte:Autoria Própria

A tela de combate apresenta os inimigos desordenados com os botões das magias (abstrações dos algoritmos de ordenação) abaixo como especificado na documentação do jogo. Ao selecionar um dos algoritmos é feita a ordenação passo a passo para a visualização do jogador. Nesta tela existe o botão de ajuda, que abre uma janela relembrando os principais pontos sobre cada algoritmo.

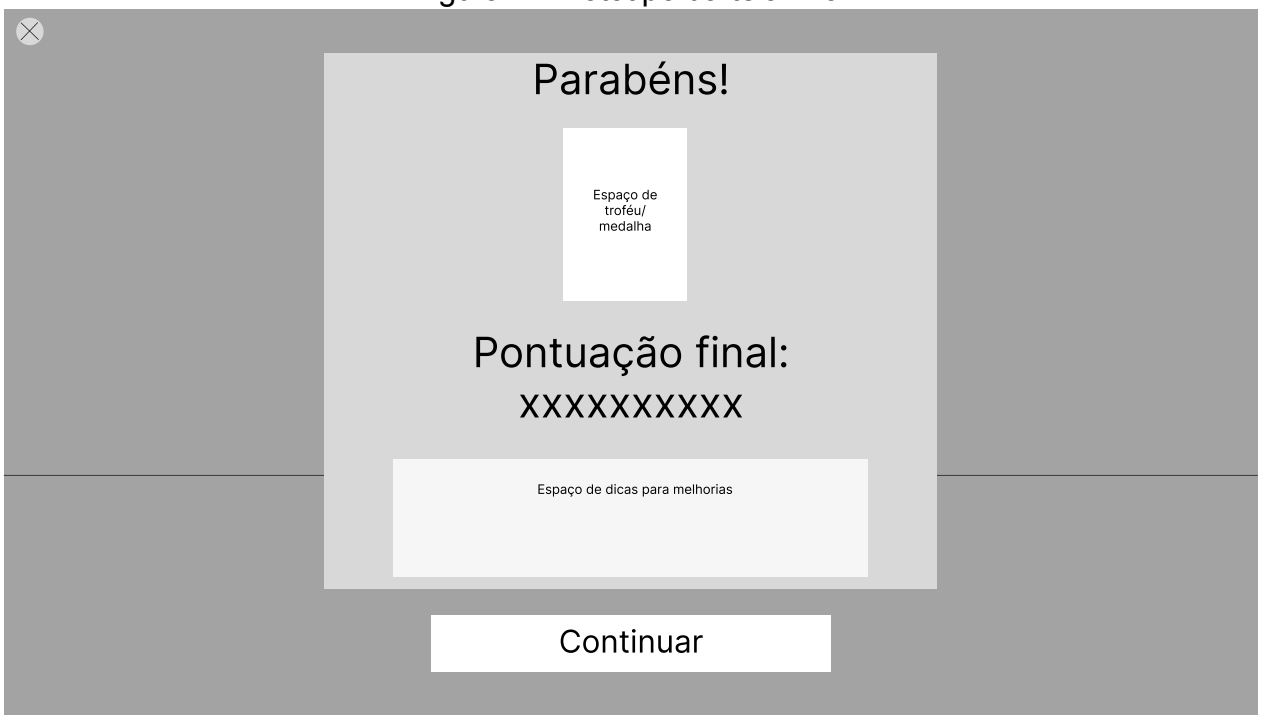
Figura 6: Protótipo Da tela de obstáculo - Torre de Hanoi



Fonte:Autoria Própria

Entre os combates, são apresentados ao jogador desafios com obstáculos que exigem a aplicação de estruturas de dados para serem superados. Para isso, o jogador deve interagir com os métodos correspondentes a essas estruturas a fim de resolver os problemas propostos. Inicialmente, foi prototipada a tela de um dos desafios, baseado na estrutura de pilha por meio da clássica Torre de Hanoi. Nesse desafio, o jogador deve manipular degraus de uma escadaria inicial utilizando os métodos *push* e *pop*, transferindo-os para uma escadaria objetivo, com o auxílio de uma terceira escadaria intermediária. A movimentação deve obedecer às regras da Torre de Hanoi, nas quais um elemento maior não pode ser colocado sobre um menor.

Figura 7: Protótipo da tela final



Fonte:Autoria Própria

Na tela final, são apresentadas ao jogador sua pontuação final e uma condecoração, que pode ser um troféu de bronze, prata, ouro ou platina, de acordo com seu desempenho ao longo do jogo. Além disso, são exibidas dicas, no espaço correspondente da prototipação, destacando os principais erros cometidos e os pontos que podem ser aprimorados pelo jogador.

5 SORTilege

Neste capítulo é apresentado o desenvolvimento do jogo SORTilege. Este nome foi escolhido para trazer uma ligação entre os dois pontos principais do jogo utilizando a palavra da língua inglesa “sortilege”, que pode ser utilizada como sinônimo de “magic” (magia) ou “sorcery” (feitiço), fazendo referência ao personagem principal, além de conter a palavra “Sort”, que tem sentido de ordenar algo, utilizada para se referir aos algoritmos de ordenação trabalhados no jogo (Sorting Algorithms).

5.1 Introdução

O jogo inicia no menu principal, onde o jogador pode optar por “Iniciar Jogo” (Modo História) ou selecionar uma fase específica na “Seleção de fases”. No modo principal, o jogo consiste em uma sequência de quatro fases de combate, intercaladas por três fases de obstáculos baseadas em estruturas de dados. O menu conta também com botões para habilitar ou desabilitar efeitos sonoros e a música do jogo.

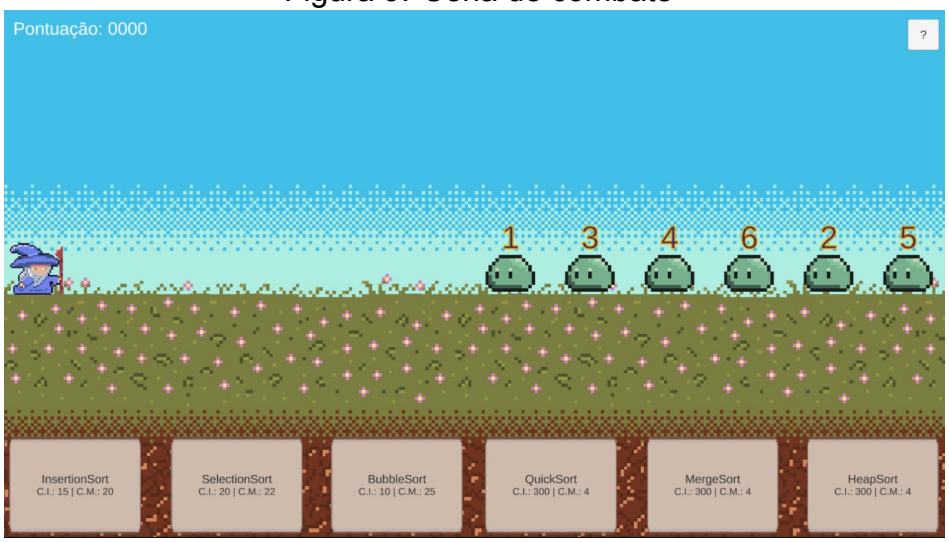
Figura 8: Menu principal

Fonte: Autoria Própria

Uma fase de combate confronta o jogador com um grupo de inimigos numerados,

cuja ordem inicial varia aleatoriamente. O jogador deve então escolher um algoritmo de ordenação, que são representados por magias que o mago pode conjurar, para derrotar as ameaças. Cada feitiço possui um “Custo de Mana Inicial” (abstração do custo de implementação) e um “Custo de Mana de Manutenção” (abstração da complexidade de tempo). Ao selecionar um feitiço, o algoritmo é executado visualmente, passo a passo, reordenando os inimigos antes de derrotá-los. A pontuação é concedida com base na eficiência da escolha do algoritmo para a situação apresentada.

Figura 9: Cena de combate



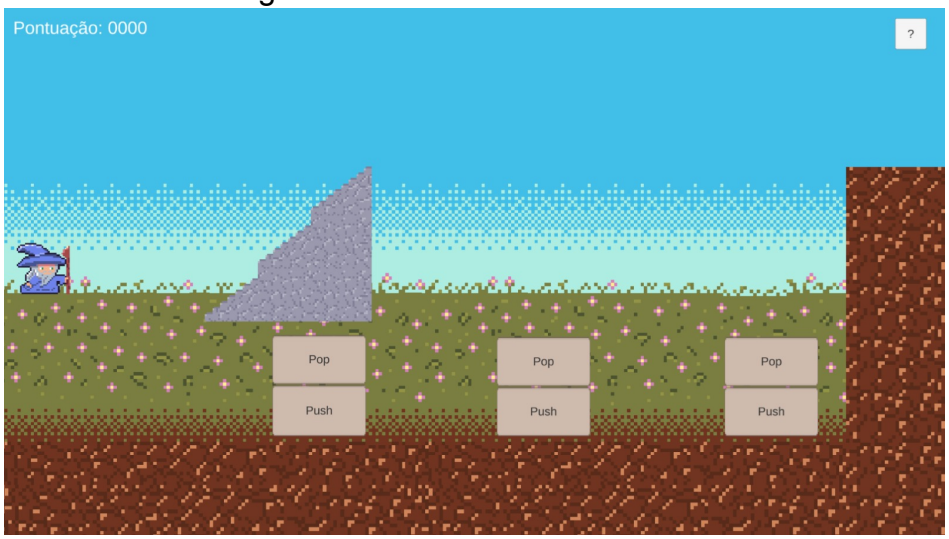
Fonte: Autoria Própria

Uma fase de obstáculo é um desafio lógico focado em uma estrutura de dados específica. O jogador deve manipular a estrutura de dados através de botões na interface que representam seus métodos fundamentais. O objetivo é usar essas operações para resolver um quebra-cabeça visual e prosseguir para a próxima fase, como construir uma escadaria, montar uma ponte ou cultivar uma Árvore de Busca Binária balanceada.

