Arqueiro Flynn: Um Jogo Educacional para o Ensino das Arquiteturas Paralelas

Archer Flynn: An Educational Game for the Parallel Architectures

Luan Q. Aracaty[1], **Thaís M. Quaresma**[1], **Josivaldo de S. Araújo**[1]

[1]Instituto de Ciências Exatas e Naturais (ICEN) – Universidade Federal do Pará (UFPA)
Rua Augusto Corrêa nº 01 - 66.075-110 - Belém - PA - Brasil

lnsantos482@gmail.com, thaismquaresma@gmail.com, josivaldo@ufpa.br

Abstract. Introduction: *Serious games are methodologies that have gained space in the teaching and learning process, because they characterize an important alternative in the construction of knowledge.* ***Objective:*** *This work presents the development and evaluation of an educational game aimed at teaching about Parallel Architectures, but specifically the classification of Flynn.* ***Methodology or Steps:*** *The game was designed based on the Tower Defense genre, where the goal is to develop skills such as strategic planning, problem solving and critical thinking. The implementation uses Python and Pygame, with progressive levels that introduce concepts such as competition, synchronization and parallelism.* ***Results:*** *The preliminary results indicate a positive and motivating evaluation of the game's usability.*
Keywords *Educational Game, Parallel Architecture, Flynn's Taxonomy*

Resumo. Introdução: *Jogos Educacionais são metodologias que ganharam espaço no processo de ensino e aprendizagem, pois caracterizam uma importante alternativa na construção do conhecimento.* ***Objetivo:*** *Este trabalho apresenta o desenvolvimento e a avaliação de um jogo educacional voltado para o ensino sobre Arquiteturas Paralelas, mas especificamente a classificação de Flynn.* ***Metodologia ou Etapas:*** *O jogo foi projetado baseado no gênero Tower Defense, onde o objetivo é desenvolver habilidades como planejamento estratégico, resolução de problemas e raciocínio crítico. A implementação utiliza Python e Pygame, com níveis progressivos que introduzem conceitos como concorrência, sincronização e paralelismo.* ***Resultados:*** *Os resultados preliminares indicam uma avaliação positiva e motivadora da usabilidade do jogo.*
Palavras-Chave *Jogo Educacional, Arquitetura Paralela, Taxonomia da Flynn.*

1. Introdução

O ensino da Programação Paralela assume um importante papel na formação de estudantes de Computação, especialmente, diante do crescente aumento por demanda em soluções computacionais que exigem alta eficiência e desempenho. A Computação de Alto Desempenho (CAD) permite executar múltiplos processos simultaneamente, otimizando o tempo de tarefas grandes e complexas [Hardasmal et al. 2020]. No entanto, muitos estudantes enfrentam dificuldades em compreender os conceitos fundamentais dessa área, particularmente nos períodos iniciais, onde predominam as abordagens da programação dita sequencial [Mullen et al. 2020].

Hoje, todos os processadores são multinúcleos e as arquiteturas paralelas passaram da exceção a regra. Entre as classificações das arquiteturas paralelas, a mais famosa é a Classificação de Flynn, pois categoriza as arquiteturas conforme o número de instruções e dados que podem ser processados em paralelo. Esta classificação define os modelos como SISD (*Single Instruction Single Data*), SIMD (*Single Instruction Multiple Data*), MISD (*Multiple Instruction Single Data*) e MIMD (*Multiple Instruction Multiple Data*), cada um com suas características e aplicações específicas. O processo de ensino de forma tradicional, talvez torne a compreensão desses conceitos um pouco abstrata.

Nesse contexto, a utilização de jogos eletrônicos educacionais têm ganhado destaque como ferramentas eficazes para o ensino e a aprendizagem em diversas áreas do conhecimento, incluindo da Computação [Amaral and Sant'Ana 2024]. Os jogos podem não apenas engajar os alunos, mas também, proporcionar um ambiente interativo onde conceitos complexos são explorados de maneira mais acessível e prática [Manoharan and Ye 2020].

