



Edição Especial

III Congresso Internacional de Ensino - CONIEN
Universidade do Minho - Braga, Portugal, 2024

PENSAMENTO COMPUTACIONAL: UM ESTUDO COM BASE NA PROGRAMAÇÃO EM BLOCOS E NA GAMIFICAÇÃO

*COMPUTATIONAL THINKING: A STUDY BASED ON BLOCK PROGRAMMING
AND GAMIFICATION*

Ana Paula Germano Stadler¹
Joyce Jaqueline Caetano²
Ana Elisa Tozetto Piekarski³
Márcio André Martins⁴

Resumo

O Pensamento Computacional (PC) pode ser entendido como um conjunto de estratégias de resolução de problemas, envolvendo conceitos de Ciência da Computação, desenvolvimento de sistemas e compreensão do comportamento humano, habilidade fundamental para todos. Considerando a necessidade de desenvolver competências em PC, cabe aos professores proporcionar aos alunos práticas apropriadas e motivadoras. Embora existam ferramentas e plataformas disponíveis, a relação entre a realização das atividades e o desenvolvimento da capacidade do PC não é clara. Assim, o objetivo desta pesquisa foi investigar o desenvolvimento do PC a partir do uso da plataforma Programaê!. Para tanto, foi realizada uma experiência de ensino, utilizando um roteiro de tarefas selecionadas disponíveis na plataforma Programaê!. A plataforma disponibiliza atividades de resolução de problemas em contexto lúdico utilizando programação em blocos. Foram delineadas algumas relações entre as tarefas do roteiro e os pilares do PC, realizados

¹ Mestra em Ensino de Ciências Naturais e Matemática.

² Professora do Programa de Pós-graduação em Ensino de Ciências Naturais e Matemática (PPGEN) da Universidade Estadual do Centro-Oeste (UNICENTRO-PR).

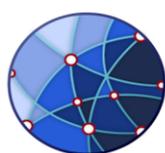
³ Professora do Departamento de Ciências da Computação (DECOMP) da Universidade Estadual do Centro-Oeste (UNICENTRO-PR).

⁴ Professor do Programa de Pós-graduação em Ensino de Ciências Naturais e Matemática (PPGEN) da Universidade Estadual do Centro-Oeste (UNICENTRO-PR).

REPPE: Revista do Programa de Pós-Graduação em Ensino

Universidade Estadual do Norte do Paraná, Cornélio Procópio (PR), v. 8, n. 2, p. 373-401, 2024

ISSN: 2526-9542



III CONIEN
Congresso Internacional de Ensino
PESQUISAS NA ÁREA DE ENSINO:
IMPACTOS, COOPERAÇÕES E VISIBILIDADE

DE 4 A 6 DE SETEMBRO
BRAGA - PORTUGAL



em cinco encontros remotos (via *Google Meet*), no contraturno escolar, totalizando 20 horas. Os encontros tinham em média três a cinco alunos do curso técnico em Informática. As informações da prática realizada foram coletadas por meio de observação participante, gravação de áudios e vídeos, e foram analisadas com base nos preceitos da Análise de Conteúdo. Durante a realização das atividades propostas, foi possível identificar, em algumas tarefas selecionadas, que os alunos utilizaram práticas e ações associadas ao PC para atingir os objetivos da atividade. Verificou-se que a programação em blocos, envolvendo a resolução de problemas em contexto de gamificação, é potencial ao desenvolvimento da capacidade de PC.

Palavras chave: Ensino Médio; Programação de Computadores; Gamificação.

Abstract

Computational Thinking (CT) can be understood as a set of problem-solving strategies involving concepts from Computer Science, system development, and understanding human behavior, making it a fundamental skill for everyone. Considering the need to develop CT skills, it is up to teachers to provide students with appropriate and motivating practices. Although there are tools and platforms available, the relationship between carrying out activities and developing CT capacity is unclear. Therefore, the aim of this research was to investigate the development of CT through the use of the Programaê! Platform. To this end, a teaching experiment was conducted using a set of selected tasks available on the Programaê! Platform. The Platform offers problem-solving activities in a playful context using block-based programming. Some relationships were outlined between the tasks in the script and the pillars of CT, conducted over five remote sessions (via *Google Meet*), during after-school hours, totaling 20 hours. The meetings averaged three to five students from the technical course in Computer Science. The information from the carried, out practice was collected through participant observation, audio and video recordings, and analyzed based on the principles of Content Analysis. During the performance of the proposed activities, it was possible to identify, in some selected tasks, that students used practices and actions associated with the CT to achieve the objectives of the activity. Our findings indicate that block programming, which incorporates problem-solving within a gamified context, enhances Computational Thinking skills.

Keywords: High School; Computer Programming; Gamification.

Introdução

O Pensamento Computacional (PC) tem sido discutido no âmbito educacional como uma capacidade a ser desenvolvida no estudante, atrelada à resolução de problemas. Neste sentido, no conjunto de aprendizagens essenciais expressas na Base Nacional Comum Curricular (BNCC), o PC é mencionado como uma habilidade específica em Matemática (Brasil, 2018). Essa proposição impacta diretamente o trabalho do professor de matemática em sala de aula, tornando relevante a busca por alternativas metodológicas de ensino e materiais didáticos adequados.

Acerca das categorias de abordagem do PC na Educação Básica, apontadas por Valente (2016), com base em um levantamento contemplando perspectivas de diferentes autores, são consideradas: a exploração de atividades lúdicas; a representação de procedimentos ou algoritmos por meio de blocos visuais; a utilização de textos, imagens, animações e outros recursos digitais. Isto posto, segundo Navarro e Souza (2023), cabe ao professor repensar os seus planejamentos de modo a criar situações pedagógicas promotoras do PC.

Assim, destacamos a importância de uma formação pedagógica continuada, de caráter exploratório e investigativo (Menezes *et al.*, 2012), com ênfase no papel do professor-pesquisador, ou seja, daquele que problematiza e propõe caminhos para a sua prática com base em uma reflexão teórica e metodológica. Nessa conjuntura, o presente trabalho admite como foco de estudo a elucidação de oportunidades potenciais para o desenvolvimento do PC. Em específico, delimitamos o âmbito do Ensino Médio e como ferramenta tecnológica a plataforma on-line Programaê!.

Para contribuir com essa discussão, primeiramente abordamos alguns elementos característicos do PC, em seguida descrevemos a metodologia adotada em nossa experiência de ensino e, por fim, apresentamos os resultados alcançados, considerando como arcabouço de análise dos dados o marco conceitual de Brennan e Resnick (2012), que correlaciona os elementos de programação utilizados, as práticas adotadas pelos aprendizes e os pilares do PC.

Aporte Teórico

Pensamento Computacional: Alguns Fundamentos

O PC é um conceito que vem sendo atualizado constantemente na literatura especializada vigente. Wing (2006), principal autora sobre o tema, destaca que assim como outras capacidades que são desenvolvidas, o PC deveria também ser considerado como uma capacidade fundamental de análise a ser desenvolvida pelos estudantes, não sendo limitada aos profissionais de ciências da computação.

Bundy (2007) define o PC como uma habilidade para programação e uma metodologia para resolver problemas. Nunes (2011) complementa esta ideia, e assume que a metodologia de trabalho inerente ao PC está relacionada à resolução de problemas no âmbito das mais diversas áreas. Nesta direção, Vicari *et al.* (2018)

destacam que esses autores foram os primeiros a descrever o PC como uma metodologia para resolver problemas. Em uma exploração a este respeito, Stadler e Martins (2021) apresentam uma revisão bibliográfica sobre a temática com destaque para o meio educacional.

Shute *et al.* (2017), em uma investigação sobre o conceito de PC, concluíram que o termo ainda não está sedimentado, mas evoluindo conforme as pesquisas sobre o assunto avançam. Os autores estabeleceram uma definição e estrutura sobre o PC para apoiar o desenvolvimento pedagógico e avaliação do PC, particularmente para o Ensino Médio, que envolvem: conhecimento prévio e habilidades cognitivas gerais; habilidades de resolução de problemas; e medidas de estilo de aprendizagem.

Com o objetivo de resolver problemas de forma eficiente e criativa, o PC envolve os seguintes processos cognitivos (Wing, 2006):

Reformulação: reenquadrar um problema na forma de uma solução familiar.

Recursão: construir um sistema incrementalmente com base nas informações anteriores.

Decomposição: dividir o problema em unidades gerenciáveis.

Abstração: modelar os aspectos centrais de problemas ou sistemas complexos.