Desafio da Pilha (Torre de Hanoi): No desafio da Pilha, baseado no clássico quebra-cabeça da Torre de Hanoi, o jogador deve mover uma pilha de degraus de uma torre de origem para uma torre de destino, utilizando uma torre auxiliar. A interação é restrita aos métodos *Push* (empilhar) e *Pop* (desempilhar), forçando o jogador a seguir a regra

fundamental de que um degrau maior nunca pode ser posicionado sobre um menor.

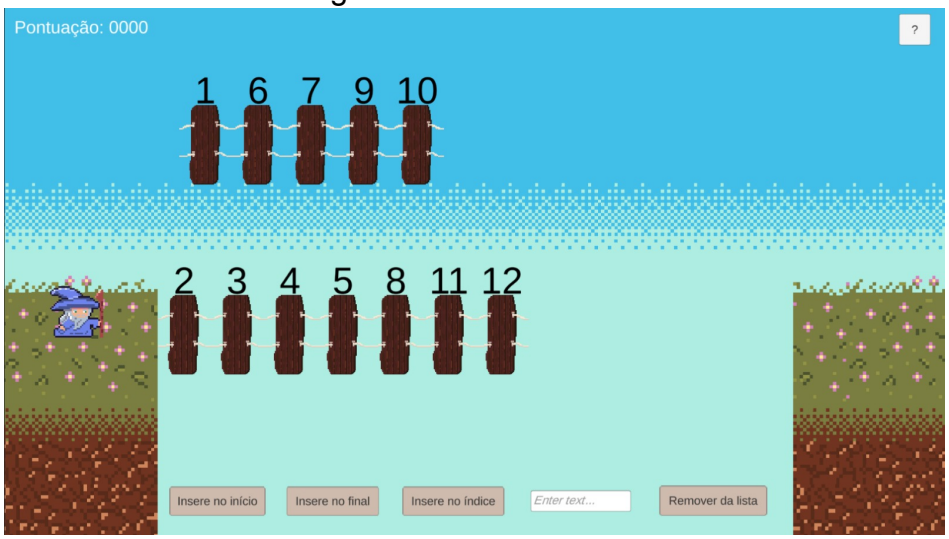
Figura 10: Cena da Torre de Hanoi



Fonte: Autoria Própria

Desafio da Lista Encadeada: No desafio da Lista Simplesmente Encadeada, o jogador precisa construir uma ponte em ordem numérica crescente, utilizando tábuas (nós) numeradas. O jogador manipula a lista, que já pode conter tábuas aleatórias no início do desafio, através de botões que representam as operações de *Inserção no Início*, *Inserção no Fim* e *Inserção em um Índice Específico*.

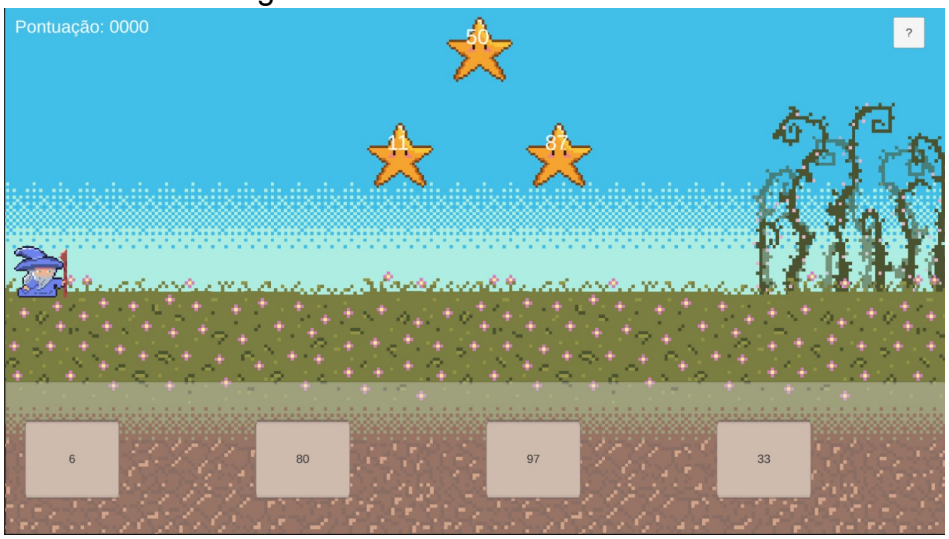
Figura 11: Cena da Lista



Fonte: Autoria Própria

Desafio da Árvore de Busca: No desafio da Árvore, o objetivo é inserir um conjunto de “frutos” (nós) numerados, seguindo a propriedade de uma Árvore de Busca Binária na qual para qualquer nó os valores menores ficam à esquerda, já os maiores à direita. O desafio central é a ordem de inserção, pois o jogador deve plantar os frutos em uma sequência que crie uma árvore estruturalmente balanceada para liberar a passagem.

Figura 12: Cena Desafio da Árvore

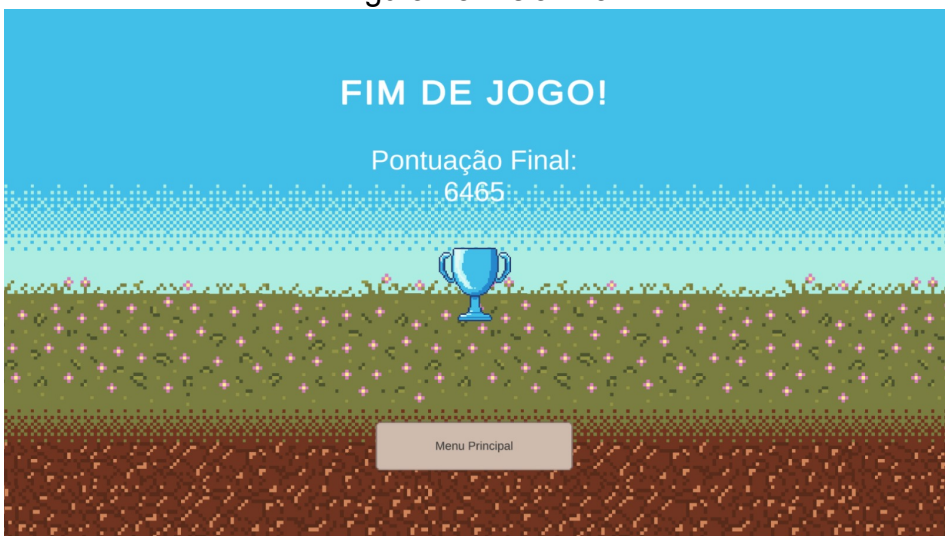


Fonte: Autoria Própria

Ao final da quarta fase de combate, o jogador é levado a uma tela final onde recebe uma pontuação e uma condecoração (troféu de bronze, prata, ouro ou platina) baseada em seu desempenho.

O desempenho é avaliado pela pontuação total, que é calculada com base na eficiência da escolha dos algoritmos nas fases de combate e no número de movimentos e tempo gasto nas fases de obstáculo.

Figura 13: Tela final



Fonte: Autoria Própria

5.2 Desenvolvimento do Software

Tomando como base os documentos de design de jogo (short GDD) criados previamente, foi iniciado o desenvolvimento do protótipo funcional. E como definido na proposta de desenvolvimento, a engine escolhida foi a Unity, utilizando a linguagem C# para a implementação das lógicas de jogo.

O desenvolvimento iniciou-se através do *Unity Hub*, *software* que gerencia projetos *Unity* e versões instaladas do *Game Engine*, onde foi criado um novo projeto utilizando o *template 2D Core*, que pré-configura o ambiente de desenvolvimento para um fluxo de trabalho bidimensional. Conforme a documentação oficial da Unity (Unity Technologies, 2025), a seleção deste *template* ajusta automaticamente as configurações principais do editor: a Câmera principal é definida com projeção *Orthographic* (Ortográfica) em vez de *Perspective*, a visualização da *Scene* (Cena) é travada no modo 2D, e o importador de *assets* é configurado por padrão para importar arquivos de imagem como *Sprite (2D and UI)*. Esta configuração inicial foi essencial para o desenvolvimento de um jogo com *pixel art*, garantindo que os *assets* fossem renderizados corretamente sem distorção de

perspectiva.

Cabe ressaltar que, embora o design de jogo, a arquitetura e a implementação do código-fonte tenham sido de autoria do responsável por este trabalho, todos os assets visuais (*sprites*) foram desenvolvidos pela artista Ana Paula Pacheco Luiz (Luiz, 2025).

A arquitetura do código foi estruturada visando a modularidade e a separação de responsabilidades (*Separation of Concerns*), permitindo que cada cena (Combate, Pilha, Lista, Árvore) tivesse seu próprio gerenciador, mas que estados globais (como pontuação e áudio) fossem controlados por serviços persistentes.