A proposta deste trabalho é apresentar o desenvolvimento de um jogo educacional que visa integrar o conceito de Programação Paralela, mais em específico, a classificação de Flynn. O objetivo é oferecer uma experiência de aprendizagem que não apenas introduza os aspectos técnicos da Programação Paralela, mas também, promova a motivação e o engajamento dos alunos, a partir da resolução de problemas reais dentro do ambiente de jogo.

2. Trabalhos Relacionados

Diversos estudos destacam o potencial dos Jogos Educacionais no ensino e como eles podem auxiliar no processo de aprendizado de diversos conteúdos, nas mais diversas áreas. O trabalho de [Mullen et al. 2020] demonstra que jogos educacionais são eficazes para introduzir conceitos de Computação de Alto Desempenho (CAD) a partir de cenários interativos que simulam problemas reais.

Em [Hardasmal et al. 2020] é explorada a gamificação em ambientes de autômatos celulares, ensinando paralelismo de forma prática. A proposta utiliza conceitos reais em uma proposta gamificada para ensinar paralelização aos estudantes em um jogo aplicado em uma turma de Bioinformática.

Já no trabalho de [Manoharan and Ye 2020] foi investigado o uso de jogos *online*, com dois jogadores, no ensino de Programação Paralela. Esse formato, incentivou a colaboração entre os alunos e facilitou a compreensão de conceitos como sincronização e balanceamento de carga.

Este trabalho avança nesses conceitos ao integrar elementos dessas abordagens em um jogo que combina aprendizado prático e progressão pedagógica, com o foco em ensinar os conceitos sobre a classificação das arquiteturas paralelas segundo Flynn.

3. Jogos como Ferramentas de Ensino

Os jogos educacionais se destacam como uma ferramenta de ensino mais dinâmica e envolvente. Eles proporcionam um ambiente interativo no qual os alunos podem aplicar conceitos teóricos em situações práticas, favorecendo o aprendizado experiencial. Segundo [Mullen et al. 2020], jogos sérios são eficazes para introduzir temas complexos,

como a Computação de Alto Desempenho (CAD), por meio de simulações e cenários interativos. Os autores também apresentam uma série de benefícios que os tornam ferramentas valiosas, especialmente para disciplinas muito teóricas, técnicas ou complexas. Alguns dos principais benefícios incluem:

- **Engajamento e Motivação**: Os elementos presentes nos jogos tornam o aprendizado mais atraente, reduzindo a ansiedade associada a temas complexos [Deterding et al. 2011].
- **Aprendizado Prático**: Os alunos têm a oportunidade de aplicar conceitos teóricos em cenários simulados, o que melhora a compreensão e a retenção de conhecimento [Mullen et al. 2020].
- ***Feedback* Imediato**: A interação com os jogos fornece respostas instantâneas às ações dos jogadores, permitindo ajustes rápidos e aprendizado autônomo [Hardasmal et al. 2020].
- **Desenvolvimento de Habilidades Interdisciplinares**: Além de abordar conteúdos técnicos, os jogos promovem habilidades como trabalho em equipe, resolução de problemas e pensamento estratégico [Hardasmal et al. 2020].

4. Ensino da Programação Paralela

Com o aumento da demanda por eficiência computacional, o ensino de Programação Paralela se tornou essencial nos cursos de graduação em Computação [Hardasmal et al. 2020], pois a Programação dita sequencial, não atende às necessidades dos problemas que necessitam de um grande poder computacional. Isso, de alguma forma, acaba por reforçar a importância de novas abordagens educacionais para ensinar conceitos complexos de paralelismo e concorrência. Como observado por [Mullen et al. 2020], estudantes costumam ter dificuldades com os conceitos básicos, como na programação sequencial. Assim os jogos educacionais oferecem uma excelente oportunidade para introduzir esses conceitos de forma prática e envolvente.

