

## PENSAMENTO COMPUTACIONAL: A CONSTRUÇÃO DO JOGO PAC-MAN NO SCRATCH

Luan Padilha dos Santos  
Universidade Estadual de Maringá - UEM  
padilha.luan16@gmail.com

Suzana Pereira do Prado  
Universidade Estadual do Paraná – UNESPAR  
suziprado@gmail.com

Noeli Teresinha de Almeida  
Universidade Estadual do Paraná – UNESPAR  
noeli.almeida@escola.pr.gov.br

Suely Maria de Souza  
Universidade Estadual do Paraná – UNESPAR  
suelymariadesouza86@gmail.com

Sérgio Carrazedo Dantas  
Universidade Estadual do Paraná – UNESPAR  
sergio.dantas@ies.unespar.edu.br

### Resumo

Este trabalho tem por objetivo analisar uma construção do jogo Pac-Man no Scratch, com base nos pilares do Pensamento Computacional. O Scratch é uma plataforma que utiliza uma linguagem de programação simplificada baseada em blocos que se encaixam, formando os algoritmos. Com ele, é possível criar histórias, jogos e animações com *scripts* que se conectam. Como aporte teórico, este trabalho baseia-se na definição de Pensamento Computacional elaborada pelo grupo de pesquisa Autômato. A motivação deste estudo surgiu a partir de um curso oferecido para professores pelo Núcleo Regional de Educação de Apucarana - PR, e a construção do jogo analisado foi proposta em um dos cinco módulos do curso. O Scratch pode propiciar diversas maneiras de raciocínio lógico, tomada de decisões, descoberta e correções dos erros, aprendendo a elaborar hipóteses para buscar as resoluções e chegar ao resultado. Dentre os pilares do Pensamento Computacional, destaca-se a *depuração*. Esse processo se fez presente entre uma ação e outra durante toda a construção do jogo, nos momentos em que foram realizados testes para verificação de seu funcionamento. Nessa etapa foi percebido equívocos e repensada a programação do jogo.

**Palavras-chave:** Educação Matemática. Linguagem de programação. Autômato.

### Introdução

O uso da computação passou a fazer parte de diversas áreas do conhecimento. Podemos encontrar exemplos de aplicação da computação em áreas como: Medicina, no gerenciamento de registros médicos, análises de tomografias e ressonâncias magnéticas, além de algoritmos que auxiliam no diagnóstico de doenças (BASÍLIO *et al.*, 2020); Engenharia, usada para modelagem e simulação de estruturas complexas, otimização de projetos, controle de processos industriais e design de sistemas eletrônicos (LOPES, TAKAHASHI, 2011); e Educação, na instrução auxiliada pelo

computador, tutoriais, jogos, simuladores, e o computador como uma ferramenta, planilhas, processadores de texto, linguagens de programação, entre outros (VALENTE, 2005).

No que diz respeito ao uso educacional do computador, os construcionistas, seguidores de Seymour Papert, têm como principal foco a aprendizagem. Eles veem o computador como uma ferramenta para explorar o mundo e aprender com ele. A linguagem Logo, por exemplo, permite uma abordagem envolvendo conceitos da matemática como recursão, decomposição, algoritmos e geometria (RAABE, ZORZO, BLIKSTEIN, 2020).

Nos últimos anos, o conceito de Pensamento Computacional (PC) tem recebido uma atenção crescente no campo da Educação e de outras áreas de conhecimento. Jeannette Wing, renomada cientista da computação e professora da Universidade de Columbia, desempenhou um papel fundamental na popularização e definição dessa expressão. Em seu artigo *Computational Thinking*, publicado em 2006, Wing apresentou uma abordagem para resolver problemas e enfrentar desafios utilizando os princípios da ciência da computação (RAABE, COUTO, BLIKSTEIN, 2020). Esse artigo é reconhecido na área de Computação como o texto que inicia a discussão sobre a expressão Pensamento Computacional.

O Pensamento Computacional refere-se a um conjunto de habilidades e processos cognitivos que os computadores e os cientistas da computação utilizam para resolver problemas. No entanto, Wing (2006) argumenta que essas habilidades não são exclusivas da área de computação e podem ser aplicadas em diversas áreas do conhecimento (RAABE, COUTO, BLIKSTEIN, 2020). Ela defende que o Pensamento Computacional é uma habilidade fundamental para todos, independentemente da profissão ou idade, em uma sociedade cada vez mais permeada pela tecnologia.

