

Paralegram: Um Jogo Sério com Programação em Blocos para Auxiliar no Pensamento Computacional Paralelo

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

¹Instituto de Ciências Exatas e Naturais – Universidade Federal do Pará (UFPA)
Rua Augusto Corrêa nº 01 Caixa Postal 66075-110 – Belém – PA – Brazil

souramon@hotmail.com, mauricio.pinheiro@icen.ufpa.br, josivaldo@ufpa.br

Abstract. *With the evolution of Parallel Programming, the challenge now is to train students who are empowered to develop the skills and competencies needed for the elaboration of efficient and optimized parallel codes. However, the teaching and learning process of Parallel Programming presents similar, or even greater, challenges than that of Sequential Programming. With the purpose of contributing to the conceptual construction of Parallel Computational Thinking, this work presents the Paralegram. A serious game, based on Block Programming, which aims to associate basic concepts of Parallel Programming with the guidelines of Computational Thinking. The game was applied and evaluated by students of the computer science course.*

Resumo. *Com a evolução da Programação Paralela o desafio agora é formar alunos que sejam capacitados a desenvolverem as habilidades e competências necessárias para a elaboração de códigos paralelos eficientes e otimizados. No entanto, o processo de ensino e aprendizagem da Programação Paralela apresenta desafios similares, ou até maiores, que o da Programação Sequencial. Com o propósito de contribuir na construção conceitual do Pensamento Computacional Paralelo, este trabalho apresenta o Paralegram, um jogo sério baseado em Programação em Blocos que tem como finalidade associar conceitos básicos da Programação Paralela com às diretrizes do Pensamento Computacional. O jogo foi aplicado e avaliado por alunos do curso de Ciência da Computação.*

1. Introdução

A Computação de Alto Desempenho, como campo multidisciplinar, evoluiu rapidamente nas últimas décadas. Essa rápida evolução se deu, em grande parte, pela crescente necessidade em se processar uma quantidade cada vez maior de dados, para se realizar análises e gerar informações em diversas áreas do conhecimento. Isso acarretou em uma maior demanda por profissionais que compreendessem a construção, funcionamento e algoritmos desse tipo de programação [Galante 2023].

Várias iniciativas foram tomadas para se definir as competências e habilidades necessárias para o aprendizado da Programação Paralela, tanto a nível internacional [ACM and IEEE-CS 2020], quanto a nível nacional [MEC 2016], [ZORZO et al. 2017] e [SBC 2019]. Dessa forma, instituições de ensino superior passaram a inserir a disciplina de Programação Paralela nas suas matrizes curriculares nos cursos de Computação. Porém, o processo de ensino e aprendizagem da Programação Paralela apresenta desafios que são próprios da área, já que a forma de programar foi modificada. Passou-se de

uma programação estritamente sequencial para uma programação que se procura obter ganhos de desempenho, com níveis elevados de paralelismo (não apenas em hardware), mas também no desenvolvimento de algoritmos [Almeida Jr. et al. 2021].

Para contornar esses desafios várias metodologias vêm sendo desenvolvidas, aplicadas e adaptadas ao processo de ensino e aprendizagem da Programação Paralela. O objetivo é auxiliar na compreensão dos conceitos e no desenvolvimento dos algoritmos. Metodologias chamadas de Ativas ajudam a viabilizar o aprendizado e colocam os alunos como protagonistas no processo de aprendizagem, como por exemplo, o uso de Jogos Sérios. Essa metodologia é utilizada como recurso para auxiliar no aprendizado, dispensando muitas vezes, uma infraestrutura própria de *hardware* para o treinamento [Nipo et al. 2022]. Além disso, introduz os conceitos, próprios da área, de forma gradativa (através de fases), o que auxilia na carga cognitiva do aluno, fazendo-o aprender de forma lúdica e até mesmo divertida [Macena et al. 2022].

Com o propósito de auxiliar na construção conceitual do chamado Pensamento Computacional Paralelo, este trabalho apresenta o Paralegram, um Jogo Sério, baseado em Programação em Blocos, que tem como objetivo 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. O jogo foi desenvolvido em fases que buscam demonstrar os principais conceitos da Programação Paralela associados às bases do Pensamento Computacional.