5.3 Estrutura Utilizada

A arquitetura central do jogo é sustentada por três padrões de design principais, adaptados das melhores práticas de desenvolvimento de jogos em Unity:

- Padrão *Singleton*: para gerenciar o estado do jogo que precisa persistir entre as cenas (como a pontuação total, o modo de jogo e a música), foram implementados os gerenciadores *GameSequenceManager* e *AudioManager* como *Singletons*. Ao utilizar *DontDestroyOnLoad*, garante-se que estes objetos não sejam destruídos durante as transições de cena, permitindo que o *AudioManager* toque música ininterruptamente e que o *GameSequenceManager* controle o fluxo correto das fases;

- Padrão *Strategy* (Scriptable Objects): o núcleo do balanceamento do jogo. Para evitar valores fixos no código dos custos de mana dos feitiços de ordenação, foi utilizado o padrão *Strategy*. Cada algoritmo possui um *SortAlgorithmData* (um *Scriptable Object*) que armazena seus custos. O *SortAlgorithmManager* apenas lê esses objetos. Isso permitiu um desacoplamento total da lógica de jogo e da lógica de balanceamento, permitindo a calibração dos custos sem a necessidade de recompilar o código-fonte;

- Padrão *Template Method*: para evitar a duplicação de código nas fases de obstáculo (Pilha, Lista, Árvore), foi criada uma classe-pai abstrata, *ObstacleManagerBase*. Esta classe define o "esqueleto" de um desafio: ela gerencia o tempo, a contagem de operações, o cálculo de pontuação e a lógica de transição de cena. As classes-filhas apenas implementam a lógica específica de sua estrutura de dados, herdando todo o comportamento comum.

5.4 Problemas e Soluções Encontradas

Durante o desenvolvimento, diversos desafios de implementação foram encontrados, exigindo soluções técnicas específicas para atender aos requisitos educacionais do jogo.

Inicialmente, o principal desafio encontrado foi com a visualização de algoritmos instantâneos, já que o jogo deveria tratar algoritmos de ordenação, que em execução normal são concluídos em milissegundos, em um processo passo a passo interativo e visual. A solução foi implementar cada algoritmo de ordenação como uma Corrotina (Coroutine) em C#. O uso de *yield return new WaitForSeconds()* permitiu pausar a execução do algoritmo em pontos críticos. O comando *yield return new WaitForSeconds(0.5f)* instrui a Unity a pausar a execução desta corrotina específica por 0.5 segundos, mas, diferentemente de um *sleep* tradicional, ele não congela o jogo. O motor da Unity continua a executar o *Update loop*, renderizando animações e processando outras entradas. Quando o tempo de espera termina, a execução da corrotina é retomada exatamente de onde parou. Isso foi essencial para criar pausas nas trocas e comparações, dando ao jogador tempo para observar o que estava acontecendo. Para reforçar visualmente a lógica, um sistema de troca de *sprites* foi implementado (utilizando as cores azul, vermelha e amarela) para destacar os inimigos que estavam sendo ativamente comparados, movidos, ou que já se encontravam em sua posição final ordenada.

Um segundo problema de design de software foi como gerenciar o balanceamento dos seis algoritmos de ordenação. Cada algoritmo possui um "Custo Inicial" e um "Custo de

Manutenção". Armazenar esses valores diretamente no código (*hard-coding*) do *SortAlgorithmManager* tornaria o processo de calibração, que é altamente iterativo, extremamente ineficiente, exigindo uma recompilação do *script* a cada pequeno ajuste. A solução foi utilizar *Scriptable Objects*. Foi criado um *script SortAlgorithmData*, que herda de *ScriptableObject*. Isso permitiu que cada algoritmo fosse representado como um *asset* independente no editor da Unity. O *SortAlgorithmManager* apenas mantém referências a esses *assets* de dados. Dessa forma, a lógica do jogo foi completamente desacoplada dos dados de balanceamento, permitindo ajustes finos nos custos de mana diretamente pelo editor, o que se provou essencial durante os testes de jogabilidade.

Um terceiro problema encontrado foi referente ao balanceamento dos custos das magias e à avaliação da escolha perfeita. O GDD especifica que o jogador é pontuado com base na sua escolha de algoritmo, e um troféu de platina é concedido para quem faz a escolha ideal em todas as fases. Como as ondas de inimigos são geradas aleatoriamente, o desafio era fazer o jogo saber qual algoritmo seria a melhor escolha em cada situação. A solução foi implementar um sistema de simulação "DryRun". Para cada algoritmo de ordenação visual, foi criada uma função *_DryRun* correspondente. Essas funções são duplicatas da lógica algorítmica, mas não contêm manipulação de *GameObjects*, *sprites* ou pausas (*yields*); elas apenas executam a ordenação em uma lista de inteiros e contam o número de operações. Assim que os inimigos aparecem, o *SortAlgorithmManager* executa todos os seis algoritmos *_DryRun*, calcula o custo total de mana de cada um com base nos *Scriptable Objects*, e armazena o algoritmo vencedor como a *_optimalChoiceData*.

6 AVALIAÇÃO DO JOGO

Após a implementação do jogo, detalhado no capítulo anterior, esta seção se dedica a avaliar a eficácia do artefato de *software* não apenas como um produto funcional, mas como uma ferramenta educacional. A simples existência do jogo não garante que ele cumpra seus objetivos pedagógicos de auxiliar no ensino de algoritmos e estruturas de dados.

Para realizar esta validação, é necessário aplicar um método de avaliação sistemático, que possa mensurar a percepção dos jogadores-alvo. Dentre os diversos modelos de avaliação de jogos educacionais, foi escolhido o modelo MEEGA+ (Petri et al., 2019).

A escolha deste modelo se justifica por ele ser um instrumento de medição sistemático, e por ter sido desenvolvido e validado especificamente dentro do contexto de avaliação de jogos para o ensino de computação (Petri et al., 2019), alinhando-se perfeitamente aos objetivos deste trabalho.

6.1 O Modelo MEEGA+

O MEEGA+ (Modelo para a Avaliação de Jogos Educacionais) é uma evolução do modelo MEEGA, desenvolvido por Petri, Gresse von Wangenheim e Borgatto (2019). O modelo se propõe a avaliar a qualidade dos jogos educacionais sob a perspectiva dos alunos no contexto de cursos superiores na área de computação e estrutura sua análise em dois fatores de qualidade principais, conforme identificado pela análise fatorial dos autores do modelo: Usabilidade e Experiência do Jogador (Petri et al., 2019).

Na categoria usabilidade é avaliada a facilidade de uso do jogo. Este fator é decomposto nas dimensões de Aprendizibilidade (quão fácil é aprender a jogar), Operabilidade (quão fácil é controlar o jogo), Estética (design da interface), Acessibilidade e Proteção contra erros do usuário (Petri et al., 2019). Já a categoria de experiência do jogador mensura a percepção geral do aluno. É decomposto em dimensões como Atenção Focada,

Diversão, Desafio, Interação Social, Confiança, Relevância (conexão com a disciplina), Satisfação e, crucialmente, a Aprendizagem Percebida (Petri et al., 2019).

Na prática, o modelo consiste em um questionário padronizado, contendo 35 itens para jogos digitais, onde os participantes respondem em uma escala Likert de 5 pontos. O MEEGA+ possui alta confiabilidade estatística (Alfa de Cronbach $\alpha = 0,928$) (Petri et al., 2019), o que o torna um instrumento robusto para a coleta de dados de avaliação do jogo.

6.2 Execução

A etapa de execução da avaliação foi conduzida aplicando o jogo em uma aula regular da disciplina de Estruturas de Dados, ministrada pelo professor orientador deste trabalho. O grupo de participantes constituiu-se de 21 estudantes do curso de Sistemas de Informação da Universidade Federal de Santa Catarina, sendo 14 estudantes do sexo masculino e 7 estudantes do sexo feminino, todos com idade entre 18 e 39 anos.

O jogo foi aplicado próximo ao final do segundo semestre letivo de 2025, portanto os alunos já possuíam conhecimento prévio dos assuntos abordados neste trabalho, como algoritmos de ordenação e estruturas de dados fundamentais, sendo assim a ferramenta foi avaliada como instrumento de revisão dos conceitos vistos no semestre.

O procedimento da aplicação do jogo em sala de aula iniciou-se com uma breve apresentação expositiva, utilizando apresentação de slides, na qual foram detalhados o conceito do jogo, narrativa e o funcionamento das mecânicas envolvidas nas fases de combate e de resolução de obstáculos.

Ao final da apresentação foi disponibilizado aos alunos o artefato de software para que estes pudessem experimentar o jogo na prática. Foi exposto aos estudantes um *QR Code* contendo o *link* para *download* do arquivo de instalação para o sistema operaciona *Android*, adicionalmente, visando a facilidade de acesso para o maior número possível de estudantes, foi disponibilizada uma versão web do jogo.

Após um período de aproximadamente 20 minutos no qual os estudantes puderam jogar livremente o jogo, foi reservado um intervalo de aproximadamente 10 minutos para a coleta das percepções dos usuários. Os participantes responderam ao questionário do modelo MEEGA+ por meio de um formulário eletônico na plataforma *Google Forms*.