Quando tratado a partir da Educação Matemática, encontramos em Navarro (2021) a defesa de que o Pensamento Computacional em ambientes escolarizáveis pode ser definido como um processo de resolução de situações problemas, plugadas e/ou desplugadas, que perpassam aspectos entre os nexos conceituais da resolução de problemas e dos pensamentos algébrico e algorítmico, desenvolvendo, dessa forma, a capacidade de interpretação, organização, generalização, abstração e produção de conhecimento matemático.

Dentre as alternativas para trabalhar com programação em ambientes educacionais está a plataforma Scratch<sup>1</sup>, que é uma plataforma pertencente ao *Media Lab do Massachusetts Institute of Technology* (MIT), de Boston - EUA, no qual os usuários utilizam blocos de programação para criar histórias, jogos e animações digitais (SOUZA, 2018). É possível usar o Scratch *online* com acesso a

---

<sup>1</sup> É possível acessar a plataforma *online* no seguinte endereço: <https://scratch.mit.edu>

uma comunidade de estudantes, educadores e interessados no assunto, ou baixá-lo em um computador ou *tablet* e utilizá-lo *offline*.

Neste trabalho, analisamos a construção de um jogo inspirado no Pac-Man utilizando o Scratch e alicerçado em pilares do Pensamento Computacional. O projeto surgiu a partir de um curso oferecido para professores pelo Núcleo Regional de Educação de Apucarana - PR, coordenado pelo professor Dr. Sérgio Carrazedo Dantas e a construção deste jogo foi proposta em um dos cinco módulos do curso.

### **O Pensamento Computacional**

Para iniciarmos nosso diálogo em torno do Pensamento Computacional, é importante deixarmos claro que esta expressão vem sendo tratada com esta nomenclatura desde as décadas de 60-70, quando Papert a utilizou, porém, sem muita ênfase, preferindo *procedural thinking*, em que destacava o pensamento procedimental como uma ferramenta intelectual poderosa, comparando-a com o computador (ALBUQUERQUE, 2021). Ao passar dos anos, percebe-se diferentes abordagens e mudanças de discursos sobre o que é e como desenvolver o Pensamento Computacional. Na presente discussão abordamos definições de diferentes autores para dialogar com a nossa compreensão sobre o tema.

Navarro e Sousa (2019), destacam que o termo Pensamento Computacional teve origem na Educação Matemática através da linguagem Logo, por meio das pesquisas de Seymour Papert, cuja abordagem visava forjar ideias mais acessíveis. No entanto, o Pensamento Computacional e a linguagem envolvida perderam força na área da Educação Matemática, e com o avanço da tecnologia, diversos aplicativos e outros termos como Tecnologias Digitais (TD), Tecnologias de Informação (TI), Tecnologias de Informação e Comunicação (TDIC), permearam os ambientes escolares e relegaram a ideia de programação e algoritmo, com foco apenas no uso das tecnologias, como o objeto em si. Desta forma, o Pensamento Computacional passou a ser objeto de estudos apenas para cientistas da computação.

A partir de 2006, com o artigo *Computational Thinking* de Jeannete Wing, noções sobre Pensamento Computacional voltaram a ser discutidas, repensadas e adaptadas, inclusive para outras áreas de conhecimento, pois Wing advém da Ciência da Computação e naquela época ocupava uma posição influente na *National Science Foundation* (NSF). Para ela

O Pensamento Computacional é uma capacidade fundamental para qualquer um, e não apenas para os cientistas informáticos. À leitura, à escrita e à aritmética, devemos acrescentar o Pensamento Computacional à competência analítica de cada criança. (WING, 2021, p. 2)

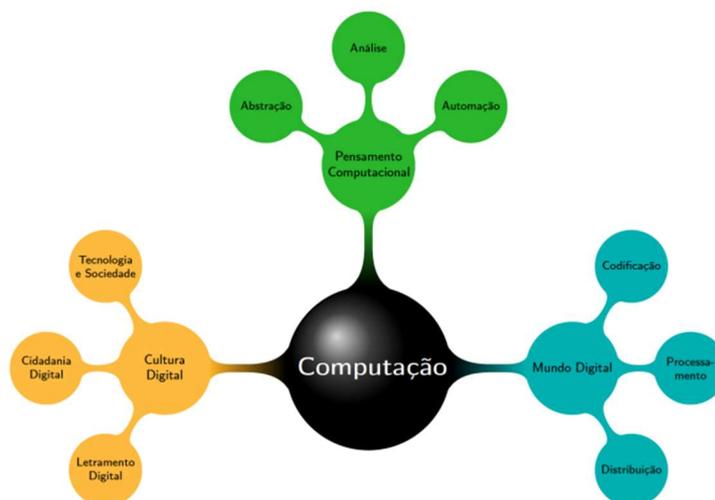
Com a visibilidade e popularização alcançada por Wing e tornando-se o computador imprescindível na sociedade e para a maioria das profissões, o Pensamento Computacional passou a ser tema de interesse de pesquisadores, governantes, instituições educacionais e empresas de tecnologia.

