

Um Jogo Educacional com Programação em Blocos para a Formação do Pensamento Computacional Paralelo

An Educational Game with Block Programming for the Formation of Parallel Computational Thinking

Flávio Ramon A. de Souza¹, Maurício A. Pinheiro¹, Josivaldo de S. Araújo¹

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

souramon@hotmail.com, mauricioaires30@gmail.com, josivaldo@ufpa.br

Abstract. Introduction: The Educational Games have been presented as an important tool in the teaching and learning process, since they provide a playful and interactive environment, favoring student engagement and motivation. **Objective:** This work presents the development and evaluation of an educational game that uses block programming for the formation of Parallel Computational Thinking. **Methodology or Steps:** The game was designed based on board games, where the goal is to develop skills such as strategic planning, problem-solving and logical reasoning. The implementation used the Game Maker Studio and Pygame tool, with levels that introduce concepts such as partitioning, communication, agglomeration and mapping. **Results:** The results indicated a positive and motivating assessment of usability and challenges presented.

Keywords Educational Game, Block Programming, Parallel Computational Thinking, PCAM methodology.

Resumo. Introdução: Os Jogos Educacionais têm se apresentado como uma importante ferramenta no processo de ensino e aprendizagem, pois proporcionam um ambiente lúdico e interativo, favorecendo o engajamento e a motivação dos alunos. **Objetivo:** Este trabalho apresenta o desenvolvimento e a avaliação de um jogo educacional que utiliza programação em blocos para a formação do Pensamento Computacional Paralelo. **Metodologia ou Etapas:** O jogo foi projetado baseado em jogos de tabuleiro, onde o objetivo é desenvolver habilidades como planejamento estratégico, resolução de problemas e raciocínio lógico. A implementação utilizou a ferramenta Game Maker Studio e Pygame, com níveis que introduzem conceitos como particionamento, comunicação, aglomeração e mapeamento. **Resultados:** Os resultados indicaram uma avaliação positiva e motivadora da usabilidade e desafios apresentados.

Palavras-Chave Jogo Educacional, Programação em Blocos, Pensamento Computacional Paralelo, Metodologia PCAM.

1. Introdução

O ensino da Programação Paralela ganhou destaque nas matrizes curriculares dos cursos de graduação em Computação, diante da crescente necessidade por soluções computacionais que exigem eficiência e alto desempenho. Soma-se a isso, a elaboração de algoritmos otimizados que consigam reduzir efetivamente o tempo de processamento das aplicações,

requer cada vez mais profissionais que compreendam a elaboração e o funcionamento desse tipo de programação [Aracaty et al. 2025]. Contudo, a construção conceitual do Pensamento Computacional Paralelo ainda é um desafio para muitos alunos, pois se passou de uma programação estritamente sequencial para uma programação com níveis elevados de paralelismo, não apenas em hardware, mas também, no desenvolvimento de códigos [Mullen et al. 2020] [Almeida Jr. et al. 2021].

Para contornar esses desafios, várias metodologias vêm sendo desenvolvidas, aplicadas e adaptadas com o objetivo de auxiliar, não apenas na compreensão dos conceitos e no desenvolvimento dos algoritmos paralelos, mas também estimular o aprendizado, colocando o aluno como protagonista. Uma das metodologias que vêm ganhando destaque é o uso dos Jogos Educacionais pois, pode-se introduzir conceitos, próprios da área alvo, de forma gradativa, através de diferentes fases. Isso acaba auxiliando na carga cognitiva do aluno, fazendo-o aprender de forma lúdica e até mesmo divertida [Macena et al. 2022]. Os jogos podem não apenas engajar, mas também, proporcionar um ambiente interativo onde definições complexas são exploradas de maneira mais acessível e prática [Manoharan and Ye 2020].

Com o propósito de auxiliar, não apenas o ensino de metodologias da Programação Paralela, mas também na formação conceitual do chamado Pensamento Computacional Paralelo, este trabalho apresenta o Paralegram. Um Jogo Educacional, baseado em Programação em Blocos, que tem como finalidade associar as diretrizes do Pensamento Computacional com os conceitos básicos da Programação Paralela. Essa associação visa preencher uma lacuna na literatura pois, embora existam diversos trabalhos que mencionem o termo Pensamento Computacional Paralelo, não foram identificados estudos que definam ou associem esses fundamentos dentro de um ambiente de jogo educacional. O objetivo é oferecer uma experiência de aprendizagem que promova a motivação e o engajamento dos alunos a partir do raciocínio lógico para a resolução dos problemas propostos em cada fase do jogo.