Testes sistemáticos: tomar ações intencionais para chegar nas soluções.

Esses processos cognitivos, também chamados de pilares do PC, são agrupados e denominados de formas distintas entre os principais trabalhos sobre o tema. No entanto, os mais frequentes são: a decomposição, o reconhecimento de padrões, a abstração e o algoritmo (Grover; Pea, 2013; Liukas, 2015; Code.org, 2022). Dessa forma, esses quatro pilares contribuem para atingir o objetivo de solucionar problemas, de maneira eficiente (Medeiros; Rabelo, 2019).

Decompor é dividir em partes, técnica denominada “Dividir para conquistar”.

O reconhecimento de padrões é o pilar que permite extrair características semelhantes de situações, tem como base a identificação de similaridades.

A abstração está relacionada com o pensamento algébrico, no que diz respeito a não dar um significado único e estanque para determinado problema, deixando suas variáveis assumirem diferentes valores.

Por fim, o algoritmo é quem integra todos os outros pilares, pois se refere ao conjunto de instruções e de regras que devem ser criadas e seguidas para a resolução de um problema (Medeiros; Rabelo, 2019).

No quadro 1 estão relacionados os quatro pilares do PC e suas características, aplicações e questões.

Quadro 1: Os Quatro Pilares do PC

	Características	Aplicações	Questões
Decomposição	<ul style="list-style-type: none"> • Quebrar problemas em partes menores • Analisar aquilo que compõe o todo de forma individualizada • Dividir tarefas para potencializar a construção final de um produto 	<ul style="list-style-type: none"> • Decompor as partes de uma bicicleta, planta, misturas de alimentos, partes de corpo, ou conjunto de dados 	<ul style="list-style-type: none"> • O problema pode ser dividido em partes menores?
Reconhecimento de padrões	<ul style="list-style-type: none"> • Pode ser entendida como generalização • Encontrar similaridades entre as partes decompostas • Realizar classificações de espécies e relevância de dados 	<ul style="list-style-type: none"> • Distinguir as similaridades em um conjunto de dados ou espécies • Padronizar receitas, fórmulas matemáticas ou elementos da natureza 	<ul style="list-style-type: none"> • Existe algum padrão entre as partes?
Abstração	<ul style="list-style-type: none"> • Filtrar das partes padronizadas: apenas aquilo que é relevante • Anular informações desnecessárias para a resolução de um problema • Facilitar o processo de solucionar uma questão levantada 	<ul style="list-style-type: none"> • Elaborar calendário, perguntas, mapas, GPS, encontrar erros em conjunto de dados • O que é mais importante para ser resolvido antes? 	<ul style="list-style-type: none"> • Há alguma informação desnecessária?
Algoritmo	<ul style="list-style-type: none"> • Tem diferentes níveis de complexidade • Usar métodos prontos para a solução de problemas • Desenvolver uma solução passo a passo, normalmente por meio de regra 	<ul style="list-style-type: none"> • Executar operações aritméticas e lógicas • Elaborar sequencias de ações 	<ul style="list-style-type: none"> • Existe um método ou regra já estabelecido que pode ser utilizado para resolver esse problema?

Fonte: Adaptado de Guimarães (2022, p. 20)

Vicari et al. (2018, 110-111) propõe:

Um conjunto de orientações para aplicar o PC em situações de ensino/aprendizagem do currículo escolar que sirva como um guia para que os professores busquem utilizar o PC nas suas disciplinas:

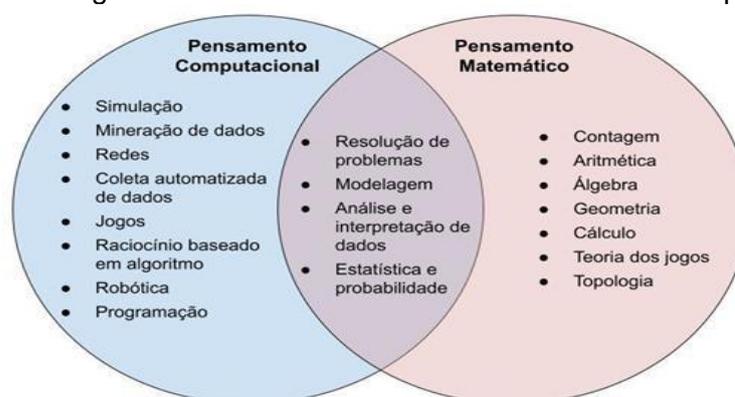
- Exploração de problemas, propondo situações em que os alunos desenvolvam algum tipo de estratégia para resolvê-las;
- Enunciado do problema, que exerce função importante, pois esse é fornecido ao aluno para que ele interprete e estruture a situação que lhe é apresentada para ser resolvida;

- Método de busca da solução, envolvendo a realização de aproximações sucessivas rumo a resolução do problema, utilizando o que o aluno já aprendeu para a resolução de outros problemas;
- Construção de um conceito para a definição de um problema pelo aluno, que também constrói conceitos articulados com outros que façam sentido num conjunto de problemas similares (que compartilham alguma etapa da sua solução).

Sobre o desenvolvimento das habilidades relacionadas ao PC juntamente com a disciplina de Matemática, podemos citar o trabalho de Dalla Vecchia (2012), que propõe a abordagem de modelagem matemática com o uso de simuladores.

Shute *et al.* (2017) relacionam as diferenças e semelhanças entre o PC e outros tipos de pensamento e comparam o PC com matemática, engenharia, *design* e pensamento sistêmico. A figura 1 mostra o conjunto comum de conceitos do pensamento matemático e computacional: resolução de problemas, modelagem, análise e interpretação de dados, estatística e probabilidade.

Figura 1: Congruências entre Pensamento Matemático e Computacional



Fonte: Shute *et al.* (2017, p.142)

Para Wing (2008), a principal semelhança entre o PC e o pensamento matemático são os processos de resolução de problemas.

Estudos realizados por Barcelos e Silveira (2012) apontam que, por meio da disciplina de Matemática, existe uma estratégia para inserção do PC na Educação Básica. Com um estudo envolvendo documentos norteadores para a elaboração dos currículos da Educação Básica brasileira, com foco no desenvolvimento do PC enquanto capacidade transversal a ser desenvolvida no estudante, esses autores analisaram a temática da tecnologia digital no ensino da Matemática. Com isso, identificaram três competências que podem ser trabalhadas: a articulação dos

símbolos e códigos; o estabelecimento de relações e identificação de regularidades; e a elaboração de modelos explicativos e representativos (Barcelos; Silveira, 2012).

Tratando-se das escolas públicas brasileiras, a inclusão do PC como conteúdo curricular parece ousada, porém um passo importante foi dado nessa direção com a inserção da terminologia Pensamento Computacional no principal documento norteador dos currículos escolares em todo o país, a Base Nacional Comum Curricular (BNCC) (Brasil, 2018). A BNCC é o documento que define o conjunto de aprendizagens essenciais que escolas públicas e privadas do país devem promover. Entre as dez competências gerais da Educação Básica citadas no documento, a quinta, sobre Cultura Digital, diz o seguinte:

(...) compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva (Brasil, 2018, p. 9).

Como já citado, o PC consiste em uma metodologia de resolução de problemas; portanto, faz-se necessário valer-se de uma metodologia de ensino que apoie esse processo. Com base nas experiências de ensino disponíveis na literatura envolvendo PC, bem como no formato das atividades das plataformas avaliadas, é possível identificar a gamificação como possível metodologia de ensino.

Gamificação e Programação em Blocos

A gamificação pressupõe a utilização de elementos tradicionalmente encontrados nos games, como narrativa, sistema de *feedback*, sistema de recompensas, conflito, cooperação, competição, objetivos e regras claras, níveis, tentativa e erro, diversão, interação, interatividade, entre outros, em outras atividades que não são diretamente associadas aos games, com a finalidade de tentar obter o mesmo grau de envolvimento e motivação que normalmente encontramos nos jogadores quando em interação com bons games (Fardo, 2013, p. 2).

Dentre as várias áreas de aplicação, a educação formal vem utilizando essa metodologia, que se constitui uma ferramenta adequada sob a perspectiva sociocultural, trazendo para o ensino um contexto habitual aos aprendizes.