Segundo a Sociedade Brasileira de Computação (SBC),

O Pensamento Computacional se refere à capacidade de sistematizar, representar, analisar e resolver problemas. Apesar de ser um termo recente, vem sendo considerado como um dos pilares fundamentais do intelecto humano, junto com leitura, escrita e aritmética, pois como estes, serve para descrever, explicar e modelar o universo e seus processos complexos. (SBC, 2017, p. 3).

Percebemos na definição dada pela SBC muita similaridade com as palavras escritas por Wing. Acreditamos ser pertinente lembrarmos os esforços da SBC em contribuir na construção de um currículo de computação voltado para a Educação Básica, bem como em complementar a Base Nacional Comum Curricular (BNCC) na definição de objetivos de aprendizagem, competências e habilidades, não só sobre o Pensamento Computacional, mas também em relação à Cultura Digital e ao Mundo Digital, que compõe os conhecimentos da área de Computação organizados em três eixos:

**Figura 1 - Pilares da Computação**



Fonte: SBC, 2017, p. 4

Para buscarmos a solução de um problema, seja ele de qualquer natureza, utilizamos estratégias no intuito de resolvê-lo de forma rápida, eficaz e eficiente, ou seja, fazemos uso de mecanismos que a partir de agora chamaremos *pilares* do Pensamento Computacional.

Na Figura 1, no eixo Pensamento Computacional, são destacados os três pilares úteis para a resolução de problemas:

**Abstração:** compreender e utilizar modelos e representações adequadas para descrever informações e processos, e técnicas para construir soluções algorítmicas;

**Automação:** ser capaz de descrever as soluções por meio de algoritmos de forma que máquinas possam executar partes ou todo o algoritmo proposto, bem como de construir modelos computacionais para sistemas complexos;

**Análise:** analisar criticamente os problemas e soluções para identificar não somente se existem soluções que podem ser automatizadas, mas também ser capaz de avaliar a eficiência e a correção destas soluções. (SBC, 2017, p. 5, negrito nosso).

Ao tratar os processos de abstração, automação e análise, Ribeiro, Foss e Cavalheiro (2022), baseadas em Wing, argumentam que a abstração é um mecanismo importante do processo de solução de problemas, que possibilita efetivar aspectos reais e representar as situações mais relevantes de um problema e sua solução; para tanto, muitas ações estarão envolvidas, perpassando a descrição dos dados (entradas e saídas) e processos que descrevem o modo de se resolver o problema dado através de técnicas utilizadas na construção de algoritmos.

“A automação é a mecanização de todas ou de parte das tarefas da solução para resolver o problema usando computadores” (RIBEIRO, FOSS, CAVALHEIRO, 2022, p. 25). As autoras destacam que nem todos os problemas são passíveis de automação, pois nem todos são computáveis e também a relevância de mecanismos tais como máquina, linguagem e modelagem computacional.

A análise é a fundamentação oriunda da Ciência da Computação, “permitindo descobrir se um problema tem ou não solução computacional e se pode existir um algoritmo eficiente que o resolva, antes mesmo de construir o algoritmo” (RIBEIRO, FOSS, CAVALHEIRO, 2022, p. 27). É através da análise que percebemos a viabilidade, podemos corrigir possíveis erros e aferir a eficiência da solução proposta ao problema inicial.

Cabe ressaltar aqui nossa compreensão a respeito do termo *problema*. Para nós, problemas são situações que geram a necessidade de modificação de uma condição inicial, o que por vezes demanda de esforço mental e de organização do raciocínio e conseqüentemente de passos capazes de transformar de maneira eficiente e eficaz esta primeira condição dada.

