

A Programação Computacional Desenvolvida na Perspectiva do Tpack no Contexto da Formação Continuada do Professor de Matemática

Tpack Perspective Computational Programming Developed in the Context of Continued Education for Mathematics Teachers

Ana Karina de Oliveira Rocha^a; Maria Elisabette Brisola Brito Prado^{b,c*}

^aUniversidade Federal de Sergipe, SE, Brasil.

^bUniversidade Anhanguera de São Paulo, Programa de Pós-Graduação Stricto Sensu em Educação Matemática, SP, Brasil.

^cUnopar, Programa de Pós-Graduação Stricto Sensu em Metodologias para Ensino de Linguagens e suas Tecnologias, PR, Brasil.

*E-mail: maria_prado@anhanguera.com.

Resumo

Este artigo tem como objetivo compreender as possibilidades da atividade de Programação computacional utilizada no curso de formação continuada de um grupo de professores, como meio de potencializar a (re)construção de conhecimentos necessários para docência com tecnologia na perspectiva do TPACK. A pesquisa, de caráter qualitativo, envolveu a participação de um grupo de 10 professores de matemática que atuam no ensino da rede pública. Para a coleta de dados utilizaram-se: questionário perfil, registros escritos e gravados em áudio e protocolos de atividades feitas pelos participantes, durante os 10 encontros de 3 horas cada, perfazendo um total de 30 horas. As ações formativas eram desenvolvidas no laboratório de informática da Diretoria de Ensino Norte-2 da cidade de São Paulo e tinham como foco principal propiciar ao grupo de professores aprender a criar programas computacionais com o uso da linguagem Scratch envolvendo conteúdos matemáticos. Durante o curso, os professores, por meio do aprender-fazendo e refletindo sobre o fazer, construíram um *software* educacional sobre generalização de padrões de sequências numéricas para ser utilizado com seus alunos. Nesse processo de criação, identificaram-se situações que favoreceram a integração dos conhecimentos tecnológico (programação), pedagógico (conhecimento sobre o aluno e estratégias de ensino) e do conteúdo (matemático), ou seja, do TPACK, no sentido da (re)construção da base do conhecimento profissional docente da era digital.

Palavras-chave: Scratch. Construcionismo. TPACK. Reconstrução do Conhecimento. Professor de Matemática.

Abstract

This article aims to understand the possibilities of the Computational Programming activity used in the continued education course of a group of teachers as a means to boost the (re)building of the required knowledge concepts for teachers with technology in the TPACK perspective. This qualitative research involved the participation of a group of 10 mathematics teachers who teach at the public school system. For data collection, the following was used: profiling questionnaire, written and audio-recorded logs and the protocols of the activities performed by the participants throughout the 10 three-hour meetings, with a total of 30 hours. The developing actions were performed at a computer lab at the North-2 Educational Division of Sao Paulo city and had as their main focus to provide the group of teachers with learning experience on how to create computer programs using Scratch language for mathematical contents. Throughout the course, the teachers, using hands-on-learning and reflecting about their practice, built an educational software about the generalization of patterns of numerical sequences to use with their students. During this creative process, we have identified situations which showed the integration of technological knowledge (programming), pedagogical knowledge (about the student and learning strategies), and content knowledge (mathematical), that is, using the TPACK to (re) build the teacher's professional knowledge basis in the digital era.

Keywords: Scratch, constructionism, TPACK, rebuilding knowledge, Mathematics teacher.

1 Introdução

A educação na cultura digital tem sido uma temática instigadora para pesquisadores de diferentes áreas do conhecimento que buscam compreender os impactos da presença das tecnologias digitais da informação e comunicação (TDIC) no cotidiano da vida dos estudantes e dos professores. O uso das tecnologias digitais na educação pode favorecer os processos de ensino e de aprendizagem, pela facilidade tanto de buscar e socializar informações como de propiciar aos estudantes construir conhecimentos, por meio de diversas formas de representação. Mas para isso é necessário repensar sobre a escola, o currículo e a prática do professor.

Em se tratando da prática pedagógica do professor para atuar na escola usando os recursos das TDIC, as pesquisas

de Almeida & Valente (2011), Prado, Lobo da Costa & Padilha (2016), entre outras, têm mostrado que essa não é uma tarefa fácil, pois requer do professor a (re)construção de conhecimentos. Isso implica rever a formação docente, segundo Fantin e Rivoltella (2012), considerando que a sociedade caracterizada pela cultura digital exige desse profissional novas competências, as quais demandam a apropriação pedagógica das tecnologias digitais, na perspectiva de integrá-las ao currículo da escola.