Este trabalho possui, além da seção de Introdução, uma seção 2 de Trabalhos Relacionados, seguido pela seção 3 onde se trata do Ensino da Programação Paralela. Na seção 4, são apresentadas as bases para a formação conceitual do Pensamento Computacional Paralelo. Na seção 5, é descrito o Jogo Sério Paralegram. Na seção 6, é discutida a metodologia desenvolvida em cada fase, bem como as habilidades esperadas. Na seção 7, a avaliação, tendo em seguida, a seção 8 dos Resultados e Discussões. Finalizando com a Conclusão e os Trabalhos Futuros.

2. Trabalhos Relacionados

No trabalho de [DeLozier and Shey 2023], buscou-se avaliar como programadores novatos resolvem problemas utilizando códigos paralelos. Para isso houve a criação de uma linguagem de programação visual e um *framework* baseado em jogos, e foram testadas as habilidades em 95 estudantes de graduação com pouco ou nenhuma experiência em programação.

A pesquisa de [Feng and Davis-Wallace 2023] propôs uma solução desenvolvendo uma versão do programa *Snap!* chamada *pSnap!*, visando 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.

Apesar das propostas apontadas, não foram identificados estudos que apresentem uma construção ou definição para a formação conceitual do Pensamento Computacional Paralelo. Este trabalho procura se diferenciar dos demais por apresentar uma definição

conceitual, utilizando as bases da Programação Paralela e os pilares do Pensamento Computacional, com o objetivo de auxiliar a construção conceitual do Pensamento Computacional Paralelo. Essa construção é usada como base para um jogo sério que utiliza programação em blocos.

3. O Ensino da Programação Paralela

Os cursos de graduação em Computação passaram a inserir nos Projetos Pedagógicos dos Cursos (PPC) a disciplina de Programação Paralela. Com o objetivo de nortear as competências e habilidades esperadas, foram utilizadas diretrizes nacionais, como a do Ministério da Educação [MEC 2016] e da Sociedade Brasileira da Computação [SBC 2019], e internacionais, como da IEEE/ACM [Force 2020].

Apesar dessas definições, uma dificuldade ainda é recorrente no processo de ensino e aprendizagem da Programação Paralela: a complexidade na construção, não apenas dos códigos paralelos, mas principalmente, da conceituação do Pensamento Computacional Paralelo. E isso passa, não apenas pelos conteúdos complexos e a forma diferente de se programar, mas também, pela falta de abordagens didáticas que despertem, no aluno, o interesse e a motivação necessários para aumentar e disponibilizar, cada vez mais, profissionais nessa área [Jr et al. 2022].

3.1. Metodologia PCAM

A metodologia criada por [Foster 1995] descreve as etapas necessárias para a implantação de um projeto de um algoritmo 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 analisar 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).

O Particionamento determina como uma tarefa pode ser dividida em subtarefas menores. Para isso, é necessário identificar e retirar possíveis gargalos e dependências. Já na Comunicação, irá se determinar uma forma de se estabelecer uma comunicação entre as subtarefas definidas na fase anterior. A Aglomeração tem a finalidade de associar as subtarefas similares, isso proporcionará uma otimização na troca de informações entre elas. Na última fase, tem-se o Mapeamento, que tem como objetivo atribuir aos nós de processamento as aglomerações definidas. Isso proporcionará um melhor balanceamento da carga, acelerando e otimizando o processamento da tarefa.

3.2. Pensamento Computacional Paralelo

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 [Freire 2025]. Está definido sobre quatro pilares: Decomposição, Abstração, Reconhecimento de Padrões e Algoritmo.

A Decomposição é similar ao Particionamento definido na metodologia PCAM. Já a Abstração, concentra-se nos elementos mais relevantes e necessários, enquanto se ignora pontos irrelevantes. É utilizado em diversas etapas da construção e da compreensão

do Pensamento Computacional [Brackmann 2017]. O Reconhecimento de Padrões define que tarefas similares podem utilizar soluções semelhantes. Estabelecer e reconhecer padrões são importantes para fazer o reuso de conceitos e soluções [Nipo et al. 2022]. Já o Algoritmo é um conjunto de estratégias ou instruções, claras e objetivas, que irão compor uma solução de um problema. É um dos principais pilares do Pensamento Computacional, pois agrupa todos os demais, abrangendo diversos momentos das atividades [Wing 2014].

O Pensamento Computacional Paralelo refere-se à abordagem estruturada de resolução de problemas que se inicia com a avaliação de um código sequencial em busca de gargalos e análise de dependência de variáveis, com o objetivo de melhorar o seu desempenho. Essa solução estruturada passa pela decomposição do problema em subtarefas e finaliza com a elaboração de um algoritmo otimizado e eficiente, e é fundamental para ajudar estudantes da área a compreenderem os conceitos fundamentais do paralelismo [De Jesus Oliveira Duraes et al. 2020].

