

Um encontro com o Pensamento Algébrico através da programação no Tinkercad

An encounter with Algebraic Thinking through programming in Tinkercad

Un encuentro con el Pensamiento Algebraico a través de la programación en Tinkercad

Rúbia Carla Pereira¹  

Alex Jordane²  

RESUMO

Este trabalho tem por objetivo analisar o desenvolvimento do pensamento algébrico durante a atividade de programação na plataforma Tinkercad. Mediante uma problemática real do cotidiano de cinco alunos de iniciação científica, do ensino médio, e de três professores, foi proposto a programação de uma simulação de semáforo com travessia de pedestre. A Teoria da Objetivação e a sua compreensão sobre o pensamento algébrico, forneceram o suporte epistemológico e teórico para a análise dos dados construídos e registrados por áudio e vídeo, por um período de nove aulas. Os dados foram analisados na perspectiva da caracterização do Labor Conjunto sobre pensamento algébrico. Durante o processo foi possível perceber o desenvolvimento de elementos cognitivos referentes à manipulação de grandezas indeterminadas, generalização e abstração, demonstrando a possibilidade de desenvolver o pensamento algébrico por meio da atividade programação.

Palavras-chave: Pensamento Algébrico; Programação; Teoria da Objetivação.

ABSTRACT

This work aims to analyze the development of algebraic thinking during the programming activity on the Tinkercad platform. Through a real problem of the daily life of five high school students and three teachers, the programming of a traffic light simulation with pedestrian crossing was proposed. The Theory of Objectification and its understanding of algebraic thinking provided the epistemological and theoretical support for the analysis of the data constructed and recorded by audio and video, for a period of nine classes. The data were analyzed from the perspective of the characterization of the Joint Work on algebraic thinking. During the process it was possible to perceive the development of cognitive elements referring to the manipulation of indeterminate quantities, generalization and abstraction, demonstrating the possibility of developing algebraic thinking through the programming activity.

Keywords: Algebraic Thinking; Programming; Theory of Objectification.

RESUMEN

Este trabajo tiene como objetivo analizar el desarrollo del pensamiento algebraico durante la programación en la plataforma Tinkercad. A partir de una problemática real de la vida cotidiana de cinco estudiantes de secundaria y tres docentes, se propuso programar un simulacro de semáforo con paso de peatones. La Teoría de la Objetivación y su comprensión del pensamiento algebraico proporcionaron el soporte epistemológico y teórico para el análisis de datos construidos y grabados por audio y video, durante un período de nueve clases. Los datos fueron analizados desde la perspectiva de la caracterización del Trabajo Conjunto sobre pensamiento algebraico. Durante el proceso se pudo percibir el desarrollo de elementos cognitivos relacionados con la manipulación de cantidades indeterminadas, la generalización y la abstracción, demostrando la posibilidad de desarrollar el pensamiento algebraico a través de la actividad de programación.

Palabras clave: Pensamiento Algebraico; Programación; Teoría de la Objetivación.

1 Mestre em Educação em Ciências e Matemática pelo Instituto Federal do Espírito Santo (Ifes). Professora do Instituto Federal do Espírito Santo, campus avançado de Viana, Espírito Santo, Brasil. Endereço para correspondência: Rodovia BR-101, Km 12, S/N, Universal, Viana – Espírito Santo, Brasil, CEP: 29134-400. E-mail: rubia.pereira@ifes.edu.br

2 Doutor em Educação pela UFES. Professor Titular e pesquisador do Programa de Pós-Graduação em Educação em Ciência e Matemática – Educimat – do Instituto Federal do Espírito Santo, Vitória, Espírito Santo, Brasil. Endereço para correspondência: Av. Vitória, 1729, Jucutuquara, Vitória – Espírito Santo, Brasil, CEP: 29040-780. E-mail: jordane@ifes.edu.br

INTRODUÇÃO

Consideramos que programação, entendida como processo de organização de uma sequência de comandos e decisões com a finalidade de resolver uma situação problematizada, usando uma linguagem simbólica (de computador ou desplugada), é uma ação incorporada às práticas sociais, culturais e históricas do mundo moderno.

Pela da programação, é possível pensar criticamente para resolver problemas, tomar decisões e fazer leituras da cultura, do trabalho e da sociedade. Della Fonte (2020) acena para a necessidade de construir uma educação tecnológica, articulando teoria e prática, feita a contrapelo da divisão do trabalho, consciente da execução do trabalho, e principalmente da sua concepção e complexidade. Dessa forma, entendemos que a educação se “amplia para o domínio teórico e prático das bases tecnológicas e gnosiológicas do processo produtivo, condição relevante para a sua condução consciente por parte das forças subjetivas” (Della Fonte, 2020, pos. 1456). A autora ainda completa que

Essa despossessão do trabalhador em relação ao saber de um trabalho concreto aprofunda a sua formação unilateral na medida em que ele torna-se um servo da máquina, desprovido dos conhecimentos científicos e tecnológicos que nela se materializam. Na condição de fragmento vivo de um mecanismo morto (máquina), o trabalhador vive uma alteração em seu processo formativo (Della Fonte, 2020, pos. 1360).

A tecnologia é parte da atividade humana, uma vez que com ela, agimos, pensamos, nos expressamos e criamos, ou seja, mudamos nossa realidade e nossas subjetividades. Sendo assim, defendemos uma educação matemática que inclui o uso das diversas tecnologias nos processos de ensino e aprendizagem, para formar sujeitos que pensem criticamente o seu papel no desenvolvimento do trabalho e da sociedade, em uma era tecnológica. Neste contexto, é necessário criar movimentos no ensino da matemática que possibilitem que estes sujeitos pensem, se comuniquem e reflitam suas práticas com a tecnologia, e o façam de forma crítica, ética e consciente. Acreditamos que a programação é um desses movimentos, uma atividade que medeia o encontro com saberes culturais, especificamente, com formas matemáticas.