Fardo (2013), avaliando relatos de aplicação da gamificação em diferentes contextos, sugere considerar, dentre outros, os seguintes aspectos para a utilização dessa metodologia:

Feedback: nos games os jogadores são sempre capazes de visualizar o efeito de suas ações em tempo real, enquanto os sistemas de avaliação nas escolas geralmente são bem mais demorados. Acelerar o processo de *feedback* estimula os aprendizes a buscar novos meios e estratégias para atingir os resultados esperados;

Dificuldade crescente: proporcionar diferentes níveis de dificuldade dos desafios propostos, conforme a habilidade dos aprendizes, permite que cada um siga o seu próprio ritmo de aprendizagem;

Decomposição: problemas são divididos em uma série de outros menores e mais fáceis de serem superados, possibilitando o aprendizado gradual e a compreensão da interrelação entre as partes do problema e o todo;

Depuração: nos games os jogadores estão acostumados a encontrar erros; portanto, aceitar o erro faz parte do processo de aprendizagem. Refletir sobre as razões do erro e buscar alternativas contribui para superar as eventuais falhas;

Narrativa: o contexto dos games justifica as ações e objetivos dos personagens; a aprendizagem contextualizada pode contribuir para motivar os aprendizes e obter melhores resultados.

Fardo (2013) destaca que a gamificação é uma ferramenta que disponibiliza elementos para criar experiências positivas para os aprendizes, seu uso contribui para motivar e potencializar as atividades educacionais.

Para Pontes (2013, p. 221-222):

A utilização de jogo como catalisador do processo de ensino pelo docente também propicia o desenvolvimento da autonomia e da confiança do aluno de forma espontânea, despertando habilidades como planejamento, seleção, organização, interpretação, decisão e socialização.

Brennan e Resnick (2012) consideram a programação em blocos com mídia amigável e interativa, um contexto poderoso para o desenvolvimento de habilidades do PC. Em uma investigação envolvendo a interação dos jovens com a plataforma digital Scratch, os autores identificaram que os jovens entrevistados adotaram uma variedade de estratégias e práticas para resolver as atividades propostas. Com base nos resultados encontrados, os autores ponderam que o PC envolve três dimensões:

Conceitos computacionais (elementos de programação) que os desenvolvedores empregam enquanto programam.

Práticas que desenvolvedores utilizam à medida que programam.

Perspectivas que os desenvolvedores formam sobre o mundo ao seu redor e sobre eles mesmos.

Segundo Brennan e Resnick (2012), são sete os conceitos de programação: sequências, *loops*, paralelismo, eventos, condicionais, operadores e dados.

Quanto às práticas, os autores observaram os seguintes conjuntos:

Incremental e iterativa: processo adaptativo, não linear, composto por ciclos iterativos. A solução vai sendo refinada; trata-se do processo em que o plano pode mudar em resposta à abordagem de uma solução em pequenas etapas.

Teste e depuração: busca da solução por tentativa e erro.

Reutilização e remixagem: a prática consiste em empregar artefatos já conhecidos (não necessariamente código pronto) para resolver novos problemas. Ajuda programadores a encontrar ideias e código, permitindo criar soluções potencialmente muito mais complexas do que poderiam ter criado por conta própria.

Abstração e modularização: construir algo grande, juntando coleções de partes menores. Todo o processo de projeto e resolução do problema, com o foco no enunciado.

As perspectivas tratam da capacidade de expressão, de conexão social e de análise crítica, analisadas nos estudantes.

Ao usar o Scratch os aprendizes seguem os desafios, da mesma forma que nos jogos; a prática de depuração fica evidente quando o aluno retorna a um estágio anterior para refazer uma atividade. Sendo assim, a gamificação torna-se uma característica própria da ferramenta.

No ensino tradicional de programação, o professor usa a forma escrita. No entanto, há diversas plataformas digitais disponíveis atualmente para auxiliar o ensino de programação, em que os alunos utilizam blocos visuais, que são arrastados e unidos na tela do computador. Por trás das ações com esses blocos, o aluno está criando códigos. Alguns exemplos dessas plataformas digitais são o Code.org e o Scratch.

A interface das atividades do Code.org é semelhante à interface do Scratch, em ambas é possível explorar e aprender programação através de blocos. O Scratch é um projeto do Lifelong Kindergarten Group do MIT Media Lab, que é disponibilizado

gratuitamente. A principal diferença entre essas plataformas é o direcionamento das atividades: enquanto o Code propõe trilhas de atividades bem definidas, com objetivos específicos, o Scratch disponibiliza recursos para projetos livres, conforme proposição do professor ou vontade do aprendiz. No contexto de plataformas com atividades em língua portuguesa, a plataforma Programaê! disponibiliza diferentes iniciativas de incentivo ao ensino de programação, incluindo atividades da plataforma Code.org.

Uma Proposta de Roteiro de Atividades

A plataforma Programaê! tem como propósito principal desenvolver estratégias para introduzir a linguagem de programação e o PC de forma mais intensa nas práticas pedagógicas, garantindo subsídios para que os professores e estudantes sejam protagonistas desse processo, bem como para auxiliar redes de ensino e escolas a desenvolverem um currículo que contemple o PC, em consonância com a Base Nacional Comum Curricular (BNCC) (Programaê!, 2018).

Na plataforma Programaê!, há duas formas para ensino, uma para discentes (Quero aprender) e outra para docentes (Quero ensinar). No módulo 'Quero ensinar', há exemplos de como elaborar atividades, códigos e desenvolvimento de aplicativos. No módulo 'Quero aprender', o aluno realiza as atividades em forma de trilhas. As trilhas são os caminhos a serem percorridos para aprender a programar, e assim desenvolver a lógica de programação (Programaê!, 2018). Além disso, a plataforma disponibiliza um guia com mais de duzentas páginas, com conteúdo sobre a importância de estudos como este e, em eixos, discute elementos significativos para a implantação da cultura digital e do PC nas escolas brasileiras.

As atividades disponibilizadas pela plataforma, fazem parte de um evento denominado "A Hora do Código" criado pela Code.org, e são ideais para uso de forma isolada, pois não necessitam de uma sequência preestabelecida. O movimento faz parte de uma ação global que busca mostrar que alguns conceitos da programação estão ao alcance de todos e que aprendê-los é mais fácil do que se imagina. Para começar, a ideia é que alunos e professores programem juntos, por ao menos uma hora, durante a semana do evento (Programaê!, 2018).

Valorizar e aplicar esse material em sala de aula vêm ao encontro de um ensino de programação que tenha significado para o estudante, para a sua

aprendizagem, ou seja, lança o olhar para estratégias de ensino e não apenas para um resultado.

Ao considerarmos que a plataforma Programaê! é ampla, disponibiliza muito conteúdo e um vasto conjunto de atividades sem um critério de níveis de dificuldade, ou trilhas como a plataforma Code.org, selecionamos dez atividades disponíveis, a fim de constituir um roteiro introdutório, a ser utilizado em sala de aula, apoiando o desenvolvimento dos vários pilares do PC.

Definimos a sequência das atividades conforme os conceitos computacionais envolvidos. Para a classificação das atividades do roteiro aqui proposto, consideramos os seguintes conceitos (também chamados elementos) de programação:

Programação sequencial: série de etapas ou instruções individuais que devem ser realizadas em sequência. Como uma receita, uma sequência de instruções de programação específica o comportamento ou ação que deve ser produzida (Brennan; Resnick, 2012).

Eventos: são gatilhos para que certas ações aconteçam para criar ambientes interativos como o uso de dispositivos (teclado, mouse, etc.) (Shute *et al.*, 2017).

Estruturas de repetição: também conhecidas como laços (ou *loops*), efetuam o processamento de um bloco de instruções quantas vezes forem necessárias. Uma estrutura de repetição pode ser usada de forma isolada (uma única estrutura) ou combinada, sendo várias estruturas em sequência (estruturas alinhadas) ou encaixadas (estruturas aninhadas) (Manzano; Oliveira, 2012).

Estruturas de seleção: esse tipo de instrução possibilita desviar o fluxo de execução, por meio da avaliação de uma condição. Também chamadas de estruturas de controle, podem ser usadas de forma isolada ou em conjunto, assim como as estruturas de repetição (Manzano; Oliveira, 2012).

Funções: são conjuntos de instruções identificados por um nome. É um bloco de instruções que pode ser reaproveitado e pode retornar um valor (numérico, lógico, literal, etc.) (Manzano; Oliveira, 2012).

Os elementos de programação são apresentados nessa ordem nas atividades da plataforma Code.org. Pode-se, então, considerar que eventos são elementos mais simples, com menor nível de dificuldade do que funções, que, quando são incluídas, estão nos desafios finais das atividades. O mesmo acontece com as estruturas de repetição e de seleção, em que aquelas aparecem antes que essas. Antes de introduzir esses elementos nas atividades, um vídeo explicativo (tutorial) anuncia o

que significa e como determinado elemento deve ser utilizado, com exemplos contextualizados.