4. Formação Conceitual do Pensamento Computacional Paralelo

A metodologia da formação conceitual foi definida para dividir a implantação dos conteúdos em etapas (essas etapas serão as fases no Jogo Paralegram). Cada etapa é responsável por demonstrar como deve ser aplicado um conceito do PCAM junto aos pilares do Pensamento Computacional (PC). No diagrama da Figura 1, as etapas Particionamento, Comunicação, Aglomeração e Mapeamento (da metodologia PCAM) estão relacionadas com a conceituação de Decomposição, Reconhecimento de Padrões, Abstração e Algoritmo (pilares do Pensamento Computacional).

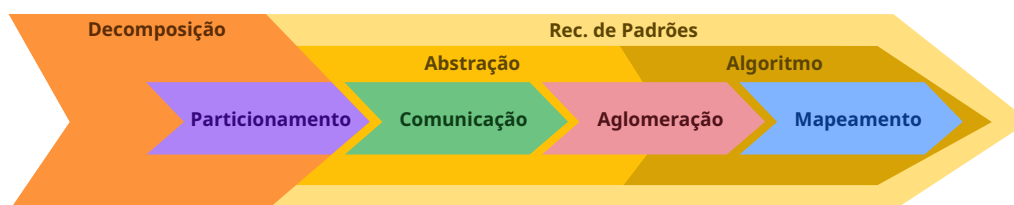


Figura 1. Ciclo do Pensamento Computacional Paralelo.

A primeira etapa descrita como **Decomposição**, no Pensamento Computacional, é descrita como **Particionamento**, na metodologia PCAM. Tem como objetivo segmentar um problema grande e complexo em tarefas menores com a finalidade de agilizar a compreensão e resolução do problema [Galante 2023]. Os dois conceitos se encontram no mesmo patamar, considerados semelhantes na prática. Portanto, podem ser desenvolvidos utilizando bases similares de conhecimentos. Considerando o Jogo Paralegram, nesta fase será necessário separar as diversas caixas, que estão misturadas, nas diversas cores diferentes, como representado na Figura 2

A segunda etapa é a **Comunicação** é planejada para que as subtarefas, resultantes da primeira etapa, possam trocar informações, caso seja necessário. Neste ponto, é identificado no problema, estruturas que poderão ser relacionadas entre si, por esse motivo a **Abstração** e o **Reconhecimentos de Padrões** encontrados no Pensamento Computacional são aplicados à **Comunicação**. Contextualizando no jogo, será necessário estabelecer uma comunicação entre duas empilhadeiras para que elas não colidam. Elas irão transportar as caixas já separadas, na etapa 1, para os caminhões, como representado na Figura 3.

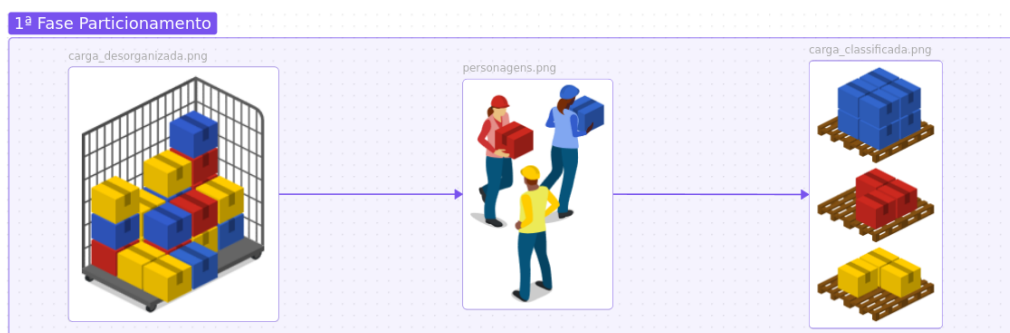


Figura 2. Primeira Fase: Particionamento.