2. Trabalhos Relacionados

Há na literatura diversos estudos que destacam o potencial dos Jogos Educacionais no processo de ensino e aprendizagem, e o quanto eles auxiliam na compreensão de diversos conteúdos, nas mais diversas áreas. O trabalho de [Aracaty et al. 2025] apresenta um jogo que tem como objetivo o ensino das Arquiteturas Paralelas, mas especificamente, a classificação de Flynn [Flynn 1972]. Desenvolvido no gênero *Tower Defense* introduz conceitos como concorrência, sincronização e paralelismo, em um ambiente que necessita de planejamento estratégico e raciocínio lógico.

A pesquisa de [Feng and Davis-Wallace 2023] propôs uma solução desenvolvendo uma versão do programa *Snap!*, chamada *pSnap!*. O estudo propôs incorporar a Programação Paralela de forma visual, mantendo a acessibilidade e a simplicidade dos blocos. Evita-se, assim, o foco em sintaxe complexa de linguagens textuais. O *pSnap!* encorajou os alunos no pensamento lógico, necessário para resolver problemas. O objetivo foi permitir com que os alunos criassem múltiplos fluxos de controle paralelos ao longo do jogo.

Este trabalho, procura se diferenciar dos demais, por apresentar uma construção para a formação conceitual do Pensamento Computacional Paralelo. Para isso, utiliza-se

das bases da Programação Paralela e dos pilares do Pensamento Computacional. Essas definições conceituais são utilizadas no jogo educacional Paralegram, como fases, e são desenvolvidas através da programação em blocos.

3. Jogos Educacionais como Ferramentas Auxiliares de Ensino

Os jogos educacionais se destacam como instrumentos pedagógicos pois, além de serem uma ferramenta de ensino dinâmica e envolvente, agregam benefícios relevantes no processo de ensino e aprendizagem ao apresentarem competências e habilidades educacionais bem definidas. Além disso, contribuem no desenvolvimento cognitivo e intelectual dos jogadores através da exploração de mundos e conceitos[Lima et al. 2025].

Diversos estudos relatam a eficácia dos jogos educacionais na introdução ou assimilação de conceitos complexos, ou no reforço na compreensão de temas confusos. Isso porque proporcionam o engajamento e a motivação, reduzindo dessa forma, a ansiedade associada a temas mais complicados. A interação com os jogos possibilita um *feedback* imediato, o que permite ajustes rápidos e um aprendizado mais autônomo. Permite, também, promover habilidades como trabalho em equipe, resolução de problemas e raciocínio estratégico e lógico [Aracaty et al. 2025].

3.1. Ensino da Programação Paralela

O ensino da Programação Paralela apresenta desafios que são próprio da área, pois envolve conceitos e o desenvolvimento de programas que sejam capazes de executar múltiplas tarefas simultaneamente. Para isso, exige que os alunos possuam raciocínio lógico, capacitação de abstração e compreensão de interações complexas entre diversos componentes do sistema [Andrade and Araújo 2025]. No entanto, muitos estudantes enfrentam dificuldades em compreender esses conceitos fundamentais dessa área, particularmente, nos períodos iniciais, onde ainda há uma predominância de abordagens da programação dita sequencial [Mullen et al. 2020].

Assim, os jogos educacionais se apresentam como uma excelente oportunidade para introduzir conceitos, bem como para auxiliar na formação do chamado Pensamento Computacional Paralelo de forma prática, envolvente e divertida. A associação desses dois eixos contribuem para um aprendizado mais acessível e motivador, ao transformar esses desafios conceituais em atividades orientadas a objetivos claros, através de uma progressão incremental e autônoma [Andrade and Araújo 2025].