Historicamente o desenvolvimento da computação, mais especificamente, a programação, está relacionado com habilidades próprias do pensamento matemático, como interpretação e organização de dados, uso de variáveis de entrada e saída e suas relações de dependência entre variáveis da programação, análise e síntese, generalização, abstração e escrita de execução de comandos lógicos (algoritmos). Assim, a computação é indissociável da matemática, uma vez que é constituída pela concatenação dos conhecimentos que envolve a resolução de problemas, o pensamento algébrico e algorítmico.

Isso exposto, percebemos que a matemática e o desenvolvimento da computação têm uma base histórica comum e a ser explorada – o pensamento algébrico, a partir de situações problemas cuja solução ocorra na dimensão da programação. Neste contexto, a programação é uma das formas de encontro com o pensamento algébrico. Assim, o viés da programação para pensar algebricamente é relevante atualmente, dada a importância tecnológica na cultura, na sociedade e no mundo do trabalho. Além disso, a programação como forma de materialização do pensamento algébrico não é algo automático, mas processual.

Na nossa concepção, pensar algebricamente na programação exige conhecer uma estrutura semiótica própria das linguagens de programação, mecanismos de comandos e linguagens algorítmicas. Portanto, o pensamento algébrico na programação é um saber que precisa de um processo de ensino e aprendizagem da educação matemática, como encontro a um tipo específico de saber cultural, originado historicamente das formas algébricas de pensamento matemático.

Nesta perspectiva, este trabalho tem o objetivo de **analisar o desenvolvimento do pensamento algébrico na programação na plataforma Tinkercad³**, para a resolução de uma problemática real do cotidiano de cinco alunos de iniciação científica do ensino médio e de três professores. A solução de parte do problema consistiu na compreensão da programação de uma simulação de semáforo com travessia de pedestre.

É importante destacar que não defendemos a inclusão da programação nas escolas com o intuito de formar programadores de computadores. Nossa defesa vai ao encontro do uso da programação para desenvolver o pensamento crítico e de compreensão de linguagens necessárias no nosso contexto cultural e social.

FUNDAMENTAÇÃO TEÓRICA

O aporte epistemológico desta proposta se fundamenta, basicamente na Teoria da Objetivação, de Luis Radford. A escolha das bases teóricas se deu pela compreensão cultural da matemática e que vai ao encontro com a dimensão cultural que a programação e a computação assumem no mundo do trabalho, nas relações entre pessoas, nas formas de agir, pensar, falar e na organização social que se desenha o mundo moderno atualmente. Todas essas características estruturam a cultura digital.

A teoria da objetivação situa-se num projeto educativo diferente: vê o objetivo da educação matemática como um esforço político, social, histórico e cultural que visa a criação dialética de sujeitos reflexivos e éticos que se posicionam criticamente em discursos e práticas matemáticas históricas e culturais constituídas, e que. Ponderam novas possibilidades de ação e pensamento. (Radford, 2021b, p. 38).

A inovação que se propõe nesta pesquisa está na proposta de desenvolvimento deste pensamento, geralmente trabalhado com conteúdos matemáticos como, por exemplo, o estudo das funções ou das equações. Nesta proposta apresentamos a programação como atividade conjunta de professores e estudantes no processo de ensino e aprendizagem do pensamento algébrico.

O ensino e a aprendizagem na TO são processos indissociáveis que alcançam o conhecimento e a formação dos sujeitos, isto é, trabalham na dimensão do saber e do ser. Dessa forma, a aprendizagem é compreendida por um processo social no qual os estudantes se encontram com formas de ações e pensamentos construídos historicamente em determinada cultura e, ao se depararem com essas formas, se transformam.

A objetivação é um processo dialético, transformador e criativo entre sujeito e objeto que se influenciam mutuamente e que vai além do saber e atinge a transformação do ser.

³ <https://www.tinkercad.com/>

Dessa forma, pode-se compreender que as salas de aulas de matemática não só produzem e reproduzem conhecimento, mas também subjetividades, baseado na “ideia de que nós, humanos, somos sempre projetos de vida inacabados, sujeitos em processo permanente de formação. Os processos de subjetivação são processos de formação incessante do sujeito, de criação contínua de um sujeito histórico e cultural e único” (Radford, 2021b, p. 61). Ou seja, os processos de subjetivação se referem às transformações pelas quais os sujeitos atravessam no encontro com o objeto cultural, onde o foco não são os processos cognitivos de aprendizagem (objetivação) e sim como o sujeito foi afetado na dimensão social.

O processo da subjetivação, concebido na TO como a transformação do ser, acontece por meio da atividade humana. A atividade ao afetar o ser o transforma e esse movimento é contínuo. Nesse movimento, em um contexto de sala de aula, aluno e professores operam na forma de trabalho conjunto, por meio de cooperação mútua com solidariedade e ética comunitária: caracterizada pela responsabilidade, compromisso e cuidado com o outro, estrutura essencial da subjetivação (Radford, 2020, p. 35).

Em síntese, os processos de objetivação promovem o encontro com o saber, enquanto os de subjetivação, transforma o ser, ambos acontecem simultaneamente e indissociavelmente por meio da atividade (labor conjunto).

A compreensão do conceito de atividade, para Radford (2020), chamada de labor conjunto, é um dos conceitos fundamentais da TO. A aprendizagem de um novo conhecimento concretiza-se por meio da atividade em sala de aula, construída historicamente, onde, matéria, corpo, movimento trazem à tona a atividade, a qual é denominada na TO como labor conjunto.

Neste contexto, entendemos que a tecnologia e a compreensão dos processos de produção de conteúdos tecnológicos e digitais são processos de produção social e de conhecimento. O que torna importante o diálogo da escola com a atividade de programação, em um espaço dialético – a sala de aula, entendendo a programação como atividade do desenvolvimento do pensamento algébrico.