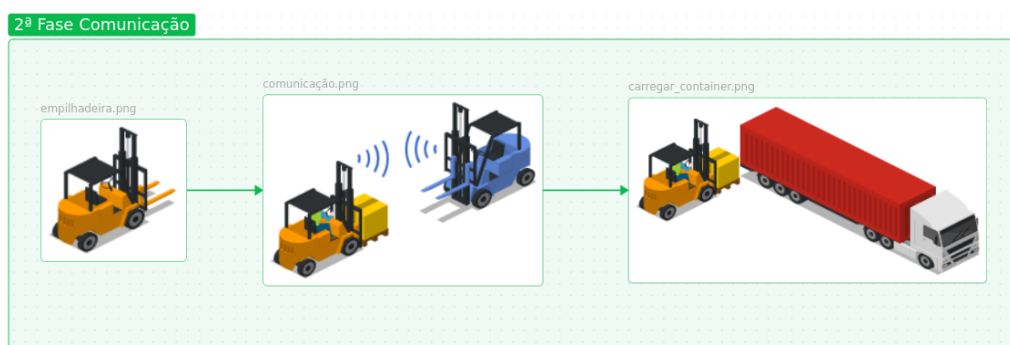


Figura 3. Segunda Fase: Comunicação

Com objetivo de evitar o gerenciamento de pequenos domínios, a **Aglomerção** reduz a sobrecarga das atividades, agrupando subtarefas, para que possam ser processados de forma estratégica. Após analisar os procedimentos do problema, as subtarefas que compartilham dados comuns são reunidas em grupos, aplicando **Reconhecimento de Padrões**. Essa otimização tem a intenção de adaptar a estrutura para execução em uma máquina real. O Reconhecimento de Padrões é essencial, nesta etapa, para verificar quais dados serão executados em conjunto, reduzindo a granulação. No jogo isso é representado pela inclusão das caixas separadas em *contêiners* de cada cor, que serão manuseados por um guindaste, como na Figura 4.

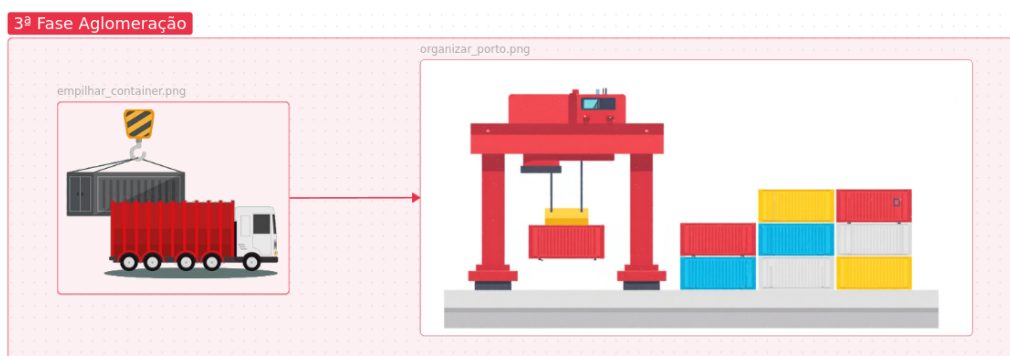


Figura 4. Terceira Fase: Aglomeração.

A última etapa é do **Mapeamento**. As tarefas agrupadas, anteriormente, devem ser atribuídas conforme a arquitetura alvo, levando em conta o menor custo de execução.

Neste ponto, serão aplicados o balanceamento de carga nos nós computacionais, onde todos devem possuir aproximadamente a mesma carga de trabalho para que o tempo de execução tenha uma média igual entre os núcleos de processamento. Além disso, a prática de minimizar a comunicação entre as unidades de execução é considerada como segunda atribuição desta etapa, a finalidade é reduzir as interações frequentes [Galante 2023]. No jogo, essa etapa é representada pela distribuição das cargas (*containers*) nos navios, como representado na Figura 5.

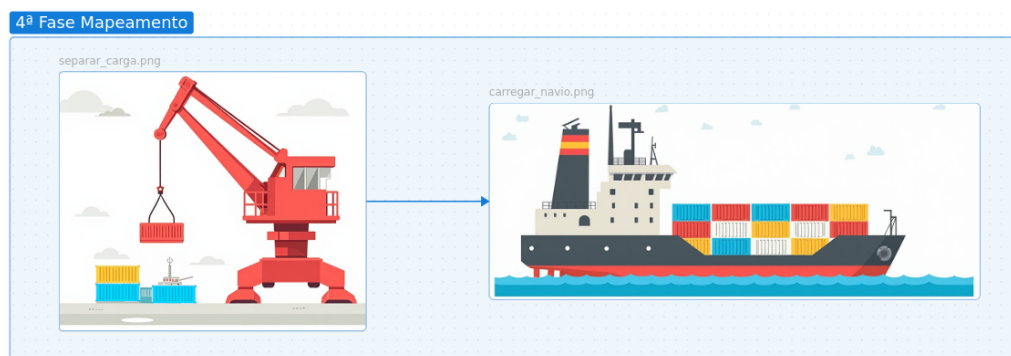


Figura 5. Quarta Fase: Mapeamento.