O ser humano, em busca de compreender o mundo e adaptar-se a ele, resolve problemas que surgem cotidianamente em sua vida. Tais problemas, muitas vezes, demandam de conhecimento matemático e do emprego de raciocínio lógico. Assim, o problema é compreendido por nós como determinante, ele

[...] pode disparar um processo de construção de conhecimento. Sob esse enfoque, problemas são propostos ou formulados de modo a contribuir para a formação dos conceitos antes mesmo de sua apresentação e linguagem formal. (ONUChic, 1999, p. 207)

No primeiro parágrafo desta subseção, afirmamos que traríamos a nossa percepção de Pensamento Computacional, sendo assim, para nós do grupo de pesquisa Autômato<sup>2</sup>, o Pensamento Computacional ocupa-se do tratamento de entes abstratos em interface com o Pensamento Matemático na busca da resolução de problemas via uma série de etapas que possam ser executadas por um agente humano ou por um dispositivo digital. Em outras palavras, podemos resumir o Pensamento Computacional em seis processos/etapas não hierárquicas:

- **Formulação do problema:** etapa em que o problema é elaborado em termos de necessidade, objeto, pergunta e é, a partir daí, que se vislumbra um possível modo de resolução. Nessa etapa se questiona o que considerar, quais são as variáveis, que ações executar, e também, um ou mais planos de ação tendo como pano de fundo técnicas e repertórios.
- **Decomposição:** consiste em obter problemas menores a partir de um problema maior ou mais complexo. Com isso, é possível concentrar a atenção na resolução de partes específicas do problema, obtendo ao fim a solução. Para orientar a prática da decomposição, podemos nos questionar:
  - Como podemos usar os detalhes para identificar partes do problema, desafio ou tarefa?
  - Que partes são familiares? Que partes são desconhecidas?
  - Quais são as diferentes formas de resolver o problema, desafio ou tarefa?
  - É possível decompor as partes em partes menores?
  - Como a decomposição do problema pode servir para resolvê-lo ou compreendê-lo? (ESPADEIRO, 2021, p. 6)
- **Reconhecimento de padrões:** processo que pode surgir a partir da decomposição, quando problemas menores podem ser solucionados com base em experiências anteriores ou via repertórios matemáticos/computacionais, ou ainda, acontece ao perceber o que é constante ou variável nos dados ou formas associadas ao problema. Para identificar o reconhecimento de padrões, podemos nos questionar:
  - Que semelhanças ou padrões encontramos no problema? Por exemplo, quantos objetos existem? Que cores são identificadas? Que repetições são identificadas?
  - Como podemos utilizar os detalhes para identificar partes do problema? Que relações existem entre as partes?

---

<sup>2</sup> Grupo de Pesquisa e Estudos em Educação Matemática da Universidade Estadual do Paraná - Campus de Apucarana, tendo como líder o Prof. Dr. Sérgio Carrazedo Dantas e linha de pesquisa *Tecnologia para o ensino e aprendizagem de Matemática*.

- Como podemos descrever os padrões?
- Como se pode utilizar o padrão para fazer previsões ou tirar conclusões? (ESPADEIRO, 2021, p. 6)
- **Abstração:** é o processo de concentrar a atenção no que é necessário e suficiente para a resolução de um problema ou subproblemas, desconsiderando dados ou informações irrelevantes. Podemos ter em mente algumas questões a responder para estabelecer nosso processo de abstração, como, por exemplo:
  - Como podemos simplificar este problema/tarefa?
  - Qual é a informação relevante para resolver este problema/tarefa?
  - Como podemos representar claramente a informação importante?
  - De que forma podemos relacionar a informação importante tendo em vista o ponto de partida e o resultado a alcançar (dar resposta a problema ou resolver a tarefa)? (ESPADEIRO, 2021, p. 6)
- **Produção de algoritmos:** é o processo de obtenção de passos ou regras de ação desenvolvidos e efetivados durante a resolução de subproblemas ou problemas. Esses algoritmos podem ser (d)escritos em códigos da língua materna, códigos matemáticos, códigos de computação/programação, entre outros. Algumas questões orientadoras na produção de algoritmos:
  - Quais são as etapas necessárias para a resolução do problema?
  - Qual é a informação necessária para a concretização de cada uma das etapas?
  - Como estruturar todos os passos necessários para a resolução do problema ou tarefa? (ESPADEIRO, 2021, p. 6)
- **Depuração:** é o processo de procura e correção de erros. Além disso, poderá assumir, de igual modo, ações de testagem, verificação, refinamento e otimização da resolução apresentada. Na depuração as ações mentais em busca da solução podem ser afirmadas, revisadas, reformuladas ou abandonadas, o que leva a rever outras ações ou a conclusão do trabalho. Durante a depuração são produtivas perguntas como:
  - Como podemos garantir que o nosso plano, modelo, ou solução funcionou, ou não?
  - O resultado corresponde ao que esperávamos?
  - Como podemos modificar a abordagem para corrigir falhas ou imprecisões?
  - Como sabemos se conseguimos corrigir o erro? (ESPADEIRO, 2021, p. 6)

O grupo Autômato também compreende que o Pensamento Computacional e o Pensamento Matemático possuem relações estreitas nos processos de resolução de problemas, ou seja, ambos recorrem à utilização do raciocínio lógico, à decomposição e à análise dos problemas em partes menores e à identificação de padrões.