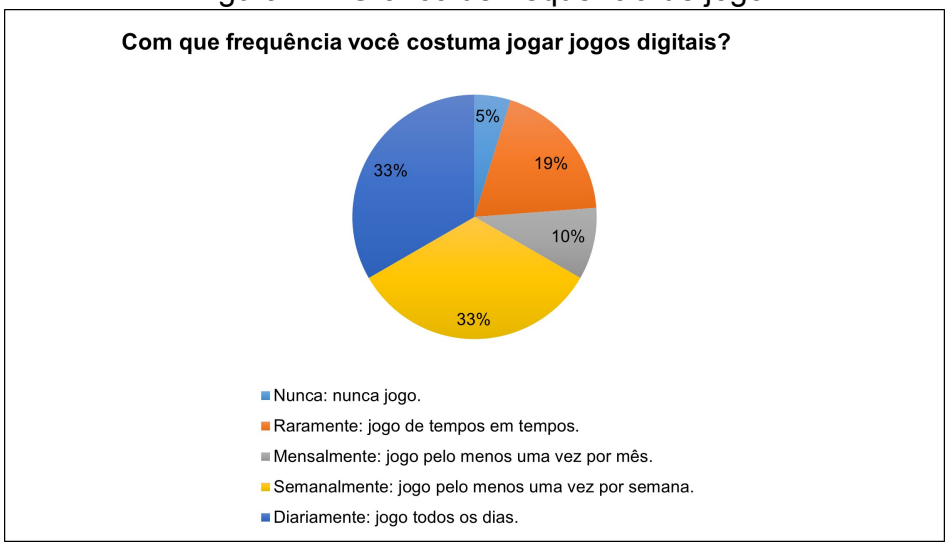
Para o processamento dos dados coletados e geração de tabelas e gráficos de resultados, foi utilizada a planilha de análise de dados e o guia de instruções fornecidos pelos autores do modelo MEEGA+. Devido à indisponibilidade do repositório oficial dos arquivos no momento da elaboração deste trabalho, o acesso aos recursos originais foi realizado por meio da ferramenta *Wayback Machine*, utilizando um *snapshot* datado de 25 de junho de 2025.

6.3 Resultados

Utilizando os formulários respondidos pelos estudantes, foi preenchida a planilha auxiliar de análise de dados disponibilizada para o MEEGA+, na qual foram atribuídos valores numéricos para os conceitos apresentados no questionário, sendo valores variando entre -2, representando o conceito “Discordo totalmente”, até 2, representando “Concordo Totalmente”. A seguir é apresentado o resultado da análise dividida em três dimensões principais: perfil dos participantes, avaliação da usabilidade e avaliação da experiência do jogador e percepção de aprendizagem.

A primeira etapa da análise dos dados consiste na caracterização do perfil da amostra. O formulário coletou informações demográficas e dados sobre o hábito de jogar dos estudantes. Nesta etapa observa-se, utilizando a figura 14, que 95% dos participantes possuem entre 18 a 28 anos enquanto 5% estão entre 29 a 39 anos. Observando os gráficos gerados pela planilha, nota-se também que apenas 5% dos participantes não têm o hábito de jogar, o que mostra que grande parte da amostra têm alguma familiaridade com jogos digitais, sendo que 66% dos participantes dizem jogar semanal ou diariamente.

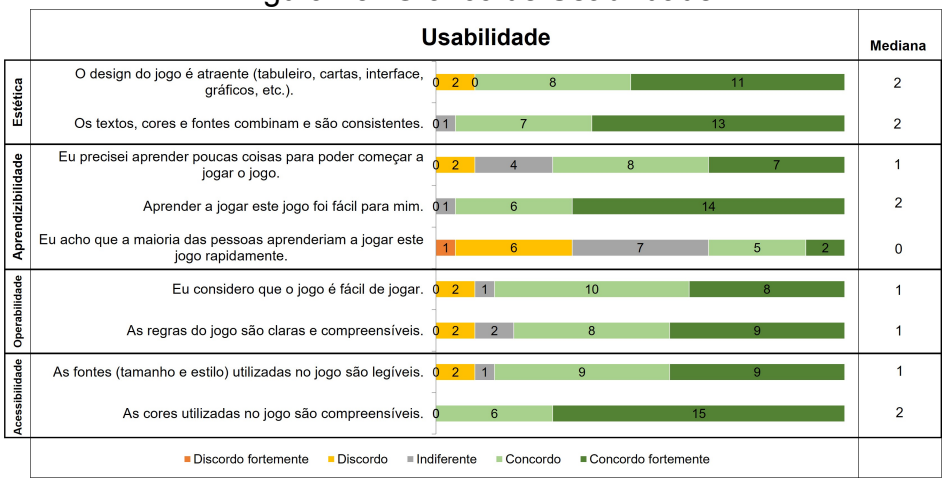
Figura 14: Gráfico de frequência de jogo



Fonte: Gerado pela planilha auxiliar do MEEGA+, utilizando dados recolhidos em questionário

Já na figura 15, que trata da avaliação da usabilidade do jogo, pode-se observar que os resultados indicam uma aceitação positiva quanto à interface e estética do jogo. Destaca-se que a categoria de Estética e as questões que tratam a facilidade pessoal para aprender a jogar e a facilidade de compreender as cores utilizadas no jogo obtiveram avaliações com a mediana 2, que representa “Concordo Totalmente”. Um destaque negativo é a mediana 0 para o item “Eu acho que a maioria das pessoas aprenderiam a jogar este jogo rapidamente”.

Figura 15: Gráfico de Usabilidade



Fonte: Gerado pela planilha auxiliar do MEEGA+, utilizando dados recolhidos em questionário

Durante a aplicação do jogo em sala, e nos campos abertos do questionário, foram feitos comentários que apontam algumas melhorias na usabilidade do jogo. Os participantes tiveram alguma dificuldade em entender rapidamente qual o objetivo ao entrar em uma nova fase, apesar do jogo contar com um botão de ajuda e explicação na parte superior direita da interface, um ponto de melhoria poderia ser dar mais destaque ao botão, já que alguns jogadores não perceberam a presença do botão em um primeiro momento, além de apresentar a ajuda e explicação na primeira vez que o jogador estiver jogando aquele tipo de fase, sem a necessidade de apertar o botão de ajuda.

Além disso foram encontrados pelos participantes alguns problemas que afetaram a usabilidade do jogo, entre eles problemas de resolução no aplicativo WEB, que fez com que alguns botões não aparecessem, especialmente os botões da fase da lista encadeada, e problemas em abrir o teclado na fase da ponte, utilizado para inserir uma tábua em um índice específico, em dispositivos da marca *Xiaomi*.

Por fim, a análise da experiência do jogador, na figura 16, aponta que o jogo trouxe diversão e desafios em um ritmo adequado, os participantes também apontaram que o conteúdo do jogo é interessante para seus interesses e está relacionado com a disciplina. Outro destaque positivo retirado da percepção dos jogadores através do questionário é que o jogo contribuiu para relembrar conceitos vistos em sala de aula. Já os principais pontos fracos percebidos pelos participantes foram nos pontos de atenção focada, algo que poderia ser melhorado com uma narrativa mais envolvente, além da falta de interação social.

Figura 16: Gráfico de Experiência do Jogador

Fonte: Gerado pela planilha auxiliar do MEEGA+, utilizando dados recolhidos em questionário

7 CONSIDERAÇÕES FINAIS

O objetivo principal deste trabalho foi desenvolver um serious game educativo, que servisse como ferramenta auxiliar na fixação de conceitos fundamentais da disciplina de Estruturas de Dados, com ênfase em algoritmos de ordenação e estruturas de dados. Para alcançar este fim, foram traçados e cumpridos objetivos específicos que guiaram a pesquisa e o desenvolvimento do artefato.

Inicialmente, foram sintetizados conceitos fundamentais sobre a evolução dos jogos digitais, gamificação e serious games, proporcionando a base teórica necessária para compreender como estas ferramentas podem engajar estudantes e facilitar o aprendizado de conteúdos complexos. Identificou-se, através do esquema de classificação de gamificação, que a proposta se alinhava ao gênero de Visualização de Ideias Abstratas (VAI), utilizando a interatividade para tornar concretos os processos algorítmicos.

A análise de trabalhos correlatos permitiu identificar lacunas e oportunidades, inspirando a criação de uma solução que unisse a visualização passo a passo dos algoritmos com mecânicas de jogo envolventes, como o combate e a resolução de quebra-cabeças, além da inspiração sobre métodos de avaliação de jogos voltados para a educação.

A escolha pela engine Unity e a linguagem C#, em conjunto com a arte em pixel art, possibilitou a criação de um ambiente 2D leve e funcional, além de ter possibilitado a exportação de artefatos para múltiplos ambientes, como plataformas web e dispositivos mobile, de forma simples e com poucas adaptações. O desenvolvimento apresentou desafios técnicos específicos, como a necessidade de transformar algoritmos de ordenação, que normalmente são de execução instantânea, em processos visuais interativos.

A avaliação do SORTilege, realizada através da aplicação do modelo MEEGA+ com estudantes da área de computação, apontou resultados majoritariamente positivos. Entre os destaques, observou-se uma boa aceitação quanto à relevância do jogo para a disciplina e a percepção de aprendizagem e diversão por parte dos alunos. A experiência

do jogador foi classificada como satisfatória, indicando que o jogo cumpriu seu papel de engajar e motivar.

Entretanto, a avaliação também revelou pontos de atenção na usabilidade. Foram relatadas dificuldades na compreensão imediata dos objetivos de novas fases e problemas de interface em situações específicas, como a impossibilidade de abertura do teclado virtual em alguns dispositivos móveis e falhas em algumas resoluções na versão Web. Estes feedbacks são valiosos e indicam que a clareza das instruções e a responsividade da interface são aspectos cruciais que demandam refinamento.