3.2. Pensamento Computacional

O Pensamento Computacional é uma habilidade transversal que pode ser aplicada nos mais diferentes contextos educacionais [Wing 2006]. Pode ser definido como um conjunto de conceitos, práticas e perspectivas que possibilitam resolver problemas de forma eficiente, através de métodos da Computação [?]. É definido em quatro pilares: Decomposição, Reconhecimento de Padrões, Abstração e Algoritmos. Esses pilares são conceituados na Figura 1.

4. Metodologia da Programação Paralela

O processo de aprendizagem da Programação Paralela não passa apenas pela complexidade no desenvolvimento de códigos paralelos eficientes, mas também, pela compreensão

de conceitos que envolvem o chamado Pensamento Computacional Paralelo. Para facilitar a compreensão destes, deve-se utilizar abordagens didáticas para despertar o interesse e a motivação nos alunos [?].

Para auxiliar na construção conceitual do Pensamento Computacional Paralelo foi utilizada a Metodologia PCAM[Foster 1995] associada às diretrizes do Pensamento Computacional [Wing 2006].

A Metodologia PCAM foi criada por [Foster 1995] e descreve as etapas necessárias para a implantação de um projeto paralelo. Para ele, a construção de um programa paralelo sempre deve ser iniciado a partir de uma solução sequencial, anteriormente implementada e funcional. Pois, a partir dessa solução sequencial, pode-se identificar pontos críticos de processamento e os resultados da execução. Essa metodologia é conhecida na literatura como PCAM, onde as letras são as iniciais de Particionamento (P), Comunicação (C), Aglomeração (A) e Mapeamento (M), conforme conceituados na Figura 1.

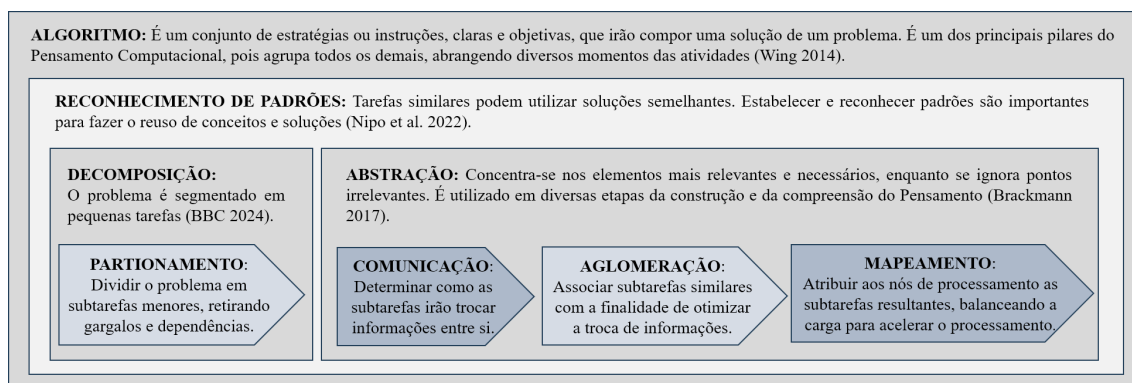


Figura 1. Conceitos do Pensamento Computacional Paralelo.

No diagrama da Figura 1, as etapas de Particionamento, Comunicação, Aglomeração e Mapeamento (da Metodologia PCAM) estão relacionadas às diretrizes da Decomposição, Reconhecimento de Padrões, Abstração e Algoritmo (pilares do Pensamento Computacional).

5. Desenvolvimento do Jogo

O Paralegram é um Jogo Educacional desenvolvido com o objetivo de auxiliar a formação conceitual do Pensamento Computacional Paralelo. Essa formação tem como base, a metodologia PCAM associada às diretrizes do Pensamento Computacional, conforme apresentados na Figura 1, e é realizada através da evolução das fases do jogo. Para cada fase, são apresentados os objetivos, as habilidades a serem desenvolvidas e os resultados esperados.