Para que as etapas de elaboração do Pensamento Computacional se efetivem, as propostas de trabalho devem estar bem elaboradas, possuindo intencionalidade pedagógica desde o momento em que se planeja até a execução, perpassando o processo de acompanhamento, possibilitando a práxis

através do estímulo da autonomia em determinados momentos, cooperação em outros períodos, criatividade e participação ativa ao longo do desenvolvimento das tarefas.

### O jogo Pac-Man no Scratch

Nesta seção apresentamos a análise de uma construção realizada no Scratch que foi inspirada no jogo de videogame Pac-Man (Figura 2). Esse jogo possui um personagem cujo objetivo é comer todas as pastilhas do labirinto sem ser atingido pelos fantasmas.

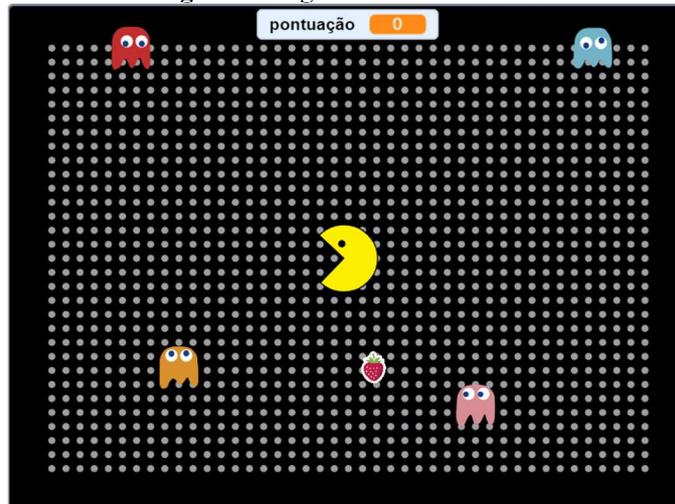
Figura 2 - Jogo Pac-Man



Fonte: Grátispng (2023).

No caso da nossa construção, substituímos as pastilhas por frutas. Ao colidir com uma fruta o Pac-Man a engole e, ao mesmo tempo emite um som, então a fruta desaparece do labirinto e a pontuação é incrementada. O personagem se movimenta no sentido horizontal e vertical, bem como os fantasmas que compõem o cenário do jogo. A seguir apresentamos o palco da nossa construção final no Scratch:

Figura 3 - Jogo Pacman no Scratch



Fonte: elaborado pelos autores

Para iniciar a construção do jogo no Scratch mobilizamos o primeiro pilar do Pensamento Computacional: a *Formulação do problema*. O problema dessa construção se caracterizou como: *Construir um jogo em que um ator se desloca pela tela nas posições horizontal e vertical em busca de capturar frutas e fugir da perseguição de quatro fantasmas*.

No que diz respeito à *Decomposição*, em nossa construção decompomos o problema geral em problemas específicos da seguinte forma:

- Construir a parte visual do jogo;
- Construção do ator principal e seus movimentos;
- Construir os movimentos do ator principal que se desloca ao toque das setas do teclado;
- Construir o movimento de quatro monstros pela tela;
- Fazer as frutas aparecerem aleatoriamente na tela;
- Marcar pontuação;
- Testar os contatos que fazem o jogo finalizar.

Cada um desses problemas elencados precisou de um tratamento diferente. No caso dessa construção, foram esses os problemas enunciados, o que pode variar de construção para construção, poderíamos ter elencado um número menor ou maior de problemas. Em nossa construção entendemos que a decomposição feita foi suficiente, pois contribuiu com a resolução do problema.

Para exemplificar o tratamento realizado para o problema: *Construir o movimento de quatro monstros pela tela*, apresentamos a programação do código utilizado em um dos fantasmas que se movimentam na tela:

**Figura 4 - Código de um dos fantasmas**



Fonte: elaborado pelos autores

Para o código desse fantasma criamos um bloco novo nomeado de *definições iniciais* (bloco rosa à esquerda), esse bloco possui a programação para fazer o fantasma ir para uma posição específica do palco e para ele mudar sua fantasia toda vez que o jogo é iniciado. Já o bloco do evento para iniciar o jogo (bloco laranja à direita) recebeu uma programação para o fantasma deslizar pelo palco no sentido dos eixos *X* e *Y* saindo da sua posição inicial (bloco *definições iniciais*). E quando esse fantasma tocar na borda do palco, ele aciona novamente o bloco *definições iniciais* e repete o processo.

No que se refere ao *Reconhecimento de padrões* podemos elencar dois padrões que estão presentes no jogo construído:

- Os quatro monstros possuem movimentos semelhantes, porém em sentidos diferentes;
- O surgimento das frutas é tratado por um simulacro de um evento aleatório.
- Com relação à *Abstração* podemos destacar dois itens:
- O Pac-Man se movimenta na vertical e na horizontal ao toque das setas do teclado;
- Testes de contatos entre os atores devem ser evitados, pois finalizam o jogo.

A respeito da *Produção de algoritmos*, em nossa construção esse pilar do PC está relacionado com a construção de blocos separados para tratar os comportamentos dos atores Pac-Man, monstros e frutas:

- Definições iniciais;
- Mudanças de fantasias;

- Pontuação;
- Testes de contatos.

Para exemplificar um dos itens enunciados, escolhemos a programação do bloco para a pontuação do jogo:

**Figura 5** - Código da pontuação



Fonte: elaborado pelos autores

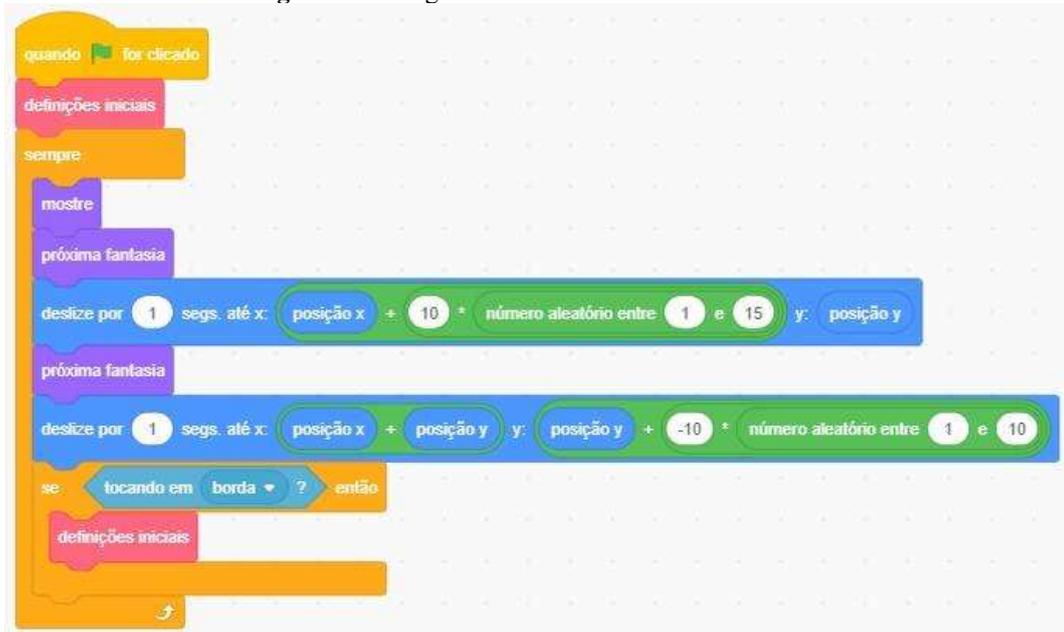
Inserimos essa programação no ator *fruta* e toda vez que a fruta for tocada pelo ator *Pac-Man* um ponto será adicionado na pontuação. Além disso, toda vez que o jogo for iniciado a pontuação é zerada.

Por fim, a *Depuração* se fez presente entre uma ação e outra durante toda a construção do jogo, nos momentos em que foram realizados testes para verificação de seu funcionamento. Nesses casos, comparamos o projeto mental do jogo com sua realização efetiva no Scratch.

Foi nessa etapa que percebemos equívocos e conseguimos ampliar e repensar aquilo que queríamos fazer. O processo de depuração nos permite revisitar nosso projeto e a construção constantemente, testando parte a parte cada passo construído, melhorando ou excluindo aquilo que não contribui com o que havíamos planejado.

A fim de podermos observar tal fato, colocaremos abaixo duas capturas de tela relacionadas a movimentação dos monstros no jogo. A Figura 6 apresenta um equívoco na linha de comando de movimentação vertical, o resultado era do fantasma se movimentando de forma muito rápida.