Por fim, como melhorias futuras, sugere-se a correção dos problemas de usabilidade identificados, como dar maior destaque ao botão de ajuda, apresentar explicações ao entrar em novas fases e otimizar a interface para diferentes resoluções de tela. Sugere-se também a geração de um artefato instalável para dispositivos iOS, a fim de evitar problemas de compatibilidade com a versão web. Além disso, poderia ser implementada uma seleção de dificuldade, apresentando maiores desafios aos jogadores, tais como cenários de ordenação mais complexos ou estruturas de dados maiores (por exemplo, uma Torre de Hanói com mais degraus ou uma árvore binária de busca com mais níveis).

REFERÊNCIAS

BECKER, Katrin. **What's the difference between gamification, serious games, educational games, and game-based learning?** Academia Letters, [S.L.], n. 209, p. 1-4, 3 fev. 2021. Academia.edu. http://dx.doi.org/10.20935/al209

BATTISTELLA, P. E. e WANGENHEIM, C. G. Von. (2016). **ENgAGED: Um Processo de Desenvolvimento de Jogos para Ensinar Computação**, In SBIE – Anais do XXVII Simpósio Brasileiro de Informática na Educação – SBIE (pp. 380-389), SBC. DOI: 10.5753/cbie.sbie.2016.380

CARVALHO, J. V. de; RENNER FILHO, A.; OLIVEIRA, A. C. R. de; SILVA, V. da; GRAÇA GOMES, M. da. **Dispositivos vestíveis aplicados no ensino**. Revista Observatório, v. 4, n. 3, p. 509–539, abr. 2018. Disponível em: `https://www.researchgate.net/publication/324833028_DISPOSITIVOS_VESTIVEIS_APLICADOS_NO_ENSINO`. Acesso em: 2 dez. 2024.

CHINN, P.; PRINS, J.; TENENBERG, J. **The role of the data structures course in the computing curriculum**. Journal of Computing in Small Colleges, p. 91–93, 2003. Disponível em: `https://faculty.washington.edu/jtenenbg/publications/chinnRoleDataStructures-ccsc2003.pdf`. Acesso em: 2 dez. 2024.

DICHEVA, Darina; HODGE, Austin. **Active Learning through Game Play in a Data Structures Course**. Proceedings Of The 49Th Acm Technical Symposium On Computer Science Education, [S.L.], p. 834-839, 21 fev. 2018. ACM. http://dx.doi.org/10.1145/3159450.3159605.

FREIRE, Gabriel Gonçalves; GUERRINI, Daniel. **Os Jogos na Sociedade Contemporânea: as influências dos avanços tecnológicos**. Revista de Ensino, Educação e Ciências Humanas, [S.L.], v. 17, n. 5, p. 463, 30 dez. 2016. Editora e Distribuidora Educacional. http://dx.doi.org/10.17921/2447-8733.2016v17n5p463-469.

FUNDAÇÃO GETULIO VARGAS. **Pesquisa revela que Brasil tem 480 milhões de dispositivos digitais em uso, sendo 2,2 por habitante**. Disponível em: `https://portal.fgv.br/noticias/pesquisa-revela-brasil-tem-480-milhoes-dispositivos-digitais-uso-sendo-22-habitante`. Acesso em: 2 dez. 2024.

GLATZ, Igor. **Desenvolvimento de um Jogo para Auxílio no Ensino de Estruturas de Dados**. 2023. 168 f. TCC (Graduação) - Curso de Sistemas de Informação, Universidade Federal de Santa Catarina, Florianópolis, 2023.

HIRA, Willian Kenji; MARINHO, Marcus Vinícius Prata; PEREIRA, Felipe Barros; BARBOZA JR., Alcides T. **Criação de um modelo conceitual para documentação de game design**. SIMPÓSIO BRASILEIRO DE GAMES E ENTRETENIMENTO DIGITAL – SBGAMES, 15., 2016, São Paulo. São Paulo: Sociedade Brasileira de Computação, 2016. Art & Design Track – Full Papers. p. 329–336. ISSN 2179-2259

INTERNET ARCHIVE. **Wayback Machine**. 2025. Disponível em: https://web.archive.org/web/20250 evaluation/meega-plus/. Acesso em: 17 nov. 2025.

KIM, Bohyun. **UNDERSTANDING GAMIFICATION**. Library Technology Reports, [S.L.], v. 51, n. 2, p. 5-35, mar. 2015. American Library Association. http://dx.doi.org/10.5860/ltr.51n2.

LEE, Michael J. et al. **Principles of a debugging-first puzzle game for computing**

education. 2014 Ieee Symposium On Visual Languages And Human-Centric Computing (Vl/Hcc), [S.L.], p. 57-64, jul. 2014. IEEE. http://dx.doi.org/10.1109/vlhcc.2014.6883023.

LUIZ, Ana Paula Pacheco. **Portfólio Ana Paula Pacheco Luiz**. Behance, 2025. Disponível em: https://www.behance.net/anapaupacheco4. Acesso em: 10 nov. 2025.

MOTTA, R. L., TRIGUEIRO JUNIOR, J., **Short game design document (SGDD): Documento de game design aplicado a jogos de pequeno porte e advergames: Um estudo de caso do advergame Rockergirl Bikeway**. SBGames 2013 - XII Symposium on Computer Games and Digital Entertainment. p. 115-121. São Paulo: 2013. Disponível em: <http://www.sbgames.org/sbgames2013/proceedings/artedesign/15-dt-paper_SGDD.pdf>. Acesso em: 06 de julho de 2025.

OKADA, Alexandra; SHEEHY, Kieron. **O VALOR DA DIVERSÃO NA APRENDIZAGEM ON-LINE: um estudo apoiado na pesquisa e inovação responsáveis e dados abertos**. Revista E-Curriculum, [S.L.], v. 18, n. 2, p. 590- 613, 26 jun. 2020. Pontificia Universidade Catolica de Sao Paulo (PUC-SP). http://dx.doi.org/10.23925/1809-3876.2020v18i2p590-613. Disponível em: https://revistas.pucsp.br/curriculum/article/view/48014. Acesso em: 2 dez. 2024.

PETRI, Giani; WANGENHEIM, Christiane Gresse von; BORGATTO, Adriano Ferreti. MEEGA+: Um Modelo para a Avaliação de Jogos Educacionais para o ensino de Computação. Revista Brasileira de Informática na Educação, [S.l.], v. 27, n. 03, p. 52-81, dez. 2019. ISSN 2317-6121. Disponível em: <https://br-ie.org/pub/index.php/rbie/article/view/v27n035281>. Acesso em: 10 nov. 2025. doi:http://dx.doi.org/10.5753/rbie.2019.27.03.52.

PGB – PESQUISA GAME BRASIL. **PGB 2024: perfil do jogador brasileiro**. 11. ed. São Paulo: Sioux Group; Go Gamers; Blend New Research; ESPM, 2024. Disponível em: https://pesquisagamebrasil.com.br. Acesso em: 6 jul. 2025.

POLITOWSKI, Cristiano; FONTOURA, Lisandra M.; PETRILLO, Fabio; GUÉHÉNEUC, Yann-Gaël. **Learning from the past: a process recommendation system for video game projects using postmortems experiences**. Arxiv, [S.L.], p. 1-19, mar. 2018. ArXiv. http://dx.doi.org/10.48550/ARXIV.2009.02445.

ROLAND, L. C.; FABRE, M.-C. J. M.; KONRATH, M. L. P.; TAROUCO, L. M. R. **Jogos educacionais**. RENOTE, Porto Alegre, v. 2, n. 1, 2004. DOI: 10.22456/1679-1916.13719. Disponível em: `https://seer.ufrgs.br/index.php/renote/article/view/13719`. Acesso em: 2 dez. 2024.

SOUSA, Tamires A. S.; MARQUES, Anna B. S.. **LEARN Board Game: A Game for Teaching Software Architecture Created through Design Science Research**. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE (SBES), 34. , 2020, Natal. Porto Alegre: Sociedade Brasileira de Computação, 2020 . ISSN 2833-0633.

SPANIER, Adam; HARMS, Sherri Weitl; HASTINGS, John. **A Classification Scheme for Gamification in Computer Science Education: discovery of foundational gamification genres in data structures courses**. 2021 Ieee Frontiers In Education Conference (Fie), [S.L.], p. 1-9, 13 out. 2021. IEEE. http://dx.doi.org/10.1109/fie49875.2021.9637447.

TODA, Armando M. et al. **Interactive learning environment for data structures with gamification concepts**. In: IADIS International Conference. 2013. Disponível em: https://www.researchgate.net/publication/321081460. Acesso em: 6 jul. 2025.

UNITY TECHNOLOGIES. **2D and 3D mode settings**. Unity Documentation. Disponível em: `https://docs.unity3d.com/Manual/2DAnd3DModeSettings.html`. Acesso em: 10 nov. 2025.

VON WANGENHEIM, Aldo; SILVA, Alexandre Goncalves. **Plano de Ensino INE5408 - Estruturas de Dados**. Universidade Federal de Santa Catarina. Centro Tecnológico. Departamento de Informática e Estatística, Florianópolis, 21 out. 2025.

ZHANG, Jinghua; ATAY, Mustafa; CALDWELL, Elvira R.; JONES, Elva J.. **Reinforcing student understanding of linked list operations in a game**. 2015 Ieee Frontiers In Education Conference (Fie), [S.L.], p. 1-7, out. 2015. IEEE. http://dx.doi.org/10.1109/fie.2015.7344132.

Apêndices

Apêndice A –Short Game Design Document

Neste jogo educativo, o jogador controla um mago que enfrenta desafios por meio da magia dos algoritmos e interação com estruturas de dados. Alternando entre batalhas contra grupos de inimigos, onde cada feitiço corresponde a um algoritmo de ordenação, e fases de obstáculos baseadas em estruturas como pilhas, listas e árvores. O jogo consiste em uma sequência de quatro fases de combate separadas por fases de obstáculos e ao final o jogador recebe um troféu baseado no seu desempenho ao otimizar a escolha dos algoritmos de ordenação e superação dos obstáculos de estruturas de dados.

O jogo inicia no menu principal, que terá a arte do background de jogo ao fundo e o botão de “Iniciar jogo” e botões para desabilitar o áudio das músicas e efeitos sonoros do jogo. Música tema. O mago, personagem principal, aparece em animação ociosa (idle) no meio da tela, sobre um chão estilo plataforma 2d. Ao clicar em “iniciar jogo” (som de clique) o mago muda para a animação de “andando” e a câmera se desloca para a direita. É sorteado um encontro de combate, o grupo inimigo surge andando a partir da borda direita da tela. Um grupo inimigo é composto de criaturas numeradas de 1 até n, onde n é o tamanho do grupo. Os inimigos variam em quantidade e na disposição inicial do grupo (grupo quase ordenado, completamente desordenado o qualquer configuração intermediária). Surgem os botões que representam os poderes do mago (algoritmos de sort) na parte de baixo da tela, cada botão contém: o nome do algoritmo de sort; o custo de energia inicial (abstração do custo de implementação); o custo de energia para manutenção do poder (abstração da complexidade do algoritmo).

Cada algoritmo de sort está ligado a um elemento ou tipo de magia clássicas de RPG. Os algoritmos de sort disponíveis serão:

- Insertion sort;
- Selection sort;

- Bubble sort;
- Quick sort;
- Merge sort;
- Heap sort.

Na tela de jogo aparece também a pontuação no canto superior esquerdo e um botão de ajuda no canto superior direito, clicar neste botão faz aparecer uma dialog contendo as informações básicas de cada algoritmo de sorting.

Ao selecionar um dos poderes o mago muda para a animação de conjuração de poderes. Simultaneamente acontece a animação do algoritmo escolhido, ordenando os inimigos seguindo o passo a passo do método de ordenação. Após o término da animação, os inimigos são derrotados, desaparecendo da tela com um fade out, toca o som de vitória, é calculada a pontuação baseada na quantidade de energia restante, que é somado à pontuação pessoal.

Em seguida o mago sai pela borda direita da tela em direção ao próximo desafio (animação de caminhar). A tela transiciona para a próxima fase, na qual o jogador deve superar um obstáculo utilizando uma estrutura de dados, este obstáculo é sorteado a partir da lista de estruturas de dados:

- Pilhas: o mago deve construir uma escadaria utilizando uma pilha de degraus, em um desafio clássico de torre de hanoi. Na tela há três espaços para uma escadaria, apenas uma levando para o caminho (seria o pino de destino na torre de Hanoi). Abaixo de cada espaço de escadaria aparecem dois botões: Pop, que remove o degrau mais acima e coloca ele no topo da tela, no espaço auxiliar e Push, que empilha o degrau do espaço auxiliar.

- Lista simplesmente encadeada: Para passar sobre um desfiladeiro o mago deve construir uma ponte de madeira, composta por tábuas (nós) numeradas, cada tábua

possui uma corda, para encadear a próxima tábua. A ponte já inicia com algumas tábuas aleatórias e deve ser construída em ordem crescente, para construir o jogador deve clicar em uma tábua, fazendo surgir os botões: inserção no início, inserção no final e inserção em um índice específico (com uma caixa de texto).

- Árvore: Para atravessar uma barreira mágica, o mago deve cultivar uma Árvore, inserindo frutos mágicos numerados em seus galhos. A árvore inicia apenas com a raiz e o jogador recebe um conjunto de frutos para plantar. O desafio é inserir todos os frutos, seguindo a propriedade de uma Árvore de Busca Binária (frutos com valor menor à esquerda, e com valor maior à direita), tendo que escolher o local correto para que a árvore cresça de forma balanceada e libere a passagem.

Esta tela segue o mesmo layout da tela anterior (Pontuação canto superior esquerdo, ajuda canto superior direito), o mago fica em animação ociosa próximo à lateral esquerda da tela, com o obstáculo centralizado, abaixo os botões que representam os métodos da estrutura sorteada. Ao completar o desafio e superar o obstáculo, o mago transiciona para a próxima fase, desta vez de combate.

O jogo segue alternando entre uma fase de combate e uma fase de obstáculo de estruturas de dados, ao final da quarta fase de combate, acontece o final de jogo, aparece a tela final, onde é apresentada a pontuação final e dada uma condecoração de acordo com o desempenho do jogador (troféu de bronze/prata/ouro/platina).

Apêndice B – MANUAL DE INSTALAÇÃO E EXECUÇÃO DO PROJETO

Para a execução e compilação do projeto, é necessário o software Unity Hub disponível em `https://unity.com/download`, com ele é possível gerenciar projetos e versões da *engine* Unity. Através do Unity Hub é necessário fazer a instalação da versão 2022.3.40f1 da engine.

O projeto encontra-se hospedado no repositório institucional da Universidade Federal de Santa Catarina. O acesso pode ser realizado através do endereço eletrônico:

`https://codigos.ufsc.br/fabio.hac/SORTilege`

Configuração do Ambiente Unity

1. Abra o **Unity Hub**.
2. Certifique-se de que a versão **2022.3.40f1** está instalada na aba *Installs*. Caso não esteja, clique em *Install Editor*, acesse a aba *Archive* ou *Official Releases* e localize a versão citada.
3. Na aba *Projects*, clique no botão **Add** (ou *Open* → *Add project from disk*).
4. Navegue até o diretório onde o repositório foi clonado ou extraído. Selecione a pasta raiz do projeto e confirme.
5. O projeto aparecerá na lista. Clique sobre o nome do projeto para abrir o editor Unity.

Execução do Jogo

Após a abertura do Editor do Unity:

1. No painel inferior, na janela **Project**, navegue até a pasta `Assets > Scenes`.

2. Localize a cena inicial do jogo chamada MainMenu. Clique duas vezes para abri-la.
3. Para iniciar a simulação, clique no botão **Play** localizado na barra de ferramentas superior central do editor.
4. O jogo será executado dentro da aba **Game**.

Apêndice C – Artigo

SORTilege: Jogo Digital para Auxílio à Aprendizagem de Estruturas de Dados e Algoritmos de Ordenação

Fábio Henrique Antunes Coelho[1], **José Eduardo de Lucca**[1]

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

Abstract. *Data Structures is a fundamental course in Computer Science but presents high difficulty rates due to the abstraction of its concepts. This paper presents SORTilege, an educational digital game of the Visualization of Abstract Ideas (VAI) genre, developed to assist in the fixation of sorting algorithms and linear and non-linear structures. The game was developed using the Unity engine and evaluated using the MEEGA+ model with 21 undergraduate students. Results indicate positive acceptance regarding pedagogical relevance and player experience, highlighting the tool's potential as a support for traditional teaching.*

Resumo. *A disciplina de Estruturas de Dados é fundamental para a Ciência da Computação, mas apresenta altos índices de dificuldade devido à abstração de seus conceitos. Este artigo apresenta o SORTilege, um jogo digital educativo do gênero Visualização de Ideias Abstratas (VAI), desenvolvido para auxiliar na fixação de algoritmos de ordenação e estruturas lineares e não lineares. O jogo foi desenvolvido na engine Unity e avaliado através do modelo MEEGA+ com 21 estudantes de graduação. Os resultados indicam uma aceitação positiva quanto à relevância pedagógica e à experiência do jogador, evidenciando o potencial da ferramenta como apoio ao ensino tradicional.*

1. Introdução

Nos últimos anos, o uso de dispositivos móveis, como *tablets* e, principalmente, *smartphones*, tem crescido muito. No Brasil, atualmente, a média de dispositivos móveis é de 1,8 por habitante; quando é considerado apenas a quantidade de *smartphones*, a média é de 1,2 dispositivos por brasileiro [Fundação Getúlio Vargas 2024]. Como o uso desses aparelhos está difundido por toda a sociedade e usá-los já é algo cotidiano, é interessante utilizar estes aparelhos como ferramenta educacional, dando outra perspectiva sobre o assunto visto em sala de aula, podendo levar mais diversão na aprendizagem, desviando para a direção oposta da ansiedade e improdutividade [Okada e Sheehy 2020].

A disciplina de Estruturas de Dados é um pilar na formação em Ciência da Computação, crucial para o desenvolvimento de habilidades de programação e resolução de problemas. No entanto, a metodologia de ensino predominante, que se baseia em aulas expositivas e exercícios práticos, tem se mantido inalterada por décadas, negligenciando o potencial de dispositivos tecnológicos modernos, como os *smartphones* [Carvalho et al. 2018]. Essa abordagem tradicional, aliada à natureza abstrata e complexa dos conceitos, e ao papel de transição da disciplina entre conteúdos básicos e avançados [Chinn et al. 2003], frequentemente resulta em dificuldades significativas para os alunos, impactando seu desempenho acadêmico.

