

# **WAlgor: um jogo de *tower defense* para o desenvolvimento do Pensamento Computacional e apresentação de algoritmos computacionais**

**Fábio Michel<sup>1</sup>, Fernanda Pires<sup>1</sup>, Marcela Pessoa<sup>1</sup>**

<sup>1</sup>ThinkTEd Lab - Pesquisa e Desenvolvimento em Tecnologias Educacionais  
Escola Superior de Tecnologia (EST) – Universidade do Estado do Amazonas (UEA)  
Av. Darcy Vargas, 1.200 – 69050-020 – Manaus – AM – Brasil

{fpires, fmm16, mspessoa}@uea.edu.br

***Abstract.** This paper proposes a game that aims to stimulate the development of Computational Thinking through the presentation and application of computational algorithms to solve problems in a playful way. WAlgor is a tower defense game with mechanics similar to Plants vs Zombies and Trolls vs Vikings, whose goal is to defend a kingdom using knights, following logical strategies. The conception of mechanics was based on the four pillars of Computational Thinking. The game is aimed at children from 10 years old, with application recommended both in school environment and for the entertainment of children, youth and adults.*

***Resumo.** Este artigo propõe um jogo que tem como objetivo estimular o desenvolvimento do Pensamento Computacional através da apresentação e aplicação de algoritmos computacionais para a resolução de problemas de forma lúdica. WAlgor é um jogo do estilo tower defense com mecânica similar a Plants vs Zombies e Trolls vs Vikings, cujo objetivo é defender um reino utilizando cavaleiros, seguindo estratégias lógicas. A concepção da mecânica foi baseada nos quatro pilares do Pensamento Computacional. O jogo é voltado para crianças a partir dos 10 anos, com aplicação recomendada tanto em ambiente escolar quanto para o entretenimento de crianças, jovens e adultos.*

## **1. Introdução**

Diante dos avanços alcançado pela ciência e tecnologia, surge o interesse de estender tais inovações para a área de educação de forma a promover a aprendizagem de conceitos computacionais desde a educação básica. A aprendizagem de computação não está limitada à utilização ou manipulação de softwares [França e Tedesco 2015] mas também aos conceitos de modelos computacionais, algoritmos e suas complexidades, aplicações para a resolução de problemas, dentre outros conteúdos [Nunes 2010].

Por outro lado, ajudar o estudante a desenvolver o raciocínio lógico é uma atividade importante. Wing (2006) apresenta o Pensamento Computacional como uma habilidade essencial para a resolução de problemas. Heinz et al. (2016) destacam a necessidade da aprendizagem de Computação desde a educação básica por auxiliar os estudantes no desenvolvimento de habilidades para a resolução de problemas computacionais, lógicos, Matemáticos e de outras ciências. Países na Europa, na

Oceania e os Estados Unidos perceberam a relevância do tema e têm se esforçado em promover iniciativas para incorporar disciplinas os conceitos de Computação na Educação Básica [Heintz et al. 2016].

O Brasil, apesar de já discutir a inclusão de conteúdos relacionados à Computação nas séries iniciais, encontra-se em um cenário distante de conseguir implementar nos currículos escolares, fazendo com que tais conteúdos sejam limitados ao Ensino Técnico ou Superior na área de Computação. Pode-se destacar o fato ocorrido no ano de 2017 quando o termo Pensamento Computacional foi integrado à Base Nacional Comum Curricular do Brasil (BNCC). Entretanto, o surgimento do termo na BNCC não sugere a introdução dos conceitos de Computação como disciplina na Educação Básica, mas que o Pensamento Computacional é uma habilidade a ser desenvolvida em meio a outras disciplinas. Por isso, surgem as discussões acerca do termo apresentado na BNCC em como utilizar a transversalidade ao incorporar Pensamento Computacional nas disciplinas curriculares da educação básica. Ao encontro destas iniciativas, a Diretoria de Educação da Sociedade Brasileira de Computação (SBC) elaborou um documento com as Diretrizes<sup>1</sup> para o ensino de Computação na Educação Básica.