Como produto do mapeamento, é gerado um **Algoritmo** fortemente ajustado ao cenário onde será executado. A meta é reduzir o tempo de execução transformando o algoritmo abstrato em algo mais próximo da solução prática [Lorenzon and Schnorr 2023]. Na Figura 1 a região delimitada para o Algoritmo engloba a atividade do Mapeamento, demonstrando assim, que ao finalizar o ciclo da metodologia PCAM, em conjunto com o Pensamento Computacional, espera-se obter como resultado não apenas um código paralelo otimizada, mas também, uma formação conceitual solidificada na construção do Pensamento Computacional Paralelo.

5. Jogos Sérios

Os Jogos Sérios são ambientes interativos que favorecem o engajamento dos alunos através de um aprendizado ativo e, por que não dizer, divertido e competitivo. Esses jogos acabam por motivá-los de forma a permitir um *feedback* relevante no apoio no processo de aprendizagem. Possuem, também, um ambiente eficaz para introduzir conteúdos específicos, como a Programação Paralela, pois supera desafios e dificuldades que podem ser característicos de cada assunto [Cardoso 2025].

5.1. Paralegram e a Programação em Blocos

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 e as diretrizes do Pensamento Computacional, e é realizada através da evolução das fases do jogo. Para cada fase, são apresentados os objetivos, a habilidade a ser desenvolvida e os resultados esperados.

O fluxo do jogo é baseado em jogos de tabuleiro, e têm 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 conceitual que o aluno vai desenvolvendo, conforme avança nas tarefas que são solucionadas, utilizando a programação em blocos. 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, conforme ilustrada na Figura 6.

Figura 6. Tela referente à Segunda Fase do Paralegram



Na Figura 6 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 (3), área onde os blocos serão estruturados para a sequência dos movimentos no tabuleiro; Em (4), são definidos botões de execução/parada da programação definida em (3), além dos botões de limpar (C) e ajuda (?) e em (5) são definidos e apresentados os objetivos de cada fase. O jogo foi desenvolvido em um ambiente gráfico 2-D através da ferramenta *Game Maker Studio* para Windows. 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* e na linguagem de programação *Python*.

6. Metodologia Desenvolvida

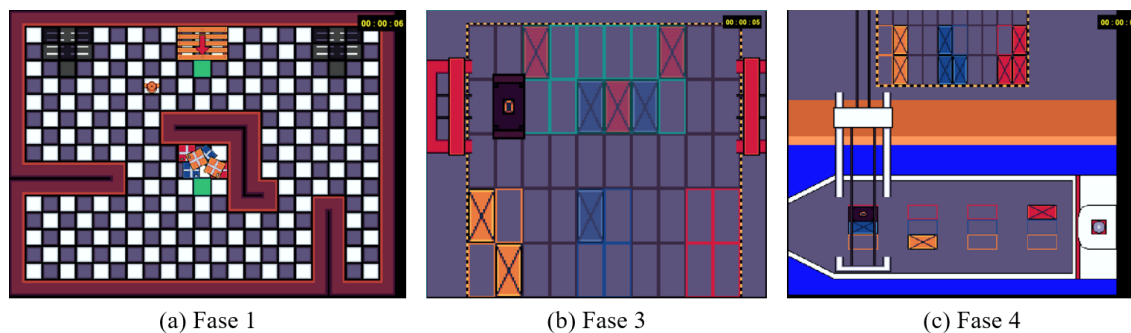
A construção conceitual foi dividida nas fases da Metodologia PCAM associadas aos pilares do Pensamento Computacional, conforme Figura 1.

6.1. Primeira Fase: Particionamento

Os produtos contidos no centro de distribuição são divididos em caixas, com diferentes cores. Essas caixas devem ser separadas e empilhadas (pelas cores). Nesta fase, o objetivo é demonstrar formas de dividir 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 7(a).

Figura 7. Fases do Paralegram



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.