5.1. Descrição da Narrativa e Temática do Jogo

O Jogo tem como cenário um centro de distribuição de mercadorias e um porto. É demonstrado, com passar das etapas, todo o fluxo de mercadorias neste ambiente. O personagem principal precisa gerenciar todas as tarefas, desde a origem dos produtos, ou seja, no centro de distribuição do fabricante, até a entrega dos contêineres nos navios. Cada fase

ilustra uma etapa da construção conceitual que o aluno vai avançando através das tarefas que são solucionadas utilizando a programação em blocos. 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

O fluxo do jogo é baseado em jogos de tabuleiro. Neste caso, o jogador necessita resolver desafios, descrevendo os movimentos do personagem, através de códigos no editor visual. Conforme vai solucionando as tarefas, o jogador vai pontuando e avançando para novas tarefas e fases. O estilo de programação é visual e preestabelecido, ou seja, programação baseadas em blocos limitados e definidos por fases, conforme ilustrada na Figura 2.

Figura 2. Tela da Fase 2 do Paralegram.



Como ferramenta pedagógica, cada elemento da jogabilidade foi cuidadosamente projetado para mapear conceitos da Programação Paralela associada a metodologia PCAM. O jogador interage com o jogo a partir de blocos predefinidos de comandos que possibilitam a movimentação no tabuleiro. Essas interações são responsáveis por manter o fluxo dinâmico do jogo. A tela da Fase 2 pode ser visualizada na Figura 2. Nela são assinalados cinco posições em círculos vermelhos: em (1) é apresentado o tabuleiro onde será realizado o movimento dos jogadores; em (2), são visualizados os blocos de comandos predefinidos, onde o jogador poderá clicar e arrastar para o ambiente (3); Em (4), são definidos botões de execução/parada do código montado em (3), além dos botões de limpar (C) e ajuda (?). Em (5), são apresentados os objetivos de cada fase do jogo.

5.3. Tecnologias e Ferramentas Utilizadas

O jogo foi desenvolvido em um ambiente gráfico 2-D através da ferramenta *Game Maker Studio*¹. A escolha dessa *engine* se baseou no fato de que diversas funcionalidades necessárias para a criação das fases do jogo já vêm disponibilizadas no ambiente, facilitando, dessa forma, o processo de criação e produção. Além de possuir uma interface amigável para a criação do editor visual. Foram desenvolvidos, também, protótipos no *PyGame*² e na linguagem de programação *Python*.

¹<https://gamemaker.io/en>

²<https://pypi.org/project/pygame/>

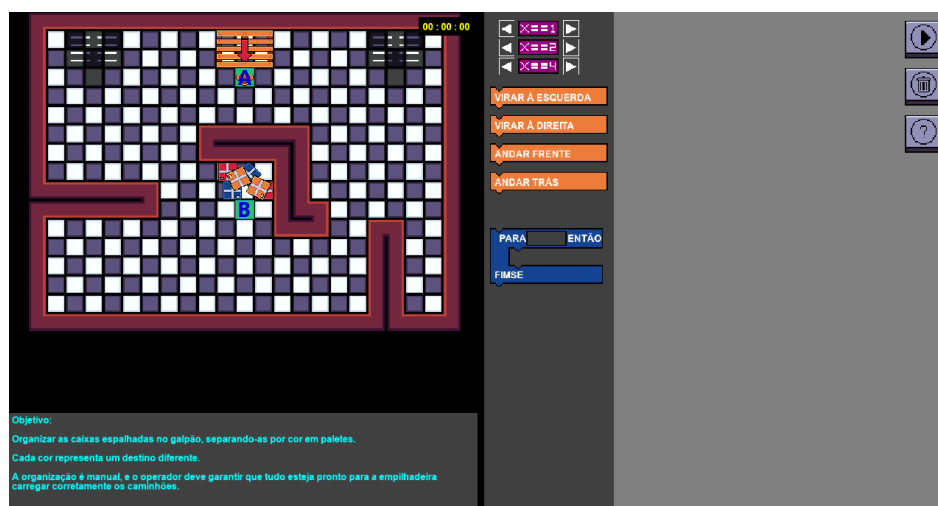
5.3.1. Análise Técnica dos Níveis

O jogo foi desenvolvido com foco em representar graficamente as quatro fases da metodologia PCAM associada as diretrizes do Pensamento Computacional. Para isso, conta com elementos de *design* como progressão por níveis e desafios crescentes. Após completar uma fase, o jogador avança para a próxima.