Aliar a aprendizagem de conceitos computacionais na Educação Básica, pode ser encarado como um desafio no cenário brasileiro, mas possibilita espaço para discussões por parte de pesquisadores e profissionais do ambiente escolar [França e Tedesco 2015]. Alguns trabalhos destacam a utilização de metodologias, jogos e outras ferramentas que podem ser utilizadas para estimular o desenvolvimento Pensamento Computacional [Vicari 2018]. Considerando este cenário, o presente trabalho propõe o desenvolvimento de um jogo digital com base nos quatro pilares do Pensamento Computacional, possuindo como finalidade a apresentação implícita de algoritmos computacionais e aplicações para a resolução de problemas.

O restante do trabalho está estruturado da seguinte forma: a Seção 2 descreve os trabalhos relacionados, seguida pela Seção 3 que aborda o aporte teórico. A Seção 4 apresenta o desenvolvimento do jogo proposto, descrição de mecânica e a avaliação. A Seção 5 registra os resultados alcançados com a mecânica proposta, seguida pela Seção 6 com as considerações finais.

## 2. Trabalho Relacionados

Para auxiliar na aplicação de mecânicas voltadas para a Computação e para desenvolvimento de habilidades que estimulem o Pensamento Computacional, Zobot et al. (2018) apresentam uma revisão sistemática da literatura analisando diversos jogos, identificando gêneros são mais adequados. Os autores classificam jogos que possuem mecânicas com desafios em forma de enigmas, como capazes de auxiliar no desenvolvimento do Pensamento Computacional por engajar os jogadores a solucionar os problemas apresentados. Destacam também jogos com elementos de interação representados por abstrações dos conceitos a serem trabalhados através de imagens e ícones.

---

<sup>1</sup> Disponível em: <https://www.sbc.org.br/educacao/diretrizes-para-ensino-de-computacao-na-educacao-basica>

Chung et al. (2016) propõem um jogo classificado como *serious game* para a resolução divertida e envolvente de problemas NP-difíceis. Os autores não especificam a faixa etária do público-alvo, mas em uma avaliação com 40 participantes relatam que os usuários que jogaram várias vezes aumentaram sua pontuação e apresentaram uma forma consistente na resolução dos problemas propostos. Os autores complementam que o processo de implementação deve contar uma interface amigável e intuitiva para motivar aqueles que possuem dificuldades em compreender problemas computacionais. O WAlgor também é uma aplicação classificada como *serious game*, sendo uma alternativa diferente das abordagens convencionais de resolução de problemas computacionalmente complexos. Como diferencial ele explora os conceitos e aplicações de algoritmos de forma implícita para a resolução de problemas, sem a necessidade de visualizar ou manipular estruturas que necessitam ser programadas.

No contexto de jogos que podem auxiliar o Pensamento Computacional no âmbito escolar, em Pires et al. (2018) é proposta a criação de um jogo eletrônico sobre sustentabilidade para coleta seletiva voltado para crianças de oito anos de idade. Os autores destacam que o processo de produção da aplicação teve como base os quatro pilares do Pensamento Computacional, com mecânica e *gameplay* contendo problemas lógicos de forma a exercitar o pensamento lógico e estimular o desenvolvimento do Pensamento Computacional através da resolução dos problemas propostos. Além dos objetivos primários, o jogo propõe a transversalidade de conteúdos por apresentar o tema sustentabilidade e auxiliar no desenvolvimento do Pensamento Computacional. O WAlgor também busca desenvolver o Pensamento Computacional sob a perspectiva de abordar aplicação de algoritmos computacionais.

### **3. Pensamento Computacional e jogos para a aprendizagem de computação**

Wing (2014,) define que quando combinado o pensamento lógico aos conceitos fundamentais da Computação, obtém-se como resultado uma metodologia de resolução de problemas, definindo tal habilidade como Pensamento Computacional. Com base nisso, o Pensamento Computacional pode ser definido como uma habilidade cognitiva inata dos seres humanos que necessita de estímulo assim como a habilidade de falar, escrever, andar etc. Após pesquisas motivadas por iniciativas de Code.Org (2016), Liukas (2015) e BBC (2015), foram estabelecidos os 4 pilares fundamentais que compõem o Pensamento Computacional: decomposição, abstração, reconhecimento de padrões e algoritmo.