No entanto, muitos estudantes enfrentam dificuldades em compreender os conceitos fundamentais dessa área, particularmente nos períodos iniciais onde predominam as abordagens da programação dita sequencial [Mullen et al. 2020].

4.1. Classificação de Flynn

A classificação de Flynn, proposta por Michael J. Flynn em 1972, [Flynn 1972], apesar de antiga, ainda é um dos modelos mais utilizados para categorizar arquiteturas de computadores com base nos fluxos de instrução e de dados. Ela organiza os sistemas computacionais em quatro categorias principais:

- **SISD (*Single Instruction, Single Data*)**: Sistemas tradicionais sequenciais, nos quais um único fluxo de instrução opera sobre um único fluxo de dados. Um exemplo típico é a arquitetura de Von Neumann [Hennessy and Patterson 2017].
- **SIMD (*Single Instruction, Multiple Data*)**: Sistemas que aplicam uma única instrução a múltiplos fluxos de dados simultaneamente. Essa abordagem é comum em processadores vetoriais e GPUs, otimizando operações matemáticas de forma paralela [Stallings 2018].
- **MISD (*Multiple Instruction, Single Data*)**: Sistemas onde múltiplas instruções operam sobre um único fluxo de dados. Embora raro na prática, pode ser encontrado em sistemas redundantes para tolerância a falhas [Flynn 1972].

- **MIMD (*Multiple Instruction, Multiple Data*)**: Sistemas paralelos nos quais *múltiplas instruções* são executadas simultaneamente sobre múltiplos fluxos de dados. Essa categoria abrange a maioria das arquiteturas de multiprocessadores modernos e sistemas distribuídos [Hennessy and Patterson 2017].

5. Desenvolvimento do Jogo

O jogo foi desenvolvido de forma a seguir uma sequência de ações, proporcionando uma experiência interativa e estratégica ao jogador.

Ao iniciar o programa, o jogador tem a opção de acessar uma tela de ajuda com instruções sobre a jogabilidade. A partir dela, pode retornar para obter mais informações ou iniciar o jogo. Com o início do nível, o jogador passa a ter controle sobre diversas interações, como pausar o jogo, liberar ou acelerar os monstros, posicionar arqueiros em locais estratégicos e evoluí-los para melhorar seu desempenho. O objetivo é impedir que os monstros cheguem ao final do caminho — caso isso ocorra, o jogador perde; caso contrário, vence o nível e progride para o próximo desafio.

5.1. Descrição da narrativa e temática do jogo

A narrativa do jogo foi projetada para maximizar a imersão dos jogadores. Ela se baseia em uma história onde um arqueiro encontra uma máquina tecnológica no meio de uma floresta e, ao interagir com ela, é transportado para dentro de um computador. Cada nível do jogo representa um desafio relacionado à Programação Paralela, como a sincronização de tarefas ou a execução em arquiteturas paralelas, permitindo que os alunos aprendam os conteúdos da disciplina conforme avancem nas fases [Mullen et al. 2020]. Essa abordagem narrativa segue princípios de *design* de jogos educativos, que sugerem que a inclusão de histórias pode aumentar o engajamento e a retenção do aprendizado [Deterding et al. 2011].

5.2. Mecânicas do jogo e interatividade

A ferramenta desenvolvida utiliza o gênero de jogos de Estratégia em Tempo Real (ETR) do tipo *Tower Defense* (Defesa de Torre), no qual o objetivo principal do jogador é impedir que ondas de inimigos alcancem um determinado ponto do mapa. Isso é feito por meio da construção e aprimoramento de torres defensivas ao longo de um caminho predefinido ou em áreas estratégicas do cenário. Como ferramenta pedagógica, cada elemento da jogabilidade foi cuidadosamente projetado para mapear conceitos de Programação Paralela. Os inimigos variam em velocidade e resistência, obrigando os jogadores a ajustar suas estratégias.