Na tabela 1 está apresentado o roteiro de atividades. Cada atividade é formada por várias trilhas (ou desafios), variando de oito a vinte trilhas. Os números apresentados no quadro correspondem à quantidade de trilhas para cada conceito computacional (na respectiva coluna). Para cada trilha é apresentado um enunciado, com o objetivo que deve ser cumprido. No início da atividade, há vídeos tutoriais que orientam os aprendizes quanto ao uso da plataforma e aos conceitos de programação explorados. Em cada desafio, há dicas que contribuem para o cumprimento do objetivo proposto, pois os cenários possuem particularidades que podem aumentar a complexidade da tarefa (como esse é um aspecto subjetivo, não foi considerado nessa classificação). Três das atividades selecionadas contém um desafio final, que pode ser compartilhado por meio de mídias sociais. A complexidade dos desafios aumenta conforme a evolução da atividade.

Tabela 1: Roteiro de Atividades da Plataforma Programaê!

Atividade	Elemento de programação						Total de trilhas
	Sequência	Evento	Repetição	Seleção	Função	Outro(s)	
1 Infinity Play Lab	2	5	3	-	-	-	10
2 Flappy Bird	-	10	-	-	-	-	10
3 Design Minecraft	1	6	2	-	-	3	12
4 Basquete NBA	-	8	-	-	-	-	8
5 Aventureiro Minecraft	4	-	6	4	-	-	14
6 Moana	5	-	8	5	-	1	19
7 Angry Birds	5	-	8	7	-	-	20
8 O Artista	3	-	4	-	3	-	10
9 Jornada do Herói Minecraft	2	-	5	-	4	1	12
10 Frozen	3	-	10	-	7	-	20

Fonte: Os Autores (2024)

As primeiras atividades elencadas foram aquelas com o maior número de desafios envolvendo o elemento evento (por ser mais intuitivo e simples). As estruturas de repetição variam entre repetir até um limite dado, repetir até que aconteça algo ou repetir infinitamente, sendo que em alguns desafios essas estruturas devem ser aninhadas ou utilizadas em conjunto com estruturas de seleção. Alguns desafios, como o ilustrado na figura 2, exploram conhecimentos matemáticos quando exigem a definição de graus nos movimentos para criação de figuras geométricas; utilizam também o conceito de pixel e estruturas de repetição aninhadas. Quanto às funções, os desafios mais simples envolvem a utilização de funções prontas,

chegando a desafios mais complexos que requerem que as funções sejam definidas. Na experiência de ensino o estudante explora o uso de funções prontas, uso de funções com parâmetros, definição de funções e desafios que não necessariamente requerem, mas podem ser resolvidos com a definição de funções.

Figura 2: Atividade Frozen: Trilha 11

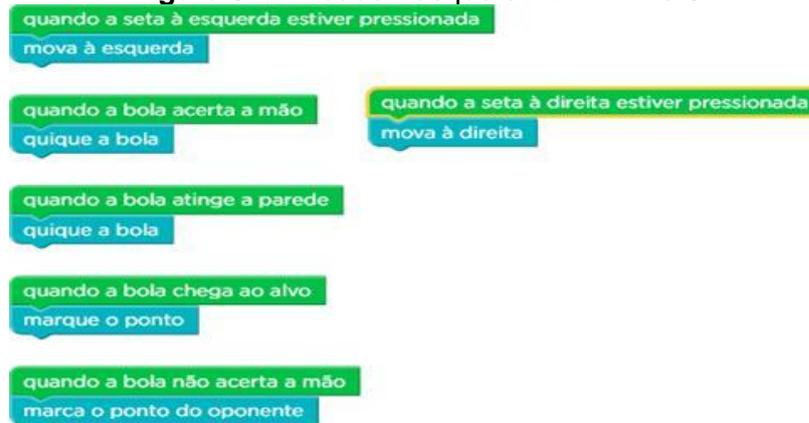


Fonte: Adaptado de Code.org (2022)

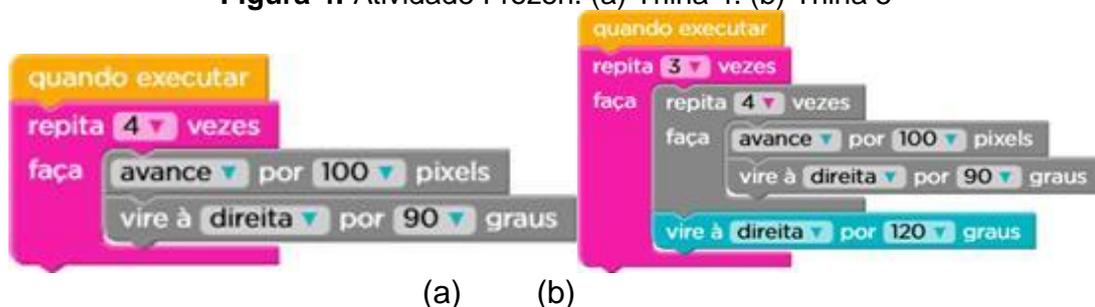
Quanto às contribuições do roteiro para o desenvolvimento das habilidades do PC, foi possível delinear algumas relações, tanto de forma mais geral (resolução de problemas a partir dos enunciados dos desafios) como mais específica (uso dos conceitos de programação):

Decomposição: ao separar os comandos de eventos em direções (esquerda, direita, cima, baixo), decompondo a interação. Um exemplo prático encontra-se na figura 2 sobre a trilha 6 da atividade Basquete NBA.

Reconhecimento de padrões: identificar blocos que se repetem e utilizar uma estrutura de repetição ou uma função, por exemplo, quando são agrupados blocos para desenhar figuras, como nas trilhas 4 e 5 da atividade Frozen (figura 4).

Figura 3: Atividade Basquete NBA: Trilha 6

Fonte: Adaptado de Code.org (2022)

Figura 4: Atividade Frozen. (a) Trilha 4. (b) Trilha 5

Fonte: Adaptado de Code.org (2022)

Abstração: esse pilar é transversal, aparece em todas as atividades. Entre as atividades propostas existe uma que se destaca por exigir com frequência o uso do mouse e estar atento aos movimentos do personagem na tela. Consiste em fazer com que o pássaro (*Flappy Bird*) desvie os obstáculos e cumpra os desafios propostos para concluir o jogo. Em vários momentos, em trilhas diferentes, exige-se fazer apenas o que se pede nas instruções, de forma isolada do contexto da atividade. Na trilha 5, por exemplo, o passarinho pode cair no chão, pode bater nos obstáculos, que nada acontece. Esse é um exemplo de atividade que apresenta características do pilar abstração, pois é necessário abstrair a informação mais relevante do enunciado e focar no resultado.

Algoritmo: utilizando o encaixe sequencial dos blocos, o algoritmo consiste em construir a solução usando os elementos de programação (blocos). As figuras que ilustram os pilares anteriores (figuras 2 e 3) são exemplos de algoritmos.

Para as atividades que constituem a Tabela 1 com o roteiro criado para aplicação da pesquisa, a habilidade de construção de algoritmos é a mais perceptível por definição, enquanto a abstração é a mais rara nesse tipo de situação, pois

precisaríamos de um número maior de atividades envolvendo casos semelhantes a fim de organizar e classificar. Cabe ressaltar que essas relações ocorrem em outras atividades das plataformas Programaê! e Code.org que seguem o mesmo padrão: resolver um desafio, conforme o enunciado, utilizando programação em blocos.

Contexto e Metodologia da Investigação

O roteiro proposto, conforme apresentado na tabela 1 (p.13), foi avaliado com base em uma investigação de caráter qualitativo e interpretativo, em que os dados obtidos, contém detalhes descritivos relacionados a indivíduos, lugares e diálogos, e referem-se a experimentos para explorar fenômenos em toda sua complexidade e em um ambiente natural (Bogdan; Biklen, 1994). Além disso, há a intencionalidade de explorar uma metodologia ou ferramenta e adotar procedimentos que contribuem com a aprendizagem dos alunos, o que é compatível com essa abordagem de investigação.

Com essa perspectiva, admite-se como características essenciais: ter um ambiente natural como fonte direta de dados; o pesquisador como ferramenta fundamental de construção e análise dos dados; a elaboração de uma descrição; a identificação do significado que as pessoas dão às coisas e a adoção de um enfoque indutivo na análise dos dados (Gomes; Gomes, 2019).

A proposta foi avaliada e aprovada nos Comitês de Ética em Pesquisa institucional, parecer Nº 4.470.975 e da Secretaria de Estado da Educação do Paraná, protocolo Nº 16.772.220-2, através de trâmites legais, e a realização foi autorizada pela Direção do Colégio.

O estudo foi realizado com alunos do curso Técnico em Informática, integrado ao nível médio, ofertado em um colégio público estadual da região central de 'Cidade', 'UF'. Embora seja uma formação específica, não há requisito para ingresso no curso, ou seja, os alunos não necessitam ter conhecimento prévio em informática ou programação de computadores, pois o Colégio não realiza meios formais de seleção de candidatos. Entretanto, é realizada uma entrevista com o objetivo de identificar interesses e afinidades compatíveis com o perfil de formação, e a situação socioeconômica do aluno.

A experiência de ensino foi desenvolvida com uma turma da primeira série do curso, envolveu inicialmente 13 inscritos, mas com o andamento da atividade, a

participação nos encontros variou entre cinco e três alunos com faixa etária entre 14 e 15 anos, durante o período letivo no ano de 2021, no contraturno, por meio de uma proposição de complemento aos conteúdos curriculares. A identidade dos alunos foi preservada, incluindo imagens e diálogos. No texto, os participantes estão identificados apenas por letras do alfabeto. A participação foi voluntária, sendo que os alunos estavam cientes da natureza do estudo.

As atividades, de acordo com o roteiro apresentado na tabela 1, foram desenvolvidas pelos alunos sob a orientação da docente da turma (autora deste texto), em ambiente virtual, devido às condições de distanciamento social impostas pela pandemia de Covid-19. Foram cinco encontros síncronos realizados via *Google Meet*, totalizando uma carga horária de 20 horas. A experiência assumiu um caráter prático, com a utilização do computador pessoal dos estudantes, em um ambiente de cooperação e colaboração entre os discentes e a docente.

A coleta de dados ocorreu com base nos princípios da observação participante e na composição de um diário de bordo, considerando-se como instrumentos de coleta de informações: vídeos com captura dos momentos através da gravação dos encontros e produção escrita dos estudantes. A partir dos dados coletados, seguimos as diretrizes da Análise de Conteúdo (Bardin, 2011), e consideramos como categorias os pilares do PC (com base no quadro 1) e os indicadores listados no quadro 2 (e relacionados na figura 5).

Na primeira coluna do quadro 2 constam os indicadores das ações desenvolvidas pelos alunos para a resolução de problemas, com base nos pilares do PC. Na segunda coluna são apresentadas as práticas relacionadas com cada pilar. As práticas podem ser avaliadas como um todo, em qualquer atividade do roteiro. Na terceira coluna estão relacionados os pilares do PC associados, também chamados de dimensões.

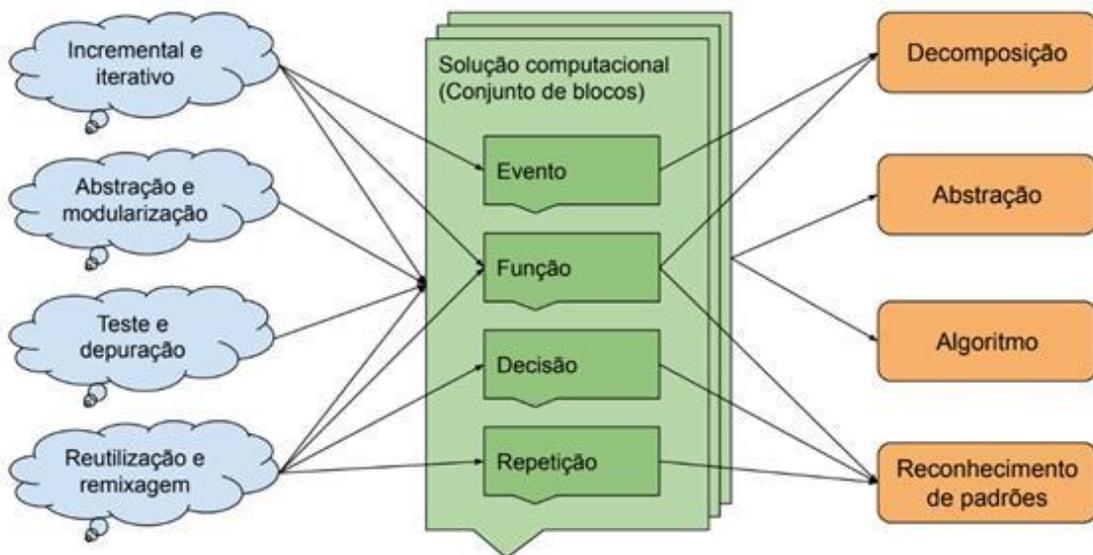
Quadro 2: Indicadores do Desenvolvimento do PC

Indicadores	Prática	Pilar
<ul style="list-style-type: none"> ● Formular uma estratégia para resolver um problema ● Adaptar o processo ● Mudar o plano em resposta à abordagem de uma solução em pequenas etapas 	Incremental e Iterativa	Decomposição
<ul style="list-style-type: none"> ● Conceituar o problema ● Usar exemplos genéricos ● Abstrair a informação mais relevante do enunciado 	Abstração e Modularização	Abstração

<ul style="list-style-type: none"> Focar no resultado 		
<ul style="list-style-type: none"> Reconhecer um padrão ou uma propriedade comum a um conjunto de objetos Usar contraexemplos 	Reutilização e Remixagem	Reconhecimento de Padrões
<ul style="list-style-type: none"> Buscar a solução por tentativa e erro Sequenciar instruções Organizar uma ação que deve ser produzida 	Teste e Depuração	Algoritmo

Fonte: Elaborada com base em Ponte et al. (2020) e Brennan e Resnick (2012)

Figura 5: Práticas, Elementos de Programação e Pilares do PC



Fonte: Os Autores (2024)

É possível perceber a adoção das práticas mediante o emprego dos elementos de programação (evento, função, estrutura de decisão, estrutura de repetição, conjunto de blocos). Neste sentido, os elementos mobilizados durante a elaboração das soluções permitem avaliar se o aluno contemplou um determinado pilar do PC. Por exemplo, ao adotar ações que levam à prática incremental e iterativa de uma solução, utilizando conjuntos de blocos, eventos ou funções, há correspondência com os pilares Decomposição e Algoritmos.

Resultados e Discussões

Não há um método simples que permita avaliar se os aprendizes desenvolveram habilidades de PC; no entanto, a análise das resoluções das atividades possibilita a identificação de características relacionadas a cada pilar do

PC. Nesta direção, conforme apontado por Brennan e Resnick (2012), a programação é um cenário valioso para o trabalho com as habilidades de PC.

As análises das interações que ocorreram durante a experiência de ensino vivenciada propiciaram a identificação de algumas práticas e ações (quadro 2), utilizadas pelos alunos para atingir os objetivos das atividades e, através delas, desenvolver algumas habilidades de PC. Essas interações aconteceram nas trilhas listadas na primeira coluna do quadro 3, selecionadas dentre as atividades que compõem o Roteiro (Tabela 1).

Quadro 3: Elementos e Práticas do PC nas Atividades do Roteiro

Atividade – Trilha	Elemento predominante	Prática
Infinity Play Lab - Trilha 10	Evento	Incremental e iterativa
Aventureiro Minecraft - Trilhas 3 e 7	Repetição	Modularização e abstração
Frozen - Trilha 10	Repetição/Função	Reutilização e remixagem
Jornada do Herói - Trilha 10	Repetição	Teste e depuração

Fonte: Os Autores (2024)

No quadro 3 estão apresentados os elementos de programação e as práticas identificados como predominantes em cada uma das atividades. As práticas com programação em blocos se concentram no processo de raciocínio e aprendizado, indo além do ‘do que você está aprendendo’ para o ‘como você está aprendendo’ (Brennan; Resnick, 2012).

Apresentamos, na sequência, a análise de exemplos representativos das práticas do PC, conforme as atividades elencadas no quadro 3.

Prática: Incremental e Iterativa com o Infinity Play Lab

A atividade Infinity Play Lab utiliza personagens da Disney. São 10 desafios ou trilhas envolvendo programação sequencial, eventos e estruturas de repetição (conforme apontado na Tabela 1). Ao longo dos desafios propostos, vão sendo apresentados os elementos de programação necessários para a resolução do último desafio (trilha 10). O enunciado da trilha 10 consiste na criação de um jogo de forma livre. O que aparece do lado esquerdo da tela, em verde (área de execução) é apenas o plano de fundo padrão, não possui personagens e nada se movimenta (Figura 6).

Figura 6: Infinity Play Lab: Trilha 10 - Solução do Aluno A



Fonte: Adaptado de Code.org (2022)

Conforme os blocos disponíveis, o aluno deve, minimamente, definir um personagem, um plano de fundo e as ações para quando a seta for acionada. O aluno também pode incluir blocos para que aconteça algum evento quando o personagem for acionado (clicado) ou quando ele encostar em outro personagem. Na Figura 6, temos a produção do aluno A. Na área de trabalho (lado direito da tela) está a solução desenvolvida por ele, constituída por 21 blocos com diferentes comandos.

Ao perceber que alguma ação era encaixada de forma inadequada e não funcionava como esperado, o aluno detectava o erro através da execução do código e concluía que para chegar ao resultado precisaria incluir ou trocar a ordem dos blocos. Também conseguimos observar, no lado direito da tela (Figura 6), vários conjuntos de blocos separados, em que o aluno precisou cuidar para que cada parte tivesse apenas um propósito, de modo que depois pudesse compor as partes de maneira eficiente para cumprir com o objetivo do desafio e, por consequência, da atividade.

A predominância do elemento de programação 'Eventos' foi constatada na produção final do aluno. Um evento pode desenvolver uma ação de maneira isolada, porém quando existe um conjunto de eventos, articulados em um mesmo cenário, ele contribui para a resolução de um problema, nesse caso, a criação de um jogo interativo.

Além disso, para atingir o objetivo e evitar falhas no algoritmo, o aluno tomou como base a observação e a forma de um processo adaptativo, e mudou de plano para que a solução acontecesse em pequenas etapas. Nesse sentido, o jogo

apresentado pelo aluno pode ser decomposto em partes menores, que envolvem a inclusão de blocos independentes para que possa atingir assim seu objetivo.

Nesse processo, o aluno A executou ciclos iterativos de imaginação e construção, desenvolvendo parcialmente e testando cada parte, com base em suas experiências e novas ideias. Foi um processo adaptativo, aquele em que o plano pode mudar em resposta à abordagem de uma solução em pequenas etapas. A modularização dos comportamentos dos personagens e objetos da atividade, por meio de eventos, facilitou a construção da solução, tornando o código organizado.

Consideramos, pois, que a prática de ser incremental e iterativa se torna perceptível na programação por meio dos eventos, tendo como base os indicadores de formulação de estratégia para resolução do problema, adaptação do processo e mudança no plano de resolução considerando uma abordagem em pequenas etapas (quadro 2).

O aluno A manifestou habilidade de PC quanto à decomposição, construindo uma possível solução para o problema, inserindo os blocos aos poucos (Figura 6).

Prática: Modularização e Abstração com o Aventureiro Minecraft

A atividade Aventureiro Minecraft é composta de 14 trilhas e tem como característica peculiar o cenário, que aparece em forma quadriculada como um plano cartesiano e indica a posição do personagem. O cenário segue o padrão do jogo e, assim como no jogo, o que torna a atividade interessante é a exploração, fazendo com que os personagens se aventurassem por grutas e ver o que conseguem encontrar.

O enunciado da trilha 3 pede que o personagem se aproxime, e tose as duas ovelhas presentes no cenário. Para isso devem ser inseridos blocos com os comandos 'avance', que significa o mesmo que 'ande'; o bloco 'tosar' e 'vire à direita'. O desafio é concluído quando o personagem se aproxima, e tosa as duas ovelhas. A atividade do Aventureiro Minecraft destacou-se por gerar um maior número de dúvidas e interação durante a experiência de ensino, o que tornou possível coletar os dados registrados através das falas dos alunos.

A fala do aluno B com relação a trilha 3 (Figura 7) foi a seguinte: "*Professora, esse eu não consegui fazer. Eu consegui fazer para chegar nas ovelhas, mas tem o limite de sete blocos, pode me mostrar como que faz?*"

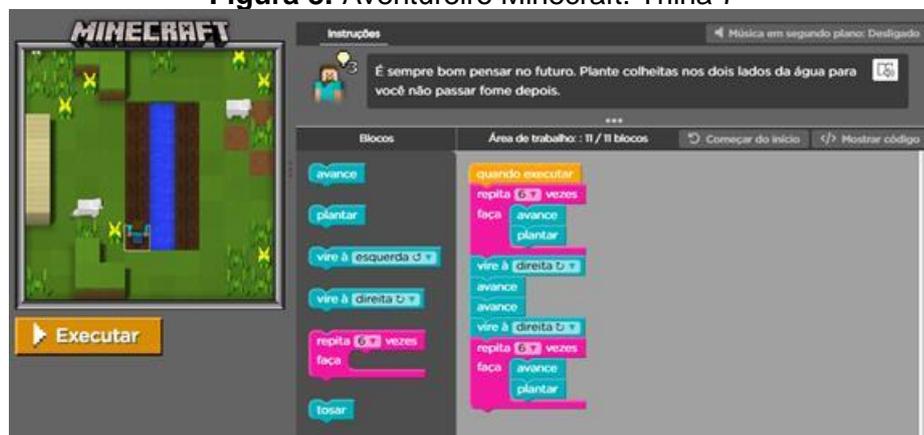
Figura 7: Aventureiro Minecraft: Trilha 3



Fonte: Adaptado de Code.org (2022)

Posteriormente, o aluno B questionou a resolução da trilha 7. Nessa trilha, o objetivo é plantar ‘colheitas nos dois lados da água’ (Figura 8).

Figura 8: Aventureiro Minecraft: Trilha 7



Fonte: Adaptado de Code.org (2022)

“Professora, você pode falar os comandos (...), o último quadradinho ficou faltando, ele plantar”. O aluno foi orientado que contasse os quadradinhos que formam o cenário. A resposta dele foi: “Eu consegui professora, mas usei 12 blocos”.

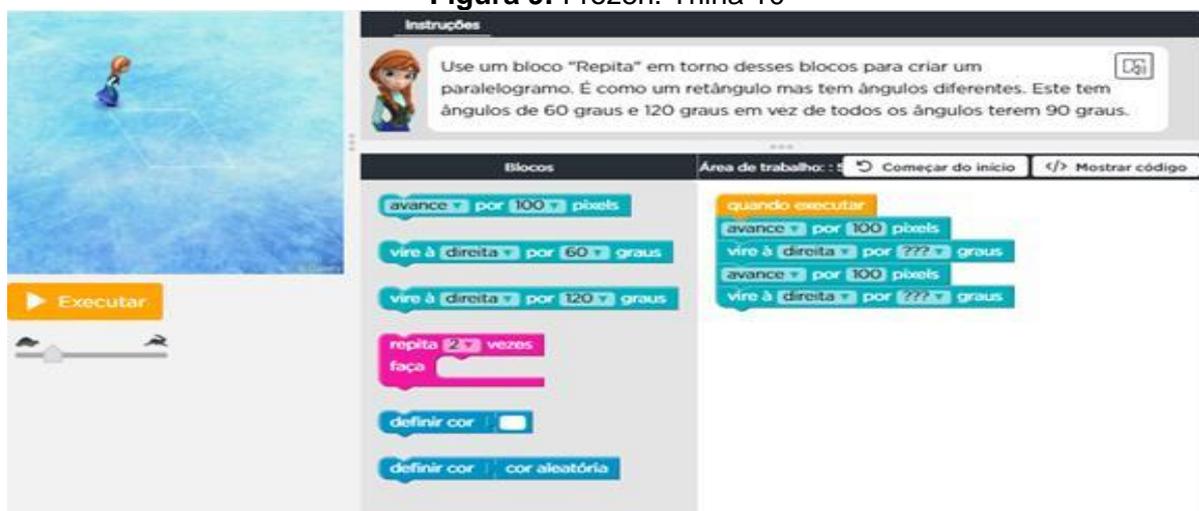
O aluno B conseguiu cumprir os desafios das trilhas, porém houve falha na construção do algoritmo por não ser otimizado. Notamos através da análise que o aluno usou a estratégia de resolução do problema com foco no enunciado; o aluno manteve a atenção em fatores relevantes ao objetivo proposto em detrimento a fatores secundários (quantidade de blocos da solução).

A falha, relacionada a utilizar mais blocos do que foi pedido no enunciado, foi resultante da não utilização de estruturas de repetição na solução. ‘Facilitar o processo’, como característica do pilar abstração apontado na Figura 1, usando uma estrutura de repetição, seria outro indício da habilidade de abstração motivada pelo objetivo da trilha.

Prática: Reutilização e Remixagem com a Frozen

O cenário da atividade da Frozen permite explorar a magia e a beleza do gelo com as personagens do filme da Disney, Anna e Elsa. Ao inserir os blocos de comandos é possível criar retas, formas geométricas e por consequência, criar figuras enquanto as personagens esquiavam no cenário (Figura 9). O enunciado da trilha 10 da atividade Frozen pede o seguinte: “Use um bloco”, “Repita” em torno desses blocos para criar um paralelogramo. É como um retângulo, mas tem ângulos diferentes. Esse tem ângulos de 60 graus e 120 graus em vez de todos os ângulos terem 90 graus”.

Figura 9: Frozen: Trilha 10



Fonte: Adaptado de Code.org (2022)

A informação no título da Área de trabalho sugere que são necessários oito blocos de código para cumprir o desafio, sendo que cinco já foram inseridos. Durante o encontro, ao chegar nessa trilha, o aluno D relatou como que ele fez: “nesse daí você coloca um bloco de ‘repita duas vezes’ e daí vire à direita 60 graus, e depois 120 no de baixo”. A Figura 10 mostra a solução da trilha de acordo com a sugestão do aluno, que moveu os blocos preexistentes para dentro de uma estrutura de repetição.

Ao ser questionado sobre como chegou ao resultado, o aluno D respondeu: “Foi como fazer o retângulo”.

Figura 10: Frozen: Trilha 10 - Solução do Aluno D



Fonte: Adaptado de Code.org (2022)

A prática de abstração e modularização foi identificada quando o aluno usou blocos de repetição, ou seja, reconheceu uma propriedade comum que aparecia na construção da solução, conforme avançava nas trilhas.

A utilização do elemento de programação repetição estimula o aluno no desenvolvimento do PC com relação ao pilar reconhecimento de padrões. As práticas de reutilização e remixagem têm sido usadas há tempos na programação de computadores, pois reusar um trecho de código que já foi testado, e está pronto, permite chegar ao objetivo mais rapidamente.

Portanto, em relação ao pilar reconhecimento de padrões, os alunos foram capazes de perceber que as regras para criar as formas, durante as trilhas da atividade da Frozen, tinham o mesmo padrão. Além disso, valeram-se da estratégia do uso de contraexemplos e das propriedades comuns aos desenhos geométricos, que se repetiam a cada fase do jogo, e que apenas se acrescentavam novas ações a cada fase, conforme os indicadores do quadro 2. Na medida em que os alunos se familiarizaram com a dinâmica das atividades, isto é, repetir a sobreposição de figuras quase sempre da mesma forma, foram capazes de realizar as atividades de maneira colaborativa, mais rápida e com maior desenvoltura.

Prática: Teste e Depuração com a Jornada do Herói

Na atividade Jornada do Herói, o cenário também é baseado no jogo Minecraft. O aprendiz precisa escrever os códigos, programando “O Agente” para limpar quaisquer obstáculos em seu caminho, para que possa coletar os itens que

precisará para a jornada. Apenas “O Agente” pode colocar e quebrar blocos, e apenas o personagem que o aluno comanda (Steve ou Alex, conforme sua escolha no início da atividade) pode coletar itens. “O Agente” pode andar, virar e ativar placas de pressão. Ele também pode destruir blocos e colocar blocos (Code.org, 2022).

Na trilha 10, o enunciado pergunta: “Você consegue ver a diferença entre as duas funções na área de trabalho?” e em seguida direciona a instrução: “Use essas funções com outro código da caixa de ferramentas para liberar o caminho até o Carrinho de Minas e, em seguida, colete a Pederneira”. A Figura 11 apresenta o cenário, onde pode-se observar o personagem e “O Agente”, os trilhos, os obstáculos e a Pederneira.

A tarefa deve ser iniciada com um comando que mude a direção do Agente para a esquerda. O aluno deve notar quantas repetições possui dentro de cada função, pois dentro dela existe um comando de ‘Avance’, mostrando quantos passos o Agente precisa andar para consertar o caminho todo. O número de passos deve ser contado através da imagem. As setas (Figura 11) servem para movimentar o personagem para que ele saia do carrinho e conclua a atividade.

Figura 11: Jornada do Herói: Trilha 10.



Fonte: Adaptado de Code.org (2022)

Na trilha 10, o aluno B relatou que não usou nenhum ‘repita’: “*Eu consegui fazer, mas eu não entendi como que faz*”. O aluno B resolveu o desafio por tentativa e erro, pois apesar de concluir o que se pedia, o personagem ficava ‘batendo na parede’, pois, havia uma orientação incorreta ou excessiva. O aluno não havia percebido a sequência das instruções, dentro das funções disponibilizadas; ao corrigir

a sua solução e testar mais vezes, conseguiu compreender o uso dos blocos disponibilizados.

Em termos de programação, utilizando a lógica aliada aos comandos adequados, o aprendiz pode modelar qualquer problema passível de ser resolvido através de algoritmos, que consiste em um processo de sequenciação de comandos válidos para executar determinada ação (Santos *et al.*, 2016). Assim, após elaborar um algoritmo, antes de executá-lo, deve-se simular a sua execução para adquirir confiança de que ele produzirá o resultado esperado. Esse processo de teste, conhecido como depuração, pode revelar erros e o algoritmo deve então ser corrigido. Em geral, para um mesmo problema existem diversos algoritmos que produzem resultados corretos.

O desenvolvimento dessa trilha consolida-se no encaixe das ações ou comandos na sequência correta para que “O Agente” destrua os obstáculos e libere a passagem para a coleta da ferramenta e conclusão com sucesso do desafio.

Percebemos que o conjunto de todas as práticas: incremental e iterativa, abstração e modularização, reutilização e remixagem, e teste e depuração estão envolvidas no desenvolvimento do pilar algoritmo. O conjunto dessas práticas contempla todos os elementos de programação: programação sequencial, eventos, estruturas de repetição e seleção, e funções. Além disso, observamos que o pilar algoritmo abrange os demais pilares do PC.

Posteriormente, depois da realização da experiência de ensino, os alunos foram questionados, se a realização das atividades, contribuíram no aprendizado de Linguagem de Programação e as respostas dos alunos A e B foram as seguintes:

Aluno A:

O curso que a professora deu ano passado foi com o intuito de usar a lógica como base para resolver os problemas propostos. Agora em 2022 que tenho aulas de programação, sei que saí na frente de meus colegas por ter feito o curso, pois me deu uma base estruturada. No curso eu aprendi lógica de programação.

Aluno B:

O curso com o Code.org me auxiliou na matéria de linguagem de programação no curso de informática. Sim, me ajudaram a ter noção básica de lógica de programação que influencia na linguagem de programação. Auxiliou na lógica e no raciocínio, não só na matéria,

mas como lógica básica inclusa no pensamento humano, fundamental esse conhecimento para as pessoas que querem ingressar na linguagem de programação.

Com base nestes relatos, consideramos que os participantes da experiência de ensino perceberam que a ferramenta estudada não estimula apenas o raciocínio lógico para um fim específico, mas para uma formação geral de aprendizado do indivíduo. Considerando Martinelli (2017), inferimos que: houve entendimento significativo dos conceitos de PC a partir dos elementos de programação apoiados pela plataforma Programaê!, de acordo com a sequência de atividades organizadas para alunos da 1º série do curso Técnico em Informática. Os alunos envolvidos apresentaram-se satisfeitos e motivados pelas atividades propostas na plataforma Programaê! Foi possível identificar que, ao longo das atividades, os alunos empregaram os elementos de programação e obtiveram os resultados esperados, concluindo os desafios propostos nas trilhas. A partir dos resultados obtidos, identificamos as práticas observadas por Brennan e Resnick (2012).

Considerações Finais

A partir dessas considerações, apontamos para a importância do papel do professor no processo de ensino e aprendizagem na utilização da abordagem de gamificação para desenvolver o PC em sala de aula.

O roteiro de atividades elaborado na pesquisa foi detalhado, gerando um produto educacional, em formato de material instrucional (disponível em link), caracterizado por um e-book, contendo atividades com foco no desenvolvimento do PC através de uma abordagem prática com programação em blocos. A plataforma utilizada segue os pressupostos da gamificação, proporcionando uma experiência de aprendizagem agradável e em contextos com os quais os alunos estão familiarizados.

Destacamos que as trilhas bem definidas da plataforma Programaê! favorecem o desenvolvimento das habilidades de PC, pois é necessário seguir cada uma das etapas para chegar ao objetivo da atividade, diferentemente de outras plataformas de programação em blocos, em que o objetivo é definido pelo professor ou pelo aprendiz.

Pode-se concluir ainda que as atividades se relacionam com o PC tanto na sua forma com a construção de jogos/programas (passo a passo para encaixe e

ordem dos blocos), quanto no seu método de resolução de problemas (decomposição, abstração, reconhecimento de padrões e algoritmo).

Identificamos que as práticas de abstração e modularização, ser incremental e iterativo e reutilização se aplicam a todas as atividades. Os elementos de programação estão diretamente relacionados com os pilares do PC: algoritmo (sequência dos comandos), decomposição (eventos) e reconhecimento de padrões (funções). O pilar Abstração foi identificado em todas as atividades, pois a abstração que dá base para o pensamento. A Abstração envolve capacidade de leitura. Se um aluno não é apto a ler e compreender o que lê, isso compromete a capacidade de abstração e, por conseguinte, de desenvolver o PC em contextos como esse.

Enfim, a programação de computadores está voltada à prática da resolução de problemas, sendo uma ciência que envolve um conjunto de princípios, técnicas e formalismos que visam a produção de *software* de qualidade. Para Brennan e Resnick (2012), a programação é um cenário valioso para o desenvolvimento de habilidades para o PC, o que pôde ser verificado na experiência de ensino vivenciada.

Agradecimento

Agradecemos o apoio financeiro recebido pela Fundação Araucária.

Referências

BARCELOS, T. S.; SILVEIRA, I. F. Pensamento computacional e educação matemática: Relações para o ensino de computação na educação básica. In: **XX Workshop sobre Educação em Computação**, 2012. p. 23.

BARDIN, L. **Análise de Conteúdo**. São Paulo: Edições 70, 2011.

BOGDAN, R. C.; BIKLEN, S. K. **Investigação qualitativa em educação: uma introdução à teoria e aos métodos**. Portugal: Porto Editora, 1994.

BRASIL, Ministério da Educação. Secretaria da Educação Básica. **Base Nacional Comum Curricular**. 2018. Disponível em: <http://basenacionalcomum.mec.gov.br/>. Acesso em: 10 mar. 2022.

BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. In: **Proceedings of the 2012 annual meeting of the American educational research association**. 2012. p. 25.

BUSARELLO, R. I.; ULBRICHT, V. R.; FADEL, L. M. A gamificação e a sistemática de jogo: conceitos sobre gamificação como recurso motivacional. In: FADEL, L. M. et al. (Org) **Gamificação na Educação**. [S.l.]: Pimenta Cultural, 2014. pp. 11-37.

CODE.ORG. 2022. Disponível em: <https://code.org/>. Acesso em: 10 mar. 2022

DALLA VECCHIA, R. A modelagem matemática e a realidade do mundo cibernético. 2012. 275f. Tese (Doutorado em Educação Matemática) – Universidade Estadual Paulista – Unesp, Rio Claro, 2012.

FARDO, M. L. A gamificação aplicada em ambientes de aprendizagem. **Renote**, v. 11, n. 1, p. 1-9, 2013. Disponível em: <https://seer.ufrgs.br/index.php/renote/article/view/41629>. Acesso em: 11 jul. 2023.

GOMES, A.; GOMES, C. R. A. Classificação dos tipos de pesquisa em Informática na Educação. In: JAQUES, P. A.; PIMENTEL, M.; SIQUEIRA, S.; BITTENCOURT, I. (Org.) **Metodologia de Pesquisa em Informática na Educação: concepção da pesquisa**. [S.l.]: SBC, 2019.

GROVER, S.; PEA, R. Computational Thinking in K-12: A Review of the State of the Field. **Educational Researcher**, v. 42, n. 1, p. 38-43, 2013. Disponível em: <https://people.cs.vt.edu/~kafura/CS6604/Papers/CT-K12-Review-State-Of-Field.pdf>. Acesso em: 11 jul. 2023.

GUIMARÃES, P. **Desenvolvimento do Pensamento Computacional no ensino da Matemática com estudantes autistas**. 2022. 94f. Dissertação (Mestrado em Ensino de Ciências Naturais e Matemática) – Universidade Estadual do Centro-Oeste do Paraná, Unicentro, Guarapuava, 2022.

LIUKAS, L. **Hello Ruby: adventures in coding**. [S.l.]: Feiwel & Friends, 2015.
MANZANO, J. A. N. G.; OLIVEIRA, J. F. **Algoritmos: lógica para desenvolvimento de programação de computadores**. [S.l.]: Érica, 2012.

MARTINELLI, S. R. **Nova Metodologia para a Aplicação e Avaliação de Práticas de Pensamento Computacional no Ensino Fundamental I**. 2017. 159 p. Trabalho de Conclusão de Curso (Especialização em Informática Aplicada à Educação) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, 2017.

MEDEIROS, I.; RABELO, H. Relato de Experiência: o pensamento computacional por meio da aplicação de Atividades Desplugadas em uma turma de 5º Ano. Natal, RN. In: **VI Conedu**. 2019.

MENEZES, L.; CANAVARRO, A. P.; OLIVEIRA, H. Teacher practice in an inquirybased mathematics classroom. **International Journal for Mathematics in Education**, n. 4, p. 357-362, 2012. Disponível em: <https://core.ac.uk/download/62454782.pdf>. Acesso em: 11 jul. 2023.

NAVARRO, E. R.; SOUZA, M. C. **Qual o conceito de pensamento computacional para a Educação Matemática?** [S.l.]: Dialética, 2023.

NUNES, D. J. Ciência da Computação na Educação Básica. **Jornal da Ciência**, n. 9, 2011. Disponível em: <http://gestaouniversitaria.com.br/artigos/ciencia-da-computacao-na-educacao-basica--3>. Acesso em: 11 jul. 2023.

PONTE, J. P.; QUARESMA, M.; MATA-PEREIRA, J. Como desenvolver o raciocínio matemático na sala de aula? **Educação e Matemática**, n. 156, p. 7-11, 2020.

PONTES, H. P. Desenvolvimento de jogos no processo de aprendizado em algoritmos e programação de computadores. *Proceedings of the XII Simpósio Brasileiro de Games e Entretenimento Digital (SBGames)*. São Paulo, 2013.

PROGRAMAÊ! **Programaê!**: um guia para construção do pensamento computacional. São Paulo: Fundação Telefônica Vivo e Fundação Lemann, 2018. Disponível em: https://fundacaotelefonicavivo.org.br/wp-content/uploads/pdfs/Guia_Final_06_09_2018.pdf. Acesso em 11 jul. 2023.

SANTOS, E. R. *et al.* Estímulo ao Pensamento Computacional a partir da Computação Desplugada: uma proposta para a Educação Infantil. **Relatec - Revista Latinoamericana de Tecnologia Educativa**, v. 15, n. 3, p. 99-112, 2016.

SHUTE, V. J.; SUN, C.; ASBELL-CLARKE, J. Demystifying computational thinking. **Educational Research Review**, n. 22, p. 142-158, 2017. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1747938X17300350>. Acesso em: 19 maio 2020.

SILVA, L. C. L. **A Relação do Pensamento Computacional com o ensino de Matemática na Educação Básica**. 2019. 131 p. Dissertação (Mestrado Profissional em Matemática) - Universidade Estadual Paulista, Presidente Prudente, 2019.

STADLER, A. P. G.; MARTINS, M. A. Pensamento Computacional: conceitos e definições no âmbito educacional. **VI Congresso Nacional de Pesquisa e Ensino em Ciências**, 5 a 7 de novembro de 2021. Disponível em: https://www.editorarealize.com.br/editora/anais/conapesc/2021/TRABALHO_EV161_MD4_SA107_ID2080_14102021143739.pdf. Acesso em: 11 jul. 2023.

VALENTE, J. A. Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação dos alunos. **Revista E-curriculum**, v. 14, n. 3, p. 864-897, 2016.

VICARI, R. M.; MOREIRA, A. F.; MENEZES, P. F. B. **Pensamento Computacional: revisão bibliográfica**. 2. ed. Porto Alegre: EDUFRGS, 2018. Disponível em: <https://lume.ufrgs.br/handle/10183/197566>. Acesso em: 19 maio 2020.

WING, J. M. Computational thinking and thinking about computing. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 366, n. 1881, p. 3717-3725, 2008.

WING, J. M. Computational Thinking. **Communications of the ACM**, v. 49, n. 13, p. 33-35, 2006. Disponível em: <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>. Acesso em: 19 maio 2020.