A cada fase, o tabuleiro se transforma, apresentando desafios diferentes que correspondem à estrutura computacional representada. A implementação da metodologia PCAM (Particionamento, Comunicação, Aglomeração e Mapeamento) foi dividida em quatro fases, cada uma representando uma etapa específica da elaboração de um código paralelo. Essa combinação de mecânicas e interatividade promove o aprendizado a partir de uma experiência lúdica que estimula habilidades como planejamento e tomada de decisão.

Fase 1 – Particionamento: Os produtos contidos no centro de distribuição estão misturados e precisam ser separados em caixas, com diferentes cores. Essas caixas devem ser agrupadas e empilhadas pelas cores. Nesta fase, o objetivo é demonstrar a divisão de uma tarefa em partes menores, ou seja, aplicar a Decomposição ou Particionamento. Isso é realizado através do Reconhecimento dos Padrões (neste caso, as cores das caixas), conforme a Figura 3. Para isso, irá se construir um Algoritmo para definir os passos que devem ser utilizados para sair do ponto A, chegar ao ponto B e retornar para A, utilizando apenas os comandos em blocos disponibilizados na Figura 3.

Figura 3. Primeira Fase: Particionamento (Decomposição, Rec. de Padrões e Algoritmo)



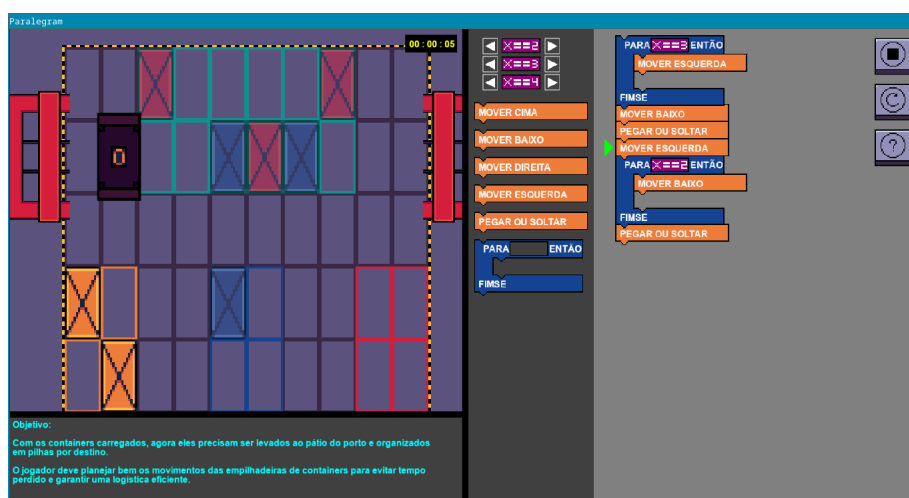
As habilidades que se espera nesta fase são: (1) Identificar as sequências de passos necessários que podem ser aplicados para resolver a tarefa; (2) Construir uma sequência, com repetições simples, com base em instruções/comandos preestabelecidos (programação em blocos); e (3) comparar e organizar objetos identificando similaridades (cores das caixas) para essa organização (reconhecimento de padrões). Como resultado, espera-se estimular o aluno a observar a melhor estratégia para, não apenas percorrer o menor caminho (menor número de passos), mas também, utilizar o menor número de instruções para resolver a decomposição do problema.

Fase 2 – Comunicação: Nesta fase, é necessário carregar os estrados, utilizando empilhadeiras, para o contêiner no caminhão. Para isso, é necessário definir um Algoritmo de forma a evitar que duas empilhadeiras colidam no corredor do centro de distribuição, conforme pode ser visualizado na Figura 2 (destacadas em um círculo vermelho). Nesta etapa, também será necessário, o reconhecimento de padrões, através das cores das mercadorias e do caminhão. O objetivo é utilizar a habilidade da Abstração para estabelecer (simular) uma Comunicação de modo a levar a empilhadeira do ponto A, ao ponto B, evitando uma possível colisão, com a segunda empilhadeira, que se movimenta em sentido perpendicular.