A decomposição consiste em formular um problema complexo dividindo-o em partes menores e que sejam mais fáceis de serem resolvidas ou gerenciadas. A análise de cada parte pode resultar na identificação de problemas parecidos que já possuam solução, sendo este o processo de reconhecimento de padrões. Na abstração, todos os detalhes irrelevantes para a solução devem ser desconsiderados e as informações essenciais são filtradas. Por fim, obtém-se o algoritmo que, ao ser seguido, resulta na solução do problema. Quando tal habilidade cognitiva é estimulada, é possível obter auxílio na compreensão de sistemas computacionais, interpretação e resolução de problemas complexos independente de área de atuação [Brackmann 2017]. Neste ponto, surge a discussão sobre quais ferramentas podem ser utilizadas como auxílio para o estímulo do Pensamento Computacional. Uma das abordagens utilizadas para o auxiliar o desenvolvimento do Pensamento Computacional são os jogos.

Huizinga (2014) destaca que jogos são atividades voluntárias do ser humano e estiveram presentes desde os primórdios de nossa história por se tratar de algo lúdico. Jogos em meios digitais surgem por volta da década de 50 em meio à ascensão dos computadores, onde os primeiros jogos eram exibidos em telas de osciloscópios [Rogers 2014]. Impulsionados pelo surgimento de novas tecnologias, jogos eletrônicos ganharam espaço dentro dos lares através de consoles, computadores e *smartphones*.

Considerando o aumento da demanda por experiências engajadoras, o mercado de jogos movimenta bilhões de dólares oferecendo novas alternativas para jogadores que desejam dedicar horas em missões, exploração de mundos virtuais, relações interpessoais etc [McGonigal 2017], [Pires et al. 2018]. Nesse mercado, existem aqueles jogos que possuem o propósito além do entretenimento dos jogadores, chamados de *serious games*. *Serious games* são jogos desenvolvidos com objetivos além do entretenimento [Alvarez 2018]. Segundo Dörner et al. (2016), *serious games* possuem um mundo próprio, que pode ter ou não relação com o mundo real, e que pretende prender a atenção de quem o está jogando. Mas a grande questão é, os jogos realmente têm o potencial de causar um impacto positivo na aprendizagem dos alunos?

Dörner et al. (2016) afirma que os jogos podem potencializar o desenvolvimento de habilidades analíticas, espaciais, estratégicas, de atenção e até mesmo *insight*. Além disso, podem melhorar as habilidades de reconhecimento e resolução de problemas, tomadas de decisão, colaboratividade, além do potencial de melhorar a memória de curto e longo prazo. Jogos podem ser observados como potenciais ferramentas para apresentar conceitos complexos de Computação no ambiente escolar da educação básica.

A Sociedade Brasileira de Computação (SBC) destaca que a aprendizagem de Computação é essencial e seu avanço tem sido um dos fatores para as mudanças no cenário tecnológico atual. A SBC reforça que a computação deve ter espaço desde o Ensino Fundamental por estar presente dentre diversas áreas do conhecimento [Raabe 2017]. Além de destacar a importância da Computação na Educação Básica, alguns autores mostram como o desenvolvimento do Pensamento Computacional é uma forma de instigar os estudantes a formular hipóteses e encontrar soluções com o uso do raciocínio lógico [Alencar 2012].

#### **4. WAlgor: jogo digital para a aprendizagem de algoritmos**

WAlgor é um jogo eletrônico para plataformas móveis, de ação e estratégia no estilo *tower defense*. O nome do jogo é uma abreviação para *Warriors of Algorithms*, em tradução livre, Guerreiros Algorítmicos. A mecânica foi baseada em sucessos consolidados no mercado de jogos como *Plants vs Zombies* e *Trolls vs Vikings*. Nesses jogos, as batalhas têm como objetivo principal a defesa de uma base dos ataques inimigos, posicionando guerreiros de acordo com os recursos disponíveis. O público-alvo são usuários a partir de 10 anos de idade.

Após a definição de estilo de jogo e mecânica, seguido pela prototipação, iniciou-se a montagem dos cenários e personagens das fases utilizando artes do repositório *Craftpix*. Em seguida, o ambiente de desenvolvimento foi definido e a *game engine Construct 2* foi escolhida por apresentar bom desempenho, qualidade e agilidade no processo de implementação de jogos 2D e por possibilitar a exportação do projeto para plataformas Web, Linux, Windows, iOS e Android.