Para Radford (2010, p. 4):

o pensamento é uma atividade que, embora realizada por um “eu” e pelo “corpo do eu”, está se valendo onipresente do kit de padrões de construção de significados da cultura, bem como de conceitos historicamente constituídos de natureza ética, política, científica e estética. O pensamento está vinculado ao contexto e à cultura em que ocorre. É por isso que é mais correto dizer que o pensamento em geral, e o pensamento algébrico em particular, é uma práxis histórica cognitiva [...].

Na perspectiva do autor, o pensamento algébrico é caracterizado não somente na natureza da grandeza, mas no tipo de raciocínio que é feito com a grandeza. Mais especificamente, três condições caracterizam o pensamento algébrico (Radford, 2021a, p. 173–174): (i) a indeterminação de grandezas – as grandezas indeterminadas como incógnitas, variáveis e parâmetros usadas na resolução de problemas; (ii) a denotação das indeterminações – utilização de signos, gestos, linguagem natural ou/e a mistura de todos para simbolizar grandezas indeterminadas; e (iii) a analiticidade – manipulação das grandezas desconhecidas, isto é, o pensar algebricamente permite que, mesmo que não se conheça uma grandeza indeterminada, seja possível operar de forma dedutiva. Na nossa concepção, essas condições

estão na estrutura algébrica da programação, uma vez que são necessárias incorporação de variáveis de entrada e saída, e estas, por sua vez, recebem um tratamento analítico, no corpo do algoritmo, nas formas que Radford propõe. Quanto à denotação, consideramos que a programação tem uma semiótica própria de comandos, definição e tratamento de variáveis (indeterminados), e que essa estrutura é base do pensamento para a composição do algoritmo. Nesse caso, ocorre o ressignificado de sistema semiótico, dado por Vigotski, que conceitua os signos como ferramentas de reflexão que permitem aos indivíduos organizar pensamentos e comportamentos, planejar ações, comunicar e expressar, dando significados aos objetos do saber cultural.

Dessa forma, a programação tem, em si, as condições de caracterização do pensamento algébrico e a linguagem de programação é uma parte do conjunto de estruturas que o estudante usa para se comunicar, pensar e compreender ideias. Nessa compreensão, as formas e linguagens de programação se tornam parte de um Sistema Semiótico de Significação Cultural (SSSC).

O conceito teórico da TO que dá conta das relações ser-subjetividade é o de Sistema Semiótico de Significação Cultural. Tais sistemas são superestruturas simbólicas dinâmicas que incluem concepções culturais sobre o mundo e os indivíduos (Morey, 2020, p. 63).

Outro elemento algébrico a ser considerado é a generalização. Para Radford (2008) a generalização algébrica, está definida como um padrão algébrico capaz de perceber uma semelhança notada em algumas particularidades (digamos $p_1, p_2, p_3 \dots p_k$); estender ou generalizar essa semelhança para todos os termos subsequentes ($p_{k+1}, p_{k+2}, p_{k+3}, \dots$), e ser capaz de usar uniformização para fornecer uma expressão direta de qualquer termo de uma sequência. O autor explica que existem diferentes aspectos envolvidos nessa definição. Primeiro, uma semelhança local – C , é notada em alguns membros da sequência – S . Essa etapa exige que se perceba o que é característico. Em segundo lugar, a semelhança C é então generalizada para todos os termos de S . Enquanto a semelhança generalizada C era uma previsão geral, portanto, algo apenas possível – na última parte do processo de generalização, C é a garantia para deduzir expressões de elementos da sequência que permanecem além do campo perceptivo. A expressão geral dos termos da sequência requer a elaboração de uma regra baseada em grandezas indeterminadas (Radford, 2008).

Radford (2008) categoriza três tipos de generalização e as examina em termos dos meios semióticos de objetivação que os alunos usam em processos de generalização matemática. São os tipos, generalização factual, generalização contextual e generalização simbólica. O primeiro tipo, generalização factual, tem o raciocínio ligado ao concreto, caracteriza-se pela percepção, sentimentos e elementos espaciais e temporais do ambiente físico do aluno ou do problema, e é demonstrada por gestos, objetos concretos, linguagem natural e corporificada, e elementos linguísticos relacionados a uma situação/enunciado de momento. A generalização contextual é caracterizada pela introdução do sistema semiótico. Isto é, o estudante ainda utiliza a sensibilidade material, mas de forma mediada pelo uso de signos, introduzindo os primeiros elementos da linguagem simbólica. Já na generalização simbólica o raciocínio já não está mais incorporado aos elementos espaciais e temporais do

ambiente ou do problema/enunciado. Nessa categoria, o pensamento assume características de abstração e do uso de sistemas semióticos.

Na programação ocorre um processo equivalente – a generalização de padrões, que consiste em buscar similaridades em sentenças ou problemas anteriormente resolvidos por determinado processo e saber adequar a outras situações.

Na nossa concepção um ambiente de movimentação de saberes matemáticos, por meio da programação, ocorre em um contexto de problematização, isto é, um espaço cultural e coletivo que envolverá conhecimentos históricos, sociais e culturais que vão além dos conteúdos matemáticos. Majmutov (1983, p. 58) define o problema como categorias da lógica dialética – a engrenagem da formação do conhecimento. Em outras palavras, para o autor, o problema determina a atividade investigativa do homem para o desenvolvimento de um conhecimento. O autor propõe um método de organização para o ensino problematizador, apresentando uma influência externa ao aluno para uma atividade interna, provocando os processos de reflexão e de contradição que provoca o conhecimento do estudante. Majmutov (1983) faz uma análise dos métodos de resolução de problemas de caráter científico elaborando um sistema de métodos de ensino por problemas que assegure o desenvolvimento das capacidades criativas dos alunos. São eles (Majmutov, 1983, p. 81-82): observação, experimentação, comparação, criação e testagem de hipótese, analogia, construção de modelos, indução e dedução e, por fim, o método do passo do abstrato para o concreto.