Nesse sentido, iniciativas do governo, por meio de programas, como, por exemplo, Mídias na Educação, Proinfo Integrado, Educação na Cultura Digital, desenvolvidos em parcerias com Secretarias de Educação dos estados e municípios, juntamente com universidades públicas e privadas,

ofertaram cursos de formação continuada de professores na área de tecnologia na educação. Embora esses cursos tenham propiciado aos professores desenvolverem práticas inovadoras com o uso das TDIC, as pesquisas, como a de Eivazian (2012) e Almeida (2014), apontam que tais práticas são pontuais e restritas ao âmbito das escolas públicas. As razões que contribuem para tal situação são diversas: uma delas, por exemplo, refere-se ao fato de o professor ter que aprender a lidar com recursos tecnológicos e a reconstruir a própria prática docente, aquela que foi construída e consolidada no seu cotidiano escolar, muitas vezes, sem o uso das TDIC. Sob esse enfoque, Almeida e Valente (2011, p.50) salientam que a formação do professor envolve

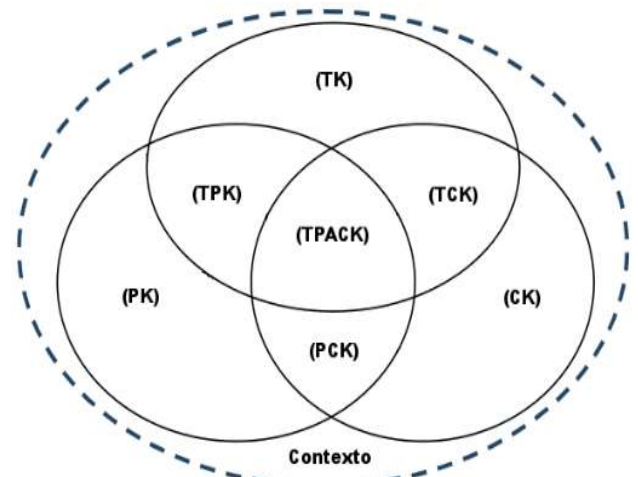
muito mais do que provê-lo com o conhecimento técnico sobre as TDIC. Ela [a formação] deve criar condições para o professor construir conhecimento sobre os aspectos computacionais; compreender as perspectivas educacionais subjacentes aos softwares em uso, isto é, as noções de ensino, aprendizagem e conhecimento implícitas no software; e entender por que e como integrar o computador com o currículo e como concretizar esse processo na sua prática pedagógica.

Para tanto, a formação do professor tem um papel fundamental neste novo cenário da educação voltada para a cultura digital. Nesse sentido, alguns estudos, como, por exemplo, de Hughes (2004), Coutinho (2011) e Lobo da Costa & Prado, (2015), têm mostrado a necessidade de a formação do professor privilegiar o desenvolvimento de ações voltadas para a base do conhecimento profissional docente, conforme o modelo teórico apresentado por Mishra e Koehler (2006), denominado de TPACK (em inglês, *Technological Pedagogical and Content Knowledge*).

Para a construção do TPACK, esses autores tomaram como ponto de partida a teoria da base de conhecimento de Shulman (1986, 1987), que introduziu o conceito de Conhecimento Pedagógico do Conteúdo (em inglês, *Pedagogical Content Knowledge - PCK*). O construto PCK refere-se à intersecção entre os conhecimentos do conteúdo (em inglês, *Content Knowledge - CK*) e o conhecimento pedagógico (em inglês, *Pedagogical Knowledge - PK*), caracterizando, dessa forma, um novo conhecimento exclusivo do profissional professor, ou seja, aquele que lhe dá condições para desenvolver o ensino que favoreça o processo de aprendizagem do aluno.

Com avanço das tecnologias disponíveis nas escolas, os pesquisadores Mishra e Koehler (2006) ampliaram a base do conhecimento de Shulman: incluíram o conhecimento tecnológico (em inglês, *Technological knowledge - TK*) e, dessa forma, consolidaram o modelo teórico do TPACK (Figura 1)

Figura 1 - Representação do modelo TPACK



Fonte: Adaptação de Koehler e Mishra (2009).

A Figura 1 mostra que, da integração dos conhecimentos (CK, PK e TK), originam três níveis de intersecções, que representam novos tipos de conhecimentos, descritos sucintamente a seguir:

- Conhecimento Pedagógico do Conteúdo (em inglês, *Pedagogical Content Knowledge - PCK*), que foi herdado das ideias de Shulman (1987) e se refere ao conhecimento do professor de quais abordagens pedagógicas são adequadas para o ensino de um determinado conteúdo. Podemos dizer que, em síntese, envolve saber: o quê, por que, para quem e como ensinar determinado conteúdo.
- Conhecimento Tecnológico do Conteúdo (em inglês, *Technological Content Knowledge - TCK*), que se refere ao conhecimento da maneira como a tecnologia e o conteúdo estão relacionados, pois o professor precisa saber não apenas o conteúdo que ensina, mas também as diferentes maneiras de representá-lo, utilizando os recursos tecnológicos disponíveis.
- Conhecimento Tecnológico Pedagógico (em inglês *Technological Pedagogical Knowledge - TPK*), relativo ao conhecimento das potencialidades e das restrições de cada recurso tecnológico utilizado e suas implicações nos processos de ensino e de aprendizagem.