WAlgor apresenta conceitos de algoritmos e suas aplicações de forma implícita, sob um cenário fictício de batalha medieval. O jogo possui quatro tipos de batalhas que abordam os problemas computacionais de busca, ordenação, coloração de grafos e o problema da mochila. A dificuldade evolui de acordo com o progresso do jogador. Em cada nível o usuário deve executar as estratégias com o objetivo de combater o progresso dos inimigos. No entanto, os movimentos são limitados sendo necessário utilizar uma fonte para continuar a execução do algoritmo da fase. Ao final de cada batalha há um ataque com o avanço de todos os guerreiros restantes do exército inimigo. Cada fase finaliza quando todos os ataques são contidos

#### 4.1. Enredo

A história do jogo se passa em um reino, chamado Reino de Turin que tem como maior orgulho a sua força militar, formada por cavaleiros que são conhecidos pelas ótimas estratégias para vencer batalhas. As estratégias foram pensadas pelo comandante Ziviani.

Um dos cavaleiros das tropas tornou-se um traidor, sendo expulso do reino. Como vingança, revelou os segredos das tropas para os inimigos das terras de Polinômia e Non-Polinômia o que resulta em um ataque contra o reino de Turin. As estratégias para combater os inimigos de Polinômia são mais fáceis de serem executadas, ao contrário dos exércitos de Non-Polinômia por serem mais difíceis de serem combatidos. Por isso, o jogador deve assumir a posição de comandante das tropas, liderando os cavaleiros na defesa dos súditos e do reino de Turin.

#### 4.2. Proposta de mecânica e *gameplay*

Para combater os ataques inimigos, o jogador necessita utilizar algoritmos computacionais para posicionar os cavaleiros frente aos castelos do reino. Para apresentar os conceitos e aplicação dos algoritmos de forma divertida e atrativa, a mecânica foi desenvolvida baseada nas atividades apresentadas no livro *Computer Science Unplugged* [Bell et al. 2015] aliada aos elementos de jogos e estilo de *gameplay* similar à *Plants vs Zombies e Trolls vs Vikings*. A mecânica provoca a necessidade de se fazer uso do raciocínio lógico para posicionar os cavaleiros na batalha, além de estimular o Pensamento Computacional através da resolução dos problemas em defesa do reino.

A Tabela 1 destaca como os 4 pilares do Pensamento Computacional são estimulados diante da mecânica proposta no jogo WAlgor. Nas fases 1 e 2 são abordados o problema de Busca. Na fase 1 o jogador possui uma lista de cavaleiros que está desordenada e deve posicionar os guerreiros no campo de batalha, buscando sequencialmente na lista pelo guerreiro específico que aparecerá na tela. Cada guerreiro da lista possui um número. No canto inferior esquerdo é destacado o cavaleiro que pode ser posicionado em batalha, ao lado do número que o identifica. O usuário pode utilizar as setas para percorrer a lista até o cavaleiro correto.

**Tabela 1. Pilares do Pensamento Computacional e *gameplay* das fases**

Pilar do Pensamento Computacional	Descrição
Decomposição	Para executar a formação de defesa do reino, o usuário precisa <b>decompor</b> a tropa, posicionando um cavaleiro por vez.
Reconhecimento de	Após posicionar o primeiro cavaleiro, o jogador começa a compreender a

padrões	estratégia a ser seguida e o padrão para buscar os cavaleiros ou objetos nas listas, montando a estratégia para a formação da defesa do reino.
Abstração	Para defender o reino, o jogador deve perceber que precisa considerar os recursos disponíveis e seguir a estratégia para que sua formação seja executada pela defesa do reino.
Algoritmo	Executar o algoritmo para posicionar os guerreiros com o uso dos recursos disponíveis e evitando que os inimigos avancem até o reino.

Na fase 2 o jogador possui uma lista ordenada de cavaleiros e pode utilizar a busca binária para encontrar o guerreiro específico que aparecerá na tela. Da mesma forma como na fase 1, no canto inferior esquerdo é destacado o cavaleiro que pode ser posto em batalha. Para iniciar a busca, o jogador deve clicar no botão “buscar”. Após esta ação, o cavaleiro do centro da lista é selecionado e em seguida o jogador deve decidir para qual lado a busca será continuada, seguindo a condição da busca binária: se maior, a mesma ação da busca é continua para o lado direito da lista; se menor, a mesma ação da busca é continua para o lado esquerdo da lista; se igual, o cavaleiro pode ser posto em batalha.