As habilidades esperadas nessa fase são: (1) Reconhecer objetos que possam ser representados através de matrizes que possam estabelecer uma organização, onde cada componente está em uma posição definida por coordenada (Abstração) sem haver conflitos entre eles; (2) Realizar manipulações simples com base em instruções/comandos pre-estabelecidos (Programação em Blocos); (3) Comparar e organizar objetos identificando similaridade (cores) para essa organização (Reconhecimento de Padrões); e (4) Associar os valores “verdadeiro” e “falso” a sentenças lógicas (colisão ou não colisão entre as empilhadeiras). Pretende-se, como resultado, avaliar a capacidade de abstração e estabelecer critérios para que os objetos (empilhadeiras) não colidam.

Fase 3 – Aglomeração: Com os produtos já separados e carregados nos contêineres, serão direcionados ao pátio da transportadora. Nessa etapa, será utilizada uma empilhadeira de contêiner, como observado na Figura 4, responsável por manuseá-los e agrupá-los. O objetivo desta fase é definir os movimentos necessários para agrupar e organizar os objetos similares (neste caso, os contêineres). Assim como na fase anterior, será utilizado um Algoritmo (para definir os movimentos da empilhadeira), o Reconhecimento de Padrões (através das cores) e a Abstração (simular o agrupamento de tarefas similares).

Figura 4. Terceira Fase: Aglomeração (Abstração, Rec. de Padrões e Algoritmo)



As habilidades esperadas nessa fase são: (1) Realizar manipulações simples com base em instruções/comandos preestabelecidos (Programação em Blocos); (2) Comparar e organizar objetos identificando similaridade (cores) para essa organização (Reconhe-

cimento de Padrões); (3) Reconhecer objetos que possam ser representados através de matrizes que possam estabelecer uma organização, onde cada componente está em uma posição definida por coordenada (Abstração); e (4) Simular algoritmos representando blocos com sequências de repetições simples e aninhadas para resolver o problema. Espera-se, como resultado desta fase, planejar movimentos (Algoritmos) para que os objetos (contêineres) sejam organizados em lotes (Reconhecimento de Padrões).

Fase 4 – Mapeamento: Nesta fase, os contêineres são retirados do pátio de expedição portuário para serem guinchados e alocados em um navio. Conforme demonstrado na Figura 5, eles devem ser colocados (mapeados) para navios específicos para que possam ser transportados. O objetivo desta fase é definir o número de estruturas necessárias (guindastes) para realizar a tarefa no menor tempo possível. Para isso, os movimentos do guindaste devem ser definidos pelos blocos de programação. (Algoritmo). Já o tempo de carregamento do navio será definido pela quantidade de guindastes que serão utilizados (Abstração). E o carregamento, continua sendo pelas cores (Reconhecimento de Padrões).

Figura 5. Quarta Fase: Mapeamento (Abstração, Rec. de Padrões e Algoritmo)



Nesta fase, as habilidades esperadas são: (1) Classificar objetos (contêineres), agrupando-os em coleções (conjuntos) e associando cada coleção a um tipo (cor) (Reconhecimento de Padrões); (2) Compreender os conceitos de paralelismo (diferentes números de guinchos); (3) Realizar manipulações simples com base em instruções/comandos preestabelecidos (Programação em Blocos); (4) Reconhecer objetos que possam ser representados através de matrizes que possam estabelecer uma organização, onde cada componente está em uma posição definida por coordenada (Abstração); e (5) Desenvolver Algoritmos fazendo uso de decomposição e do reuso no processo da solução do problema, automatizando-o através das unidades de processamento. Como resultado, se deseja definir estratégias para que os objetos (contêineres) sejam organizados em lotes (Reconhecimento de Padrões) utilizando vários guindastes (mapeamento) simultaneamente (paralelamente).