6.2. Fase 2: Comunicação

Na segunda fase, é necessário realizar a programação de forma a evitar que duas empilhadeiras colidam no corredor do centro de distribuição, simulando dessa forma, uma comunicação, conforme pode ser visualizado na Figura 6, marcado por um quadrado amarelo. Para classificação é necessário, novamente, o reconhecimento dos padrões, através das cores das mercadorias. O objetivo dessa fase é utilizar a habilidade da Abstração para estabelecer (simular) uma comunicação entre diferentes objetos, evitando uma possível colisão.

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 preestabelecidos (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.

6.3. Fase 3: Aglomeração

Nesta fase, com os produtos já separados e carregados nos contêineres, serão direcionados ao pátio da transportadora. Nessa etapa, será utilizada uma empilhadeira de contêiner, simulada na Figura 7(b), responsável manuseá-los e agrupá-los. O objetivo desta fase é definir os movimentos necessários para agrupar e organizar objetos similares.

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 (Reconhecimento 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).

6.4. 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 7(c), eles devem ser colocados na ordem em que serão retirados, caso contrário a extração pode ser inviabilizada. O objetivo desta fase é definir o número de estruturas necessárias (guindastes) para realizar a tarefa no menor tempo possível.

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).

7. Avaliação

O jogo foi avaliado por 28 alunos do curso de Bacharelado em Ciência da Computação. Para a realização da avaliação, foram considerados os seguintes passos: (1) Apresentação dos pilares do Pensamento Computacional e da Metodologia PCAM; (2) Acesso ao Tutorial do Jogo; (3) Interação com o jogo Paralegram; (4) Aplicação de um questionário para avaliar o **Aprendizado** (quanto as atividades propostas no jogo estimularam a evolução do aprendizado com as tarefas) e a **Usabilidade** (facilidade ou dificuldade de uso e a experiência quanto a interface geral). Para isso, foi utilizada a escala *Likert* com cinco alternativas, variando de 1 (Discordo Totalmente) a 5 (Concordo Totalmente). Vale observar que os alunos conseguiram realizar as quatro fases do jogo em um tempo médio total de 39 minutos. O que pode ser considerado positivo, considerando um tempo de aula tradicional, entre 45 e 50 minutos.

7.1. Cuidados Éticos

O trabalho foi desenvolvido em conformidade com as resoluções do Conselho Nacional de Saúde (CNS), especificamente a Resolução CNS nº 466/2012 (diretrizes gerais) e a Resolução CNS nº 510/2016 (específica para Ciências Humanas e Sociais, aplicável a

muitos estudos de IHC/Computação). Dessa forma, foi aprovado pelo Comitê de Ética de Pesquisa da UFPa o processo Nº 92975125.5.0000.0018, enviado em 02/07/2025 e aprovado em 17/10/2025. Vale ressaltar que os participantes assinaram o Termo de Consentimento Livre e Esclarecido (TCLE).

8. Resultados e Discussões

O primeiro item avaliado foi o Aprendizado (A1-A7). As perguntas realizadas estão apresentadas no Quadro 1, e as respostas, podem ser visualizadas na Figura 8.

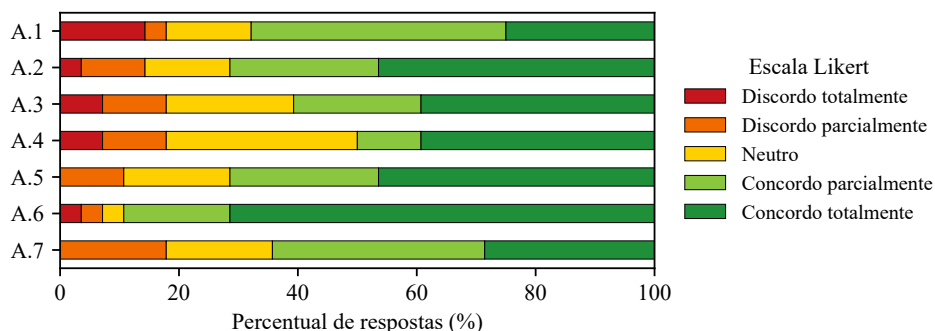
ID	Aprendizado
A.1	O jogo aprofundou a minha compreensão dos conceitos teóricos de Programação Paralela (Metodologia PCAM e Pensamento Computacional Paralelo)?
A.2	O editor de código em blocos, integrado na plataforma, facilitou a prática e a experimentação dos códigos paralelos?
A.3	Os conteúdos teóricos e exemplos fornecidos no jogo foram eficazes para o meu aprendizado sobre Programação Paralela?
A.4	A utilização do jogo me motivou a me envolver mais ativamente com os conceitos da Programação Paralela?
A.5	O jogo me auxiliou a aplicar diferentes conceitos de Programação Paralela de forma prática (Metodologia PCAM e Pensamento Computacional Paralelo)?
A.6	A estrutura de ensino do jogo (editor de códigos, animações e conteúdos) tornou o aprendizado mais dinâmico?
A.7	O jogo me ajudou a entender as vantagens e desvantagens da aplicação da Programação Paralela?