Nas fases 3 e 4, o problema apresentado é de ordenação. Na fase 3 o jogador possui uma lista desordenada e precisa posicionar os cavaleiros em batalha utilizando a estratégia baseada no *Selection Sort*. O jogador poderá posicionar um cavaleiro em batalha se estiver na posição de início e for o cavaleiro de menor índice na lista. O jogador poderá percorrer a lista com as setas da barra inferior e realizar a troca da carta do menor cavaleiro e em seguida vai para a posição de destaque do início da lista, no canto inferior esquerdo da tela. A cada cavaleiro posicionado corretamente em batalha, a lista diminui um elemento. Quando a lista possuir 0 elementos, uma nova lista de cavaleiros é gerada.

Na fase 4 o jogador possui uma lista desordenada e precisa posicionar os cavaleiros em batalha utilizando a estratégia baseada no *Merge Sort*. Para iniciar, o jogador deverá tocar no botão “Ordenar” o que resultará em duas sublistas ordenadas. Em seguida o usuário poderá utilizar as setas para percorrer entre os cavaleiros nas primeiras posições de cada sublista. O jogador poderá posicionar em batalha o cavaleiro de menor índice dentre aqueles nas primeiras posições de cada sublista. Quando um cavaleiro é posto em batalha, a sublista que o cavaleiro pertencia perderá um elemento. Se as duas sublistas possuírem 0 elementos, uma nova lista de cavaleiros é gerada.

Nas fases 5 e 6, são retratados problemas classificados pela teoria da complexidade como solucionáveis em tempo polinomial, por um algoritmo não determinístico. Os inimigos a serem enfrentados são das terras de Non-Polinômia. A fase 5 aborda uma adaptação do problema de coloração de vértices de um grafo. Para defender o reino, o jogador deve posicionar os guerreiros na batalha, buscando em uma lista desordenada, de forma sequencial, pelo arqueiro específico que aparecerá na tela. Cada arqueiro terá uma cor e atira flechas para frente ( $0^\circ$ ) e para as diagonais ( $45^\circ$  e  $315^\circ$ ). O jogador não poderá posicionar arqueiros da mesma cor se estiverem na mesma direção de ataque. Por exemplo, se um arqueiro de cor vermelha for atingido por uma flecha de outro arqueiro da mesma cor, será excluído da batalha.

A fase 6 aborda o problema da mochila. Para defender o reino, o jogador deve posicionar os guerreiros no campo de batalha, buscando em uma lista desordenada, pelos acessórios que o cavaleiro irá utilizar. O guerreiro ficará em destaque no canto inferior esquerdo da tela, ao lado de um número que indica o peso que é capaz de

suportar. Cada acessório também possui um peso e o valor que pode fortalecer o cavaleiro. O jogador deve escolher qual a melhor combinação dos acessórios, sem ultrapassar o peso suportado pelo guerreiro e com os itens que possuem o melhor valor para fortalecê-lo para a batalha.



Figura 1. Telas de *gameplay* das fases

Em todas as fases, o jogador possui um número limitado de recursos para percorrer as listas. O número mostrado na parte superior das listas, indica o número de recursos disponíveis para o jogador. A cada ação de percorrer alguma das listas, o jogador perderá um recurso. Por isso, na parte direita da barra inferior, ao lado da lista de cavaleiros ou objetos, existe uma carta que é uma fonte de recursos que poderá ser posicionada na batalha. Após ser posicionada, a cada 15 segundos é oferecida uma joia ao jogador. Quando a joia é coletada, o jogador ganha mais 5 recursos. Esta mecânica foi pensada com o intuito de que os jogadores percebam que uma estratégia gasta mais recursos do que outra. Por exemplo, a busca sequencial irá exigir mais recursos do que a busca binária. Ou ao utilizar a ordenação com a *Merge Sort*, irá perceber que menos recursos são exigidos em comparação ao *Selection Sort*.

### 4.3. Apresentação de recursos computacionais

WAlgor foi desenvolvido para apresentar algoritmos computacionais e de forma atrativa e divertida aos jogadores. Por isso, os elementos computacionais presentes no jogo fazem parte da experiência de *gameplay* para que os jogadores possam manipular tais estruturas enquanto estão imersos na tentativa de proteger o Reino. A Tabela 2 apresenta os recursos computacionais presentes nas fases e suas respectivas funções entre os recursos estão citados: chave de busca, lista de cavaleiros, lista de objetos, carta de recursos, número de recursos, setas, botão buscar, botão ordenar, índice de cavaleiro, peso suportado pelo soldado, peso e poder do objeto.