Esses três tipos de conhecimentos (PCK), (TCK) e (TPK) se interseccionam, resultando no Conhecimento Tecnológico e Pedagógico do Conteúdo (em inglês, *Technological Pedagogical Content Knowledge - TPACK*). Portanto, situado em um determinado contexto, este conhecimento requer a compreensão das diferentes formas de representação de conceitos relacionados ao conteúdo específico, com o uso das tecnologias e das estratégias pedagógicas que possam favorecer o processo de aprendizagem dos alunos.

Esse modelo teórico TPACK, nas últimas décadas, vem sendo estudado no contexto da formação continuada de professores de matemática, como, por exemplo, por Niess, Sadri e Suharwoto (2006), Vieira (2013), Rocha (2015), Prado & Lobo da Costa (2016), entre outros, que destacam a complexidade do processo de formação e, ao mesmo tempo, anunciam que este pode ser um caminho possível para propiciar o desenvolvimento profissional docente, levando em

conta os artefatos tecnológicos existentes na sociedade atual.

Considerando tais estudos, este artigo, que se refere a um recorte da pesquisa de doutorado da primeira autora, tem como objetivo compreender as possibilidades da atividade de Programação computacional utilizada no curso de formação continuada de um grupo de professores, como meio de potencializar a (re)construção de conhecimentos necessários para a docência com tecnologia na perspectiva do TPACK.

2 Desenvolvimento da Pesquisa

Esta pesquisa de caráter qualitativo, baseado nos princípios de Bogdan e Biklen (1994), envolveu a participação de um grupo de dez professores de matemática que atuam no Ensino Fundamental II e Médio da rede estadual pública na cidade de São Paulo. Por meio de uma parceria entre o Programa de Pós-Graduação em Educação Matemática da Universidade Anhanguera de São Paulo e a Diretoria de Ensino – Norte 2 da Secretaria de Educação do Estado de São Paulo, foi oferecido um curso de formação continuada sobre Linguagem de Programação Scratch no contexto de Matemática.

Essa formação foi desenvolvida no laboratório de informática da Diretoria de Ensino Norte-2 da cidade de São Paulo, em 10 encontros de 3 horas cada, perfazendo um total de 30 horas de atividades. Além dos encontros presenciais, os participantes tiveram acesso ao ambiente virtual no Moodle, que foi customizado para ser um espaço complementar de aprendizagem. Nesse espaço virtual, eram disponibilizados os materiais de apoio, tutoriais e textos, além das ferramentas de comunicação para viabilizar as interações entre todos os envolvidos, favorecendo a troca de ideias, o esclarecimento de dúvidas e o compartilhamento de reflexões.

Para a coleta de dados foram utilizados os seguintes instrumentos: questionário perfil, registros escritos e gravados em áudio dos encontros de formação e protocolos de atividades realizadas pelos professores participantes.

A dinâmica adotada nos encontros formativos envolvia os professores na realização de atividades com resolução de problemas, por meio da criação de programas que exploravam conteúdos matemáticos, ao mesmo tempo em que aprendiam a linguagem computacional Scratch.

3 Atividade de Programação Scratch

O Scratch é uma linguagem de programação desenvolvida pelo grupo de pesquisa Lifelong Kindergarten no MIT Media Lab, com o objetivo de propiciar a participação de pessoas das mais variadas idades, formações e nacionalidades na criação de *softwares* como: jogos, animações e simulações (Resnick et al., 2009). Segundo este autor, o Scratch foi criado não com o intuito de formar programadores profissionais, mas para que as pessoas pudessem se expressar de forma criativa por meio da programação computacional (Resnick, 2012).

A Figura 2, a seguir, apresenta a tela inicial da página do Scratch (<https://scratch.mit.edu/>), onde é possível fazer uma

inscrição gratuita para ter acesso à versão *online* da linguagem de programação.

Figura 2 - Página inicial do Scratch na Internet

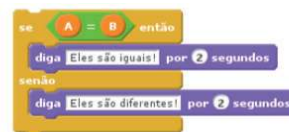


Fonte: Os autores.

Para os seus criadores, a programação com o Scratch oferece suporte ao pensamento computacional, proporcionando às pessoas aprender sobre resoluções de problemas na matemática e desenvolver estratégias de *design* que podem ser aproveitadas em outros domínios do conhecimento (Resnick et al., 2009).

A gramática da linguagem Scratch baseia-se em uma coleção de blocos de comandos organizados em várias categorias (com cores distintas), que podem ser encaixados e encadeados de forma a produzir ações desejadas.

Figura 3 - Exemplo de blocos de comandos da linguagem Scratch



Fonte: Os autores.

Esses blocos são constituídos por comandos, de acordo com suas funções, por exemplo, comandos de movimentos, de controle, sensores, operadores, variáveis, entre outros.

As Figuras 4 e 5 mostram blocos de comandos relacionados ao movimento do Scratch. Os comandos que aparecem na figura 4 podem ser empregados em atividades envolvendo conteúdos da geometria e trigonometria, pois possibilitam a utilização de conceitos como, por exemplo, ângulos e direção, cujas medidas estão definidas em graus.

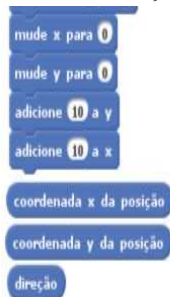
Figura 4 - Comando para a utilização de ângulos



Fonte: Os autores.

Os comandos mostrados na Figura 5 podem ser utilizados em conteúdos da matemática que explorem o eixo cartesiano, possibilitando, por exemplo, construir figuras, delimitar na tela espaços, via programação, que podem servir de cenários para produzir eventos e/ou efeitos de animação.

Figura 5 - Comandos para movimentação no plano cartesiano



Fonte: Os autores.

Na Figura 6, pode-se ver um bloco de comandos utilizado na atribuição de valores para as variáveis, criadas no ambiente de programação Scratch.

Figura 6 - Comandos para atribuição de variáveis



Fonte: Os autores.

Outro recurso da linguagem Scratch, que serve para armazenar valores, preservando a ordem linear entre eles, é a estrutura de dados denominada Lista. A figura 7 mostra comandos para preencher uma lista com as letras A, B e C, mantendo a ordem de inserção.

Figura 7 - Comandos da estrutura de dados Lista



Fonte: Os autores.

Além dos comandos existentes, a linguagem possui operadores aritméticos: adição, subtração, multiplicação e divisão; operadores de comparação: menor que, igualdade, maior; operadores lógicos: <E>, <OU>, <NÃO>; e as funções matemáticas pré-definidas que executam operações como: raiz quadrada, logaritmo, arredondamento de um número, entre outros. Na linguagem Scratch, os operadores lógicos são utilizados em blocos de comandos do tipo: “Se <sentença lógica for verdadeira> então (execute A) senão (execute B)”, enquanto A e B são procedimentos ou comandos, definidos por quem está programando com a linguagem.

A escolha da linguagem de programação nesta pesquisa se deu pelo fato de os comandos do Scratch possuírem relação com conceitos da matemática, que podem ser utilizados em diversas atividades envolvendo resolução de problemas e criação de programas de diferentes níveis de complexidade.

4 Aprendizagem do Scratch e o Tpack

O curso de formação continuada teve como foco principal propiciar aos professores participantes aprenderem a criar programas computacionais envolvendo a generalização de padrões com sequência numéricas.

Os participantes da pesquisa eram graduados em

Licenciatura em Matemática, e alguns tinham mais de uma graduação. A maioria do grupo já tinha feito cursos de extensão e de especialização. No entanto, os professores não conheciam o Scratch, tampouco tinham vivenciado a atividade de programação; apenas um professor do grupo tinha trabalhado com a linguagem Pascal durante seu curso de graduação. A maioria conhecia alguns *softwares* específicos de matemática, tais como Winplot e Geogebra, mas nunca os tinha utilizado na prática com os alunos. Todos tinham pouca familiaridade com os recursos tecnológicos: nove professores alegaram ter conhecimento moderado, e apenas um dos participantes comentou ter pouca familiaridade.

Considerando o perfil dos participantes, o desenvolvimento do curso privilegiou, nos primeiros encontros formativos, a exploração e a apresentação de alguns comandos e do ambiente do Scratch. Em seguida, foram propostas atividades para serem desenvolvidas em dupla, abordando a resolução de problemas, mais especificamente a criação de pequenos programas voltados para explorar conteúdos matemáticos, tais como geometria, teorema de Pitágoras, números primos, entre outros.

Dessa forma, os participantes desenvolveram programas a partir da construção do algoritmo em linguagem natural; em seguida, elaboraram sua representação no fluxograma; e, finalmente, em linguagem computacional. Nesse processo de aprender-fazendo o programa, os professores foram se apropriando dos comandos da linguagem computacional Scratch. A seguir, apresentamos um exemplo da atividade realizada pela dupla de professores (S07 e S10):

Quadro 1 - Exemplo de atividade “Par ou ímpar?”

Dado um número $x \in \mathbb{N}$ fornecido pelo usuário, diga se ele é par ou ímpar.

1. Escrever o algoritmo em linguagem natural
2. Elaborar o fluxograma
3. Criar um programa que descobre se o número x fornecido pelo usuário é par ou ímpar.

Fonte: Dados da pesquisa.

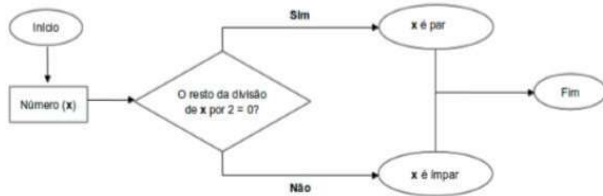
Para resolver a atividade “Par ou ímpar?”, primeiramente os professores escreveram o algoritmo em linguagem natural e, em seguida, construíram o fluxograma, como mostram os exemplos a seguir:

Quadro 2 - Algoritmo em Linguagem Natural

Seja $x \in \mathbb{N}^*$ um número fornecido pelo usuário, calcule o resto da divisão do número x por 2.

Se o resto da divisão do número x por 2 for igual a zero, o número x é par. Senão, o número x é ímpar.

Fonte: Dados da pesquisa.

Figura 8 - Algoritmo representado no Fluxograma

Fonte: Dados da pesquisa.

O próximo passo no curso foi propiciar ao grupo de professores a vivência do aprender-fazendo a programação, ou seja, ensinar o computador a resolver a situação proposta na atividade “Par ou ímpar?”. Nesse momento, a dupla (S07 e S10) propôs que o programa fizesse uma pergunta para o usuário e ficasse esperando uma resposta, a qual seria o valor de N. Na programação, esse valor dado pelo usuário precisaria ser armazenado em uma variável para ser testado. A partir do teste – que envolve conceitos numéricos –, o programa deveria emitir uma mensagem, dizendo: “O número N é par” ou “O número N é ímpar”. A Figura 9 mostra um exemplo dos comandos utilizados no programa interativo da atividade “Par ou ímpar?”:

Figura 9 - Comandos do programa na linguagem Scratch

Fonte: Dados da pesquisa.

Nessa dinâmica do curso, os professores foram conhecendo os conceitos de programação, a sintaxe e os recursos do Scratch em atividades matemáticas pontuais. O interessante é que neste tipo de atividade os professores participantes evidenciaram na prática a integração dos conhecimentos matemáticos e do Scratch, que representa o Conhecimento Tecnológico do Conteúdo (em inglês, *Technological Content Knowledge - TCK*).

Ressaltamos que, no contexto do curso, o papel da mediação da formadora/pesquisadora foi fundamental, uma vez que ela procurou intervir no grupo de professores com base nos princípios de Vygotsky (1991). Ela buscava entender a zona de desenvolvimento proximal dos professores, para que fosse possível, por meio do diálogo, provocar a reflexão sobre os conteúdos matemáticos e tecnológicos envolvidos na atividade. Além da mediação direta presencial e/ou a distância, a interação entre a formadora/pesquisadora e os professores permitiu redesenhar as atividades do curso entre um encontro e outro, de modo a favorecer ao grupo engajar-se no processo de programação, dando significado para a própria aprendizagem.

Assim, em continuidade às ações formativas, foi proposta

aos professores a construção de um *software* educacional envolvendo o conteúdo de generalização de padrões com sequências numéricas, voltado para a realidade de seus alunos.

A atividade de criação do *software* educacional foi feita individualmente pelos participantes. A maioria dos professores sentiu a necessidade de fazer um planejamento do *software* por meio da linguagem natural, levantando os questionamentos e as ações que seriam desenvolvidas nas fases de programação. O desafio colocado para o grupo foi de pensar como o *software* poderia instigar seus alunos a refletir e construir conhecimento sobre o conteúdo abordado.

No processo de programar o *software*, os professores foram descobrindo novos recursos da linguagem de programação e aprendendo novos conceitos computacionais necessários para representar os conceitos matemáticos de forma diferente daquela que habitualmente fazem no seu cotidiano. Além disso, como era a intenção da maioria do grupo criar o *software* contemplando o aspecto interativo, isso demandou dos professores reflexões relacionadas aos aspectos pedagógicos. Para o programa dialogar com os usuários (os alunos), alguns professores incluíram no *software* a criação de um personagem do tipo avatar, que, durante a interação, apresentava o problema e fazia questionamentos sobre o conteúdo matemático, conforme exemplifica o trecho do planejamento do professor S07 no Quadro 3 a seguir:

Quadro 3 - Trecho do planejamento do *software* de S07

Devo idealizar estratégias para que o aluno descubra as posições solicitadas e fazer com que ele consiga escrever a lei de formação da sequência.