Para enfrentar esse desafio, as tecnologias educacionais, como os jogos educativos, têm sido cada vez mais exploradas como ferramentas eficazes para engajar e facilitar o aprendizado; afinal, elas podem trazer diversão junto com a aprendizagem e aumentar a retenção do conteúdo ensinado [Roland et al. 2004]. Os jogos podem criar ambientes interativos e imersivos, nos quais os alunos experimentam, aplicam e aprendem conceitos de forma prática, criando um vínculo afetivo, o que pode não acontecer com os métodos tradicionais, já que, quando estão em aula, os alunos podem não alcançar um resultado satisfatório, criando um bloqueio no seu processo de aprendizagem, o que os afasta desse vínculo [Roland et al. 2004].

A sociedade contemporânea testemunha uma interação cada vez mais dinâmica com os avanços tecnológicos, que têm moldado e sido moldados por diversas práticas culturais, incluindo os jogos. A era digital, iniciada por volta da década de 1970, impulsionou uma explosão de jogos eletrônicos, que hoje integram múltiplas formas de arte e tecnologias gráficas avançadas, oferecendo uma vasta gama de experiências culturais [Rodrigues *apud* Freire & Guerrini 2016]. Essa evolução transformou a indústria de jogos eletrônicos em um gigante econômico global, superando o faturamento da indústria cinematográfica [Bauman; Reis; Cavichiolli; Carvalho *et al.* *apud* Freire & Guerrini 2016]. No Brasil, a Pesquisa Game Brasil 2024 [PGB 2024] revelou que o hábito de consumir jogos digitais atingiu 73,9% da população, evidenciando a consistência e o alcance massivo desse mercado no país.

Nesse contexto, a gamificação surge como uma abordagem promissora. Ela não se define como a criação de um jogo completo, mas sim como a aplicação de elementos de jogos, como pontos, regras, *feedback* e participação voluntária, em contextos não relacionados a jogos, com o propósito de tornar atividades cotidianas mais envolventes e divertidas [Kim 2015].

Outro conceito que surge neste contexto é o de jogos sérios e, mais especificamente, os jogos educacionais [Becker 2021], categoria proeminente dentro dos *serious games*.

A literatura acadêmica corrobora os múltiplos benefícios associados ao emprego de jogos na educação. Eles são capazes de aumentar a eficácia da aprendizagem, o interesse, a motivação e a persistência dos alunos. Por serem intrinsecamente divertidos, os jogos facilitam o aprendizado e elevam a capacidade de retenção do conteúdo, funcionando como uma "ginástica mental" [Roland et al. 2004]. Um dos aspectos mais relevantes é a promoção da aprendizagem ativa, na qual os alunos se engajam em atividades como diálogo, resolução de problemas e pensamento crítico, em vez de serem meros receptores passivos de informação. Essa abordagem alinha-se diretamente com as teorias construtivistas da educação, que valorizam a participação e a experimentação do sujeito na construção de seu próprio conhecimento. Especificamente, jogos digitais oferecem vantagens distintivas na simulação de sistemas complexos, na automação de regras e no fornecimento de *feedback* imediato e individualizado, tornando-os particularmente adequados para o ensino de conceitos abstratos [Roland et al. 2004].

Este trabalho propõe o **SORTilege**, um jogo digital educativo focado na visualização interativa de algoritmos de ordenação e na manipulação de estruturas de dados fundamentais (pilhas, listas e árvores), com o objetivo de servir como ferramenta

complementar ao ensino da disciplina.

2. Trabalhos Relacionados

A literatura apresenta diversas iniciativas que utilizam jogos para o ensino de computação. No âmbito específico de estruturas de dados, destacam-se:

- **Stack Game**: Focado no ensino de Pilhas, integrando conceituação e implementação de código [Dicheva e Hodge 2018].
- **Space Traveler**: Utiliza a mecânica de "Snake" para ensinar operações em listas encadeadas, exigindo que o aluno complete trechos de código [Zhang et al. 2015].
- **Sort It Out**: Uma ferramenta para o ensino de algoritmos de ordenação que permite a execução passo a passo dos métodos [Glatz 2023].

O *SORTilege* se diferencia ao integrar tanto a ordenação quanto as estruturas de dados em um único fluxo de jogo ("loop"), combinando mecânicas de combate RPG com quebra-cabeças lógicos do gênero de Visualização de Ideias Abstratas (VAI) [Spanier et al. 2021].

3. Desenvolvimento do SORTilege

O jogo foi desenvolvido utilizando a *game engine* **Unity** e a linguagem **C#**. O jogo é bidimensional e o estilo de arte é *pixel art* visando simplicidade e compatibilidade com diversos dispositivos.

3.1. Mecânicas de Jogo

No modo principal, o jogo consiste em uma sequência de quatro fases de combate, intercaladas por três fases de obstáculos baseadas em estruturas de dados. Cabe ressaltar que, embora o design de jogo, a arquitetura e a implementação do código-fonte tenham sido de autoria do responsável por este trabalho, todos os assets visuais (*sprites*) foram desenvolvidos pela artista Ana Paula Pacheco Luiz [Luiz 2025].

Uma fase de combate confronta o jogador com um grupo de inimigos numerados, cuja ordem inicial varia aleatoriamente. O jogador deve então escolher um algoritmo de ordenação, que são representados por magias que o mago pode conjurar, para derrotar as ameaças. Cada feitiço possui um "Custo de Mana Inicial" (abstração do custo de implementação) e um "Custo de Mana de Manutenção" (abstração da complexidade de tempo). Ao selecionar um feitiço, o algoritmo é executado visualmente, passo a passo, reordenando os inimigos antes de derrotá-los. A pontuação é concedida com base na eficiência da escolha do algoritmo para a situação apresentada.

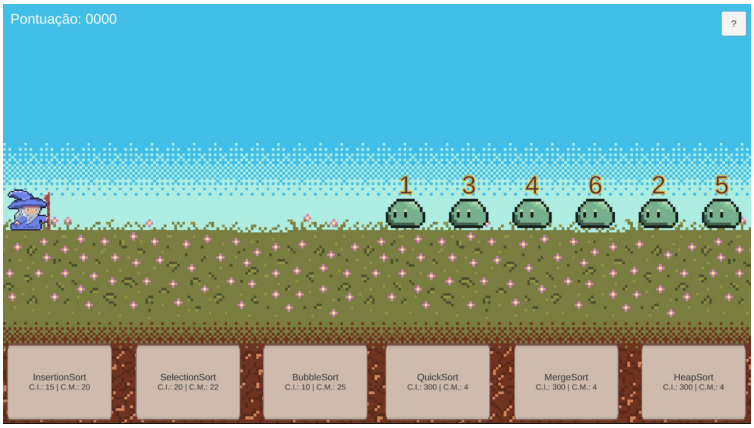


Figura 1. Cena de combate

Uma fase de obstáculo é um desafio lógico focado em uma estrutura de dados específica. O jogador deve manipular a estrutura de dados através de botões na interface que representam seus métodos fundamentais. O objetivo é usar essas operações para resolver um quebra-cabeça visual e prosseguir para a próxima fase, como construir uma escadaria, montar uma ponte ou cultivar uma Árvore de Busca Binária balanceada.

Desafio da Pilha (Torre de Hanoi): No desafio da Pilha, baseado no clássico quebra-cabeça da Torre de Hanoi, o jogador deve mover uma pilha de degraus de uma torre de origem para uma torre de destino, utilizando uma torre auxiliar. A interação é restrita aos métodos *Push* (empilhar) e *Pop* (desempilhar), forçando o jogador a seguir a regra fundamental de que um degrau maior nunca pode ser posicionado sobre um menor.

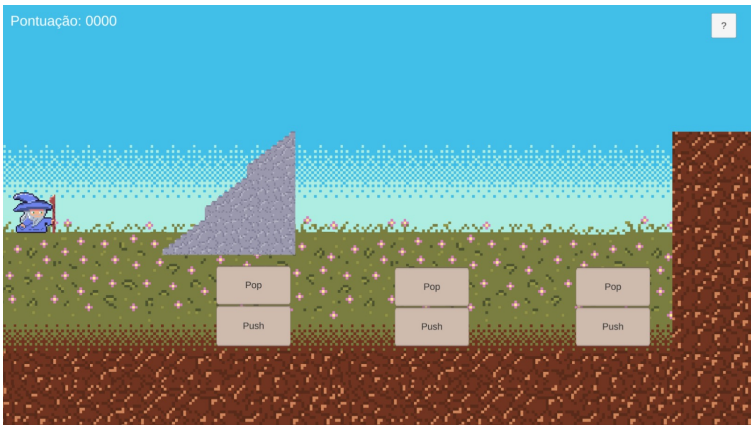


Figura 2. Cena da Torre de Hanoi

Desafio da Lista Encadeada: No desafio da Lista Simplesmente Encadeada, o jogador precisa construir uma ponte em ordem numérica crescente, utilizando tábuas (nós) numeradas. O jogador manipula a lista, que já pode conter tábuas aleatórias no início do desafio, através de botões que representam as operações de *Inserção no Início*, *Inserção no Fim* e *Inserção em um Índice Específico*.

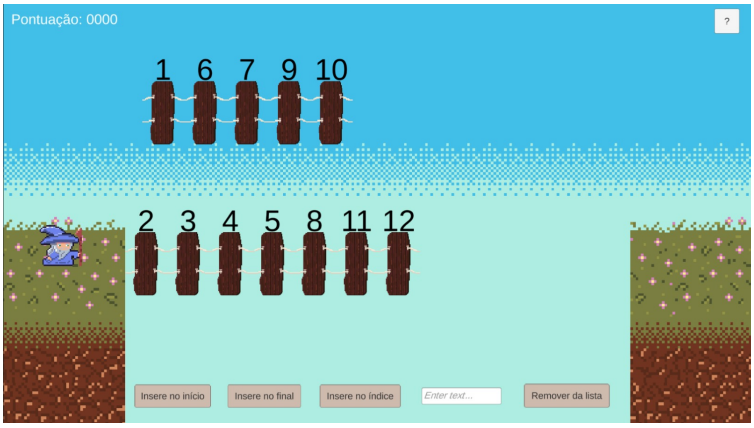


Figura 3. Cena da Lista

Desafio da Árvore de Busca: No desafio da Árvore, o objetivo é inserir um conjunto de "frutos" (nós) numerados, seguindo a propriedade de uma Árvore de Busca Binária na qual para qualquer nó os valores menores ficam à esquerda, já os maiores à direita. O desafio central é a ordem de inserção, pois o jogador deve plantar os frutos em uma sequência que crie uma árvore estruturalmente balanceada para liberar a passagem.

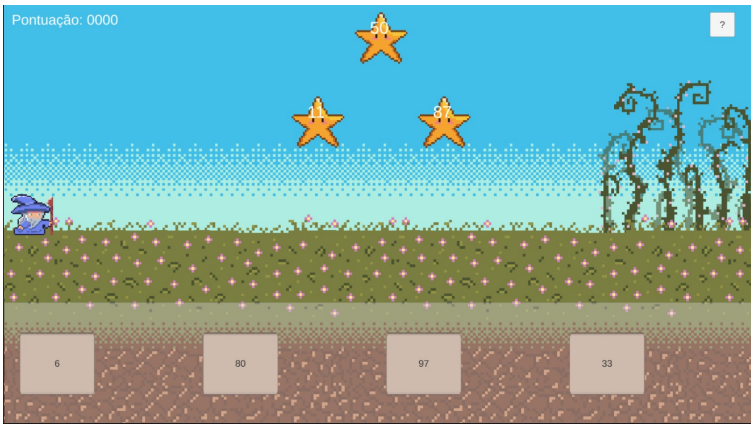


Figura 4. Cena Desafio da Árvore

Ao final da quarta fase de combate, o jogador é levado a uma tela final onde recebe uma pontuação e uma condecoração (troféu de bronze, prata, ouro ou platina) baseada em seu desempenho.

O desempenho é avaliado pela pontuação total, que é calculada com base na eficiência da escolha dos algoritmos nas fases de combate e no número de movimentos e tempo gasto nas fases de obstáculo.

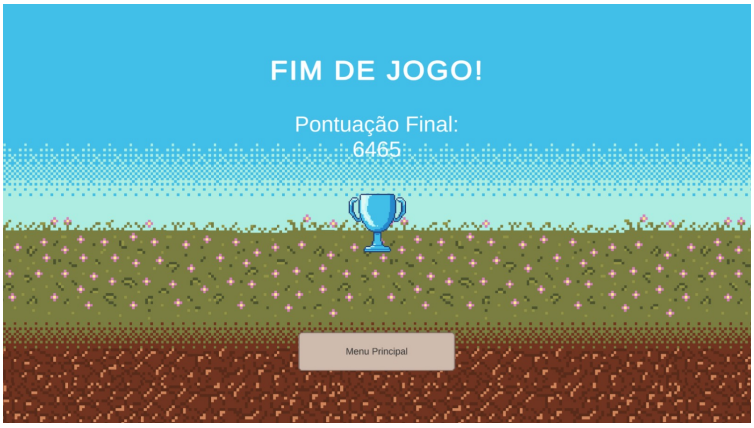


Figura 5. Tela final

4. Metodologia de Avaliação

A avaliação do jogo foi realizada através de um estudo de caso com 21 estudantes do curso de Sistemas de Informação da Universidade Federal de Santa Catarina (UFSC). O procedimento consistiu em uma sessão de jogo de 20 minutos seguida pela aplicação do questionário **MEEGA+** [Petri et al. 2019], um modelo padronizado para avaliação de jogos educacionais em computação que analisa as dimensões de Usabilidade e Experiência do Jogador.

5. Resultados e Discussão

Os dados coletados apontaram resultados majoritariamente positivos.

- **Experiência do Jogador**: A maioria dos participantes concordou que o jogo é relevante para a disciplina e auxilia na revisão de conceitos. As dimensões de "Diversão" e "Satisfação" obtiveram médias altas, indicando que o jogo cumpriu o objetivo de ser engajador.
- **Usabilidade**: O jogo foi considerado fácil de jogar, mas foram identificados pontos de melhoria. Alguns participantes relataram dificuldades em entender imediatamente o objetivo de novas fases e problemas de interface na versão *Web* e em dispositivos móveis específicos (sobreposição do teclado virtual).

O feedback qualitativo sugeriu que, embora a mecânica seja sólida, a clareza das instruções iniciais (tutoriais) e a responsividade da interface precisam de refinamentos para garantir uma experiência fluida em diferentes resoluções.

6. Conclusão

A avaliação do SORTilege, realizada através da aplicação do modelo MEEGA+ com estudantes da área de computação, apontou resultados majoritariamente positivos. Entre os destaques, observou-se uma boa aceitação quanto à relevância do jogo para a disciplina e a percepção de aprendizagem e diversão por parte dos alunos. A experiência do jogador foi classificada como satisfatória, indicando que o jogo cumpriu seu papel de engajar e motivar.

Entretanto, a avaliação também revelou pontos de atenção na usabilidade. Foram relatadas dificuldades na compreensão imediata dos objetivos de novas fases e problemas de interface em situações específicas, como a impossibilidade de abertura do teclado virtual em alguns dispositivos móveis e falhas em algumas resoluções na versão Web. Estes feedbacks são valiosos e indicam que a clareza das instruções e a responsividade da interface são aspectos cruciais que demandam refinamento.

Por fim, como melhorias futuras, sugere-se a correção dos problemas de usabilidade identificados, como dar maior destaque ao botão de ajuda, apresentar explicações ao entrar em novas fases e otimizar a interface para diferentes resoluções de tela. Sugere-se também a geração de um artefato instalável para dispositivos iOS, a fim de evitar problemas de compatibilidade com a versão web. Além disso, poderia ser implementada uma seleção de dificuldade, apresentando maiores desafios aos jogadores, tais como cenários de ordenação mais complexos ou estruturas de dados maiores (por exemplo, uma Torre de Hanói com mais degraus ou uma árvore binária de busca com mais níveis).

Referências

Becker, K. (2021). **What's the difference between gamification, serious games, educational games, and game-based learning?** Academia Letters, 209:1–4.

Carvalho, J. V. de, Renner Filho, A., Oliveira, A. C. R. de, Silva, V. da, e Graça Gomes, M. da (2018). **Dispositivos vestíveis aplicados no ensino.** Revista Observatório, 4(3):509–539.

Chinn, P., Prins, J., e Tenenberg, J. (2003). **The role of the data structures course in the computing curriculum.** Journal of Computing in Small Colleges, p. 91–93.

Dicheva, D. e Hodge, A. (2018). **Stack Game: A way to learn stacks.** In Proceedings of the 49th ACM Technical Symposium on Computer Science Education, p. 834–839. ACM.

Freire, G. G. e Guerrini, D. (2016). **Os Jogos na Sociedade Contemporânea: as influências dos avanços tecnológicos.** Revista de Ensino, Educação e Ciências Humanas, 17(5):463.

Fundação Getúlio Vargas (2024). **Pesquisa revela que Brasil tem 480 milhões de dispositivos digitais em uso, sendo 2,2 por habitante**. Disponível em: `https://portal.fgv.br/noticias/pesquisa-revela-brasil-tem-480-milhoes-dispositivos-digitais-uso-senc` Acesso em: 2 dez. 2024.

Glatz, I. (2023). **Desenvolvimento de um Jogo para Auxílio no Ensino de Estruturas de Dados**. TCC (Graduação) - UFSC, Florianópolis.

Kim, B. (2015). **Understanding Gamification.** Library Technology Reports, 51(2):5–35.

Luiz, A. P. P. (2025). Portfólio Ana Paula Pacheco Luiz. Behance. Disponível em: `https://www.behance.net/anapaupacheco4`. Acesso em: 10 nov. 2025.

Okada, A. e Sheehy, K. (2020). **O valor da diversão na aprendizagem on-line: um estudo apoiado na pesquisa e inovação responsáveis e dados abertos.** *Revista E-Curriculum*, 18(2):590–613.

PGB (2024). **PGB 2024: perfil do jogador brasileiro**. 11. ed. São Paulo: Sioux Group.

Petri, G., Wangenheim, C. G. v., e Borgatto, A. F. (2019). **MEEGA+: Um Modelo para a Avaliação de Jogos Educacionais para o ensino de Computação.** Revista Brasileira de Informática na Educação, 27(03):52–81.

Roland, L. C. et al. (2004). **Jogos educacionais na construção do conhecimento.** RENOTE, 2(1).

Spanier, A., Harms, K., e Hastings, J. (2021). **Gamification in Data Structures: A Classification Scheme.** In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, p. 1–9.

Zhang, Y., Atay, M., Caldwell, E. R., e Jones, E. J. (2015). **Space Traveler: A Game for Teaching Linked Lists.** In Proceedings of the 2015 IEEE Frontiers in Education Conference, p. 1–7.