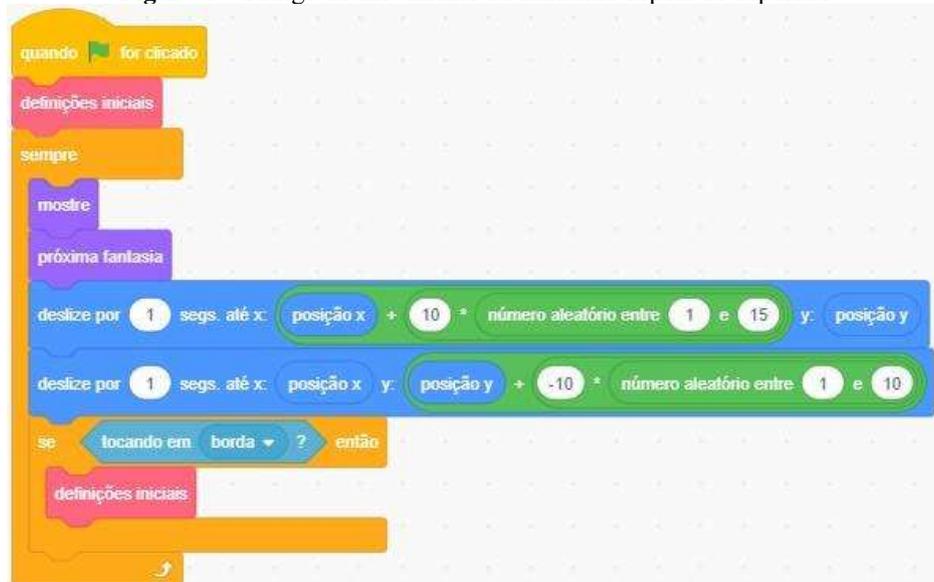
**Figura 6** - Código inicial do movimento do fantasma



Fonte: elaborado pelos autores

Já a figura a seguir mostra o código após depurado e testado, obtendo um movimento do fantasma de forma compassada, de acordo com o que pretendíamos.

**Figura 7** - Código do movimento do fantasma depois de depurado



Fonte: elaborado pelos autores

O processo de construção do jogo Pac-Man no Scratch nos permitiu explorar os pilares do Pensamento Computacional. E nessa discussão apresentamos uma análise de parte desses pilares presentes em nosso jogo devido à limitação de páginas.

## Considerações finais e reflexões

Diante da problemática exposta sobre os pilares que são mobilizados na construção do jogo Pac-Man no Scratch, tornou-se possível esboçar ao longo dessa discussão considerações que demonstram quais são esses pilares, segundo a definição de Pensamento Computacional do grupo de pesquisa Autômato.

Nesse sentido, é perceptível que são seis os pilares aqui estudados e demonstrados: Formulação do problema, Decomposição, Abstração, Reconhecimento de padrões, Produção de algoritmos e Depuração.

Dentre os pilares do Pensamento Computacional, destacamos a Depuração. Esse processo se fez presente entre uma ação e outra durante toda a construção do jogo, nos momentos em que foram realizados testes para verificação de seu funcionamento. Foi nessa etapa que percebemos equívocos e conseguimos ampliar e repensar aquilo que queríamos fazer. O processo de depuração nos permitiu visitar nosso projeto e a construção constantemente, testando parte a parte cada passo construído, melhorando ou excluindo aquilo que não contribui com o que havíamos planejado.

O reconhecimento de padrões nos permitiu otimização de tempo ao passo que ao percebermos o *reaproveitamento* de um mesmo bloco ou linha de códigos, utilizamo-nos não só da ferramenta copiar e colar, mas também o recurso *mochila* que proporciona copiar partes inteiras de um projeto para reutilizá-las em outros atores ou em outros projetos, de acordo com a necessidade.

Destacamos ainda sobre a Produção de algoritmos para que um computador realize certas tarefas, é preciso que os algoritmos estejam escritos em uma linguagem de programação, que é uma maneira de mostrar como acontece as sequências dos códigos para ele ser executável. Uma entrada de passos finitos que executa determinada sequência, produzindo uma saída que chegue ao objetivo proposto, daí a nossa opção pelo Scratch que fornece um conjunto de ferramentas visuais e intuitivas pelo seu formato de encaixe, além do feedback imediato viabilizando a verificação do projeto.

Dessa forma, o Scratch foi essencial por facilitar diferentes maneiras de abordar uma mesma proposta valendo-se do raciocínio lógico, tomada de decisões, descoberta e correções dos erros, proporcionando elaborar hipóteses para buscar as resoluções e chegar a um resultado. Percebemos que um mesmo problema pode ser observado, analisado e resolvido de formas diversas, mobilizando as etapas do Pensamento Computacional e sua utilização na construção de jogos como o do Pac-Man, por exemplo.