Conscientes das teorias do ensino problematizador e da objetivação, esta pesquisa apresenta uma situação de ensino-aprendizagem por meio da problematização do cotidiano dos alunos que fazem diariamente a travessia de uma rodovia federal em frente à escola para acesso ao transporte público. Diante desta situação problema foi proposto o estudo do funcionamento de um semáforo com travessia de pedestre nesta rodovia, considerando o impacto no tráfego. Neste contexto foi elaborado uma simulação para resolução do problema na plataforma Tinkercad, que detalharemos a seguir.

PERCURSO METODOLÓGICO

Esta pesquisa começou em 2020 e os sujeitos são cinco estudantes de iniciação científica, do curso técnico em logística integrado com o ensino médio, e dois professores de matemática, todos do Instituto Federal do Espírito Santo, campus Viana (Ifes Viana) e campus Vitória (Ifes Vitória). A proposta é investigar o movimento do pensamento matemático na atividade de programação. É importante ressaltar que entendemos que atividade, segundo a TO, “é um esforço conjunto, por meio do qual os indivíduos produzem sua subsistência. Inclui noções de autoexpressão, desenvolvimento racional e prazer estético” (Radford, 2021b, p. 54).

Entre as várias situações problemas que envolveram os estudos deste coletivo, vamos analisar, especificamente neste trabalho, o problema do semáforo com travessia para pedestre, descrito a seguir.

Para chegar ou sair do campus Ifes – Viana, os estudantes atravessam uma rodovia federal, a BR-262, Km 12, no município de Viana. Esse trecho da rodovia é muito movimentado com transporte de cargas em caminhões e há grande fluxo de carros em geral. A queixa dos estudantes se tratava do tempo que esperavam para fazerem a travessia e dos riscos de atropelamento. Ao serem questionados sobre uma solução, sugeriram a instalação de um semáforo com faixa para pedestre. A Figura 1 representa a situação.

Figura 1 – Layout da Situação Problema



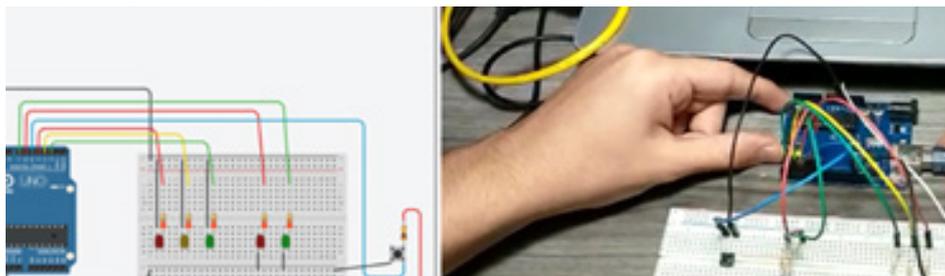
Fonte: GoogleMaps (<https://www.google.com/maps/@-20.3643062,-40.4438889,324m/data=!3m1!1e3!5m1!1e4?entry=ttu>) editado pelos autores

Os professores pontuaram sobre a importância de se pensar na temporização do semáforo, uma vez que o maior fluxo para travessia se concentra nos horários de entrada e saída dos estudantes e caso o temporizador do semáforo fosse contínuo poderia gerar engarrafamento no tráfego dos veículos. Então, os estudantes idealizaram o temporizador do semáforo acionado por botão de pedestre. Dessa forma, o luminoso ficaria vermelho para os veículos de acordo com a necessidade de travessia dos indivíduos.

Diante da problemática, os professores perceberam que havia um *motivo*⁴ para modelar uma situação usando a programação. A proposta foi feita aos estudantes para que a simulação do semáforo fosse programada na plataforma Tinkercad. Assim, foi estruturado o projeto didático, segundo a estrutura de atividade de ensino-aprendizagem, segundo Radford (2021b, p. 125).

O Tinkercad é uma plataforma de programação em blocos e em código (linguagem escrita) para simulação de projetos no Arduino⁵, que por sua vez, é uma plataforma programável de prototipagem eletrônica que permite a criação de sistema de automação. A escolha foi feita, pela possibilidade de o projeto do semáforo, estruturado no Tinkercad, se materializar no Arduino, como realmente aconteceu posteriormente (Figura 2).

Figura 2 – Montagem do Projeto Semáforo no Tinkercad (a esquerda) e no Arduino (a direita)



Fonte: Arquivos dos autores.

⁴ Relacionado ao conceito de necessidade segundo Leont'ev. Radford (2021b, p.123).

⁵ <https://www.arduino.cc/>

Assim, a montagem do projeto semáforo se deu em etapas. A primeira etapa consistiu no simples acionamento de um LED. A segunda na construção de um semáforo simples. A terceira no aprimoramento do semáforo, inserindo sinalização para pedestres. A última etapa propunha incluir um botão de acionamento para pedestres, atendendo aos questionamentos dos professores acerca da temporização do semáforo.

A execução do projeto aconteceu em um período de nove horas, sendo as primeiras no formato presencial e as últimas no formato não presencial pelo aplicativo de reuniões Meet, da Google, devido as medidas de isolamento da pandemia da Covid-19. Neste trabalho, analisaremos uma situação da última etapa: construção de um sistema de semáforo para veículos com acionamento por botão para travessia de pedestres. No Meet, um dos alunos compartilhava a tela do Tinkercad e todos interagiam, assistidos pelos professores, que intervinham quando necessário. Os dados da pesquisa foram registrados em áudio e vídeo e tornaram-se unidades de análise na perspectiva das categorias do labor conjunto no desenvolvimento do pensamento algébrico. Tais dados estão limitados aos diálogos e às programações, pois os dados da corporificação não foram possíveis de serem capturados. Nos diálogos apresentados na seção a seguir foram atribuídos nomes fictícios aos estudantes.