O jogador interage com o jogo a partir de botões que iniciam as fases, compram ou cancelam a colocação de torres e ajustam a velocidade do jogo. Essas interações são responsáveis por manter o fluxo dinâmico do jogo. O jogo é dividido em quatro níveis com mapas específicos, cada um oferecendo desafios progressivamente mais complexos. Essa combinação de mecânicas e interatividade promove o aprendizado a partir de uma experiência lúdica que estimula habilidades como planejamento e tomada de decisão.

5.2.1. Análise Técnica dos Níveis

O jogo foi desenvolvido com foco em representar graficamente as quatro arquiteturas clássicas de Flynn, e conta com elementos de *design* como progressão por níveis, desafio crescente, ajuda contextual, pausa, e variedade de interações. Após completar um nível, o jogador tem a opção de reiniciar ou avançar para o próximo, mas não pode pular fases livremente sem concluí-las.

Na dinâmica geral do jogo, o jogador assume o papel de defensor de um caminho (ou vários), utilizando arqueiros como elementos de ataque. Ele pode interagir com os inimigos pausando o jogo, acelerando-os, posicionando arqueiros e evoluindo-os. Na Figura 1, tem-se, por exemplo, os diferentes tipos de arqueiro. À medida que o jogador progride, pode adquirir arqueiros mais potentes, que atacam mais rápido e possuem um maior alcance de mira.

Figura 1. Nível dos Arqueiros disponíveis no jogo.

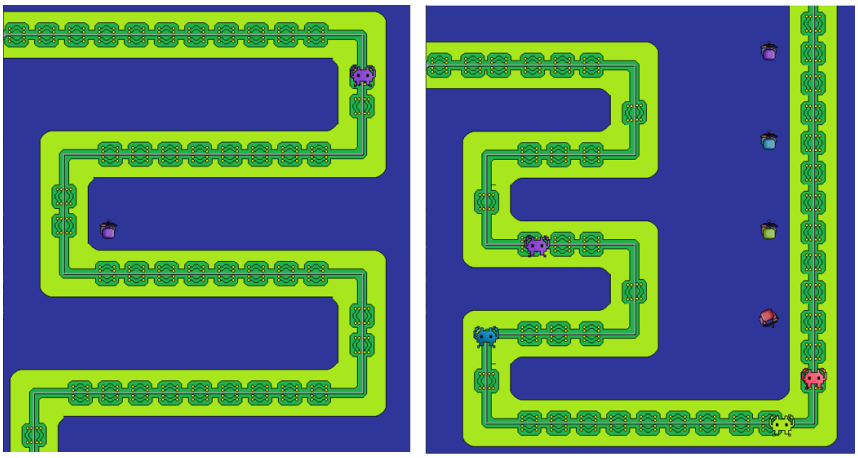
Os monstros, por sua vez, aumentam sua resistência e a velocidade de locomoção conforme a Figuras 2, e isso exige mais estratégia por parte do jogador.

Figura 2. Tipos de Monstros disponíveis no jogo.

A cada nível, o mapa (ou ”mundo”) se transforma, apresentando desafios diferentes que correspondem à estrutura computacional representada. A implementação dos diferentes modelos da Classificação de Flynn neste jogo foi dividida em quatro níveis, cada um representando uma arquitetura específica de processamento. Cada fase foi projetada para ilustrar como os dados são consumidos em cada modelo computacional.

- **Nível 1 – Modelo SISD:** Para o primeiro nível, há um arqueiro (que representa a figura do processador) e um único monstro (que representa o dado a ser consumido/processado), conforme a Figura 3 (a). O arqueiro realiza uma única ação (disparar) e lida com apenas um dado de entrada (o monstro). O arqueiro atira no único monstro e após isso a fase é encerrada . O objetivo central desse nível é mostrar que, nessa arquitetura, tudo ocorre de forma sequencial, sem paralelismo.