Tabela 2. Conceitos computacionais presentes em WAlgor.

Fase	Item	Caraterísticas
1, 2 e 5	Chave de Busca	A chave de busca indica ao jogador qual cavaleiro deve ser buscado na lista para ser posto em batalha.
1, 2, 3, 4, 5	Lista de cavaleiros	As listas armazenam os cavaleiros disponíveis para busca ou ordenação (dependendo da fase).
6	Lista de objetos	A lista de objetos está presente na fase 6 e armazena os objetos para fortalecer o cavaleiro para a batalha.
Todas as fases	Carta de Recursos	A carta de recursos é disponibilizada em batalha para fornecer recursos ao jogador para executar a ação de percorrer as listas.
Todas as fases	Número de recursos	O número de recursos informa ao jogador quantas vezes pode ocorrer a ação de percorrer os elementos da lista.
1, 3, 4, 5 e 6	Setas	As setas são utilizadas pelo jogador para percorrer as listas.

2	Botão Buscar	O botão buscar na fase 2, executa a ação de selecionar o elemento do centro da lista ao seguir a busca binária.
4	Botão Ordenar	O botão ordenar na fase 4, ordena, divide a lista principal de cavaleiros, gerando duas sublistas ordenadas.
1, 2, 3, 4, 5	Índice do cavaleiro	O índice do cavaleiro é informado em cada carta nas listas que podem ser percorridas pelo jogador.
6	Número de peso suportado pelo soldado	O número de peso na fase 6 informa ao jogador o número máximo que o cavaleiro irá suportar, levando em conta o peso dos objetos que podem ser carregados por ele na batalha.
6	Peso do objeto	O peso do objeto é informado em cada carta da lista de objetos na fase 6.
6	Poder do objeto	O poder do objeto está em cada carta da lista de objetos na fase 6 para informar o poder o cavaleiro obtém ao ir para a batalha.

Em cada fase o jogador realiza a execução de um algoritmo para que a estratégia seja executada corretamente. A Figura 4 apresenta como o algoritmo a ser seguido pelo jogador na execução da fase 1:

```

reino_sob_ataque <- Verdadeiro
lista_cavaleiros = [10]
recursos_disponíveis <- 10
enquanto reino_sob_ataque == Verdadeiro faça:
    cavaleiro_posicionado <- Falso
    gerar_lista_aleatória (lista_cavaleiros, (1, 20))
    chave_busca <- lista_cavaleiros [numero_randomico(0,9)]
    carta_cavaleiro <- 0
    enquanto cavaleiro_posicionado == Falso faça:
        escreva (lista_cavaleiros[carta_cavaleiro])
        se (lista_cavaleiros[cavaleiro_lista] == chave_busca) então:
            escreva (“O cavaleiro pode ser posicionado na batalha”)
            cavaleiro_posicionado <- Verdadeiro
            lista_cavaleiros.limpar()
        se (recursos_disponíveis > 0) então:
            se (seta_esquerda_clicada == Verdadeiro && carta_cavaleiro != 0) então:
                carta_cavaleiro <- carta_cavaleiro - 1
                recursos_disponíveis <- recursos_disponíveis - 1
            se (seta_direita_clicada == Verdadeiro && carta_cavaleiro != 9) então:
                carta_cavaleiro <- carta_cavaleiro + 1
                recursos_disponíveis <- recursos_disponíveis - 1

```

Figura 2. Algoritmo em para a resolução da fase 1.

## 5. Resultados e discussões

O jogo apresentado ainda está em fase de implementação e testes com os desenvolvedores para que depois sejam realizados os testes e validação com o público-alvo. Para que os jogadores se sintam imersos ao ambiente proposto, a mecânica visa aliar um estilo de jogo consolidado no mercado com o estímulo ao Pensamento Computacional para a resolução de problemas através da utilização de conceitos computacionais.

Apesar de ainda não ter sido testado com o público-alvo, o jogo foi apresentado para um profissional especialista na área de desenvolvimento de jogos, para profissionais de Licenciatura em Computação e especialistas em *design* de



aprendizagem, cientistas de computação da área de estruturas de dados e complexidade computacional, tendo mecânica aprovada para que seja finalizada, testada e posteriormente disponibilizada para o público geral.

