revista PARANAENSE DE EDUCAÇÃO MATEMÁTICA

PROGRAMAÇÃO APLICADA AO ENSINO DE MATEMÁTICA SOB A PERSPECTIVA DA TEORIA DOS REGISTROS DE REPRESENTAÇÃO SEMIÓTICA: UMA EXPERIÊNCIA DIDÁTICA COM AS NOÇÕES DE VARIÁVEIS DEPENDENTE E INDEPENDENTE NA FUNÇÃO AFIM

DOI: https://doi.org/10.33871/22385800.2023.12.28.287-310

Mônica Romana de Oliveira Santos[1]
Lúcio Souza Fassarella[2]

Resumo: Apresentamos um recorte de uma pesquisa qualitativa com fim exploratório fundamentada na Teoria dos Registros de Representação Semiótica, que teve como objetivo geral investigar as possíveis contribuições proporcionadas pelo uso da programação no ensino e aprendizagem da função afim, tendo como ênfase as ideias de variáveis dependentes e independentes. A pesquisa inclui a aplicação e análise de uma experiência didática realizada em 2021 com onze estudantes do primeiro ano do Ensino Médio de uma escola da rede pública estadual do Município de Teixeira de Freitas, Bahia. Nessa experiência, aplicamos uma sequência didática para o ensino do conceito de função afim, em que os estudantes tiveram que resolver problemas utilizando a linguagem de programação *Scratch*. Da análise dos dados referentes a uma atividade dessa sequência, concluímos principalmente que as conversões envolvendo representações na língua natural e nos registros algébrico ou computacional contribuem para compreensão de aspectos relacionados ao conceito de função, tais como a noção de variável, a relação de dependência entre grandezas, o uso de letras ou comandos para representar possíveis valores numéricos e realizar generalizações.

Palavras-chave: Educação Matemática. Resolução de Problemas. Registro Computacional. *Scratch*.

PROGRAMMING APPLIED TO THE TEACHING OF MATHEMATICS FROM THE PERSPECTIVE OF THE THEORY OF THE REGISTERS OF SEMIOTIC REPRESENTATION: A DIDACTIC EXPERIENCE WITH THE NOTIONS OF DEPENDENT AND INDEPENDENT VARIABLES IN THE POLYNOMIAL FUNCTION OF THE FIRST GRADE

Abstract: We present an excerpt from a qualitative research with exploratory purposes based on the Theory of the Registers of Semiotic Representation, which had as its general objective to investigate the possible contributions provided by the use of programming in the teaching and learning of polynominal function of the first grade, with emphasis on the ideas of dependent and independent variables. The research includes the application and analysis of a teaching experience carried out in 2021 with eleven students in the first grade of High School at a public school in the city of Teixeira de Freitas, Bahia. In this experiment, we applied a didactic sequence for teaching the concept of polynominal function of the first grade in which students had to solve problems using the *Scratch* programming language. From the analysis of data referring to an activity of this sequence, we mainly concluded that conversions involving representations in natural language to an algebraic or computational registers contribute to the understanding of aspects related to the concept of function, such as the notion of variable, the dependency relationship between magnitudes, the use of letters or commands to represent

[1] Mestre em Ensino na Educação Básica pela Universidade Federal do Espírito Santo (UFES). Professora do Colégio da Polícia Militar Anísio Teixeira, Teixeira de Freitas – BA, Brasil. E-mail: monicaromanasantos@gmail.com - Orcid: https://orcid.org/0000-0002-3052-0731
[2] Doutor em Ciências Físicas pelo Centro Brasileiro de Pesquisas Físicas (CBPF). Professor do Programa de Pós-Graduação em Ensino na Educação Básica da Universidade Federal do Espírito Santo. São Mateus - ES, Brasil. E-mail: lucio.fassarella@ufes.br - Orcid: https://orcid.org/0000-0001-6339-8958

possible numerical values and to make generalizations.
Keywords: Mathematics Education. Problem Solving. Computational Register. *Scratch*.

Introdução

O uso da programação no ensino da Matemática justifica-se pelas diversas habilidades que podem ser desenvolvidas quando os estudantes programam, uma vez que o ato de programar geralmente mobiliza a imaginação e o raciocínio lógico, a organização de ideias e a elaboração, implementação e avaliação de estratégias para resolução de problemas (WING, 2016; FASSARELLA, 2020).

Já na década de 1980, Papert (2020, p.34, tradução nossa) observou que as crianças programando um computador "[...] embarcam em uma exploração sobre o seu próprio pensamento". Esse aspecto reflexivo proporcionado pelo ato de programar foi também destacado por Valente (1999) ao afirmar que o computador, quando usado como máquina para ser ensinada, possibilita que o estudante descreva o seu raciocínio, reflita sobre os resultados de suas ações e depure suas ideias.

O ensino e aprendizagem por meio da programação encontra ressonância na Base Nacional Comum Curricular (BNCC), a qual salienta as possibilidades didáticas das tecnologias de informação e comunicação, preconizando seu uso desde o Ensino Fundamental e orientando, entre outras coisas, que os estudantes sejam estimulados a desenvolver o pensamento computacional (BRASIL, 2018).

Entre as competências gerais que devem ser obtidas no percurso pela Educação Básica, a BNCC considera que os estudantes devem ser capazes de:

> [...] compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) 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).

Abordando especificamente o ensino da Matemática, o documento ressalta que é por meio das representações que se tem acesso aos objetos matemáticos e é por meio delas que se compreende os fatos, as ideias e os conceitos. Mais especificamente,

> [...] na Matemática, o uso dos registros de representação e das diferentes linguagens é, muitas vezes, necessário para a compreensão, resolução e comunicação de resultados de uma atividade. Por sua vez, o trânsito entre os diversos registros de representação pode favorecer que os estudantes tenham

> maior flexibilidade e fluidez na área e, ainda, promover o desenvolvimento do raciocínio (BRASIL, 2018, p. 519).

É possível perceber uma proximidade do que é proposto na BNCC com a Teoria dos Registros de Representação Semiótica (TRRS) proposta por Duval (2009, 2012, 2015), embora essa aproximação envolva algumas distinções importantes (SIMONETTI; MORETTI, 2021). Destacamos especialmente o que consta sobre a aprendizagem de Matemática e suas Tecnologias no Ensino Médio na Competência Específica 4:

> [...] compreender e utilizar, com flexibilidade e fluidez, diferentes registros de representação matemáticos (algébrico, geométrico, estatístico, computacional etc.), na busca de solução e comunicação de resultados de problemas, de modo a favorecer a construção e o desenvolvimento do raciocínio matemático (BRASIL, 2018, p. 523).

Diante do exposto e considerando que os algoritmos ou programas de computador constituem um registro de representação semiótica (FASSARELLA, 2020), buscamos com base na TRRS investigar as possíveis contribuições da programação para o ensino-aprendizagem da Matemática. Mais especificamente, desenvolvemos uma pesquisa qualitativa com fim exploratório e objetivo geral de investigar as possíveis contribuições proporcionadas pelo uso da programação no ensino e aprendizagem de função afim (função polinomial do primeiro grau).

Para atingirmos esse objetivo, realizamos uma experiência didática com estudantes da Educação Básica. Por se tratar de uma pesquisa envolvendo seres humanos, esclarecemos que a intervenção ocorreu somente após a proposta ter sido apreciada e aprovada pelo Comitê de Ética em Pesquisa de nossa instituição (parecer n° 4.941.135 do Comitê de Ética em Pesquisa do Centro Universitário Norte do Espírito Santo da Universidade Federal do Espírito Santo).

Na experiência didática, propusemos a resolução de problemas no ambiente de programação *Scratch*. Trata-se de uma linguagem de programação desenvolvida pelo *Massachusetts Institute of Technology* (MIT), disponível para uso e download gratuito no site oficial https://scratch.mit.edu/ (acesso em 13 jul. 2023). Possui um visual interativo e intuitivo, tornando-a adequada para uso de pessoas não familiarizadas com programação. Possui blocos de comandos que substituem a digitação de códigos e fornecem instruções que controlam objetos gráficos (chamados de atores) disponíveis para seleção no software. No *Scratch*, os programas são criados pelo encaixe e encadeamento de blocos, os quais estão divididos em dez categorias visualmente distinguidas por cores e formatos: blocos de movimento, de aparência, de som, de sensores, de variáveis, de eventos, de controle, de operadores, de caneta e outros recursos. Esclarecemos também que alguns blocos, como os de aparência e sensores, exigem informações de entrada que dizem ao computador o que deve ser feito. O número 10 no bloco “mova 10 passos” é um exemplo de informação que pode ser inserida, fazendo com que o ator se movimente na janela de visualização (palco). Os blocos de operadores aritméticos e os blocos de variáveis também permitem a entrada e armazenamento de informações que retornarão um dado de saída ao usuário conforme o programa elaborado, além de possuírem comandos que, quando combinados, possibilitam a programação de diversas funções matemáticas.

Esta pesquisa se junta aos estudos que focam o papel desempenhado pelas conversões na aprendizagem de funções, como a apresentada por Silva *et al.* (2022), em que são consideradas conversões entre diversos registros de representação da função afim, com a exceção do *registro computacional* (o qual será discutido adiante).

Neste recorte, damos ênfase à contribuição da programação para a apreensão dos conceitos de variáveis dependentes e independentes da função afim em atividades envolvendo conversões entre os registros de representação semiótica da língua natural, algébrico e computacional. Cabe esclarecer que o software *Scratch* também foi utilizado na pesquisa para gerar representações gráficas da função afim, mas verificamos que ele não se mostrou adequado para esta finalidade devido às dificuldades de ampliação ou redução da janela de visualização.

Na próxima seção e na seguinte, apresentamos a TRRS e o registro computacional, mostrando particularmente sua aplicabilidade à análise das contribuições da programação à aprendizagem da Matemática. Na sequência, discutimos o ensino de funções na Educação Básica, considerando aportes BNCC e da perspectiva semiocognitiva. Depois, apresentamos breve relato de nossa experiência didática e discutimos ilustrativamente um dos problemas propostos. Por último, fizemos algumas considerações finais.

A Teoria dos Registros de Representação Semiótica

A TRRS considera que não existe acesso direto ou instrumental aos objetos matemáticos (números, funções, relações geométricas etc.), ou seja, não podemos percebê-los diretamente ou por meio de instrumentos como ocorre com os objetos concretos, incluindo aqueles estudados nas diversas disciplinas científicas, como física, química e biologia. Consequentemente, as representações são indispensáveis para o pensamento matemático (DUVAL, 2009; 2012; 2015; 2017).

Duval (2012, p. 270, 2013, p.15) afirma que “[...] o funcionamento cognitivo do pensamento humano se revela inseparável da existência de uma diversidade de registros semióticos de representação" e que "[…] as dificuldades de compreensão na aprendizagem da Matemática não estão relacionadas aos conceitos, mas à variedade de representações semióticas utilizadas e o uso 'confuso' que fazem delas". Assim, a apreensão conceitual de um objeto matemático depende da *coordenação* de diversos registros de representação semiótica, ou seja, da capacidade de reconhecimento de um mesmo objeto matemático em registros distintos e "[…] se manifesta pela rapidez e a espontaneidade da atividade cognitiva de conversão" (DUVAL, 2012, p.282).

Conforme explicitamos anteriormente, um sistema semiótico é considerado um registro de representação semiótica quando admite três atividades cognitivas fundamentais: a *formação de uma representação identificável*, a qual "[...] implica seleção de relações e de dados no conteúdo a representar" (DUVAL, 2012, p. 271), seja para expressar uma representação mental ou para evocar um objeto real. Pode ser comparada à realização de uma tarefa de descrição, devendo respeitar as regras internas do sistema semiótico considerado a fim de assegurar as condições de identificação e reconhecimento do objeto matemático representado no respectivo registro. Exemplos de formação de uma representação identificável são: a enunciação de uma frase, a expressão algébrica de uma função, o desenho de uma figura geométrica.

Os *tratamentos* são as transformações das representações de um registro em representações diferentes, no mesmo registro em que ela foi formada. Exemplos de tratamentos são os cálculos aritméticos envolvidos numa expressão algébrica e planificação de uma figura tridimensional.

As *conversões* mobilizam dois registros de representação semiótica diferentes e consiste na transformação da representação de um objeto em dado registro (registro inicial ou de saída) em uma representação desse mesmo objeto num outro registro (registro final ou de chegada). Isso ocorre, por exemplo, quando informações constantes no enunciado de um problema são colocadas na forma de uma equação algébrica no processo de resolução: nesse caso, convertemos a representação de um (ou mais) objetos matemáticos na língua natural para uma representação algébrica desse(s) objetos.

A atividade cognitiva de conversão envolve análise e comparação das representações nos registros de partida e de chegada, subentendendo a identificação de unidades significantes elementares, ou seja, toda a unidade que se destaca do léxico de um registro, nas representações envolvidas. Duval (2009) sugere que as conversões devem ocorrer em ambos os sentidos para que haja melhor apreensão dos conceitos matemáticos. Essa atividade é simples e realizada quase imediatamente quando as representações de saída e chegada são congruentes, sendo a congruência caracterizada pelas seguintes condições (DUVAL, 2009):

1. *Possibilidade de uma correspondência semântica* entre os elementos significantes das representações inicial e final: cada elemento significante da representação no registro de partida corresponde a uma unidade significante da representação no registro de chegada.

2. *Univocidade semântica terminal*: cada elemento na representação de partida corresponde a um só elemento no registro de representação de chegada.

3. *Correspondência da ordem de organização das unidades significantes* que compõem a representação inicial com a ordem de organização das unidades significantes que compõem a representação final.

Em síntese, o grau de dificuldade em uma conversão depende da ocorrência do fenômeno de não congruência – situação em que também dizemos que o registro de chegada não transparece o registro de partida (DUVAL, 2009; 2012; 2017). Assim, “[...] ou a representação terminal transparece na representação de saída e a conversão está próxima de uma situação de simples codificação – diz-se então que há congruência –, ou ela não transparece absolutamente e se dirá que ocorre a não congruência” (DUVAL, 2017, l. 210).

Observamos que *representações semióticas auxiliares* podem ser mobilizadas como coadjuvantes em processos cognitivos típicos, como a conversão. Nesse caso, uma representação auxiliar atuaria como intermediária entre a representação inicial e a representação final, servindo para alguma finalidade específica, a saber: apresentar as informações dadas na representação principal, indicando uma interpretação explicativa para o problema; explicitar e organizar uma seleção de informações pertinentes em situações em que a representação principal contém muitas informações; ou substituir um objeto matemático. Conforme definem Moretti e Baerle (2022), essas possibilidades correspondem a representações auxiliares dos tipos Interpretação Explicativa, Organização–Seleção de Elementos Pertinentes e Material, respectivamente.

Finalmente, podemos dizer que a TRRS caracteriza a compreensão conceitual em Matemática em termos da habilidade para realizar conversões, cabendo citar novamente Duval (2012, p. 282): “A compreensão (integral) de um conteúdo conceitual repousa sobre a coordenação de ao menos dois registros de representação, e esta coordenação se manifesta pela rapidez e a espontaneidade da atividade cognitiva de conversão”.

Abordamos agora os registros de representação semiótica definidos pelos algoritmos ou programas de computador, escritos na forma de pseudocódigo ou codificados em alguma linguagem de programação específica.

O Registro Computacional

Inicialmente, definimos alguns termos importantes para a compreensão desta seção. *Algoritmos* designam sequência de ações ou procedimentos que visam cumprir objetivos pré-determinados. *Programas* são algoritmos escritos de modo a serem executados por computadores. *Pseudocódigos* são formas padronizadas de escrever algoritmos utilizando a linguagem natural e alguns símbolos extras, como a seta "←" usada para indicar atribuição. *Linguagens de programação* são linguagens formais desenvolvidas para escrever programas de modo a serem executados por computadores, sendo geralmente constituídas por um conjunto de símbolos e um conjunto de regras gramaticais.

Concordamos com Fassarella (2020) ao afirmar que os programas (algoritmos escritos na forma de pseudocódigos ou codificados numa linguagem de programação) formam um registro de representação semiótica. Para o autor, os programas cumprem as três características fundamentais dos registros de representação semiótica: 1) a formação de uma representação identificável é dada pela codificação de algoritmos (programação), o que requer conformidade com a lógica e as regras sintáticas rígidas da linguagem de programação que for empregada; 2) os tratamentos dos programas podem ser realizados mediante modificações diversas, visando preservar ou alterar suas finalidades, como as proporcionadas pela especialização, generalização e simplificação; 3) as conversões envolvendo programas podem trazê-los como representações iniciais, finais ou auxiliares, sendo mais comum que figurem como auxiliares.

Denominamos registro computacional o sistema semiótico definido por um pseudocódigo ou linguagem de programação, enquanto uma representação nesse registro denominamos programa ou representação computacional. Para exemplificar, apresentamos no Quadro 1, um algoritmo descrito como um pseudocódigo e em duas linguagens de programação (*Python* e *Scratch*), que recebe dois números inteiros de entrada, ou seja, inseridos pelo usuário, e devolve a soma destes números.

Quadro 1: Pseudocódigo e linguagens de programação

Pseudocódigo	Início Passo 1. Faça a ← entrada ("Digite um número inteiro") Passo 2. Faça b ← entrada ("Digite um número inteiro") Passo 3. Faça soma ← a + b Fim
Linguagem *Python*	1 | a = int (input ("Digite um número inteiro: "))

<table>
<tr><td></td><td>2 | b = int (input (“Digite um número inteiro: ”))
3 | soma = a + b
4 | print (soma) |</td></tr>
<tr><td>Linguagem Scratch</td><td>
</td></tr>
</table>

Fonte: Elaborado pelos autores.

Observando o Quadro 1, nota-se que, embora o problema esteja representado em diferentes linguagens, trata-se da representação do mesmo objeto matemático.

Conforme já explicitamos anteriormente, do ponto de vista cognitivo é a coordenação entre diversos registros de representação e a habilidade em realizar conversões que se apresentam como fundamental para a apreensão dos conceitos em Matemática. Aqui, nos interessa compreender as conversões envolvendo representações computacionais, especialmente quando relacionadas à resolução de problemas matemáticos.

A resolução de um problema matemático por intermédio da programação (elaboração e codificação de um algoritmo) envolve uma conversão da linguagem natural (enunciado do problema) para o pseudocódigo ou linguagem de programação. No contexto do ensino da Matemática, o resultado de um programa formulado para resolver um problema pode gerar um terceiro registro, como uma tabela ou um gráfico, o que também pode contribuir para a apreensão dos conceitos matemáticos envolvidos.

A elaboração de uma representação computacional que resolva um problema matemático mobiliza diversos registros de representação e o processo de construção do algoritmo pode ser assim descrito

> [...] partindo de um registro multifuncional, a linguagem natural, na qual possivelmente foi elaborado o enunciado, passando para um registro monofuncional, que pode ser: algébrico, numérico ou simbólico. Nessa fase, pode haver a utilização de um registro misto, pois em alguns momentos a linguagem natural é utilizada em conjunto com a linguagem algébrica, simbólica e com o pseudocódigo. Finalmente, o algoritmo é escrito, utilizando um registro também monofuncional, o qual chamamos de *registro computacional* (SETTI, 2009, p. 50, destaque no original).

A fim de discutir os fenômenos de congruência na conversão da representação algébrica para a representação computacional, vamos considerar a seguinte tarefa ilustrativa: *Elaborar um programa que expresse a área de um retângulo cujo comprimento é o dobro da largura*.

Para elaborar o algoritmo, podemos partir tanto da representação do retângulo na linguagem natural, quanto da algébrica ou ainda, da representação figural. Em qualquer caso, podemos fazer as seguintes designações: L para a largura e S para a área do retângulo. O Quadro 2 mostra as representações algébrica e computacional para a área do retângulo:

Quadro 2: Representações algébrica e computacional da área do retângulo da tarefa.

Representação algébrica	Representação computacional
$S = 2.L^2$	01. **Início** 02. Escreva: “Digite o comprimento do menor lado do retângulo” 03. Leia L 04. Crie S 05. S ← 2*L^2 06. Retorne S 07. **Fim**

Fonte: Elaborado pelos autores.

Analisando as representações do Quadro 2, percebemos que a representação algébrica é mais sintética do que a representação computacional, posto que esta envolve comandos para definir variáveis e determinar ações a serem realizadas (linhas 01, 02, 03, 04, 06 e 07). Excluindo esses comandos, notamos que existe uma correspondência direta entre os elementos significantes na representação algébrica e os elementos significantes da linha 05 do algoritmo, conforme explicitado no Quadro 3:

Quadro 3: Univocidade semântica terminal nas representações algébrica e computacional.

Representação algébrica (de partida)	S	=	2	.	L	^{2}
Representação computacional (de chegada)	S	←	2	*	L	^2

Fonte: Elaborado pelos autores.

Nessa correspondência, além das variáveis S e L serem idênticas em ambas as representações, os caracteres “=” e “.” da expressão algébrica “$S = 2.L^2$” têm as mesmas funções dos caracteres “←” e “*” do comando “S←2*L^2”: os dois primeiros caracteres (=, ←)[3]

[3] Destacamos que algumas linguagens de programação, como o Python, empregam o símbolo “=” para atribuição de valor a variáveis e o símbolo “==” (duplo igual) para denotar a comparação dos valores de duas variáveis. Mais precisamente, “==” é uma função a duas variáveis que verifica se elas possuem o mesmo valor, retornando ‘Verdadeiro’ ou ‘Falso’ conforme o caso.

denotam igualdade e atribuição, respectivamente, enquanto os segundos (., *), denotam a operação de multiplicação. Quanto à notação da potenciação (dada por um caractere sobrescrito na representação algébrica e indicada pelo acento circunflexo na representação computacional), essa conversão do registro algébrico para o registro computacional apresenta univocidade semântica terminal, visto que os elementos significantes na representação de partida (representação algébrica) possuem apenas um elemento significante correspondente no registro de representação de chegada.

Também observamos que a conversão preserva a ordem de arrumação das unidades significantes nas representações algébrica e computacional. Portanto, a atividade de conversão da representação algébrica para a representação computacional cumpre as três condições necessárias para que seja uma conversão congruente.

Em síntese, reconhecemos que a representação terminal deixa transparecer a representação de saída. Além disso, o sentido inverso da conversão também é congruente, pois, se partirmos da conversão do registro de representação computacional (Calcule S ← 2*L^2) para o registro de representação algébrico ($S = 2.L^2$), o objeto matemático em estudo possivelmente será reconhecido, considerando que são preservadas as características fundamentais: correspondência, univocidade semântica terminal e a mesma ordem de arrumação entre os elementos significantes correspondentes.

Se o leitor acha estranho que seja congruente a conversão da representação algébrica para a representação computacional apresentada (visto que são representações tão diferentes na forma), é oportuno comparar com o que acontece numa conversão da representação algébrica de uma função para sua representação gráfica: não se considera que os eixos coordenados sejam, *per se*, unidades significantes da representação gráfica; antes, os eixos são tratados como elementos estruturais do registro gráfico[4]. O mesmo podemos dizer dos comandos nas linhas 01, 02, 03, 04, 06 e 07 do algoritmo no Quadro 2: são elementos estruturais ou convencionais de qualquer algoritmo que, como no caso, precise receber e retornar informações.

Ressaltamos, entretanto, que as conversões para representações computacionais que não envolvam fórmulas serão geralmente não congruentes em razão da ausência da representação algébrica, podendo implicar ao não cumprimento das três condições necessárias para uma conversão congruente e, consequentemente, ao aumento da distância cognitiva entre os

---

[4] Observamos que os eixos cartesianos têm significado nas representações gráficas, mas esse significado é intrínseco do Registro Gráfico (no sentido de ser parte da própria definição desse registro), sendo particularmente independente dos objetos nele representados.

registros. Setti (2009), por exemplo, apresenta um exemplo disso com o algoritmo para o cálculo de Máximo Divisor Comum (MDC). De acordo com sua análise, a conversão da representação do MDC do registro na linguagem natural (algoritmo de Euclides) para a representação computacional não é congruente.

Agora, passamos a discutir a função do registro computacional em sua relação com a representação principal de objetos matemáticos no contexto de um problema matemático. Se o problema é enunciado na linguagem natural, com eventual subsídio de representações auxiliares, a representação computacional pode assumir duas funções: representação auxiliar ou representação final (também denominada representação de chegada). Barcelos e Silveira (2012) destacam o uso auxiliar da representação computacional:

> [...] representar um problema na forma algorítmica [computacional] pode se constituir como uma etapa intermediária entre a narração verbal e a linguagem algébrica, podendo promover uma transição mais "suave" para a compreensão da linguagem matemática (BARCELOS; SILVEIRA, 2012, n.p.).

Assim, a representação computacional corresponde à representações auxiliares dos tipos Interpretação Explicativa (quando apresenta, de outra forma, as informações contidas na representação principal), Organização–Seleção de Elementos Pertinentes (quando explicita e organiza uma seleção de informações pertinentes em situações em que a representação principal contém muitas informações) e Material (quando substitui o objeto matemático), conforme definem Moretti e Baerle (2022).

A representação computacional será uma representação final no contexto de um problema quando ela constituir um algoritmo-solução, definido como "[...] um algoritmo cuja execução gera a solução exata ou uma aproximação arbitrariamente precisa da solução exata [de um dado problema]" (FASSARELLA, 2021, p. 242). Os algoritmos-soluções tipicamente possuem laços recursivos, mas não necessariamente.

Conforme afirma Duval (2017), a possibilidade de mobilizar os diversos tipos de registro é o que caracteriza a originalidade da atividade Matemática e proporciona a apreensão dos conceitos. Entretanto, o registro computacional não apenas diversifica a atividade de conversão, mas também constitui uma alternativa estratégica para a resolução de problemas matemáticos que pode ser muito eficiente numa ampla classe de problemas.

Para terminar esta seção, cabe observar que a Matemática e a Computação são áreas que se entrelaçam em diversos aspectos. Como a Matemática nos permite utilizar um conjunto amplo de métodos e procedimentos para solucionar problemas, a Computação também fornece meios para sistematizarmos o processamento de informações mediante o emprego de objetos computacionais. Assim como os objetos matemáticos,

> [...] os objetos computacionais não são acessíveis diretamente. Eles são entes abstratos, que não podem ser tocados, mas não deixam de ser tão reais quanto os elementos físicos. São onipresentes e essenciais, mas somente podem ser acessados por meio de suas representações (S.B.C., [s.d], p. 3).

Considerando nosso foco no ensino e aprendizagem da Matemática, cabe também lembrar que as representações computacionais servem para evidenciar o raciocínio dos estudantes, permitindo a eles e aos professores perceberem e corrigirem erros pela reflexão e depuração dos programas (PAPERT, 2020; VALENTE, 1999).

Discutimos na próxima subseção aspectos específicos do ensino e aprendizagem de Álgebra à luz da TRRS, considerando que a ênfase da pesquisa está na conversão da representação algébrica da função afim para a representação computacional e nas relações de interdependência entre as variáveis.

O ensino de funções na Educação Básica

Na BNCC, o tópico funções está incluído na unidade temática Álgebra, que tem como finalidade desenvolver o pensamento algébrico, considerado fundamental para que se possa utilizar modelos matemáticos para compreensão, representação e análise de relações quantitativas entre grandezas, de situações e de estruturas matemáticas, fazendo uso de letras e outros símbolos. Para desenvolver esse tipo especial de pensamento, é necessário que os estudantes

> [...] identifiquem regularidades e padrões de sequências numéricas e não numéricas, estabeleçam leis matemáticas que expressem a relação de interdependência entre grandezas em diferentes contextos, bem como criar, interpretar e transitar entre as diversas representações gráficas e simbólicas, para resolver problemas por meio de equações e inequações, com compreensão dos procedimentos utilizados (BRASIL, 2018, p. 268).

Geralmente, essa temática é introduzida formalmente nos anos finais do Ensino Fundamental e torna-se motivo de dificuldade para grande parte dos estudantes. No cotidiano das salas de aula, essas dificuldades persistem nas demais etapas do ensino, podendo se estender até o final da Educação Básica. Como exemplos de dificuldades típicas, Brandt e Moretti (2018) destacam alguns erros que aparecem com frequência ao longo da escolaridade, como: $a + b = ab$ ou $(a + b)^2 = a^2 + b^2$. Equívocos como estes são decorrentes da ideia de que o pensamento algébrico é uma extensão do pensamento aritmético:

> Esta ideia justificaria o erro apresentado pelos alunos de que a + b = ab, pois dois algarismos de um numeral justapostos são somados conforme o valor posicional no campo aritmético: 32 = 3 x 10 + 2. De igual maneira, explica logicamente que, se 2 + 3 = 5, então a + b tem que ser igual a algum valor, então a + b = ab. Se a álgebra for introduzida como extensão do pensamento aritmético será difícil aceitar que a + b = a + b (BRANDT; MORETTI, 2018, p. 2).

Especificamente a respeito do conceito de função, a BNCC orienta para que a noção intuitiva seja explorada em todo o percurso do Ensino Fundamental – Anos Iniciais e nas séries iniciais do Ensino Fundamental – Anos Finais, bem como na resolução de problemas relacionados à grandezas diretamente proporcionais:

> A noção intuitiva de função pode ser explorada por meio da resolução de problemas envolvendo a variação proporcional direta entre duas grandezas (sem utilizar a regra de três), como: "Se com duas medidas de suco concentrado eu obtenho três litros de refresco, quantas medidas desse suco concentrado eu preciso para ter doze litros de refresco?" (BRASIL, p. 268, 2018)

Nas séries finais, o conhecimento algébrico, que foi inserido informalmente na etapa inicial, deve ser aprofundado. É a etapa em que serão introduzidas as equações e as expressões algébricas de maneira formal. De acordo com a BNCC,

> [...] Nessa fase, os alunos devem compreender os diferentes significados das variáveis numéricas em uma expressão, estabelecer uma generalização de uma propriedade, investigar a regularidade de uma sequência numérica, indicar um valor desconhecido em uma sentença algébrica e estabelecer a variação entre duas grandezas. É necessário, portanto, que os alunos estabeleçam conexões entre variável e função e entre incógnita e equação (BRASIL, 2018, p. 269).

No ensino do tópico função, entendemos que deve ser considerado o ponto de vista cognitivo discutido anteriormente, observando a atividade de designação nas relações algébricas, as operações de substituição semiótica e os aspectos relacionados aos diferentes registros de representação de uma função: língua natural, representação algébrica, representação gráfica e representação computacional.

Conforme exposto por Duval (2012), para que haja apreensão do conceito de função, como de qualquer outro conceito matemático, é necessário coordenar os diversos registros de representação semiótica desse objeto. A coordenação dos diversos registros de representação se manifesta principalmente na atividade cognitiva de conversão. No entanto, essa coordenação não ocorre de maneira natural, pois a grande dificuldade da maioria dos estudantes na aprendizagem de função está em reconhecê-la em suas diversas representações.

Dessa forma, ensino de função deve ser planejado privilegiando a resolução de problemas que possibilitam a conversão entre as representações, pois, é o reconhecimento pelos estudantes da variedade de registros de representação e a possibilidade de conversão a ser realizada entre elas que pode proporcionar a apreensão integral do conceito. Para Duval *et al*. (2015), o que é preciso reconhecer na compreensão das escritas algébricas são as variadas e heterogêneas operações de substituição, tais como, uma letra por um número desconhecido, vários números por uma letra e duas maneiras diferentes de designar a mesma quantidade. De modo geral, "[...] a álgebra permite *a generalização da operação semiótica de substituição*, não a de um sinal em um objeto real, mas a de um sinal em outro sinal e mais globalmente de uma expressão em outra expressão" (DUVAL *et al*., 2015, p. 54, destaque no original).

Em relação à designação dos objetos na conversão de um problema em uma equação, Duval *et al.* (2015) apontam para a tomada de consciência sobre três operações a serem efetuadas: (1) redesignar todos os objetos já designados no enunciado do problema, diretamente ou em relação a outros objetos; (2) designar ou redesignar objetos duas vezes; (3) colocar na forma de equação duas designações de um mesmo objeto, escrevendo uma equivalência referencial entre elas.

As designações envolvem a articulação significativa de números, letras e símbolos de operações aritméticas. Elas abrangem a maior parte das relações existentes no campo algébrico e podem ser de vários tipos: "[…] designação verbal, designação numérica, designação indireta (descritiva ou funcional), dupla designação, designação direta, dentre outras" (BRANDT; MORETTI, 2018, p. 15). Todavia, colocar um problema na forma de equação é uma atividade cognitivamente difícil não só porque depende da designação, mas também porque as variáveis cognitivas são modificadas quando transitamos entre diferentes problemas (DUVAL *et al.*, 2015).

Entendemos que o processo de ensino e aprendizagem de função afim pode ser explorado em todo o percurso escolar, inicialmente, de maneira intuitiva, como ponto de partida para o desenvolvimento do pensamento algébrico. Posteriormente, faz-se necessário a introdução das escritas simbólicas que, geralmente, acarreta em dificuldades de compreensão. Entender as dificuldades dos estudantes e propor um ensino a partir do ponto de vista semiocognitivo pode contribuir para que os estudantes adquiram as condições necessárias para apreensão dos conceitos algébricos, bem como do conceito de função afim.

A Experiência Didática

Nossa experiência didática ocorreu no âmbito de uma pesquisa qualitativa e exploratória realizada em 2021 (OLIVEIRA, 2022). Nessa pesquisa, procuramos responder à seguinte questão norteadora: *Quais as possíveis contribuições proporcionadas pelo emprego da programação no ensino-aprendizagem de função afim?* Por se tratar de uma pesquisa com fim exploratório, visamos estabelecer “[...] maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a constituir hipótese[s]” (GIL, 2002, p. 41).

A experiência envolveu um grupo de onze estudantes do 1º ano do Ensino Médio, de uma escola da rede pública estadual, localizada em Teixeira de Freitas, Bahia. Constituiu-se na aplicação de uma sequência didática sobre função afim com problemas para serem resolvidos com o software de programação *Scratch*. As aulas foram conduzidas pela autora deste artigo ao longo dos meses de outubro a dezembro de 2021. Das nove aulas destinadas para a vivência da sequência didática, sete foram remotas em consequência das restrições impostas pela pandemia de COVID-19, e duas presenciais na Unidade de Ensino. Como suporte para as aulas, elaboramos e fornecemos aos estudantes uma apostila contendo breves explicações dos tópicos abordados (noções básicas de programação no *Scratch* e função afim) bem como exemplos, exercícios e problemas. A plataforma usada para as aulas remotas foi *Google Meet*.

Os dados coletados incluem todos os algoritmos elaborados para resolver os problemas propostos, além de gravações de áudio e vídeo e registros no diário de bordo. Nossa análise focou o processo cognitivo mobilizado pelos estudantes na resolução dos problemas, com especial atenção nas conversões para o registro computacional.

Como recorte dessa pesquisa, apresentamos e discutimos a seguir um dos problemas propostos na Aula 3, baseado na Figura 1: “Invente uma "máquina" que triplica e adiciona 1 a um dado número de entrada. Escreva a lei dessa função e elabore, no *Scratch*, um algoritmo que ela poderia utilizar”.

Figura 1: Apostila: Questão 3.2 da Aula 3.

c) A "máquina" de dobrar

Observe a seguir a representação de uma "máquina" que recebe um número como entrada e devolve como saída o dobro desse número.

O algoritmo a seguir, apresentado usando pseudocódigo, foi construído para obter os números que saem da "máquina" a partir dos números que entram.

Início
Nomeie de x o valor de entrada
Crie n
Calcule n ← 2 · x
Saída: n
Fim

Fique atento

Pseudocódigo é uma linguagem simples e escrita sem utilizar uma linguagem de programação específica.

Fique atento

- Algoritmos são sequências de passos ou regras simples e ordenadas, elaboradas para obter soluções gerais de determinados problemas, sendo escritas de maneira clara e objetiva.
- Nesse algoritmo, x e n são as variáveis e a seta ← indica que uma variável do algoritmo vai receber um valor (um número explicitado no algoritmo, o valor de outra variável ou o resultado de um cálculo). Por exemplo, em n ← 2 · x, a variável n do algoritmo recebe o valor do cálculo 2 · x.

Observando o algoritmo que representa o funcionamento da "máquina" de dobrar, temos $n = 2 \cdot x$, que é a fórmula matemática da função.

Fonte: Dante e Viana (2020, p.15).

O objetivo da aula foi apresentar o conceito de função, identificar diferentes registros de representação desse objeto matemático e resolver problemas utilizando o software *Scratch*. A aula foi realizada remotamente e contou com a participação de dez estudantes. Nomeamos os participantes da pesquisa como Aluno 1, Aluno 2, ..., substituindo os seus nomes reais.

Ao longo da resolução do problema, observamos primeiramente que surgiram dúvidas quanto ao significado de "triplica (…) um dado número", com alguns deles achando que seria "elevar ao cubo". Para esclarecer a questão, explicamos as operações de triplicar e elevar à terceira potência, destacando suas diferenças. Entendemos que esta dificuldade não diz respeito à programação, sendo uma dúvida decorrente do efeito do ensino dos conceitos de triplo e cubo de um número.

Depois dos esclarecimentos iniciais, não demorou para o Aluno 1 sugerir que a expressão que responde à questão era n = 3.x + 1, e os demais concordaram com essa resposta. A partir de então, a dificuldade concentrou-se na programação do algoritmo no *Scratch*, começando pela escolha dos blocos que utilizariam para definir a entrada de dados e manipular as variáveis envolvidas. Presumimos que essa dificuldade surgiu tanto de suas incertezas em relação ao conceito de função quanto da inexperiência com o *software*.

Observando os estudantes, percebemos que eles buscaram por uma correspondência entre os termos da expressão algébrica e os comandos do *Scratch*. Ou seja, esperavam encontrar um bloco correspondente a cada elemento da expressão “n = 3.x + 1”: um para a letra n, um para a letra x etc. Depreendemos que eles pressupunham tacitamente que a conversão do registro algébrico para um algoritmo no *Scratch* se resumiria a substituir elementos da primeira representação por elementos da segunda, de um modo simples e direto.

Para auxiliar na resolução, sugerimos que começassem escrevendo um pseudocódigo, como exemplificado no enunciado do problema. Nesse caso, a conversão da expressão algébrica (n = 3.x + 1) para a representação computacional (n ← 3.x + 1) é congruente por motivos análogos aos que foram apresentados para a conversão nos Quadros 2 e 3, da seção anterior, ou seja, a conversão cumpre as três condições necessárias para que seja congruente: correspondência semântica entre os elementos significantes das representações inicial e final, univocidade semântica terminal e correspondência na ordem de organização das unidades significantes inicial e final. Não observamos dificuldades significativas. Transcrevemos o algoritmo elaborado por um participante no Quadro 3:

Quadro 3: Algoritmo elaborado pelo Aluno 5.

01. **Início**
02. Pergunte (valor x de entrada) e espere
93. Crie n
04. Calcule n ← 3.x + 1
05. Saída: n
06. **Fim**

Fonte: Arquivos da pesquisa.

Posteriormente, solicitamos ao participante da pesquisa que explicasse o significado do pseudocódigo criado por ele:

> Aluno 5: Professora, eu usei o pseudocódigo da questão como modelo, mas fiz pensando nos blocos do *Scratch* também. Vai funcionar assim: quando o programa inicia, ele pede para a pessoa informar um valor que vai ser o valor de “x”, daí o programa recebe esse valor de “x” e calcula o valor de “n” de acordo com o valor de “x” que foi informado anteriormente. A cada valor de “x” informado toda vez que o programa é iniciado, será exibido um valor para “n” no palco.
>
> **Fonte:** Recorte de diálogo ocorrido com o Aluno 5, durante a aula 3, no dia 25/10/2021.

Como indica a fala do Aluno 5, na medida em que criavam algoritmos para resolver problemas envolvendo funções, eles iam compreendendo o aspecto da dependência e independência de variáveis. No caso da função definida nesta situação, a variável "x" é independente e a função denotada por "n" assume o papel de variável dependente. Presumimos que a possibilidade de informar diversos valores no campo de entrada e obter instantaneamente um valor correspondente no campo de saída (palco) auxiliou na compreensão da relação entre essas variáveis.

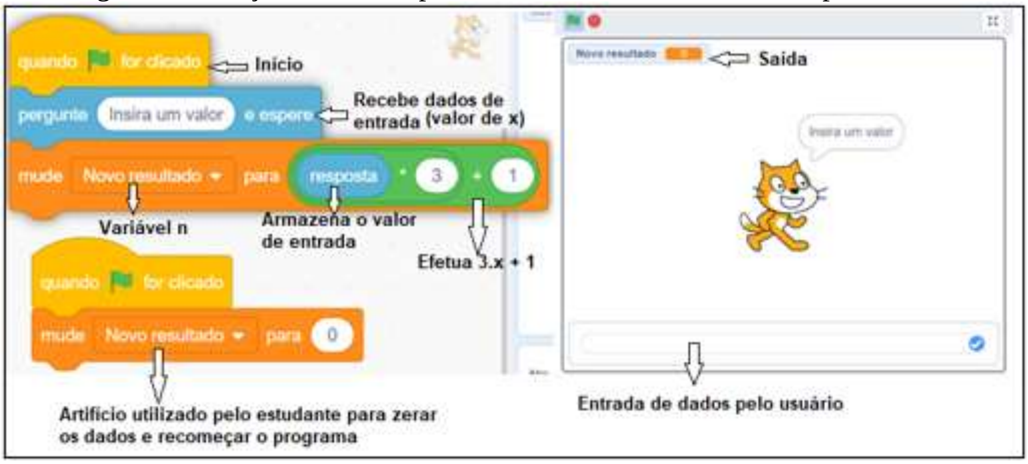
De certa forma, essas constatações corroboram à ideia fundamental do Construcionismo[5], de que a programação proporciona aos estudantes aliarem criatividade e aprendizagem (PAPERT, 2020; VALENTE, 1999) – ainda que, no nosso caso, a criatividade estivesse restrita à elaboração de algoritmos. Na aprendizagem sob o ponto de vista Construcionista, a criança aprende descobrindo por si mesma, com o intermédio do computador, sendo a programação uma ferramenta auxiliar no processo de manifestação das ideias e da reflexão, que possibilita observar com maior proximidade o desenvolvimento intelectual do estudante, visto que o programa reflete o seu raciocínio.

A partir do pseudocódigo, foi mais fácil realizar a tarefa de escrever no *Scratch* um algoritmo para uma "máquina" que "triplica e adiciona 1 a um dado número de entrada". Influenciados pelos exemplos dados, os participantes utilizaram os mesmos blocos nas programações, pois se basearam nas orientações e no pseudocódigo elaborado anteriormente. Após terminarem a tarefa, alguns estudantes se voluntariaram para apresentar seus *scripts*[6]. Na Figura 2 mostramos o compartilhamento de um dos *scripts*, indicando a função de cada bloco utilizado:

[5] Dado o papel central da programação nesta pesquisa, consideramos também os aportes do Construcionismo para nortear o emprego didático do computador em sala de aula.

[6] Em computação, *script* é uma sequência de instruções para serem executadas de maneira ordenada num algoritmo, sendo também uma denominação comum para os programas elaborados no *Scratch*.

Figura 2: Algoritmo-solução elaborado pelo Aluno 6, com comentários explicativos dos autores.



Fonte: Arquivos da pesquisa.

No *script* mostrado na Figura 2, o bloco “Novo resultado” pôde ser criado e nomeado pelo estudante no próprio software. O bloco “resposta” encontra-se na lista de blocos de sensores do *Scratch* e é responsável pelo armazenamento do dado de entrada descrito no pseudocódigo como “x”, enquanto “Novo resultado” corresponde à variável dependente “n”. Assim, cada valor de entrada informado pelo usuário é armazenado na variável “resposta” e, após o processamento do cálculo da função pelos blocos de operadores matemáticos, é exibido o valor de saída no palco (espaço onde aparecem os atores à direita) em “Novo resultado”. O *script* não possui comando para exibir o dado de saída porque, no caso do *Scratch*, o resultado é exibido automaticamente no palco.

Destacamos que o estudante utilizou um bloco de comandos para zerar o “Novo resultado” antes de ser inserido um novo valor de entrada, mas esse artifício não é necessário porque o *script* desconsidera quaisquer valores inseridos ou retornados em execuções anteriores. Solicitamos que ele explicasse o funcionamento do seu *script*:

> Aluno 6: Quando a bandeira for clicada, o ator vai pedir para inserir um valor. Esse valor que a gente coloca aqui na entrada é variável independente da função. Daí eu criei um bloco de variável e dei o nome de “Novo resultado” que é a variável dependente. Para cada valor que a gente coloca aqui [entrada] aparece um novo resultado aqui [saída] que corresponde ao triplo do valor informado aqui [entrada], mais um. Funciona, professora! Já testei para muitos números.
>
> **Fonte:** Recorte de diálogo ocorrido com o Aluno 6, durante a aula 3, no dia 25/10/2021.

Para testar os *scripts*, os participantes compararam os dados de saída com os resultados esperados para diversos valores de entrada, ou seja, verificando a compatibilidade do algoritmo com o que obtinham calculando de forma independente a expressão “n = 3.x + 1”. Conforme observado por Barcelos e Silveira (2012, n.p.) “[...] a linguagem narrativa do algoritmo pode permitir que o estudante identifique e teste suas próprias hipóteses, construindo ativamente o seu próprio formalismo matemático a partir da construção do algoritmo”.

Lembramos que a TRRS estabelece que a oportunidade de coordenar diferentes registros de representação do objeto matemático (no caso, uma função afim), potencializa sua apreensão. Ora, para elaborar o algoritmo-solução do problema no *Scratch*, foi necessário que os estudantes percebessem os mesmos objetos representados em diferentes registros de representação semiótica. Exemplificamos isso no Quadro 4:

Quadro 4: Respostas do Aluno 1 sobre a análise da função.

Designação verbal (língua natural)	Redesignação literal (representação algébrica)	Redesignação computacional *(Scratch)*
“... triplica e adiciona 1 a um dado numérico...”	n = 3.x + 1	

Fonte: Arquivos da pesquisa.

Vale dizer que essa atividade exige apenas conversões dos registros da língua natural e algébrico para o registro computacional, embora a apreensão conceitual seja favorecida quando há conversões nos dois sentidos. Acontece que o modo como a resolução do problema foi proposto, ela orienta a elaboração de um algoritmo-solução que acaba figurando como representação final e não requer uma conversão do resultado para o registro inicial ou outro registro. De qualquer forma, isso não significa que a atividade seja inadequada, desde que seja complementada por outras atividades que requeiram conversão em ambos os sentidos.

Enfim, o processo reflexivo proporcionado pelo ato de programar conduziu os estudantes a entenderem que “resposta” corresponde a um conjunto de possibilidades que pode ser informado como dado de entrada no programa criado, o qual gera dados de saída correspondentes. Esse processo viabilizou a compreensão de que uma letra pode substituir não apenas um número, mas, sim, diversos números, preservando a condição própria do problema em questão. Consequentemente, depreendemos que a programação contribuiu para apreensão das noções de variáveis dependentes e independentes no conceito de função.

Aspectos semelhantes foram verificados na pesquisa de Ventorini e Fiorese (2018, p. 583), na qual concluíram que a linguagem de programação aliada ao conceito de função possibilitou “[…] que os alunos se apropriassem de algoritmos (programações) que contemplassem não só uma única situação, mas várias situações, contribuindo para o processo de generalização”.

Cabe destacar que os erros nas programações não foram encarados como fracasso, mas como parte do processo de resolver o problema em questão, servindo como estímulo para reflexão sobre o que havia sido feito.

Considerações Finais

Neste artigo apresentamos um recorte de uma pesquisa que teve por objetivo investigar as possíveis contribuições proporcionadas pelo emprego da programação ao ensino e aprendizagem de função afim. Além de nossa fundamentação teórica na TRRS, apresentamos nossa concepção a respeito do registro computacional e relatamos um recorte de uma experiência didática, realizada com estudantes do 1º ano do Ensino Médio de uma escola da rede pública do Estado da Bahia.

Como ilustração dos dados coletados e análises realizadas, apresentamos algumas de suas produções relacionadas a um problema proposto na sequência didática aplicada. Discutimos as operações cognitivas mobilizadas pela tarefa, dando especial atenção à redesignação de objetos e conversões da língua natural para o registro algébrico e deste para o registro computacional. Por exemplo, para redesignar a variável “x” que ocorria numa expressão algébrica, os estudantes utilizaram o bloco de comando “resposta” do *Scratch*, o qual armazena dados de entrada.

A partir do que observamos, constatamos que a substituição de letras (variáveis) e sinais (operações aritméticas) próprios da representação algébrica por correspondentes blocos de comando do *Scratch* associada à possibilidade de inserir quaisquer valores no campo de entrada, pelo que se obtêm instantaneamente o resultado do cálculo de uma expressão algébrica na tela do computador, auxiliou na compreensão dos conceitos de variável e função, bem como na tomada de consciência de aspectos relacionados às operações cognitivas de designação e substituição, as quais são necessárias para a compreensão do funcionamento das escritas algébricas (DUVAL *et al.*, 2015).

Além disso, observamos o desenvolvimento da habilidade de fazer generalizações, pois ao informar diversos dados de entrada e obter, imediatamente, um valor de saída correspondente, os estudantes notaram que as variáveis de uma função afim podem assumir diversos valores. Esta percepção contribuiu para a apreensão de propriedades e padrões envolvidos nas relações de interdependência entre as grandezas nos problemas propostos de uma maneira que, provavelmente, levaria mais tempo para perceberem apenas por meio de tratamentos no registro algébrico, utilizando o cálculo manual.

Pontuamos, por fim, que não tínhamos a intenção de tornar os alunos programadores experientes, nem tão pouco presumimos que a programação é capaz de resolver todos os problemas relacionados às questões de aprendizagem em Matemática. Mas, por meio desta pesquisa, percebemos que os estudantes se engajaram na resolução de problemas da forma como propusemos e que tiveram um aproveitamento satisfatório. Eles se envolveram num processo de reflexão, de investigação, de abstração e de aplicação do conceito de função afim elaborando e testando os seus algoritmos, o que, presumivelmente, resultou num ganho cognitivo tanto nos aspectos relacionados exclusivamente ao conhecimento matemático, quanto naqueles específicos da Ciência da Computação.

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 e XXX Congresso da Sociedade Brasileira de Computação (n.p.). **Anais** [...]. Curitiba, 2012. Disponível em: https://www.researchgate.net/publication/256439343/. Acesso em: 15 mar. 2023.

BRANDT, C. F.; MORETTI, M. T. Aprendizagem da Álgebra segundo Raymond Duval. **Revista Brasileira de Educação em Ciências e Educação Matemática**, v. 2(1), 1-26, 2018. Disponível em: https://e-revista.unioeste.br/index.php/rebecem/article/download/19419/12599/70932. Acesso em: 15 mar. 2023.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular.** Brasília, 2018. Disponível em : http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 15 mar. 2023.

DANTE, L. R.; VIANA, F. **Matemática em contextos**: função afim e função quadrática. 1ª ed. São Paulo: Ática, 2020.

DUVAL, R. **Semiósis e Pensamento Humano**: Registros semióticos e aprendizagem intelectuais. Tradução de Lênio Fernandes Levy e Marisa Rosâni Abreu da Silveira. São Paulo: Livraria da Física, 2009.

DUVAL, R. Registros de Representação Semiótica e Funcionamento Cognitivo do Pensamento. **Revemat**: R. Eletr. de Edu. Matem., v. 7(2), 266-297. Florianópolis, 2012. Disponível em: https://periodicos.ufsc.br/index.php/revemat/article/view/1981-1322.2012v7n2p266. Acesso em: 15 mar.2023.

DUVAL, R.; CAMPOS, T. M. M.; BARROS, L. G. X.; DIAS, M. A. **Ver e ensinar a matemática de outra forma**: introduzir a álgebra no ensino: qual o objetivo e como fazer isso? Organização: Tânia M. M. Campos. 1ed. São Paulo: PROEM, 2015.

DUVAL, R. Registros de Representações Semióticas e Funcionamento Cognitivo da Compreensão. In: MACHADO, S. D. A. (Org.). **Aprendizagem em Matemática:** Registros de Representação Semiótica. eBook Kindle, 2017. ASIN: B07LB7DQRJ.

FASSARELLA, L. S. Desconexão Procedimental e Programação no Ensino-Aprendizagem da Matemática: Considerações a partir da Teoria dos Registros de Representação Semiótica. **Revemat**: R. Eletr. de Edu. Matem., v. 15(2), p.1-24. Florianópolis, 2020. Disponível em: https://periodicos.ufsc.br/index.php/revemat/article/view/76610. Acesso em: 15 mar. 2023.

FASSARELLA, L. S. Estimando probabilidades por simulações computacionais. **Professor de Matemática On Line**. v. 9(2), p.240-251, 2021. Disponível em: http://pmo.sbm.org.br/wp-content/uploads/sites/16/2021/02/art17_vol9_PMO_SBM_2021.pdf. Acesso em: 15 mar. 2023.

FREITAS, J. L. M; REZENDE, V. Entrevista: Raymond Duval e a Teoria dos Registros de Representação Semiótica. **Revista Paranaense de Educação Matemática**, Campo Mourão, PR, Brasil, v.2(3), p.10-34, 2013. Disponível em: https://periodicos.unespar.edu.br/index.php/rpem/article/view/5946. Acesso em: 15 mar. 2023.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4ª ed. São Paulo: Atlas, 2002.

MORETTI, M. T; BAERLE; L. D. M. O Uso de Representações Auxiliares na Aprendizagem da Matemática: Um olhar semiocognitivo segundo Raymond Duval. **Educação Matemática Pesquisa.** v. 24, n. 1, pp. 582-610, 2022. Disponível em: 141 https://revistas.pucsp.br/index.php/emp/article/view/55409. Acesso em: 27 de Abril de 2022.

PAPERT, S. **Mindstorms**: Children, Computers, and Powerful Ideas. eBook Kindle, 2020. ISBN: 978-1-5416-7510-0.

OLIVEIRA, M.R.S. **Programação Aplicada ao Ensino de Função Afim**: Investigação baseada na Teoria dos Registros de Representação Semiótica. 2022. 237 f. Dissertação (Mestrado em Ensino na Educação Básica) - Universidade Federal do Espírito Santo, Centro Universitário Norte do Espírito Santo, São Mateus - ES, 2022.

SBC – Sociedade Brasileira de Computação. **Ensino de Computação na Educação Básica**. (s.d.). Disponível em: https://www.sbc.org.br/documentos-da-sbc/send/203-educacao-basica/1220-bncc-em-itinerario-informativo-computacao-2. Acessado em: 15 mar.2023.

SETTI, M. O. G. **O Processo de Discretização do Raciocínio Matemático na Tradução para o Raciocínio Computacional**: Um Estudo de Caso no Ensino/Aprendizagem de Algoritmos. Tese (Doutorado em Educação) – Universidade Federal do Paraná, Curitiba, 2009.

SILVA, A.S.; PACHÊCO, F.F.F.; SILVA, E.C.R.T da; MACHADO, G.B.B.; ALBUQUERQUE, M.S de. Introduzindo o estudo de função polinomial do primeiro grau: um olhar à luz da teoria dos registros de representação semiótica. **Revista Paranaense de Educação Matemática**, Campo Mourão, PR, Brasil, v.11, n.25, p.325-344, maio-ago. 2022. Disponível em: https://periodicos.unespar.edu.br/index.php/rpem/article/download/5148/4996/20285. Acesso em: 15 mar. 2023.

SIMONETTI, D.; MORETTI, M. T. Base Nacional Comum Curricular do Ensino Médio e Registros de Representação Semiótica: uma articulação possível? **Revista Internacional de Pesquisa em Educação Matemática**, v. 11 (1), p.99-117, 2021. Disponível em: http://funes.uniandes.edu.co/27643/1/Simonetti2021Base.pdf. Acesso em: 15 mar.2023.

VALENTE J. A. Informática na Educação no Brasil: Análise e Contextualização Histórica. *In*: VALENTE, J. A. (Org.), **O Computador na Sociedade do Conhecimento**. Campinas, SP: UNICAMP/NIED, 1999.

VENTORINI, A. E.; FIOREZE, L. A. Funções e Programação no Scratch**. Revista Novas Tecnologias na Educação.** v. 16, n. 2, p.576-585, 2018. Disponível em: https://seer.ufrgs.br/renote/article/view/89295. Acesso em: 13 de Set. 2020.

WING, J. Pensamento Computacional – um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar. **Revista Brasileira de Ensino de Ciência e Tecnologia**, v. 9, n. 2, 2016. Disponível em: https://periodicos.utfpr.edu.br/rbect/article/view/4711. Acesso em: 13 jul. 2023.

Recebido em: 15 de março de 2023
Aprovado em: 23 de junho de 2023