Quadro 1. Avaliação do Aprendizado

Percebe-se que, no geral, a avaliação foi positiva, se forem consideradas as escalas "Concordo Parcialmente" e "Concordo Totalmente". Porém, nos itens (A.4) e (A.3), percebe-se uma maior incidência de avaliações "Neutras". Isso pode ser justificado, pois os itens avaliaram o envolvimento dos conceitos e o aprendizado da Programação Paralela, respectivamente. Como os alunos tiveram apenas uma introdução simples sobre a metodologia PCAM, e não tiveram uma formação completa sobre esse conteúdo (um curso durante todo um semestre), isto é, os alunos não cursaram uma disciplina de Programação Paralela, isso acabou não despertando, naquele momento, uma maior compreensão por parte deles.

Já com relação ao conjunto de instruções e comandos pré-estabelecidos, quase 70% indicaram "Concordar Totalmente ou Parcialmente" que auxiliaram na elaboração do algoritmo, para a solução dos problemas propostos, item (A.2). Mesmo resultado do item (A.5), com relação a forma como o jogo auxiliou a aplicação dos diferentes conceitos de forma prática, na associação da Metodologia PCAM e Pensamento Computacional Paralelo. Outra avaliação positiva, com quase 90%, foi com relação a estrutura definida no jogo, item (A.6).

O segundo item avaliado foi a Usabilidade (U1-U7). As perguntas realizadas estão apresentadas no Quadro 2, e as respostas, podem ser visualizadas na Figura 9.

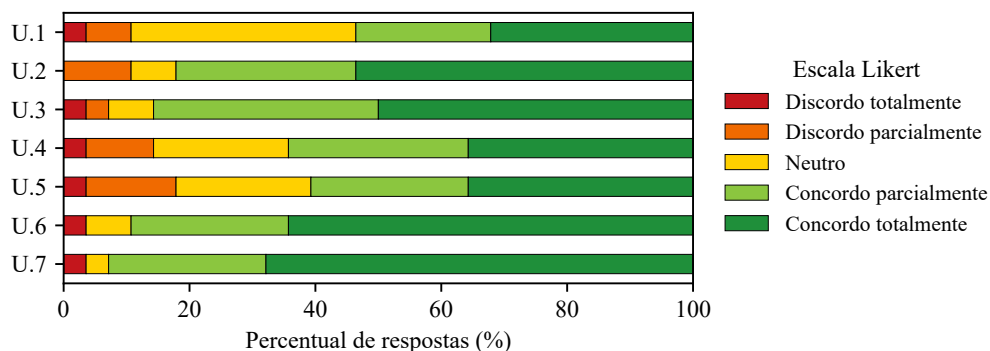
Figura 8. Resultados do Aprendizado

ID	Usabilidade
U.1	A Interface geral do jogo é clara, organizada e fácil de navegar?
U.2	A funcionalidade do editor de códigos para criar, modificar e executar código é intuitiva?
U.3	Encontrou facilidades em aprender a utilizar todas as funcionalidades do jogo?
U.4	O desempenho técnico do jogo (rapidez, estabilidade) é satisfatório durante a compilação e execução dos códigos?
U.5	O <i>feedback</i> do jogo (mensagens de erros, avisos) é compreensível e ajuda na resolução dos problemas?
U.6	O jogo mantém uma consistência na terminologia e nos elementos de interação?
U.7	O <i>design</i> visual (cores, <i>layout</i> , elementos gráficos) é agradável e contribui para a experiência de uso?

Quadro 2. Avaliação da Usabilidade

Percebe-se, que de uma forma geral, o jogo teve uma avaliação positiva nos quesitos avaliados. No entanto, pode-se perceber, também, que as avaliações mais negativas foram para as questões (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 acesso ao tutorial do jogo, muitos alunos acabaram não finalizando a leitura, que explicava vários pontos importantes que deveriam ser levados em consideração, e isso acarretou em dúvidas durante a execução do jogo, bem como, um maior tempo de compreensão das tarefas para serem executadas.