ANÁLISE DOS DADOS

Na última etapa do projeto, os estudantes avançaram na construção do sistema semafórico, incluindo um botão para travessia de pedestre. Esse botão foi necessário visto que na problematização, um semáforo com um temporizador contínuo poderia gerar atraso no fluxo dos veículos. Assim, foi pensado um modelo, já existente na vida real, travessia de pedestre por acionamento de um botão que avisa ao sistema semafórico dos veículos a informação de interrupção do fluxo para a travessia de pedestre.

Uma questão amplamente discutida e que gerou maior atenção dos alunos, foi uma situação bem específica de acionamento sequencial do botão. Isto consiste na situação de que um indivíduo que tenha acionado o botão e, o sistema de sinalização autorizou a travessia do pedestre, imediatamente após ser liberado a travessia dos carros, o botão seja pressionado por outro pedestre. Neste caso, os estudantes interpretaram que havia a necessidade de aguardar um período para que o luminoso vermelho dos veículos fosse novamente aceso, para não gerar um longo período de tempo com o sinal fechado para os veículos.

Os estudantes idealizaram a solução programando um cronômetro usando os blocos de programação, mas se depararam com dificuldades visto que a programação em blocos era limitada. Decidiram, então, migrar para a programação em códigos. O Tinkercad escreve a programação em bloco na forma de códigos automaticamente, no caso, a linguagem usada é a C++. Como os estudantes já tinham programado em bloco, foi fácil compreenderem a semiótica da linguagem de programação C++. Antes de partir diretamente para C++, os professores e estudantes pensaram e construíram uma programação em uma linguagem escrita da forma como se fala (LPF), que consiste em usar palavras-chave de comandos ("quando", "se", por exemplo), montando toda a estrutura da programação com as variáveis, condicionais e operações usando a língua materna. A Figura 3 mostra como foi construída a programação do semáforo em LPF.

Figura 3 – Programação do Semáforo em LPF

```

1  Pedestre
2      Carros = vermelho
3      Pedestre = verde
4      Esperar 5s

5  Normal
6      Carro = verde
7      Pedestre = vermelho

8  VAR Botao
9  VAR Tempo_Botao
10 VAR Tempo_inicio (começar com 0)

11 LOOP

12 Se pino_botao == Alto
13     Então Botao = Alto
14     Senão Botao = Baixo

15 Quando Botao == alto
16     Tempo_Botao = Tempo_inicio
17     Se o Tempo_Botao >= 10s
18         Então chamar pedestre
19         Senão
20             Normal

21 Quando Botao == baixo
22     Chamar Normal

```

Fonte: Acervo dos autores.

As linhas 1 a 4 e 5 a 7 são, respectivamente, os procedimentos⁶ Pedestre e Normal, que executam os funcionamentos da sinalização de travessia dos pedestres e dos veículos, respectivamente. As linhas de 8 a 10 são definições das variáveis para a programação do tempo de espera sobre o acionamento do botão. Especialmente na linha 10, a variável Tempo_inicio assume o valor da contagem de tempo da execução da simulação, então na compreensão do coletivo ela já tem um valor inicialmente agregado por um comando específico do Tinkercad. As linhas de 12 a 22 é a programação da situação problema que estamos tratando nesta pesquisa. O pino_botao faz referência à conexão do botão na placa do Arduino. Ao acionar o botão, o pino_botao é alto e não acionar é baixo.

O professor inicia o processo de testagem da LPF da Figura 3.

Professor – Então vamos simular aqui. Eu começo com o procedimento normal. (o professor vai passando o mouse para indicar qual linha está executando). O pino_botao é alto? (linha 15) Não. O pino do botão é baixo, então ele está executando o [procedimento] normal (linha 20) e o tempo está passando. Passou 10 segundos, 20 segundos, passou um minuto, você apertou o botão, então vou colocar aqui que o meu Tempo_inicio agora, no meu teste, é 60 segundos. (professor escreve 60s em vermelho ao lado da 10). Ele fica no Loop o tempo todo e depois de 60s alguém apertou o botão. Se o pino do botão for alto (linha 12), no caso é alto porque eu apertei o botão, então [a variável] botão é alto (linha 13). Quando [a variável] botão é alto o Tempo_Botao é o Tempo_inicio (linha 16), que é 60 segundos, no meu teste o Tempo_Botao é maior

⁶ Procedimentos são estruturas de programação completas que são executadas dentro de outros algoritmos. É um programa dentro e executando em outro programa.

ou igual a 10 segundos? (linha 17)

Alunos – Sim. (Acompanhando a testagem/simulação que o professor está fazendo).

Professor – Então ele chama o [procedimento] pedestre (linha 18), ou seja, o sinal de carro vai ficar vermelho, de pedestre vai ficar verde (fazendo referência sobre o que o procedimento Pedestre executa). Quando o botão por baixo, [procedimento] normal (linhas 21 e 22). No caso, meu botão está alto, aí ele vai voltar aqui para cima (linha 12), o pino_botao é alto? Você já soltou o botão, certo? Porque você não fica lá, segurando, você já soltou. Aí vai perguntar aqui: o botão é alto? (linha 15). Não, ele é baixo. Quando o botão é baixo, [procedimento] normal então ele vem para cá (aponta a linha 22).

Julia – Mas e quando eu apertar o botão e for menor que 10s?

Fabrcio – Isso, o que vai acontecer? Assim, a pessoa apertou o botão, e aí não é maior que 10s. Vai esperar dar 10s para o sinal de pedestre abrir?

Júlia e Fabrcio demonstram uma preocupação em relação ao tempo de 10s (linha 17). Estão em dúvida quanto o funcionamento do tempo de espera no caso do acionamento sequencial do botão.

Professor – Vamos ver aqui? Vamos fazer agora o seguinte, era 60 segundos quando a pessoa apertou o botão, isso aqui é o Tempo_inicio (sinalizando que a variável Tempo_inicio é 60s, linha 10). Agora vamos colocar que o botão foi apertado novamente em 69s (Escreve 69s ao lado da linha 10). Então ele está aqui no Loop, a pessoa aperta o botão, a variável botão é alto (passa o mouse na linha 13), quando a variável o botão é alto, meu Tempo_Botao **é igual o** Tempo_inicio (linha 16), ou seja, agora meu Tempo_Botao vale 69, é maior que 10s?

Júlia – Sim.

Professor – Então ele vai chamar o [procedimento] pedestre (linha 18. Segue um silencio de alguns segundos). E era para chamar?

Júlia – Não.

Professor – Então meu Tempo_Botao está com algum problema. Quanto vocês acham que deveria ser o Tempo_Botao aqui?

Igor – 70.

Wal – É, 70.

Igor e Wal respondem fazendo referência que só poderia executar a linha 18 no tempo igual a 70s, obedecendo o tempo de espera do acionamento do botão, que é de 10s. O professor os provoca em relação aos 69s e 70s, e Fabrcio começa a organizar as ideias.

Júlia – Eu pensei em 9. Tipo... depois de 9s que apertou.

Professor – Isso. E como a gente pode fazer para aparecer um nove aqui? (Linha 16. Com Tempo_Botao igual a 9s o procedimento pedestre não seria executado). Lembra que meu Tempo_inicio vale 69, mas, o Tempo_Botao que eu quero é 9. O que tenho que fazer?

Fabrcio – O tempo, no caso, é 69 e a gente quer que ele seja 9 segundos, o Tempo_Botao?

Professor – Isso. Esse tempo do botão é o tempo que passou de quando até quando? Quando que começou a contar entre os 9 segundos?

Wal e Fabrcio – Quando eu apertei o botão.

Júlia – Pela segunda vez.

Fabrcio – Então eu tenho que fazer alguma subtração aí [na linha 16], né? De um Tempo_inicio menos alguma coisa. Nesse caso seria menos 60, só que se for 49 segundos? Isso não valeria.

Igor – *Isso que eu pensei, se é sempre uma coisa certa para subtrair. Nesse caso subtraio 60, mas não é sempre isso.*

Neste momento do diálogo, Fabrício e Igor percebem a necessidade de uma variável para o cálculo do tempo de espera entre um acionamento e outro do botão. Aqui o processo de objetivação acontece quanto a identificação da natureza do elemento que é demandado na resolução do problema. Eles percebem que a natureza do elemento é indeterminada, uma vez que o tempo em que o botão é acionado é um elemento independente e de valor variável no processamento do programa. Além disso, Fabrício e Igor têm a compreensão da necessidade de se operar analiticamente esse termo indeterminado com as demais variáveis.

Os alunos estão em um encontro com as formas de pensar algebricamente, por meios semióticos de denotações e linguagens da programação e da resolução de problemas, conforme as condições apontadas por Radford (2021a). Esse encontro com o saber acontece em um ambiente de colaboração entre estudantes e professores. Além disso, o professor tem a preocupação de não apresentar a solução, mas de deixar que os estudantes concluam seus processos de pensamento usando o funcionamento do semáforo da vida real para explicar como descrever o programa usando signos e elementos da programação e do pensamento algébrico. Esse cuidado do professor com os alunos é um dos vetores da ética comunitária, dando indícios de processos de subjetivação, na dimensão professor-estudante, e que observa as formas de movimento do saber e da colaboração humana, isto é, o labor conjunto.

Os diálogos continuam:

Professor – *Não, não pode. Nesse caso aqui era 60, mas isso vai variar. (silêncio por alguns segundos). Aqui tem uma diferença que é 69 menos 60 igual 9 segundos. 69 é o Tempo_inicio e 60 é o quê?*

Fabrício – *60 é uma outra variável que é o valor do tempo de uma rodada anterior.*

Professor – *Esse 60 estava onde na rodada anterior?*

Igor – *No Tempo_Botao.*

Professor – *Isso. Vejam bem, se já tiver 2 horas que a programação começou, o Tempo_Botao vai ser 2 horas, isso sempre vai ser maior do que 10 segundos, porque ele está pegando o tempo desde o começo. Está faltando aí comparar o tempo desde o começo com o tempo que foi apertado o botão da última vez. Entenderam?*

Fabrício – *Então, no caso, seria o Tempo_inicio menos o tempo do botão que foi apertado na última vez, só que o tempo de botão apertado na última vez é a variável Tempo_Botao.*

Igor – *O 60 vai ter que ser uma variável.*

O professor percebe que os estudantes compreendem a necessidade e que estão pensando na presença do indeterminado, porém, ainda não completaram as formas de analiticidade com este indeterminado para modelar o problema. Até que Fabrício percebe que qual o valor temporal – “o tempo que foi apertado pela última vez”, mas que já é a variável Tempo_Botao. Nesse momento, Igor volta ao contexto da primeira simulação, indicando que está em uma forma de generalização contextual, ao falar “60 vai ter que ser uma variável”, e explicando ao Fabrício que as variáveis Tempo_Botao e Tempo_inicio não são suficientes

para modelar o problema, é necessário criar uma outra variável que, neste momento, Igor a denota de 60.

Professor – *Exatamente, só que a gente vai ter um problema aí, porque vejam bem, na próxima vez, isso já vai dar problema, porque a gente está estabelecendo agora, nesse caso, nessa rodada, o Tempo_Botao vai passar a valer 9 segundos e aí eu já não vou mais conseguir fazer a comparação com o Tempo_inicio. Porque quando ele (o processamento) pergunta: O Tempo_Botao é maior ou igual do que 10s (linha 17)? A resposta é não, porque está 9, então ele chama o [procedimento] normal (linha 20) e ao executar novamente o loop, ele (o processamento) pergunta, o pino_botao é alto (linha 12)? E aí já vai passar para baixo e vai ficar [no procedimento] normal para sempre e essa pessoa nunca vai atravessar. Vamos fazer uma outra simulação? (Silêncio) A pessoa apertou o botão de novo no tempo igual a 81s. (Professor escreve 81 ao lado da linha 10). O pino_botao é alto (linha 12) e a variável botão fica alto (linha 13), aí ele entra nesse Quando (linha 15), o Tempo_Botao agora é quanto? 81 menos...??*

Fabício – 9.

Professor – *Ah, deu ruim. Ele tinha que colocar 81 menos quanto?*

Fabício – *69 (silêncio por uns segundos). Então o tempo do botão não seria qualquer, é no caso, na matemática aqui óh, Tempo_inicio menos Tempo_Botao não seria qualquer Tempo_Botao, a referência dele deveria ser última vez ele ativou o sinal do pedestre, né?! Mas a última vez que o pedestre ficou verde é o 9s, não? (Estão discutindo sobre a linha 16)*

Professor – *Não, passaram-se 9s, porque o Tempo_inicio ele vai correndo. Vocês estão percebendo aqui na nossa simulação que eu tenho 81 que é o Tempo_inicio, eu tenho o 69 que é a última vez que o botão foi apertado e eu tenho o 11 segundos, ou seja, três números que não são fixos na minha programação.*

Fabício – *Então são 3 variáveis, né?*

Neste momento Fabício toma consciência da necessidade de um novo elemento indeterminado. E essa percepção aparece pela compreensão sobre a analiticidade dos demais indeterminados da modelagem. O que ocorreu, neste caso, foi um processo de abstração, porque ao ter consciência das necessidades para resolver o problema, Fabício vai ao encontro de um saber conceitual. É ir além de raciocinar analiticamente dado um indeterminado, é pensar analiticamente e ao fazer esse movimento, conceber um elemento indeterminado.

O professor confirma a compreensão de Fabício sobre a modelagem.

Professor – **É uma saída.**

Júlia – *A gente precisa definir essa variável, tempo da última vez que apertou.*

Wal – *Eu acho que é só a gente fazer Tempo_Botao igual o Tempo_inicio, que é do programa, menos Tempo da última vez. Eu só não sei quanto que essa variável vai valer e onde a gente vai colocar ela.*

Fabício – *Acho que a gente define a variável Tempo_ultimo, começa valendo zero e toda vez que a gente executar o [procedimento] pedestre, a gente faz ela valer o Tempo_inicio.*

Júlia e Wal acompanham Fabício e o professor e o diálogo mostra indícios dos processos de objetivação com o pensamento algébrico. Júlia compreende qual o valor que essa nova variável deve assumir e a Wal completa a fórmula para o tempo de espera do aciona-

mento sequencial do botão do pedestre. No entanto, ainda mostra estar perdida quanto ao posicionamento dentro da execução do programa, o que mostra que a aluna tem compreensão de que a programação tem uma estrutura de execução. Fabrício responde a Wal quanto as suas dúvidas, mostrando o cuidado com o outro, vetor da ética comunitária, e superando o individualismo. Neste contexto, é possível perceber que os processos de objetivação e subjetivação se complementam. O professor fez as alterações indicadas pelos estudantes e fez a simulação, que revelaram outros problemas de execução, rapidamente solucionado pelos próprios alunos, que não são focos deste trabalho.

Os alunos finalizaram a programação em C++ tranquilamente depois dessa etapa usando a LPF. A Figura 4 mostra como ficou a programação finalizada.

Figura 4 – Programa final em C++, Plataforma Tinkercad

```

1 int BOTAO = 0;
2 int VERDE = 2;
3 int AMARELO = 3;
4 int VERMELHO = 4;
5 int VERDE_P = 7;
6 int VERMELHO_P = 6;
7 int PINO_BOTAO = 5;
8 int TB = 0;
9 int TI = 0;
10 int TU = 0;
11
12
13 void setup()
14 {
15   pinMode(5, INPUT);
16   pinMode(2, OUTPUT);
17   pinMode(3, OUTPUT);
18   pinMode(4, OUTPUT);
19   pinMode(7, OUTPUT);
20   pinMode(6, OUTPUT);
21   Serial.begin(9600);
22 }
23
24 void normal() {
25   digitalWrite(VERDE, HIGH);
26   digitalWrite(AMARELO, LOW);
27   digitalWrite(VERMELHO, LOW);
28   digitalWrite(VERDE_P, LOW);
29   digitalWrite(VERMELHO_P, HIGH);
30
31 }
32 void pedestre(){
33   digitalWrite(VERDE, LOW);
34   digitalWrite(AMARELO, LOW);
35   digitalWrite(VERMELHO, HIGH);
36   digitalWrite(VERDE_P, HIGH);
37   digitalWrite(VERMELHO_P, LOW);
38 }
39
40 void amarelo(){
41   digitalWrite(VERDE, LOW);
42   digitalWrite(AMARELO, HIGH);
43   digitalWrite(VERMELHO, LOW);
44   digitalWrite(VERDE_P, LOW);
45   digitalWrite(VERMELHO_P, HIGH);
46 }
47
48 void loop()
49 {
50   TI = millis();
51   Serial.print(BOTAO);
52   Serial.print(" - ");
53   Serial.print(TB);
54   Serial.print(" - ");
55   Serial.println(TU);
56
57   if (digitalRead(PINO_BOTAO) == LOW) {
58     BOTAO = 1;
59   }
60   if (BOTAO == HIGH) {
61     TB = TI - TU;
62   }
63   if (BOTAO == HIGH && TB >= 10000) { // Pedestre
64     amarelo();
65     delay(2000); // Wait for 5000 millisecond(s)
66     pedestre();
67     delay(3000); // Wait for 5000 millisecond(s)
68     normal();
69     TU = TI;
70     BOTAO = 0;
71   } else {
72     normal();
73   }
74 }
75 }

```

Fonte: Acervo dos autores.

Na programação em C++ da Figura 4, TB é a variável Tempo_Botao usada na LPF, TI é a variável Tempo_inicio, usada na LPF, que assume o cronômetro da simulação na linha 50. E TU é a variável Tempo_ultimo, usada na LPF, ao qual os alunos tratam no diálogo. Ela representa o tempo em que o botão foi pressionado pela última vez.

CONSIDERAÇÕES FINAIS

Neste trabalho identificamos indícios do desenvolvimento do pensamento algébrico na programação produzidos pelos estudantes ao modelar a simulação de um semáforo.

Além disso, os processos de objetivação e subjetivação identificados denotam o labor conjunto estabelecido na atividade. Assim percebemos que é possível o desenvolvimento de elementos cognitivos referentes à manipulação de grandezas, generalização e abstração, demonstrando a possibilidade de desenvolver o pensamento algébrico por meio da programação.

Dessa forma, este trabalho inicia a discussão de uma proposta de um método para o ensino-aprendizagem do pensamento algébrico. Tal proposta consiste em planejar problemas que estejam no contexto dos estudantes. Entendemos “contexto” como sendo as situações que afloram dos próprios alunos, da comunidade escolar e social. Dessa forma, os estudantes estarão mais motivados e imersos nessas situações para a reflexão sobre a realidade, por meio de modelos cerebrais de imagens ideais que correspondem a objetos reais. Tal reflexão contribui para o desenvolvimento de processos mentais de comparações, assimilação, categorização, criação de hipóteses, entre outros, como foi possível observar nesta pesquisa. Através do pensamento criativo, da imaginação e da abstração, o indivíduo elabora elementos e conceitos baseados na reflexão sobre a realidade, que se materializam na atividade prática e caminham na direção da elaboração de um conceito geral. Dessa forma, a ativação do conhecimento do homem se relaciona com a atividade teórica, caracterizando os processos de subjetivação.

Como citado, propomos a programação como atividade mediadora da aprendizagem de formas de pensar algebricamente na modelagem de uma situação problema e defendemos esta forma pelo fato de que, na programação, os estudantes trabalham com o pensamento algébrico aplicado às situações simuladas desses contextos e, assim, os saberes são materializados em conhecimentos. Este método, ainda em construção, é uma proposta de produto educacional da pesquisa sobre o desenvolvimento do pensamento algébrico por meio da programação, no programa de Doutorado em Educação em Ciências e Matemática – Educimat, do Instituto Federal do Espírito Santo e que está em desenvolvimento. Os dados deste trabalho são parte de um protótipo deste estudo.

REFERÊNCIAS

- DELLA FONTE, Sandra Soares. **Formação omnilateral e a dimensão estética em Marx**. Curitiba, PR: Appris, 2020. Livro Digital – Kindle.
- MAJMUTOV, Mirza I. **La Enseñanza Problemática**. Pueblo y Educación. Havana: [s. n.], 1983.
- MOREY, Bernadete. Abordagem semiótica na Teoria da Objetivação. In: GOBARA, Shirley; RADFORD, Luis (Org.). **Teoria da Objetivação: fundamentos e aplicações para o ensino e aprendizagem de ciências e matemática**. São Paulo, SP: Livraria da Física, 2020, p. 43-70.
- RADFORD, Luis. Iconicity and contraction: A semiotic investigation of forms of algebraic generalizations of patterns in different contexts. **ZDM–International Journal on Mathematics Education**, [s. l.], v. 40, n. 1, p. 83–96, 2008. <https://doi.org/10.1007/s11858-007-0061-0>
- RADFORD, L. Algebraic thinking from a cultural semiotic perspective. **Research in Mathematics Education**, 12(1). 2010, p. 1-19. <https://doi.org/10.1080/14794800903569741>
- RADFORD, L. On the cognitive, epistemic, and ontological roles of artifacts. In G. Gueudet, B. Pepin, & L. Trouche, (Eds.), **From text to ‘lived’ resources**. New York: Springer. p. 283-288, 2012.

RADFORD, L. Un recorrido a través de la Teoría de la Objetivación. In: GOBARA, Shirley; RADFORD, Luis (Org.). **Teoria da Objetivação: fundamentos e aplicações para o ensino e aprendizagem de ciências e matemática**. São Paulo, SP: Livraria da Física, 2020, p. 15-42.

RADFORD, Luis. O Ensino-aprendizagem da Álgebra na Teoria da Objetivação. In: MORETTI, Vanessa; RADFORD, Luis (org.). **Pensamento algébrico nos anos iniciais: Diálogos e complementaridades entre a teoria da objetivação e a teoria histórico-cultural**. São Paulo, SP: Livraria da Física, 2021a, p. 171–195.

RADFORD, Luis. **Teoria da Objetivação: Uma perspectiva Vygotskiana sobre conhecer e vir a ser no ensino e aprendizagem da matemática**. São Paulo, SP: Livraria da Física, 2021b.

Histórico

Recebido: 14 de agosto de 2024.

Aceito: 13 de dezembro de 2024.

Publicado: 26 de dezembro de 2024.

Como citar – ABNT

PEREIRA, Rúbia Carla; Jordane, Alex. Um encontro com o Pensamento Algébrico através da programação no Tinkercad. **Revista de Matemática, Ensino e Cultura – REMATEC**, Belém/PA, n. 50, e2024008, 2024.

<https://doi.org/10.37084/REMATEC.1980-3141.2024.n50.e2024008.id697>

Como citar – APA

Pereira, R. C., & Jordane, A. Um encontro com o Pensamento Algébrico através da programação no Tinkercad. *Revista de Matemática, Ensino e Cultura – REMATEC*, (50), e2024008.

<https://doi.org/10.37084/REMATEC.1980-3141.2024.n50.e2024008.id697>

Número temático organizado por

Juliana Martins  

Jadilson Ramos de Almeida  