## 6. Considerações finais

Este trabalho teve como proposta apresentar o desenvolvimento do jogo WAigor sob uma perspectiva multidisciplinar com a finalidade de apresentar a aplicação de algoritmos computacionais na resolução de problemas, além do estímulo ao Pensamento Computacional. Vale ressaltar que o jogo pode ser utilizado como ferramenta de auxílio em processos de aprendizagem ou para entretenimento de jovens e adultos.

Além disso, tem como objetivo contribuir para a discussão acerca da inserção de metodologias para o desenvolvimento do Pensamento Computacional sob uma perspectiva transversal nas disciplinas na Educação Básica, por ser essencial na formação de habilidades para a resolução de problemas em diversas áreas do conhecimento.

Ainda estão sendo elaborados ajustes e inserção de novos elementos na aplicação. Como trabalhos futuros, a mecânica proposta pode ser implementada para a criação de novas fases ou novos jogos que desejam apresentar outros tipos de algoritmos computacionais de forma implícita.

## Referências

- Alvarez, J. (2018). Les Serious Games. *Revue de l'APÉMu*, 83.
- BBC. (2015). Introduction to computational thinking. Retrieved from <https://www.bbc.com/bitesize/guides/zp92mp3/revision/1>
- Bell, T., Witten, I. H., Fellows, M., Adams, R., McKenzie, J., Powell, M., & Jarman. (2015). CS Unplugged: An enrichment and extension programme for primary-aged.
- Brackmann, C. P. (2017). *Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica*. Universidade Federal do Rio Grande do Sul. Porto Alegre,
- Chung, C., Matsuoka, A., Yang, Y., Rubin, J., & Chechik, M. (2016). *Serious games for NP-hard problems: challenges and insights*. Paper presented at the 2016 IEEE/ACM 5th International Workshop on Games and Software Engineering (GAS).
- Code.Org. (2016). *Code Studio Lesson Plans for Courses 1,2, and 3*.
- de Sousa Pires, F. G., Melo, R., Machado, J., Silva, M. S., Franzoia, F., & de Freitas, R. (2018). *EcoLogic: um jogo de estratégia para o desenvolvimento do pensamento computacional e da consciência ambiental*. Paper presented at the Anais dos Workshops do Congresso Brasileiro de Informática na Educação.
- de Sousa Pires, F. G., Michel, F., Bernardo, J. R. S., Melo, R., & de Freitas, R. (2018). *O livro do conhecimento: Um jogo de aventura para exercitar a ortografia*. Paper presented at the Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE).
- Dörner, R., Göbel, S., Effelsberg, W., & Wiemeyer, J. (2016). *Serious games: foundations, concepts and practice*: Springer.

- França, R., & Tedesco, P. (2015). *Desafios e oportunidades ao ensino do pensamento computacional na educação básica no Brasil*. Paper presented at the Anais dos Workshops do Congresso Brasileiro de Informática na Educação.
- Heintz, F., Mannila, L., & Färnqvist, T. (2016). *A review of models for introducing computational thinking, computer science and computing in K-12 education*. Paper presented at the 2016 IEEE Frontiers in Education conference (FIE).
- Huizinga, J. (2014). *Homo Ludens IIs 86*: Routledge.
- Liukas, L. (2015). *Hello Ruby: Adventures in Coding*: Feiwel & Friends.
- McGONIGAL, J. (2017). *A realidade em jogo*: Editora Best Seller.
- Nunes, D. J. (2010). Computação ou informática. *Jornal da Ciência*, 30.
- Raabe, A. (2017). Referenciais de formação em computação: Educação básica. *Sociedade Brasileira de Computação*.
- Rogers, S. (2014). *Level Up! The guide to great video game design*: John Wiley & Sons.
- Vicari, R. M., Moreira, A. F., & Menezes, P. F. B. (2018). Pensamento computacional: revisão bibliográfica.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. (2014). Computational thinking benefits society. *40th Anniversary Blog of Social Issues in Computing, 2014*.
- Zabot, D., Andrade, S., & Matos, E. (2018). *Raciocínio Computacional e Jogos Digitais: categorias e mecânicas*. Paper presented at the Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE).