

O Modelo Tpack e a formação de professores de Matemática: repercussões de um Curso de Robótica Educacional

*Cleia Alves Nogueira**

*Cleyton Hércules Gontijo***

*Maria Elisabette Brisola Brito Prado****

*Américo Junior Nunes da Silva*****

Resumo

Este artigo objetiva analisar o processo de construção das atividades desenvolvidas por duas professoras de matemática, participantes de um curso de formação continuada, sobre a inserção das tecnologias em seus processos formativos e construção de novas perspectivas de atuação, a partir do modelo teórico do Conhecimento Tecnológico Pedagógico do Conteúdo (TPACK). As informações analisadas fazem parte da pesquisa de doutorado da primeira autora e foram constituídas em uma perspectiva narrativa. Do processo investigativo, pode-se concluir que a utilização pedagógica da robótica amplia as possibilidades de reflexão acerca do que pode ser feito em sala de aula, promovendo um ambiente de aprendizagem mais interativo e colaborativo, além de estimular o desenvolvimento

* Doutora e Mestre em Educação, pela Universidade de Brasília (UnB). Especialista em Tecnologias na Educação pela PUC-RJ e em Arte, Educação e Tecnologias Contemporâneas pela UnB. Orcid: <https://orcid.org/0000-0003-0983-2631>. Email: cleianog@gmail.com.

** Professor Associado IV na Universidade de Brasília (UnB). Orcid: <https://orcid.org/0000-0001-6730-8243>. Email: cleyton@unb.br.

*** Doutora em Educação: Currículo pela PUC-SP. Docente do Programa de Pós-Graduação em Metodologias do Ensino de Linguagens e suas tecnologias – UNOPAR. Orcid: <https://orcid.org/0000-0002-8595-4203>. Email: bette.prado@gmail.com

**** Professor Adjunto do Departamento de Educação da Universidade do Estado da Bahia – (Uneb - Campus VII). Orcid: <https://orcid.org/0000-0002-7283-0367>. Email: ajnunes@uneb.br.

de habilidades críticas e criativas no campo da matemática. Decorre desse processo, a recomendação de que a formação continuada dos professores esteja apoiada no desenvolvimento de conhecimentos para trabalhar com tecnologias digitais, proporcionando experiências que possam reverberar em suas práticas com os estudantes.

1. Introdução

As tecnologias têm se tornado cada vez mais presentes em vários aspectos da vida contemporânea, e na educação isso não é diferente. Com a popularização de dispositivos eletrônicos, da internet e de plataformas online, a utilização de tecnologias educacionais ampliou significativamente as possibilidades de organização do trabalho pedagógico, favorecendo o acesso a informações, recursos educacionais e atividades interativas, colaborando para melhorar os processos de ensino e de aprendizagem.

No entanto, para que o uso das tecnologias digitais impacte positivamente na qualidade do ensino, os professores precisam compreender os conhecimentos que integram essas ferramentas e como elas refletem em sua prática pedagógica; levando em consideração a interação entre conteúdo, pedagogia e tecnologia. Dessa forma, destaca-se um modelo teórico, criado por Mishra e Koehler (2006) denominado de Technological Pedagogical Content Knowledge (TPACK), que descreve os conhecimentos necessários para a prática de professores que utilizam tecnologias digitais como suporte para o ensino, tornando-se algo essencial para promover uma formação adequada.

Dessa forma, é fundamental que instituições de ensino e professores considerem as possibilidades oferecidas pelas tecnologias educacionais, buscando utilizá-las de forma crítica e criativa, a fim de maximizar seu potencial inclusivo e a qualidade do ensino. A formação continuada de professores, considerando não apenas o uso das tecnologias digitais, mas também o conhecimento sobre como integrá-las aos processos de ensino e de aprendizagem, é um passo importante nessa direção.

Objetivando compreender esse processo de integração das tecnologias na formação do professor, em especial o de matemática, este artigo organiza-se da seguinte forma: i) Introdução, apresenta a temática e objetivo de escrita; ii) Encaminhamentos Metodológicos, descreve a metodologia adotada para a realização da pesquisa; iii) Breve Fundamentação Teórica, aborda o TPACK e a integração de tecnologias ao processo de ensino; iv) Análise dos Dados produzidos e v) Considerações, com algumas conjecturas de fim de texto.

2. Encaminhamentos metodológicos

Este estudo, pautado em uma perspectiva de pesquisa narrativa, analisou narrativas de duas participantes do curso de formação continuada em robótica educacional intitulado “Robótica Educacional: Arduino e suas Funcionalidades” (REAF); bem como as estratégias utilizadas para construção do projeto final. O curso foi oferecido por um dos Núcleos de Tecnologia Educacional do Distrito Federal (NTE/DF), na modalidade virtual e teve uma carga horária de 120 horas.

Ressaltamos que o curso foi desenvolvido em 2021, no contexto da pandemia da Covid 19 e, por esta razão, a maioria das atividades foi realizada de modo online. Durante o curso, os participantes tiveram atividades no Ambiente Virtual de Aprendizagem - AVA Moodle, que incluíram fóruns de discussão e simulação de experimentos no Tinkercad. Além disso, tiveram a oportunidade de colocar em prática os experimentos, utilizando as placas Arduino e *Robot Linking*, na elaboração de um projeto final. Para cada encontro, preencheram um Diário de Bordo, relatando como foi a experiência na realização de cada atividade.

Compreendemos que o uso de projetos para ensinar matemática revela cenários ou situações que ampliam as possibilidades de investigação, como sugerido por Skovsmose (2000). Nesse sentido, o espaço dado aos cursistas para desenvolverem seus projetos foram chamados de cenários durante a pesquisa realizada.

Neste artigo, a análise é conduzida de forma narrativa, com foco nas descobertas compartilhadas por duas professoras participantes da pesquisa, por meio de seus diários e na elaboração dos projetos finais. A escolha das duas participantes se deu pelo fato terem respondido todos os instrumentos de construção dos dados durante a pesquisa. A primeira delas é a professora cursista Bixby, que nunca teve experiência prévia com a robótica educacional. A segunda é a Professora Formadora, que não estava familiarizada com a placa *RobotLinking*, no entanto, após oferecer suporte à sua cursista Bixby, sentiu-se inspirada a desenvolver seu próprio projeto.

3. Tecnologias digitais utilizadas no desenvolvimento da pesquisa

O presente texto aborda um recorte específico da pesquisa, que envolveu a utilização de duas tecnologias distintas para a construção das atividades desenvolvidas pelas professoras participantes. Essas tecnologias foram introduzidas aos docentes por meio de encontros remotos e videoaulas. A seguir, detalhamos a origem dessas tecnologias e como elas interagem.

3.1 A placa *MakeyMakey*

A placa utilizada para a realização de uma das atividades do curso foi a *RobotLinking*, construída a partir da placa *MakeyMakey* que, embora apresente a mesma estrutura, é pouco conhecida no Brasil. A *MakeyMakey* foi idealizada por integrantes do MIT Media Lab, Jay Silver e Eric Rosenbaum, em parceria com a empresa *Sparkfun*, e objetivou tornar o uso da eletrônica acessível a todas as pessoas, incluindo as que não possuem conhecimento em programação. Com essa placa é possível transformar objetos do mundo físico em teclas, de modo mais fácil, por meio de seus conectores do tipo “jacarés”¹ (Figura 1).

¹ Fios de conexão que possuem em suas extremidades garras no formato de uma boca de jacaré. Esses cabos são apelidados de “jacarés” devido à sua semelhança com a boca do réptil, com dentes que se abrem e fecham para agarrar e fixar os pontos de conexão.

Figura 1 – Placa Makey Makey com conectores jacarés



Fonte: Site do MakeyMakey²

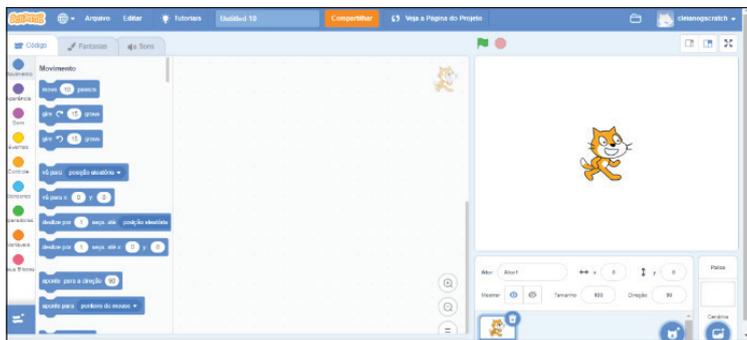
O nome *MakeyMakey* se origina de Make + Key = MaKey, ou seja, Fazer + Tecla. Na parte frontal da placa encontramos imagens de setas com as conexões para as teclas “direita”, “esquerda”, “cima” e “baixo”, e dois círculos para as conexões com as teclas “space” e o “click”. Ao conectarmos estes espaços a objetos condutivos do mundo real, transferimos a função das teclas do computador para os respectivos objetos.

3.2 O Scratch como ferramenta de interação com o MakeyMakey

A placa *MakeyMakey* já tem uma programação interna que possibilita a sua interação com o mundo físico e para desenvolver projetos mais complexos será necessário dizer o que as setas e portas deverão fazer. Desse modo, o *Scratch* surge enquanto ambiente de programação em blocos, capaz de interagir com o *MakeyMakey* (Figura 2).

² Site do *MakeyMakey*. Disponível em: https://cdn.shopify.com/s/files/1/0162/8612/files/image1_1800x.jpg?v=1548871772. Acesso em: 5 mai. 2024.

Figura 2 – Tela inicial do Scratch 3.0 disponibilizado em 2019



Fonte: <https://scratch.mit.edu/projects/329459328/editor>

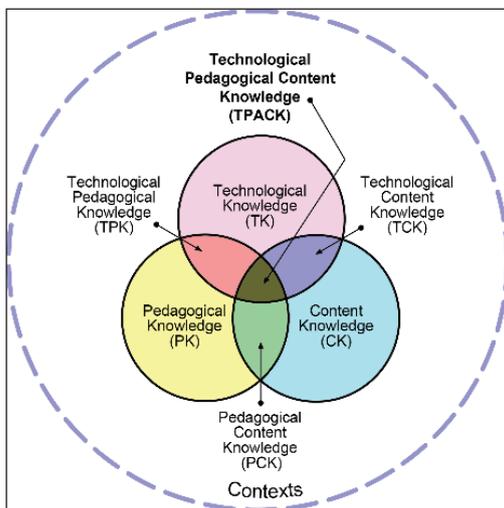
A linguagem foi criada em 2007, pelo Media Lab do MIT, idealizada pelo professor e pesquisador Mitchel Resnick no projeto do grupo *Lifelong Kindergarten*, que traduzido significa “Jardim de infância para toda vida”. O *Scratch*, segundo Resnick et al. (2009), foi desenvolvido mantendo os princípios da teoria Construcionista de Papert, que norteou a criação da Linguagem de Programação Logo em 1968. O Construcionismo enfatiza a importância do estudante aprender-fazendo e refletindo sobre o seu fazer em ensinar o computador (Papert; Harel, 1991).

4. O Tpack e a integração de tecnologias ao processo de ensino

O uso de tecnologias digitais faz parte do cotidiano e muitas escolas buscam incorporá-lo em sua prática pedagógica; no entanto, para que isso ocorra, o professor assume papel importante, o que se exige conhecimentos específicos. Dentre estes conhecimentos, destacamos o Technological Pedagogical Content Knowledge (TPACK) ou Conhecimento Tecnológico Pedagógico do Conteúdo (TPACK), um conjunto de conhecimentos teóricos que auxiliam os professores na compreensão das especificidades que permeiam o uso das tecnologias no ensino (Prado; Lobo da Costa, 2016; Mishra; Koehler, 2006 e Cibotto; Oliveira, 2017).

O TPACK é composto por três conhecimentos principais: conhecimento do conteúdo (CK), conhecimento pedagógico (PK) e conhecimento tecnológico (TK) (Figura 3). No entanto, cabe-nos evidenciar as suas intersecções, como o Conhecimento Pedagógico de Conteúdo (PCK), o Conhecimento Tecnológico de Conteúdo (TCK) e o Conhecimento Tecnológico de Pedagogia (TPK); sendo que são elas que representam a compreensão integrada e interdependente dos conhecimentos necessários para o ensino com o uso das tecnologias (Mishra; Koehler, 2006).

Figura 3 – Representação em diagrama do conceito TPACK



Fonte: Reproduzido com permissão do editor, © 2012 por tpack.org

O PCK refere-se à habilidade do professor de adaptar o conteúdo ao contexto de ensino, utilizando estratégias pedagógicas variadas para promover a aprendizagem dos estudantes (Shulman, 1987). O TCK envolve a capacidade de utilizar as tecnologias para representar os conteúdos de maneiras diversas, facilitando a compreensão dos estudantes e permitindo a seleção apropriada das ferramentas tecnológicas (Koehler; Mishra, 2009). Já o TPK diz

respeito à compreensão de como as tecnologias podem impactar o processo de ensino e aprendizagem, exigindo uma integração adequada entre a tecnologia e a pedagogia (Mishra; Koehler, 2006) e o TPACK se constitui pela integração dessas três interseções PCK, TCK e TPK.

Dessa forma, esse modelo nos evidencia que o ensino é uma atividade complexa, que ocorre em um ambiente dinâmico e pouco estruturado; porém, ressalta-se a importância de não considerar os conhecimentos de forma isolada, mas compreendê-los em suas interrelações, a fim de melhorar o processo educacional (Mishra; Koehler, 2006). Essas reflexões teóricas sobre o TPACK fornecem uma base para a análise dos dados produzidos durante a pesquisa, permitindo relacionar a teoria com a prática e investigar a efetividade do uso das tecnologias na sala de aula.

5. Análise das narrativa e dos projetos construídos

Neste espaço, realizaremos a análise dos dois projetos desenvolvidos pelas professoras participantes da pesquisa. Denominamos esses momentos de “cenários”, considerando-os espaços de exploração da atividade, nos quais o participante se torna protagonista de seu processo de aprendizagem, com o formador atuando como orientador.

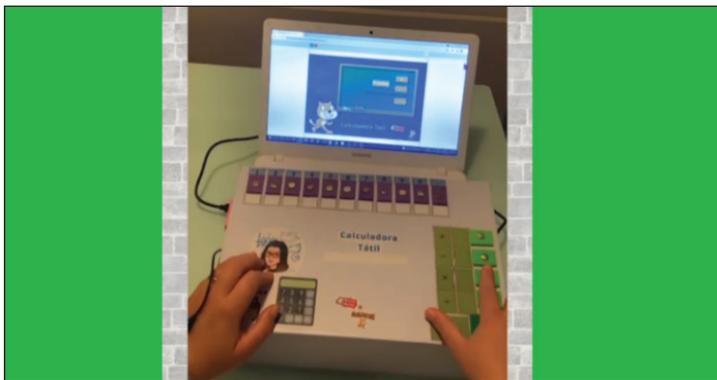
5.1. Cenário 1 – Projeto Calculadora Tátil (Bixby)

O objetivo principal do projeto de Bixby, “Calculadora Tátil”³ (Figura 4), foi “Construir uma calculadora que trabalhasse as quatro operações básicas (+, -, x e ÷), acessível para todos os estudantes, em especial, para pessoas cegas ou com deficiência visual” - Relatório, Bixby (Nogueira, 2021, p. 176).

Uma vez que a Base Nacional Comum Curricular (BNCC) orienta que os estudantes “desenvolvam diferentes estratégias para

³ Vídeo do projeto final da professora cursista Bixby – Calculadora Tátil. Disponível em: https://youtu.be/C_qRnFtokBw. Acesso em: 5 mai. 2024.

Figura 4 – “Calculadora Tátil” construída por Bixby



Fonte: Nogueira (2021, p. 177)

a obtenção dos resultados, sobretudo por estimativa e cálculo mental, além de algoritmos e uso de calculadoras” (Brasil, 2018, p. 268), vimos que o projeto de Bixby, com o suporte de uma Tecnologia Assistiva, atenderia esta orientação e auxiliaria estudantes com deficiência para um melhor entendimento do que está sendo executado em seu processo de aprendizagem (Geller; Sganzerla, 2014).

Para começar o projeto, Bixby e sua Professora Formadora, dividiram o tempo em três frentes, sendo elas: (1) Definir o formato da calculadora, de modo a atender os estudantes, em especial, os cegos. (2) Explorar a programação escolhida para a calculadora. (3) Realizar os primeiros testes da placa *RobotLinking*, interagindo com a programação.

Com o formato da calculadora definido (escolhido por Bixby como retangular e com números organizados de modo linear), Bixby e a Professora Formadora começaram a estudar a programação (Frente 2), registrando suas descobertas.

No site do *MakeyMakey* existe um *link* para a programação de uma calculadora tátil⁴, mas o utilizado por Bixby foi outro⁵, que apre-

⁴ Código da calculadora tátil no site Makey Makey. Disponível em: <https://scratch.mit.edu/projects/471598668/editor/>. Acesso em 5 mai. 2024.

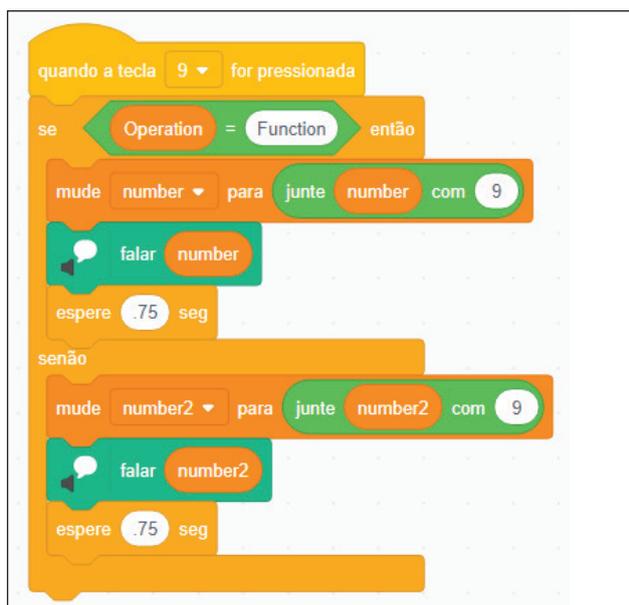
⁵ Código utilizado por Bixby para construção de seu projeto final. Disponível em: <https://scratch.mit.edu/projects/334824249>. Acesso em 5 mai. 2024.

sentou uma estrutura mais fácil de compreensão para um primeiro contato com programação. Essa análise foi realizada por Bixby e pela Professora Formadora, quando da escolha da programação.

Como a programação escolhida fazia parte de outro projeto, fizeram o remix⁶ no *Scratch*, adaptando-o para a Calculadora Tátil de Bixby. Ressaltamos que o código escolhido já existia na internet, mas, associar o código à placa *RobotLinking* tornou-se uma novidade para elas.

Durante a análise da programação do projeto, a dupla identificou que existia um bloco de comando para cada número da calculadora a ser acionado (Figura 5).

Figura 5 – Bloco de programação para o número 9



Fonte: *Scratch - Talking MakeyMakey Cardboard Calculator*

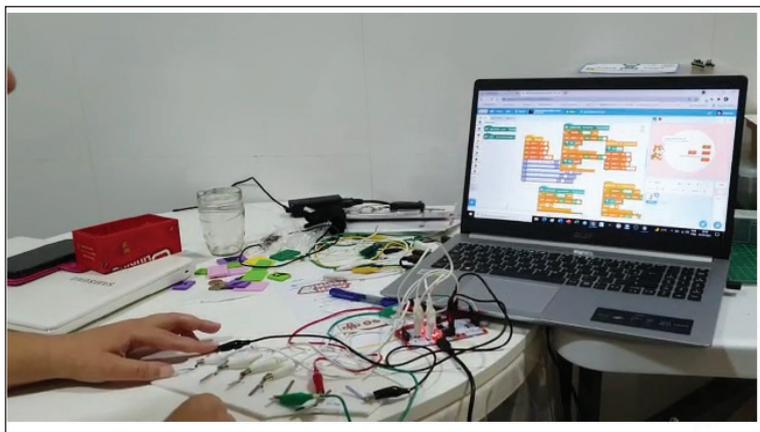
⁶ Remix é uma versão alterada de um projeto existente. Os usuários podem modificar, adicionar recursos ou personalizar projetos de outros, mantendo uma conexão com o projeto original na comunidade Scratch.

Examinando o bloco responsável pelo acionamento da tecla nove, entenderam que ao clicar no número nove, o programa entrava em uma condicional (SE) que o encaminhava para a variável NUMBER, se ele for o primeiro número a ser clicado, ou (SENÃO), para a variável NUMBER2, se for o segundo número a ser clicado. Isso acontecia com os demais blocos também, pois eram todos iguais. Desse modo, qualquer número ao ser clicado no teclado do computador ou da calculadora, seria reconhecido pelo programa.

Em seguida, a dupla observou que não existia o bloco para inserção do número **zero** e nem para o **ponto**, caso o estudante quisesse fazer uma operação com números decimais, notando também, que alguns blocos não eram completos em relação a ter o bloco **Falar** (alguns tinham e outros não). Mas não era possível, apenas olhando para a programação, saber de fato como a ausência dos blocos interferiria na calculadora.

Com uma análise geral da programação, partiram para a terceira frente, que era testar a calculadora com alguns números e operações, confirmando se a programação interagia com a placa (Figura 6).

Figura 6 – Bixby iniciando o projeto Calculadora Tátil



Fonte: Nogueira (2021, p. 182)

Fizeram, então, as conexões iniciais dos jacarés com as portas w, a, s, d, f e g, com os três números (4, 5 e 6), a operação de (+), o (=) e a opção para limpar a operação. Com as conexões prontas, ligaram a placa e fizeram o teste. Como tudo funcionou como o esperado, conectaram as demais portas da placa, incluindo os números de 0 a 9, as operações e o .(ponto). Ressaltamos que a dupla precisou buscar conhecimento extra para ativar todas as portas da placa, mas não detalharemos esta busca neste artigo, por não ter sido um conhecimento que iria interferir diretamente na programação da calculadora.

Analisando os conhecimentos necessários para esta atividade, percebemos que o conhecimento matemático (conteúdo) da dupla foi suficiente e necessário para compreensão da programação. No início, apesar de menos complexa do que a original, foi difícil identificar a matemática por trás da programação. Somente ao realizarem as operações, conseguiram visualizar o que estava acontecendo na prática. Foi nesse momento que o conhecimento tecnológico (da montagem e programação) tomou significado para a dupla. Ao compreender a relação entre o conteúdo matemático e a programação, associada às conexões feitas na calculadora, Bixby demonstrou ter desenvolvido o *Conhecimento Tecnológico do Conteúdo* (TCK), representando o conhecimento matemático por meio da programação. A compreensão do uso da calculadora e da metodologia utilizada com os estudantes, por sua vez, ancora-se enquanto *Conhecimento Pedagógico da Tecnologia* (TPK). Com este conhecimento, foi possível para Bixby compreender como as tecnologias (e neste caso específico, a calculadora) podem mudar o processo de ensino e de aprendizagem, não só trazendo novas possibilidades, como também, dificultando este processo, caso não sejam utilizadas de forma adequada (Mishra; Koehler, 2006).

No momento da montagem Bixby ressaltou que “Conhecer a realidade dos alunos com deficiência visual me ajudou a tornar o designer da calculadora mais acessível” - Relatório, Bixby (Nogueira, p. 185, 2021). Entendemos que os conhecimentos que fo-

ram trazidos por Bixby foram além do conhecimento de conteúdo e tecnológico. Segundo Almeida e Valente (2011), este é o papel da formação de professores que, além de prover ao docente o conhecimento tecnológico, precisa criar condições para que ele construa conhecimentos, compreendendo as perspectivas educacionais, quanto ao ensino, aprendizagem, integração ao currículo e outras. No caso de Bixby, temos ainda os conhecimentos necessários aos processos de inclusão de estudantes com deficiência.

Não tive grandes dificuldades durante o processo de construção da calculadora. Apesar de não ter tido nenhuma experiência anterior, semelhante a esta, me senti muito feliz em realizar de forma tão acessível, gostei das reflexões que tive que fazer para pensar no que tornaria a calculadora mais acessível possível, foi desafiador, mas muito gostoso - Relatório, Bixby (Nogueira, 2021, p. 186).

Como formadora docente para o uso de tecnologias eu já havia tido a experiência de dizer ao computador o que ele deveria fazer, como por exemplo, no uso da Linguagem Logo. Mas acompanhar Bixby e ser sua parceira de trabalho me deu a dimensão de formadora e de colega de dupla, que descobrem juntas e ressignificam seus conhecimentos (Nogueira, 2021, p. 186).

De acordo com a narrativa de Bixby, observamos que ela identifica habilidades que podem ser desenvolvidas a partir da programação em blocos e, também, a partir do uso dela, com a realização das operações. Segundo Prado *et al.* (2020, p. 205), a atividade de programação tem um papel importante quando o “[...] estudante, enquanto ensina o computador a resolver o problema, colocar em ação, compreender na ação e refletir sobre as propriedades e a construção de fórmulas e regras matemática de forma ativa e criativa” e ainda ressaltam que “[...] atividade de programação desenvolvida na educação básica precisa estar pautada nos princípios construcionistas, os quais concebem o aluno como participante ativo e o professor como mediador nos processos de ensino e aprendizagem”.

Apresentaremos na próxima seção, o Cenário 2, com o projeto construído pela Professora Formadora, utilizando a placa *RobotLinking* e a linguagem de programação *Scratch*, bem como análise e reflexões da construção realizada.

5.2 Cenário 2 – Projeto Conversor de Medidas (Professora)

Durante a pesquisa, Professora Formadora e a cursista construíram um diário de bordo com relatos de todas as atividades realizadas durante o curso. Além do diário, a Professora Formadora se sentiu motivada a participar da construção do Projeto Final, por também ser o seu primeiro contato com a placa *RobotLinking*.

Devido ao contexto pandêmico, a professora formadora do curso tornou-se dupla de trabalho de seus cursistas, uma vez que o isolamento social não permitia encontro presenciais entre os integrantes da turma e, naquele momento, foi muito complicado montar as duplas para realização da tarefa.

Desse modo, vendo Bixby pensando em seu projeto, a professora formadora se sentiu desafiada a construir seu projeto final.

Depois de muitas pesquisas, encontrei o projeto Conversor de Medidas⁷ e gostei muito da proposta. Agora, o meu desafio era transformar um projeto todo elaborado para o computador, em um projeto que interagisse com a placa *RobotLinking*. Realmente eu não tinha certeza se era possível, mas resolvi tentar! Talvez teria que fazer muitas alterações na programação, e não sabia se o pouco conhecimento na linguagem *Scratch* seria um obstáculo para a construção (Nogueira, 2021, p. 198).

O objetivo principal do projeto “Conversor de Medidas”⁸ (Figura 7) foi “Construir uma máquina que possibilite converter medidas inteiras entre as unidades de comprimento, interagindo com a placa *Robot Linking* e o computador” - Cleia (Nogueira, 2021, p. 198).

⁷ Conversor de unidades de longitud (m). Disponível em: <https://scratch.mit.edu/projects/527525484>. Acesso em: 5 mai. 2024.

⁸ Conversor de medidas construído pela professora formadora. Disponível em: <https://youtu.be/FgeNaM9Lzlw>. Acesso em: 5 mai. 2024.

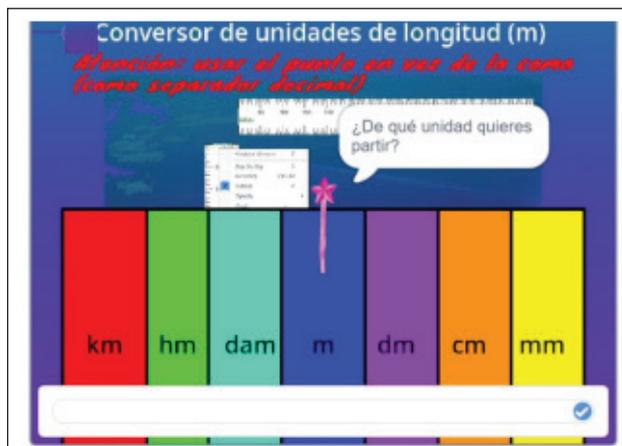
Figura 7 – Projeto “Convertor de Medidas” da Professora Formadora



Fonte: Nogueira (2021, p. 198)

Para iniciar o projeto, a Professora Formadora estudou a programação do código original, *Convertor de Unidades de Longitud (m)*, e logo percebeu que o código foi elaborado para ser executado apenas no computador, conforme observado na (Figura 8).

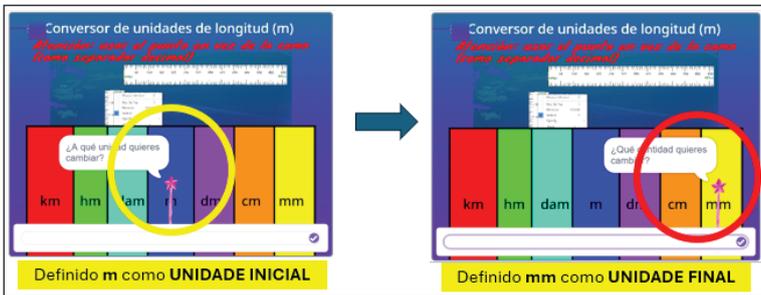
Figura 8 – Tela inicial do Convertor de Unidades de Longitud (m)



Fonte: Print da tela inicial do projeto no *Scratch* no computador (com adaptação)

Ao executar o programa, ele solicita de imediato que o usuário digite a **unidade inicial** (Posição Inicial) e, logo em seguida, a **unidade final** (Posição Final) para qual o valor deva ser convertido. Para finalizar, o usuário deve digitar o **valor a ser convertido**. Toda a programação que recebe as unidades inicial e final é realizada a partir do objeto “Varinha Mágica”. Ao digitar a **unidade inicial** e **unidade final**, o objeto “Varinha Mágica” se move para as coordenadas informadas na programação. Como por exemplo: definida a **unidade inicial** igual a **m** (metro) a Varinha Mágica se move para a coordenada (0, -100) e para **unidade final** igual a **mm** (milímetro) a Varinha Mágica se move para a coordenada (180, -100) conforme Figura 9.

Figura 9 – Movimentação do objeto “Varinha Mágica”



Fonte: Arquivo dos autores

A partir deste objeto (a Varinha Mágica), toda a programação é construída em um único bloco, recebendo as unidades de conversão (inicial e final) e o valor a ser convertido, de modo a executar o cálculo e informar o valor final convertido.

O bloco geral é composto de três partes principais, sendo ela: a primeira, para receber a **unidade inicial**, o segundo, para

9 Objetos no Scratch são elementos visuais programáveis, como personagens ou cenários. Eles executam ações específicas conforme instruções de código. Os usuários controlam seu movimento, tamanho e interações. No caso do Conversor de Medidas o objeto é uma varinha rosa, que serve de marcador da unidade selecionada.

receber a **unidade final** e o último, para receber o **valor a ser convertido**. Em seguida, o programa realiza a conversão.

Para finalizar, o usuário insere o **valor a ser convertido**. A partir desse momento, o programa passa por três condicionais, sendo elas: 1) **Se** a posição do **x**, da coordenada da **unidade inicial** e coordenada da **unidade final** forem iguais ($PI = PF$), o resultado do valor a ser convertido será ele mesmo 2). **Se** a posição do **x**, da coordenada da **unidade inicial** for menor que a posição do **x**, da coordenada da **unidade final** ($PI < PF$), o resultado do valor a ser convertido será igual a: **valor a ser convertido** * 10 elevado ao módulo da [posição **x** (INICIAL) – posição **x** (FINAL)] / 60 e, 3). **Se** a posição do **x**, da coordenada da **unidade inicial** for maior que a posição do **x**, da coordenada da **unidade final** ($PI > PF$), o resultado do valor a ser convertido será igual a: **valor a ser convertido** / 10 elevado ao módulo da [posição **x** (INICIAL) – posição **x** (FINAL)] / 60.

Vejamos um exemplo: Posição Inicial = m (metro), Posição Final = mm (milímetros) e Valor a ser convertido = 8. Pela programação, a unidade metro está na coordenada (0, -100) e a unidade milímetros está na coordenada (180, -100).

É possível observar que o **x** da **Posição Inicial** é menor do que o **x** da **Posição Final**, ou seja, $0 < 180$. Desse modo, aplicando a fórmula presente na 3ª condicional teremos que o valor convertido é igual a: **Valor a ser convertido** * [10 elevado ao módulo de (**x** da **Posição Inicial** – **x** da **Posição final**) / 60]. No caso do nosso exemplo, teremos então, $8 * 10^3 = 8000\text{mm}$.

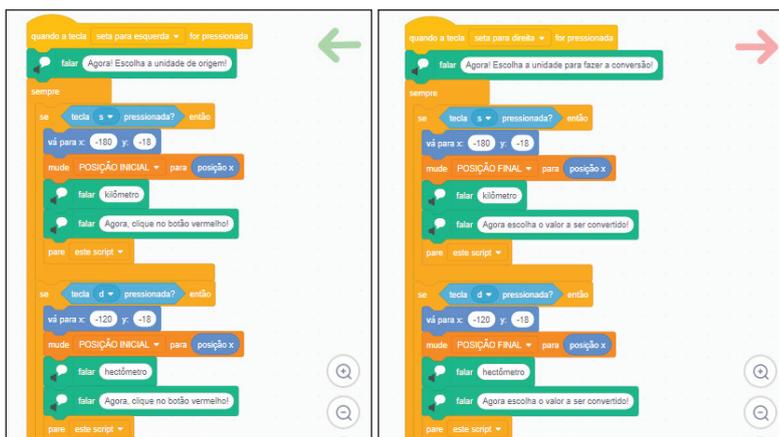
Compreendida a parte da programação, a Professora Formadora começa a pensar em como colocar este código para interagir com a placa *RobotLinking*. Para iniciar, era preciso partir da programação onde, o que inicialmente era digitado, agora seria informado por meio do toque nos colchetes (as unidades e os valores) conectados as portas da placa *RobotLinking*.

A Professora Formadora então dividiu o problema em partes menores, observando que, agora, seria necessário dividir os blocos

para enviar as informações para a placa *RobotLinking*. Foram criados dois novos objetos (seta para esquerda e seta para direita) e associou os dois primeiros blocos a eles, respectivamente. Ou seja, para a “seta para a esquerda”, será realizada a programação responsável por receber os dados referentes a unidade inicial para conversão e, para a “seta para a direita”, será realizada a programação para os dados referentes a unidade final para conversão.

Detalhando melhor os dois blocos (Figura 10), foi criado um objeto dentro do *Scratch* (seta para esquerda) e associado o primeiro bloco de programação a ele. Assim, ao tocar no objeto “seta para esquerda”, a placa *RobotLinking* entenderia que, em seguida, a informação recebida seria da **unidade inicial**. Na sequência, criou-se o objeto (seta para direita) e associou o segundo bloco de programação a ele. Assim, ao ser tocado, a placa *RobotLinking* entenderia que, em seguida, a informação recebida seria da **unidade final**.

Figura 10 – Blocos iniciais do Projeto Conversor de Medidas



Fonte: Nogueira (2021, p. 202)

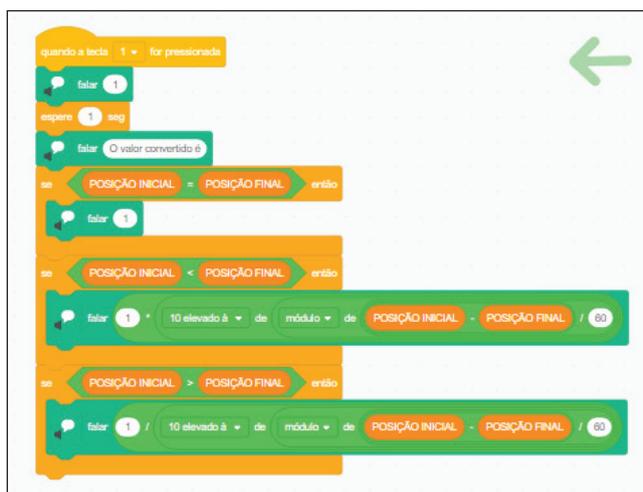
Dentro de cada bloco, após tocar na “seta para esquerda”, o usuário deveria escolher a unidade inicial e, em seguida, clicaria

na “seta para direita” para escolher a unidade final. Desse modo, as seis unidades foram definidas nos dois blocos.

Ao chegar neste momento da programação, a Professora Formadora sente falta do bloco responsável em fazer o cálculo do valor convertido e, depois de vários testes, percebe que ao tocar no valor a ser convertido, precisaria associar um bloco de conversão para cada valor informado, ou seja, clicou no número 1, a programação vai buscar a posição x da unidade inicial, em seguida, a posição x da unidade final (Figura 11).

Com as duas posições definidas, a programação realiza as três condicionais que vimos anteriormente: $PI = PF$, $PI > PF$ e $PI < PF$.

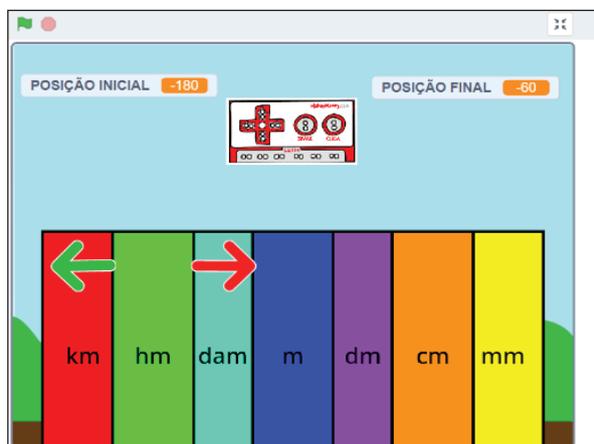
Figura 11 – Bloco de programação para cálculo do valor a ser convertido



Fonte: Nogueira (2021, p. 203)

Com a programação quase finalizada, ou seja, máquina física pronta para enviar os valores e o *Scratch* pronto para informar o valor a ser convertido, a Professora Formadora faz alguns ajustes no projeto, incluindo na tela do *Scratch* os objetos (Seta para esquerda e Seta para direita), no lugar da “Varinha Mágica”, que se movimentam de acordo com as unidades selecionadas, além de deixar visível na tela a posição **x** da unidade escolhida (Figura 12).

Figura 12 – Tela inicial do Projeto Conversor de Medidas



Fonte: Nogueira (2021, p. 204)

Com o algoritmo finalizado, conectou cada porta da placa *RobotLinking* as setas (para esquerda e para direita), as unidades de medida (km, hm, dam, m, dm, cm, mm) e aos números (1, 2, 3, 4, 5, 6, 7, 8, e 9), totalizando 18 portas a serem utilizadas (Figura 13).

Figura 13 – Imagem interna e externa do Projeto Conversor de Medidas



Fonte: Nogueira (2021, p. 204)

Dentro deste cenário, é possível observar que além de trabalhar os conceitos matemáticos relacionados à conversão de unidades de medidas, a Professora Formadora partiu de um projeto pronto e o transformou em um novo projeto, que atenderia outras necessidades, interagindo com uma máquina construída por ela

mesma. Com certeza, revisou o conteúdo sobre conversão de medidas, em um cenário provocativo de investigação. Neste contexto, desenvolveu o projeto em busca de resolver o problema inicial, que era transformar um código que programado para o computador pudesse interagir com uma “máquina de papelão”, com suporte da placa *RobotLinking*. Segundo Skovsmose (2000, n. p.), “o trabalho de projeto está localizado em um ambiente de aprendizagem que difere do paradigma do exercício. É um ambiente que oferece recursos para fazer investigações”. Com certeza, a Professora Formadora já tinha conhecimento do conteúdo matemático, mas no momento de identificá-lo na programação, não foi tão simples como imaginava.

Em um primeiro momento, eu não conseguia entender a necessidade das coordenadas cartesianas, para que a máquina realizasse a conversão. Só depois de estudar e executar a programação várias vezes, percebi que no código original o autor utilizava as coordenadas (em especial a posição do x) para poder informar a placa a posição da unidade a ser convertida. Quando fazemos essa conversão manualmente, precisamos saber se a unidade a ser transformada está a direita ou esquerda e, também, quantas casas de “distância” da unidade a ser convertida. Na programação, significa dizer se ela é maior ou menor do que a outra. Estes dados vão informar ao código se ele vai multiplicar ou dividir o valor indicado para a conversão - Cleia (Nogueira, 2021, p. 205).

Na execução do projeto, a Professora Formadora não identificou dificuldades em relação ao conteúdo matemático, porém, o conhecimento tecnológico foi um obstáculo no início, mas superado, no próprio ambiente do *Scratch*. Quando não sabia como fazer algo, buscava respostas visitando outros projetos e procurava compreender como funcionavam. A construção da programação, associando conteúdo e tecnologia, deu suporte ao Conhecimento Tecnológico de Conteúdo (TCK). Segundo Koehler e Mishra (2009) e Rocha, Prado e Valente (2020), a ação da Professora Formadora foi representar o conteúdo matemático de outro modo, viabilizado por recursos tecnológicos, como a linguagem em blocos *Scratch*.

A partir dos conhecimentos necessários à docência por Shulman (1986), é possível observar neste momento de execução do projeto, a Professora Formadora utilizou o *Conhecimento Pedagógico de Conteúdo* (PCK), melhorando a programação, permitindo identificar o que precisava acrescentar no código e, também, a ordem de execução dos blocos, tal como: Quero fazer uma conversão? Quem vem primeiro? O que a “máquina” fará com o valor recebido? Para entender o código, o PCK favoreceu a interpretação do código inicial de modo a ser utilizado corretamente no projeto.

Tenho afinidade com a linguagem de programação em texto. Pretendo no futuro me aprofundar e estudá-la mais. Mas, durante a pandemia, senti-me mais segura para trabalhar com os blocos no Scratch. Mesmo com conhecimento básico na programação em blocos, a organização dos blocos no Scratch me ajudou a pensar em estratégias para que o projeto funcionasse como eu estava imaginando. Perceber o conteúdo matemático na programação só dependeu de parar e estudar. E para associar o Scratch as portas da placa, dependeu de pesquisas e testes - Cleia (Nogueira, 2021, p. 206).

Por meio do *Conhecimento Pedagógico da Tecnologia* (TPK) a Professora Formadora pode mergulhar na programação do projeto e, mesmo com pouco conhecimento da linguagem *Scratch*, teve uma noção do que era possível fazer ou não com os blocos e, também, com a placa *RobotLinking*. Segundo Koehler e Mishra (2009) e Rocha, Prado e Valente (2020), a partir do TPK é possível identificar as potencialidades das tecnologias a nossa disposição, tanto para promover o processo de ensino como o de aprendizagem.

Considerações finais

Este artigo, que toma como base a pesquisa de doutorado da primeira autora, busca refletir sobre os conhecimentos necessários à docência para se trabalhar com robótica educacional no ensino de matemática. Diante da análise das narrativas dos textos produzidos por duas participantes da pesquisa, sendo uma delas a professora formadora, entendemos a importância do modelo TPACK na formação de professores, especialmente no contexto do ensino

de matemática através da robótica educacional. A aplicação desse modelo em projetos como a “Calculadora Tátil” e o “Conversor de Medidas” mostra como a integração de tecnologia, pedagogia e conteúdo pode transformar práticas educacionais e promover uma aprendizagem com mais significado, tanto para quem ensina quanto, conseqüentemente, para quem aprende.

A experiência de Bixby com a “Calculadora Tátil” exemplifica a relevância de adaptar tecnologias para atender às necessidades de todos os estudantes, especialmente aqueles com deficiência visual. O trabalho colaborativo com sua professora formadora foi essencial para superar os desafios técnicos e pedagógicos, demonstrando a eficácia do TPACK em criar soluções educacionais capazes de atender às necessidades dos estudantes de Bixby. Esse processo evidenciou como o Conhecimento Tecnológico do Conteúdo (TCK) e o Conhecimento Pedagógico da Tecnologia (TPK) podem ser aplicados na prática para melhorar os processos de ensino e de aprendizagem.

A criação do “Conversor de Medidas” pela professora formadora ilustra como a inovação e a adaptação de tecnologias existentes podem resolver problemas práticos e educativos, mesmo em situações desafiadoras. A transformação de um código existente para interagir com novas ferramentas tecnológicas destaca a flexibilidade e a criatividade necessárias no ambiente educacional moderno, reforçando a importância do TPACK na compreensão do projeto desenvolvido.

Esses projetos sublinham a necessidade de formação adequada para os educadores, capacitando-os a utilizar e integrar tecnologias de maneira eficaz. O desenvolvimento de conhecimentos tecnológicos, aliado à compreensão pedagógica e de conteúdo, é essencial para criar ambientes de aprendizagem inclusivos, interativos e que promovam o desenvolvimento integral dos estudantes.

Finalizando, o curso de formação de professores para o uso da robótica educacional, fundamentado no modelo TPACK, demonstrou que é possível construir projetos educacionais que facilitam o aprendizado e garantem a inclusão de todos os estudantes,

integrando tecnologia e pedagogia para o avanço de uma educação matemática mais equitativa e eficiente, promovendo uma educação que seja ao mesmo tempo inovadora e acessível.

Referências

ALMEIDA, M. E. B.; VALENTE, J. A. **Tecnologias e currículo: trajetórias convergentes ou divergentes?** São Paulo: Paulus, 2011.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular (BNCC)**. 2018. Disponível em: <http://basenacionalcomum.mec.gov.br/>. Acesso em: 9 jul. 2024.

CIBOTTO, R. A. G.; OLIVEIRA, R. M. M. A. TPACK - Conhecimento Tecnológico e Pedagógico do Conteúdo: uma revisão teórica. **Imagens da Educação**, v. 7, n. 2, p. 11-23, 2017.

RESNICK, M. et al. Scratch: programming for all. **Communications of the ACM**, New York, v. 52, n. 11, p. 60-67, 2009.

GELLER, M.; SGANZERLA, M. A. R. Reflexões de professores sobre Tecnologias Assistivas e o processo de ensino e aprendizagem de matemática. **Acta Scientiae**, Canoas, v. 16, n. 4, p. 116-137, Edição Especial, 2014. Disponível em: <http://www.periodicos.ulbra.br/index.php/acta/article/view/1275/1023>. Acesso em: 9 jul. 2024.

KOEHLER, M. J.; MISHRA, P. **What is technological pedagogical content knowledge? Contemporary Issues in Technology and Teacher Education**, 2009.

MISHRA, P.; KOEHLER, M. J. **Technological Pedagogical Content Knowledge: a framework for teacher knowledge**. Teachers College Record, 2006. Disponível em: <https://journals.sagepub.com/doi/10.1111/j.1467-9620.2006.00684.x>. Acesso em: 9 jul. 2024.

NOGUEIRA, Cleia Alves. **Narrativas de professores de matemática: experiências com aprendizagem criativa em um curso de robótica educativa**. 2021. 227 f. Tese (Doutorado em Educação) – Universidade de Brasília, Brasília, 2021.

PAPERT, S.; HAREL, I. **Constructionism**. Norwood: Ablex Publishing, 1991.

PRADO, M.E.B.B.; LOBO DA COSTA, N.M. O papel da atividade de programação no processo de construção de conhecimentos para a docência. **Revista**

e-Curriculum. Programa de Pós-graduação Educação: Currículo – PUC/SP, São Paulo, v.14, n.03, p. 898 – 918 jul./set.2016 Currículo – PUC/SP. Disponível em: <https://revistas.pucsp.br/index.php/curriculum/article/view/28696>. Acesso em: 9 jul. 2024.

PRADO, M. E. B. B.; GARCIA SILVA, A. F.; PIETROPAOLO, R. C.; SILVA, S. F. K. Pensamento computacional e atividade de programação: perspectivas para o ensino da matemática. **Revista Sergipana de Matemática e Educação Matemática.** ReviSeM, 2020, N° 2, p. 195 – 208. Disponível em: <https://periodicos.ufs.br/ReviSe/article/view/14422>. Acesso em: 9 jul. 2024.

RESNICK, M. et al. Scratch: programming for all. **Communications of the ACM**, New York, v. 52, n. 11, p. 60-67, 2009.

ROCHA, A. K. de O.; PRADO, M. E. B. B.; VALENTE, J. A. A linguagem de programação Scratch na formação do professor: uma abordagem baseada no TPA-CK. **Revista Sergipana de Matemática e Educação Matemática.** ReviSeM, 2020, N° 2, p. 19–36. Disponível em: <https://periodicos.ufs.br/ReviSe/article/view/14421>. Acesso em: 9 jul. 2024.

SHULMAN, L. S. Those who understand: knowledge growth in teaching. **Educational Researcher**, v. 15, n. 2, p. 4-14, 1986.

SHULMAN, L. S. Knowledge and teaching: foundations of the new reform. **Harvard Educational Review**, Cambridge, v. 57, p. 1-22, 1987.

SKOVSMOSE, O. Cenários para investigação. **Bolema**, Rio Claro – SP, v.13, n. 14, 2000. Disponível em: <https://www.periodicos.rc.biblioteca.unesp.br/index.php/bolema/issue/view/693>. Acesso em: 9 jul. 2024.

Dados dos autores

CLEIA ALVES NOGUEIRA

Doutora e Mestre em Educação, pela Universidade de Brasília (UnB). Especialista em Tecnologias na Educação pela PUC-RJ e em Arte, Educação e Tecnologias Contemporâneas pela UnB e graduada em Matemática (licenciatura) pela Universidade Católica de Brasília - UCB. Coordenadora e multiplicadora/formadora de Núcleos de Tecnologia Educacional (NTE) da Secretaria de Estado de Educação do DF. Integrante dos grupos de pesquisa: Grupo de Pesquisa e Investigações em Educação Matemática (PI) e Grupo de investigação em Ensino de Matemática (GIEM) da UnB. Orcid: <https://orcid.org/0000-0003-0983-2631>.

Email: cleianog@gmail.com.

CLEYTON HÉRCULES GONTIJO

Professor Associado IV na Universidade de Brasília (UnB), com lotação no Departamento de Matemática. Professor do Mestrado Profissional em Matemática em Rede Nacional - ProfMat e do Programa de Pós-Graduação em Educação da UnB. Líder do Grupo PI: Pesquisas e Investigações em Educação Matemática. Coordenador do subprojeto Matemática do Programa de Bolsas de Iniciação à Docência (PIBID) da UnB (2020 - 2024). Tem experiência na área de Matemática, com ênfase em Educação Matemática, atuando principalmente nos seguintes temas: criatividade em matemática e pensamento criativo em matemática e avaliação em matemática. Orcid: <https://orcid.org/0000-0001-6730-8243>. Email: cleyton@unb.br.

MARIA ELISABETTE BRISOLA BRITO PRADO

Doutora em Educação: Currículo pela Universidade Católica de São Paulo (PUCSP). Mestre em Educação na área de Psicologia Educacional pela Universidade Estadual de Campinas (UNICAMP), graduada em Licenciatura em Ciências e Matemática e Licenciatura em Pedagogia pela Faculdade de Educação (UNICAMP). Pesquisadora colaboradora do Núcleo de Informática Aplicada à Educação (NIED) da (UNICAMP). Docente e Pesquisadora do Programa de Pós-Graduação em Metodologias para o Ensino de Linguagens e suas Tecnologias da Universidade Pitágoras Unopar Anhanguera de Londrina, PR. Orcid: <https://orcid.org/0000-0002-8595-4203>. Email: bette.prado@gmail.com.

AMÉRICO JUNIOR NUNES DA SILVA⁴

Professor Adjunto do Departamento de Educação da Universidade do Estado da Bahia – (Uneb - Campus VII) e docente permanente do Programa de Pós-Graduação em Educação, Cultura e Territórios Semiáridos - PPGESA (Uneb - Campus III). Atualmente realiza estágio de pós-doutoramento no Departamento de Educação e Psicologia da Universidade de Aveiro (UA), Portugal. Doutor em Educação pela Universidade Federal de São Carlos (UFSCar) e Mestre em Educação pela Universidade de Brasília (UnB). Pesquisa na área de formação de professores que ensinam matemática, ludicidade e narrativas. Orcid: <https://orcid.org/0000-0002-7283-0367>. Email: ajnunes@unb.br.