Como último ponto que pode ser destacado da Figura 9 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*, 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.

Figura 9. Resultados da Usabilidade

8.1. 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.

Outro ponto que pode ser destacado é que antes de iniciar o jogo Paralegram os alunos eram incentivados a acessarem o tutorial, porém como o mesmo era definido em termos de textos, percebeu-se que nem todos chegavam até o final. Acabavam abandonando pela metade e iniciavam o jogo. Isso foi bastante prejudicial, pois quando chegavam em determinadas tarefas, não sabiam o que tinha que ser executado. Isso resultava em pedidos de ajudas ou retornos ao tutorial, atrasando, dessa forma, a conclusão do jogo.

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, incluir um contador de instruções e de passos, para avaliar a quantidade de instruções utilizadas pelos alunos, bem como, o número de passos dado por cada componente do jogo. Além disso, pretende-se inserir um tutorial que seja mais dinâmico, com vídeos curtos mais didáticos e interativos.

10. Agradecimentos

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

Declaração sobre Uso de Inteligência Artificial

Os autores declaram que não foram utilizadas ferramentas de Inteligência Artificial (IA) no processo de elaboração ou revisão deste artigo. Todo o conteúdo científico, análise de dados, interpretação dos resultados e a seleção bibliográfica foram realizados integralmente pelos autores.

Referências

- ACM and IEEE-CS (2020). Computing curricula: Paradigms for global computing education.
- 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).
- Brackmann, C. (2017). *Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica*. PhD thesis.
- Cardoso, A. e. a. (2025). Avaliação de um jogo sério com minijogos baseados no desafio bebras para apoio ao ensino do pensamento computacional. pages 837–848, 20 a 24 de julho de 2025. XXXIII Workshop sobre Educação em Computação (WEI).
- De Jesus Oliveira Duraes, T., Sergio Lopes De Souza, P., Martins, G., Jose Conte, D., Garcia Bachiega, N., and Mazzini Bruschi, S. (2020). Research on parallel computing teaching: State of the art and future directions. volume 2020-October. cited By 3.
- DeLozier, C. and Shey, J. (2023). Using visual programming games to study novice programmers. *International Journal of Serious Games*, 10(2):115–136.
- 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.
- Force, C. T. (2020). *Computing Curricula 2020: Paradigms for Global Computing Education*. Association for Computing Machinery, New York, NY, USA.
- Foster, I. (1995). Designing parallel algorithms. In *Designing and Building Parallel Programs*, page 26–82. Addison-Wesley Longman Publishing Co. cap. 2.
- Freire, C. A. e. a. (2025). Desenvolvendo habilidades do pensamento computacional em crianças: Uma oficina de criação de jogos no scratch. pages 341–352, 20 a 24 de julho de 2025. XXXIII Workshop sobre Educação em Computação (WEI).
- Galante, G. (2023). Projeto de aplicações paralelas. *Minicursos da XXIII Escola Regional de Alto Desempenho da Região Sul*, pages 25–32.
- Jr, R., Farias, H., and Araújo, J. (2022). Um mapeamento sistemático da literatura sobre o ensino e aprendizagem da programação paralela com o uso de placas gráficas. pages 439–450.

- Lorenzon, A. and Schnorr, L. (2023). *Pensamento Computacional Paralelo: Desafios do Presente e do Futuro*, pages 53–78. Sociedade Brasileira de Computação.
- 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)*.
- MEC (2016). Resolução nº 5, de 16 de novembro de 2016. (2016). In 220, editor, *Diário Oficial da União (DOU)*, 1, page 22.
- Nipo, D. T., Rodrigues, R. L., and França, R. (2022). Jogando e pensando: Aprendendo pensamento computacional com jogos de entretenimento. *Anais do XXXIII Simpósio Brasileiro de Informática na Educação (SBIE 2022)*.
- SBC (2019). Diretrizes para ensino de computação na educação básica. In *Diretrizes para ensino de Computação na educação Básica Relatório Técnico nº 001/2019*.
- Wing, J. (2014). Computational thinking with jeannette wing. *Columbia Journalism School*.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.
- ZORZO, A. F., NUNES, D., MATOS, E., STEINMACHER, I., de ARAUJO, R. M., CORREIA, R., and MARTINS, S. (2017). Referenciais de formação para os cursos de graduação em computação. In *Sociedade Brasileira de Computação (SBC)*, page 153.