- **Nível 2 – Modelo SIMD:** Nesta etapa, múltiplos processadores (representados por arqueiros) executam simultaneamente a mesma instrução (atirar), porém aplicando-a a diferentes dados (monstros). A Figura 3 (b) ilustra que cada processador realiza a mesma operação ao mesmo tempo, mas sobre um elemento distinto do conjunto de dados. Essa analogia ilustra o funcionamento do modelo SIMD , no qual uma única instrução é distribuída para diversos núcleos, que operam paralelamente sobre dados distintos.



Modelo SISD
(a)

Modelo SIMD
(b)

Figura 3. Arquitetura de Flynn: Em (a) Modelo SISD e (b) Modelo SIMD

- **Nível 3 – Modelo MISD:** No terceiro nível, é abordado o modelo MISD, onde os diferentes arqueiros atacam o mesmo monstro, mas usando diferentes tipos de ataque. Cada arqueiro pode utilizar ataques distintos, mas todos estão mirando no mesmo monstro (mesmos dados). Com isso, ilustra-se a ideia de múltiplas instruções sendo aplicadas a um único dado (Figura 4).
- **Nível 4 – Modelo MIMD:** No quarto e último nível, que representa o modelo MIMD, possui vários arqueiros atacando diferentes dados (diferentes monstros). Cada arqueiro pode ter comportamentos distintos, atacando monstros diferentes ao mesmo tempo. Isso demonstra o cenário mais complexo de paralelismo, onde cada núcleo processa dados e instruções de forma independente (Figura 4).

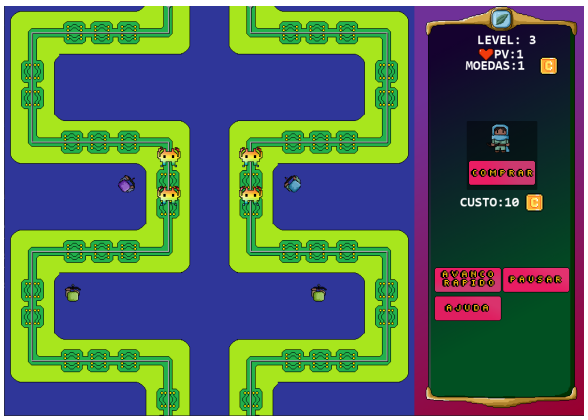


Figura 4. Nível 3: Modelos MISD/MIMD

5.3. Descrição dos Controles do Jogo

A interface do jogo apresenta um conjunto de botões que proporcionam a interação do jogador com os personagens. Os seguintes comandos são descritos a seguir e podem ser visualizados na Figura 5:

Figura 5. Botões do jogo

- **Iniciar**: Libera os monstros e inicia a fase;
- **Pausar**: Pausa o jogo. Com um novo clique, o jogador retorna para o jogo;
- **Comprar**: Executa a função de comprar um arqueiro diferente para destruir os diferentes tipos monstros;
- **Concluir**: Conclui a operação de compra de personagem;
- **Upgrade**: Evolui o nível de alcance do ataque do arqueiro nos monstros;
- **Avanço rápido**: Aumenta a velocidade de avanço dos monstros;
- **Reiniciar**: Reinicia o jogo;
- **Ajuda**: Abre a tela de informações sobre o jogo;
- **Voltar**: Fecha a tela de ajuda.

5.4. Tecnologias e Ferramentas Utilizadas

O Jogo foi desenvolvido usando a biblioteca *PyGame*, da linguagem *Python*. Esta linguagem foi escolhida por proporcionar mais flexibilidade no desenvolvimento e por ser amplamente utilizada no ensino de programação. A partir da biblioteca foi possível gerenciar elementos gráficos, sons e eventos de interação, facilitando o desenvolvimento. A ferramenta utiliza uma estruturação de dados no formato JSON (*JavaScript Object Notation*) para armazenar os dados. Essa técnica foi utilizada para o armazenamento de cada nível, permitindo uma manipulação fácil de informações como mapas e trajetórias.

Por fim, para a criação do material visual, foram utilizadas duas ferramentas gratuitas: uma plataforma *Web* chamada *Pixilart*[1] que possibilita a criação de desenhos e o *software* livre chamado Tiled [2], que serve para a criação de mapas em duas dimensões.

6. Metodologia da Avaliação

O público-alvo do jogo são alunos de graduação ou pós-graduação da área de Computação que possuam noções básicas de programação sequencial e estejam iniciando seus estudos sobre a Classificação de Flynn. Foram realizados testes para avaliar a usabilidade. Esses

[1]https://www.pixilart.com/
[2]https://www.mapeditor.org/

testes foram conduzidos de duas formas: um teste presencial e outro realizado de forma *on-line*.

O teste presencial contou com a participação de quatro pessoas, enquanto o teste *on-line* foi realizado com três participantes. Dentre os sete participantes, haviam dois alunos de mestrado, um de doutorado e os quatro demais eram de graduação. Quatro dos entrevistados já possuíam algum conhecimento prévio sobre Programção Paralela, e todos pertenciam ao curso de Ciência da Computação. Os testes *on-line* foram enviados aos participantes e realizados no dia 24 de março de 2025, enquanto os testes presenciais foram realizados no dia 25 de março.

No teste presencial, os voluntários foram recebidos em um laboratório da Instituição. Eles utilizaram um computador com a ferramenta do jogo já instalada e seguiram um roteiro de atividades[3] estruturado para avaliar a usabilidade do sistema. Os testes *on-line* foram conduzidos de maneira diferente. Os participantes receberam dois arquivos: um arquivo compactado contendo o código-fonte do jogo e um documento com o mesmo roteiro de testes utilizado na experiência presencial. Nessa modalidade, foi realizado o suporte para a instalação do jogo, visto que cada testador realizava os testes a partir do seu computador pessoal.

Em ambos os testes, não houve intervenção por parte dos organizadores durante o processo, exceto quando solicitado pelos participantes. Ao final da experiência, todos os usuários responderam a um questionário[4] de usabilidade baseado na Escala SUS (*System Usability Scale*).

6.1. *System Usability Scale (SUS)*

A *System Usability Scale* (SUS), criada por John Brooke, é uma ferramenta amplamente utilizada para avaliar a usabilidade de produtos e serviços digitais [Brooke 1996]. Consiste em um questionário de 10 itens, onde os participantes expressam seu grau de concordância em uma escala de 1 (discordo completamente) a 5 (concordo completamente). A pontuação final varia de 0 a 100, conforme Tabela 1. A SUS avalia três aspectos principais:

- **Eficácia**: Capacidade dos usuários em completar tarefas e atingir objetivos.
- **Eficiência**: Quantidade de recursos despendidos para alcançar esses objetivos.
- **Satisfação**: Nível de conforto e aceitação ao utilizar o sistema.

A utilização dessa métrica permite uma avaliação quantitativa da usabilidade do sistema, oferecendo uma visão objetiva sobre a experiência dos usuários.

6.1.1. Cálculo da Pontuação SUS

A pontuação SUS de cada participante foi calculada da seguinte forma:

1. Para as questões ímpares (1, 3, 5, 7, 9): subtrai-se 1 da resposta do participante.
2. Para as questões pares (2, 4, 6, 8, 10): subtrai-se a resposta do participante de 5.
3. A soma dos valores obtidos é multiplicada por 2,5 para obter a pontuação final.

[3] https://drive.google.com/file/d/15Mb-hsJPDsSKFe6XQyR2NTeRydAITilw/view

[4] https://forms.gle/NhK1mJ3uW4ZCAfdV6

Tabela 1. Tabela de classificação do SUS.

Pontuação	Adjetivo
90,0 - 100	Melhor Imaginável
80,0 - 89,9	Excelente
70,0 - 79,9	Bom
60,0 - 69,9	Ok
50,0 - 59,9	Ruim
$<$ 49,9	Muito Ruim

7. Resultados

Como citado anteriormente, os testes presenciais foram conduzidos com quatro participantes, cujas respostas foram registradas e analisadas conforme a metodologia SUS. A Tabela 2 apresenta as respostas individuais de cada participante e a média geral.

Tabela 2. Média da Pontuação SUS por Participante Presencial

Participante	Pontuação SUS
1	65,0
2	72,5
3	57,5
4	52,5
Média Geral	**61,88**

Já os testes *on-line* foram conduzidos com três participantes, cujas respostas foram registradas e analisadas, também, conforme a metodologia SUS. A Tabela 3 apresenta as respostas individuais de cada participante e a média geral.

Tabela 3. Média da Pontuação SUS por Participante *on-line*

Participante	Pontuação SUS
1	67,5
2	55,0
3	60,0
Média Geral	**60,83**

7.1. Discussão dos Resultados

A média geral da Escala SUS, entre os testes presenciais e *on-line*, foi de 61,355. Essa nota enquadra o jogo em uma classificação ”OK”da tabela SUS. Isso significa que o jogo tem uma usabilidade aceitável, mas com espaço para melhorias, já que uma pontuação acima de 60 geralmente indica uma experiência de uso satisfatória. No entanto, ao analisar os resultados por modalidade de teste (presencial e *on-line*), nota-se algumas diferenças nas respostas, o que sugere que o formato do teste pode ter influenciado a experiência dos usuários.

Nos testes presenciais, realizados com quatro participantes, a média da pontuação SUS foi de 61,88, indicando uma experiência de uso bem positiva. A maioria dos participantes elogiou a fluidez do jogo e sua capacidade de ensinar os conceitos de maneira clara e acessível. Entretanto, alguns participantes sugeriram melhorias, como a inclusão de um *feedback* mais robusto nas interações do jogo e a possibilidade de reiniciar apenas a fase perdida, o que evitaria frustrações causadas pela necessidade de reiniciar o jogo desde o início.

Nos testes *online*, realizados com outros três participantes, a média da pontuação SUS foi de 60,83, o que é um valor um pouco abaixo da média geral, mas ainda dentro de um intervalo aceitável. O *feedback* dos participantes *online* indicou que, embora o jogo tenha sido bem recebido, houve mais desafios relacionados à configuração e ao uso do jogo, o que pode ter impactado as pontuações. Foi sugerido, também, que a experiência *on-line* poderia ser aprimorada com um guia de instalação mais claro, além de melhorias na comunicação visual e sonora durante o jogo.

8. Ameaça à Validade da Pesquisa

A principal percepção entre os testes presenciais e *online* foi a interação dos participantes com a ferramenta. Nos testes presenciais, não houve quaisquer interrupções relacionadas à configuração do ambiente. Já nos testes *online*, a configuração inicial e a falta de *feedback* imediato nas interações geraram algumas dificuldades que impactaram a experiência geral. Essa diferença sugere que o formato *on-line* exige mais suporte e ajustes para garantir uma experiência igualmente satisfatória.

Apesar dos resultados positivos, o projeto também enfrentou algumas limitações. Entre elas, destaca-se o número reduzido de participantes nos testes de usabilidade, o que limita a generalização dos resultados. Além disso, a ausência de um acompanhamento impede avaliar os efeitos do jogo no aprendizado em longo prazo. Restrições de tempo e recursos também influenciaram na complexidade dos desafios implementados e na profundidade dos conteúdos abordados.

9. Conclusão e Trabalhos Futuros

O objetivo deste trabalho foi apresentar e aplicar um jogo educacional para servir de suporte no processo de ensino e aprendizagem das Arquiteturas Paralelas. O projeto combinou conceitos pedagógicos com mecânicas de jogos interativas, com a finalidade de proporcionar aos estudantes uma experiência de aprendizado engajante e eficaz, além de estimular a resolução de problemas e o pensamento crítico.

Como trabalhos futuros, pode-se indicar uma expansão nas funcionalidades do jogo, o que poderia incluir novos níveis e desafios que explorem tópicos mais avançados em Programação Paralela, como balanceamento dinâmico de carga, escalabilidade e uso de Placas Gráficas. Além disso, pode-se diferenciar as instruções dos dados, com objetivos, estratégias e níveis diferentes em cada etapa.

Referências

Amaral, H. F. and Sant'Ana, A. R. (2024). Onde estão os jogos educacionais: Uma revisão de literatura. In *Anais do XXIII Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, Manaus, Brasil. SBC.

Bergmann, J. and Sams, A. (2012). *Flip your classroom: Reach every student in every class every day*. International Society for Technology in Education, Eugene, EUA.

Brooke, J. (1996). Sus: A quick and dirty usability scale. In Jordan, P. W., Thomas, B., Weerdmeester, B., and McClelland, A. L., editors, *Usability Evaluation in Industry*. Taylor & Francis, London, Reino Unido.

Deterding, S., Dixon, D., Khaled, R., and Nacke, L. (2011). From game design elements to gamefulness: Defining "gamification". In *Proceedings of the 15th International Academic MindTrek Conference*, Tampere, Finlândia.

Flynn, M. J. (1972). Very high-speed computing systems. *Proceedings of the IEEE*, 60(2).

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., and Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23).

Grover, S. and Pea, R. (2013). Computational thinking in k-12: A review of the state of the field. *Educational Researcher*, 42(1).

Hardasmal, T., Salguero, A. G., and Hardasmal, A. J. (2020). Ensinando paralelismo com gamificação em ambientes de autômatos celulares. Trabalho não formalmente publicado.

Hennessy, J. L. and Patterson, D. A. (2017). *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, San Francisco, EUA.

Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? In *Educational Psychology Review*, volume 16, pages 235–266.

Hunsaker, E. and Madden, M. (2020). Implementing computational thinking through game design. *International Journal of Computer Science Education in Schools*, 4(2).

Kokotsaki, D., Menzies, V., and Wiggins, A. (2016). Project-based learning: A review of the literature. *Improving Schools*, 19(3).

Krajcik, J. S. and Blumenfeld, P. C. (2006). Project-based learning. In Sawyer, R. K., editor, *The Cambridge Handbook of the Learning Sciences*. Cambridge University Press, Cambridge, Reino Unido.

Manoharan, S. and Ye, X. (2020). Ensino de programação paralela introdutória usando jogos online para dois jogadores. Trabalho não formalmente publicado.

Mullen, J., Milechin, L., and Milechin, D. (2020). Ensino e aprendizagem de hpc por meio de jogos sérios. Autopublicação.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York, EUA.

Prince, M. (2004). Does active learning work? a review of the research. *Journal of Engineering Education*, 93(3).

Rodrigues, C. F., Santos, S. M., and Goulart, C. F. (2018). *Programação concorrente: Mecanismos de sincronização e comunicação de processos*. Instituto de Matemática e Estatística, São Paulo, Brasil.

Santos, A. I., Reis, F. G., and Zaidan, A. M. (2017). A utilização de metodologias ativas no ensino de programação. *Revista Brasileira de Informática na Educação*, 25(2).

Savery, J. R. (2015). Overview of problem-based learning: Definitions and distinctions. In *Essential Readings in Problem-Based Learning*. Purdue University Press, West Lafayette, EUA.

Selby, C. C. and Woollard, J. (2013). Computational thinking: The developing definition. `https://eprints.soton.ac.uk/356481/`. Acesso em: abr. 2025.

Shute, V. J., Sun, C., and Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22.

Stallings, W. (2018). *Computer Organization and Architecture: Designing for Performance*. Pearson, Boston, EUA.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3).