Por meio dessa discussão, esperamos fornecer uma possibilidade de trabalho com o uso do Pensamento Computacional e pretendemos inspirar educadores e pesquisadores a explorarem novas abordagens para o ensino e aprendizagem com o emprego do Pensamento Computacional.

Portanto, entendemos que é fundamental haver maior incentivo para o desenvolvimento de mais debates sobre o tema, viabilizando a participação da comunidade escolar e pesquisadores para que tenha realmente resultados satisfatórios para a educação. Que o Construcionismo de Papert, deve ser retomado como um desafio, para *aprender a aprender*.

## Referências

ALBUQUERQUE, Carlos. Pensamento Computacional e Matemática. **Educação e Matemática: Revista da associação de professores de matemática**, N.º 162 (2021): Revista temática sobre Pensamento Computacional, p. 31-38.

BASÍLIO, Arthur; DA SILVA, Pedro; MACHADO, Vinícius; ALDEMANN, Nayze. Estudo e Implementação de Algoritmos de Agrupamento e de Rotulação Aplicados no Diagnóstico por Imagens de Patologias Renais. In: **ESCOLA REGIONAL DE COMPUTAÇÃO DO CEARÁ, MARANHÃO E PIAUÍ (ERCEMAPI)**, 8., 2020, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 189-196.

ESPADEIRO, Rui Gonçalo. **O Pensamento Computacional no currículo de Matemática**. Educação e Matemática: Revista da associação de professores de matemática, N.º 162 (2021): Revista temática sobre Pensamento Computacional, p. 5 - 10.

GRATISPNG. **Pac-Man**. 2023. Disponível em: <https://www.gratispng.com/png-66aevr/download.html>. Acesso em: 17 jul. 2023.

LOPES, Heitor Silveiro; TAKAHASHI, Ricardo Hiroshi Caldeira. **Computação Evolucionária em Problemas de Engenharia**. 1 ed. Curitiba: Omnipax, 2011.

MARCATTO, F. S. F. Uma reflexão sobre resolução de problemas na formação de professores que ensinam Matemática. **Revemop**, v. 1, n. 3, p. 458-475, 1 set. 2019. Disponível em: <https://periodicos.ufop.br/revemop/article/view/1947/1673>. Acesso em: 20 jul. 2023.

NAVARRO, Eloisa Rosotti. **O desenvolvimento do conceito de pensamento computacional na educação matemática segundo contribuições da teoria histórico-cultural**. 2021. Tese (Doutorado em Educação) – Universidade Federal de São Carlos, São Carlos, 2021. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/15112>. Acesso em: 05 jul. 2023.

NAVARRO, Eloisa Rosotti; SOUSA, Maria do Carmo de. O Pensamento Computacional na educação matemática: um olhar analítico para teses e dissertações produzidas no Brasil. In: **Anais do Encontro Nacional de Educação Matemática**. Anais. Cuiabá – Mato Grosso, 2019. Disponível em: <https://www.sbemmatogrosso.com.br/xiiienem/anais.php>. Acesso em: 02 maio 2023.

RAABE, André; ZORZO, Avelino F.; BLIKSTEIN, Paulo. **Computação na Educação Básica: fundamentos e experiências**. Porto Alegre: Penso, 2020.

RESNICK, Mitchel. **Jardim de infância para a vida toda:** por uma aprendizagem criativa, mão na massa e relevante para todos. Penso Editora, 2020.

SBC. Sociedade Brasileira de Computação. **Referenciais de Formação em Computação:** Educação Básica. 2017. Disponível em: <https://bit.ly/3qSqzur>. Acesso em: 06 jul. 2023.

SCRATCH. **Scratch.** 2020. Disponível em: <https://scratch.mit.edu/>. Acesso em: 19 jul. 2023.

RIBEIRO, Leila; FOSS, Luciana. CAVALHEIRO, Simone André da Costa. Entendendo o pensamento computacional. In: RAABE, André; ZORZO, Avelino F.; BLIKSTEIN, Paulo. **Computação na educação básica:** fundamentos e experiências. Porto Alegre: Penso, 2020. p. 16-30.

SOUZA, Michel Figueiredo de; COSTA, Christine Sertã. **SCRATCH:** Guia Prático para aplicação na Educação Básica - 1. ed. - Rio de Janeiro: Imperial, 2018.

WING, Jeanette Marie. Pensamento Computacional. **Educação e Matemática: Revista da associação de professores de matemática**, N.º 162 (2021): Revista temática sobre Pensamento Computacional, p. 2 - 4.