Devo pensar em como programar o acerto e o erro do aluno.

1ª fase: criar um personagem (tipo um avatar) para introduzir o assunto, por exemplo: “Olá nesta atividade vamos investigar os padrões e regularidades na formação de uma sequência matemática”.

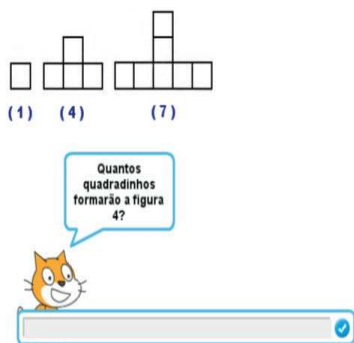
2ª fase: perguntar ao aluno quantos quadradinhos formam a Figura 1. Idem para a Figura 2, até a Figura 3. Programar para que as respostas do aluno sejam armazenadas e, após determinado tempo, deve aparecer em forma de sequência numérica. A partir da Figura 4, o aluno deverá dizer quantos quadradinhos ela possui sem vê-la. Se ele acertar, a figura aparecerá na tela. Idem para a Figura 5.

3ª fase: o próximo passo é obter uma expressão que nos permita obter o termo geral desta PA conhecendo o 1º termo e a razão.

Fonte: Dados da pesquisa

A Figura 10 mostra o efeito da programação na tela do computador, em que aparece o personagem (Sprite) fazendo uma pergunta para o usuário (o aluno). Nesse caso, o programa aguarda a resposta da pergunta que o aluno deverá digitar, a partir da observação da imagem que representa a sequência.

Figura 10 - Tela da 2ª fase do *software* educacional elaborada pelo S07



Fonte: Dados da pesquisa.

Neste exemplo, podemos observar que o professor S07 em seu planejamento demonstrou preocupação com os aspectos pedagógicos, tanto na forma de abordar os conteúdos matemáticos como na escolha da ordem em que cada conceito deveria aparecer no *software*. Além disso, S07 sinalizou estar atento à maneira de os alunos aprenderem, procurando antecipar suas resoluções, prevendo seus erros e acertos.

No coletivo do grupo ficou evidenciado, durante a construção do *software* educacional, que os professores levantavam uma série de conjecturas sobre situações-problema que pudessem evidenciar as regularidades de sequências numéricas. E nesse processo também foram construídos seus algoritmos, usando os comandos da linguagem, articulando conceitos matemáticos e computacionais. Interessante, nesse processo, foi observar que o conhecimento do conteúdo matemático foi realimentando o conhecimento da programação Scratch e viceversa. Este fato nos permitiu identificar que esse grupo de professores conseguiu realizar a integração entre os conhecimentos tecnológicos e do conteúdo, por ter compreendido como descompactar o conhecimento matemático para representá-lo de uma nova forma, usando a linguagem Scratch. Por sua vez, essa integração retrata o Conhecimento Tecnológico do Conteúdo (em inglês, *Technological Content Knowledge - TCK*) pelo grupo de professores.

Outro fato ocorrido durante a programação do *software* refere-se ao momento em que alguns professores se depararam com a necessidade de programar um tipo de *feedback* para o aluno, a partir da sua resposta, ao interagir com o *software* educacional. Para isso, o programa teria que checar possíveis (variáveis) respostas que seriam dadas pelos alunos e enviar um *feedback* pedagógico, ou seja, para orientar a aprendizagem do aluno. Essa situação foi crucial, pois desencadeou o processo de (re)construção do Conhecimento Tecnológico Pedagógico (em inglês, *Technological Pedagogical Knowledge - TPK*) desse grupo de professores.

Nesse momento do curso, ficou evidente que essa situação causou no grupo um desequilíbrio cognitivo, e, na busca de solução para dar continuidade à atividade de programação do *software*, alguns professores resolveram programar

um *feedback* genérico e muito simples do ponto de vista pedagógico. Por exemplo, se a resposta dada pelo aluno não fosse correta, deveria aparecer na tela a frase: “Você errou, tente novamente!”.

A solução encontrada neste exemplo não foi adequada pedagogicamente, pois não retrata uma intervenção que possa orientar o processo de aprendizagem do aluno. Nesse momento, a mediação pedagógica da pesquisadora-formadora foi fundamental; e, para provocar a reflexão dos professores sobre os aspectos pedagógicos, foi colocada a seguinte questão no grupo: “O que vocês fariam nessa situação, para interagir com os alunos em sala de aula?”. Os professores ficaram alguns instantes em silêncio, pois essa situação gerou momentos de introspecção no grupo e, aos poucos, os participantes começaram a trocar ideias sobre as possíveis resoluções dos alunos e de que forma poderiam programar suas intervenções pedagógicas. Esse momento foi extremamente importante, pois desencadeou um processo reflexivo sobre a prática pedagógica dos professores, ou seja, sobre situações semelhantes que acontecem em sala de aula, quando seus alunos não acertam a resposta de determinadas atividades.

A situação de propiciar aos professores que refletissem sobre a própria prática pedagógica não tinha sido prevista no *design* do curso, mas emergiu na ação durante os encontros formativos, como consequência da atividade de programação do *software*. Esse fato nos revela o potencial da programação sendo desenvolvida com base nos princípios construcionistas de Papert (2008) e ampliados em diversas situações pelos estudos de Valente (2016), os quais priorizam o aprender-fazendo e refletindo sobre o fazer no contexto de uma atividade em que o aprendiz esteja afetiva e cognitivamente envolvido na sua produção. Nesse caso, os professores estavam aprendendo-fazendo e refletindo sobre a programação no contexto da matemática, produzindo um *software* que seria utilizado por seus alunos.

No entanto, essa atividade também requereu uma base referencial dos processos de ensino e aprendizagem que precisavam ser tratados de forma articulada para dar conta de programar os *feedbacks*, a fim de contribuir para a aprendizagem dos alunos.

Isso ficou visível no momento em que alguns professores tentaram virtualizar de alguma maneira a sala de aula real no *software*, ou seja, sem ressaltar o erro do aluno, mas procurando compreendê-lo. Foi necessário a formadora/pesquisadora retomar essa situação no coletivo do grupo para discutir a questão do erro do aluno, que pode ser um indicativo para o professor se inteirar sobre o conhecimento do aluno e fazer as intervenções adequadas para favorecer sua aprendizagem.

Essa situação instigou alguns professores a revisitarem a própria prática de sala de aula, identificando as lacunas conceituais de seus alunos em relação aos conteúdos matemáticos. A partir daí, os professores compartilhavam e refletiam no grupo sobre o conhecimento prévio do aluno e

sua importância para o estabelecimento de relações. Trocavam ideias, avaliando as estratégias de ensino que poderiam contribuir para aprendizagem dos alunos. Esse momento do grupo evidenciou a integração do Conhecimento Pedagógico do Conteúdo (em inglês, *Pedagogical Content Knowledge - PCK*).

No entanto, a atividade dos professores centrava-se no processo de criação do *software*, o qual, no decorrer do curso, foi ganhando uma nova dimensão, pois a reflexão do grupo de professores passou a enfatizar o quanto sabiam sobre o conhecimento do aluno. Esse processo deixou evidente que o grupo de professores reconheceu a necessidade de articular esse conhecimento com as possíveis estratégias pedagógicas que poderiam ser programadas para envolver os alunos no aprendizado sobre generalização de padrões com sequências numéricas, ao interagirem com o *software* educacional.

Esse momento, portanto, marcou a integração dos Conhecimentos Tecnológico e Pedagógico do Conteúdo (em inglês, *Technological Pedagogical Content Knowledge - TPACK*) e destacou, em relação aos conhecimentos (PCK e TPCK), a importância de estarem ancorados em princípios teóricos e práticos educacionais.

5 Conclusão

Esta pesquisa mostrou o potencial da atividade de programação utilizada no contexto de formação continuada do professor de matemática, em termos de propiciar a apropriação das tecnologias digitais; no caso, a linguagem de programação Scratch indo além do domínio operacional.

Esse aprendizado da linguagem de programação Scratch pelo grupo de professores participantes da pesquisa revelou a importância do tipo de atividade que deve ser proposta durante a formação. As atividades podem envolver pequenos projetos, tal como a atividade “Par ou ímpar?”, que representou uma porta de entrada significativa para o aprendizado do professor sobre a linguagem Scratch; ou projetos mais elaborados em termos computacionais, como aconteceu na criação do *software* educacional.

Vale ressaltar que as ações de formação tiveram como base os princípios construcionistas, trabalhados na perspectiva de projeto, oportunizando ao grupo de professores a vivência da autoria durante o processo de aprender-fazendo e a reflexão sobre o fazer. Ficou evidente, nos momentos formativos, a presença do caráter reflexivo do grupo, que a cada encontro se tornava mais abrangente e articulada, envolvendo o conhecimento da prática vivenciada no curso (construindo o *software*) e o conhecimento da prática na atuação profissional (em sala de aula).

Esta forma de aprender, vivenciando a atividade de programação e assumindo autoria na produção de um *software* sobre conteúdo matemático destinado aos alunos da educação básica, pode ser um caminho que oportunize aos professores integrar os Conhecimentos Tecnológico e Pedagógico do Conteúdo (em inglês, *Technological Pedagogical Content*

Knowledge - TPACK), a fim de reconstruir a base do conhecimento profissional docente para a era digital.

Referências

- Almeida, M. E. B. (2014). Integração currículo e tecnologias: concepção e possibilidades de criação de web currículo. In: M. E. B. Almeida, R. M. Alves, & S. D. V. Lemos (Org.), *Web currículo. Aprendizagem, pesquisa e conhecimento com o uso de tecnologias digitais*. Rio de Janeiro: Letra Capital
- Almeida, M. E. B., & Valente, J. A. (2011). *Tecnologias e currículo: trajetórias convergentes ou divergentes?* São Paulo: Paulus.
- Bogdan, R., & Biklen, S. (1994). *Investigação qualitativa em educação: uma introdução à teoria e aos métodos*. Porto: Porto.
- Coutinho, C. P. (2011). Tpack: em busca de um referencial teórico para a formação de professores em tecnologia educativa. *Rev Cient Educ Dist*, 2(4).
- Eivazian, A. M. B. (2012). *O computador móvel e a prática de professores que ensinam matemática em uma escola do projeto UCA*. Dissertação de Mestrado em Educação Matemática, Universidade Bandeirante de São Paulo, São Paulo.
- Fantini, F., & Rivoltella, P. C. (2012). *Cultura digital: pesquisa e formação de professores*. Campinas: Papirus.
- Hughes, J. (2004). Technology learning principles for preservice and in-service teacher education. *Contemporary Issues in Technology and Teacher Education*, 4(3), 345-362.
- Koehler, M. J., & Mishra, P. (2009). What is technological pedagogical content knowledge? *Contemporary Issues in Technology and Teacher Education*, 9(1), 60-70.
- Lobo da Costa, N. M., & Prado, M. E. B. B. (2015). Integração das tecnologias digitais ao ensino de matemática: desafio constante no cotidiano escolar do professor. *Perspectivas da Educação Matemática*, 8(16), 121-139.
- Mishra, P., & Koehler, M. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6), 1017-1054.
- Niess, M. L., Lee, K., Sadri, P., & Suharwoto, G. (2006, April). Guiding inservice mathematics teachers in developing a technology pedagogical content knowledge (TPCK). In *American Educational Research Association Annual (AERA) Conference*, San Francisco, CA.
- Papert, S. (2008). *As máquinas das crianças – repensando a escola na era da informática*. Porto Alegre: Artmed.
- Prado, M. E. B. B., & Lobo da Costa, N. M. (2016). O papel da atividade de programação no processo de construção de conhecimentos para a docência. *E-Curriculum*, 14(3), 898-918.
- Prado, M. E. B. B., Lobo da Costa, N. M., & Padilha, W. R. (2016). Tecnologias digitais móveis e o processo de apropriação do professor para a reconstrução da prática pedagógica. In *Atas do IV Congresso Internacional TIC na Educação - Tecnologias digitais e a Escola do Futuro* (pp. 39-51). Lisboa, Portugal: Instituto de Educação da Universidade de Lisboa.
- Resnick, M. (2012, July/August). Reviving Papert's dream. *Educational Technology*, 52(4), 42-46.
- Resnick, M., Maloney, J., Hernández, A. M., Rusk, N., Eastmond, E., Brennan, K., Rosenbaum, E. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11),

60-67.

- Rocha, A. K. de O. (2015). *A programação de computadores como meio para integrar diferentes conhecimentos: uma experiência com professores de Matemática*. Tese de Doutorado, Universidade Anhanguera de São Paulo – UNIAN, São Paulo.
- Shulman, L. S. (1986). Those who understand: knowledge growth in teaching. *Educational Researcher*, 15(2), 4-14.
- Shulman, L. S. (1987). Knowledge and teaching: foundations of the new reform. *Harvard Educational Review*, 57(1), 1-22.
- Valente, J. A. (2016). Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. *E-Curriculum*, 14(3), 864-897.
- Vieira, E. R. (2013). *Grupo de estudos de professores e a apropriação de tecnologia digital no ensino de Geometria: caminhos para o conhecimento profissional*. Tese de Doutorado, Universidade Anhanguera de São Paulo, São Paulo.
- Vygotsky, L. S. (1991). *A formação social da mente*. São Paulo: Martins Fontes.