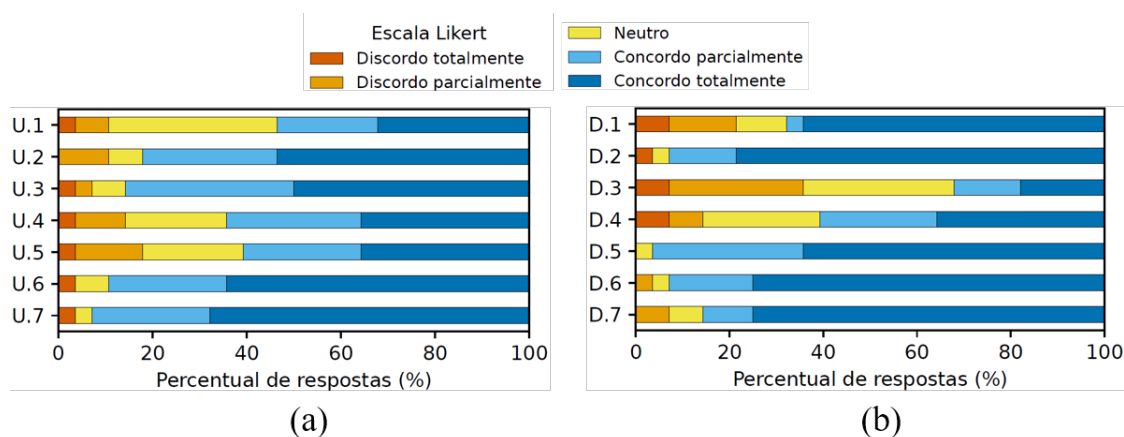
6. Metodologia da Avaliação

O jogo foi aplicado e avaliado por uma turma da Disciplina de Sistemas Distribuídos, sendo 26 alunos do curso de Bacharelado em Ciência da Computação e 2 do curso de Bacharelado em Sistemas de Informação. O projeto foi aprovado pelo Comitê de Ética da Universidade Federal do Pará em 17/10/2025 sob o processo nº 92975125.5.0000.0018. Após finalizar o jogo os alunos responderam um questionário para avaliar a **Usabilidade** (facilidade ou dificuldade de uso e a experiência quanto a interface geral) e os **Desafios** (quanto as atividades propostas no jogo estimularam a evolução do aprendizado com as tarefas). As perguntas foram baseadas no *EGameFlow* [Fu et al. 2009] e utilizavam a escala *Likert*, com cinco alternativas, variando de 1 (Discordo Totalmente) a 5 (Concordo Totalmente). O questionário pode ser acessado em: <https://forms.gle/7jVSe7wbK99zJnMR7>

7. Resultados e Discussões

O gráfico da Figura 6(a) apresenta os resultados da Usabilidade, de (U.1) a (U.7). Já o gráfico da Figura 6(b) apresenta os resultados dos Desafios, de (D.1) a (D.7). Esses itens podem ser visualizados em: <https://url-shortener.me/GDJX>

Figura 6. Resultados da Usabilidade e do Desafio



Percebe-se, que de uma forma geral, o jogo teve uma avaliação positiva nos quesitos avaliados. Contudo, nota-se, também, algumas avaliações negativas. Na Figura 6(a), por exemplo, as avaliações mais negativas foram para os itens (U.1) e (U.5), onde as perguntas eram direcionadas à interface geral do jogo (clareza, organização e facilidade de navegação), e se o *feedback* das mensagens ajudavam na resolução dos problemas, respectivamente. Isso pode ser justificado pois, percebeu-se que durante o jogo, muitos alunos acabaram não finalizando o acesso ao tutorial, que explicava vários pontos importantes que deveriam ser levados em consideração. Isso acarretou em dúvidas durante a execução do jogo, bem como, um maior tempo de compreensão das tarefas para serem executadas.

Essas avaliações podem ser ratificadas, também, na Figura 6(b), mais precisamente no item (D.3). Esse item, pergunta se existem "dicas" que possam auxiliar nas tarefas. Como muitas das instruções ("dicas") estavam disponíveis no tutorial porém, apenas como texto, muitos alunos não as leram na sua totalidade. Com isso, as dúvidas acabaram surgindo durante as tarefas. Esse grande número de instruções em forma de

texto, foi outro ponto, apontado negativamente, no item (D.4), onde a pergunta era sobre a presença de vídeos ou áudios para auxiliar nas tarefas.

Outro ponto que pode ser destacado da Figura 6(a) é o item (U.4). Esse item trata do desempenho (rapidez, estabilidade) com que o jogo era executado. Pode-se, também, justificar esse item, já que o mesmo foi desenvolvido através da ferramenta *Game Maker Studio* para Windows, e foi aplicado em um laboratório com máquinas com sistema operacional Linux. Foi necessário, então, gerar executáveis compatíveis com o Linux. Para isso, foi utilizado o programa *Wine 9.0*³, responsável por implementar uma camada de compatibilidade de *softwares* do Windows para Linux. Dessa forma, foram traduzidas todas as chamadas da biblioteca gráfica e da entrada/saída nativas de um sistema operacional para o outro. E essa tradução acabou deixando o executável um pouco mais lento.

8. Ameaça à Validade da Pesquisa

Uma das principais observações que podem ser apontadas como uma ameaça aos resultados dessa pesquisa foi que o jogo Paralegram, voltado para o ensino da Programação Paralela, foi aplicada em uma turma de Sistemas Distribuídos. Os alunos não tinham um conhecimento preliminar sobre o conteúdo apresentado e desenvolvido no jogo. Os únicos conceitos que tiveram conhecimento foram apresentados sobre a metodologia PCAM e o Pensamento Computacional antes de realizarem os testes do jogo. Isso pode ter influenciado bastante os resultados da pesquisa.

9. Conclusão e Trabalhos Futuros

Com os altos investimentos e o avanço da Computação de Alto Desempenho houve a necessidade de se desenvolver a formação na área de Programação Paralela. Assim como na Programação dita sequencial, as dificuldades no processo de ensino e aprendizagem são grandes. Pois, muda-se a forma de pensar, a estrutura dos códigos, as ferramentas utilizadas e além disso, muda-se a arquitetura onde essas aplicações serão executadas. Para amenizar essas dificuldades, o objetivo deste trabalho foi apresentar uma formação conceitual do Pensamento Computacional Paralelo através de um jogo sério baseado em programação em blocos. Para isso, foram utilizados conceitos como a metodologia PCAM e do Pensamento Computacional.

Como trabalhos futuros, espera-se desenvolver novas funcionalidades, com novos desafios e níveis, incluindo mais comandos em blocos, com uma Programação Paralela mais avançada. Onde se possa discutir conceitos de sincronização, escalabilidade, balanceamento dinâmico e uma programação com um número maior de unidades de processamento. Pretende-se, também, aprimorar funções para se contabilizar instruções e passos, além de um tutorial que seja mais dinâmico, com vídeos curtos, que sejam mais didáticos e interativos.

10. Agradecimentos

Este trabalho teve apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

³<https://www.winehq.org/>

Referências

- Almeida Jr., R., Farias, H., Saavedra, M., and Araújo, J. (2021). Metodologias de ensino na programação paralela com placas gráficas: Uma revisão sistemática da literatura. page 1243–1252, 22 a 26 de novembro de 2021. XXXII Simpósio Brasileiro de Informática na Educação (SBIE).
- Andrade, A. H. G. and Araújo, J. S. (2025). Gamificação no ensino de programação paralela em cursos de computação: Um mapeamento sistemático da literatura. In *Revista Novas Tecnologias na Educação (RENOTE)*, volume 24, pages 441–452.
- Aracaty, L. Q., Quaresma, T. M., and Araujo, J. d. S. (2025). Arqueiro flynn: Um jogo educacional para o ensino das arquiteturas paralelas. In *Anais do XXIV Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, Salvador, Brasil. SBC.
- 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.
- Feng, W. and Davis-Wallace, L. (2023). Parallel programming with pictures: Choosing your own adventure. In *2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 256–261.
- Flynn, M. J. (1972). Very high-speed computing systems. *Proceedings of the IEEE*, 60(2).
- Foster, I. (1995). Designing parallel algorithms. In *Designing and Building Parallel Programs*, page 26–82. Addison-Wesley Longman Publishing Co. cap. 2.
- Fu, F.-L., Su, R.-C., and Yu, S.-C. (2009). Egameflow: A scale to measure learners’ enjoyment of e-learning games. volume 52, pages 101–112. Computers and Education (ACM).
- Lima, C. d. C., Souza, A. L., and Xexéo, G. (2025). Tempo mestre: Um jogo educacional para o ensino de gestão do tempo. In *Anais do XXIV Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, Salvador, Brasil. SBC.
- Macena, J., Pires, F., and Melo, R. (2022). Hello food: uma jornada de aprendizagem lúdica em algoritmos, programação e pensamento computacional. *Anais do XXXIII Simpósio Brasileiro de Informática na Educação (SBIE 2022)*.
- 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.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3).