



UNIVERSIDADE ESTADUAL PAULISTA

PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO MATEMÁTICA

ÁREA DE ENSINO E APRENDIZAGEM DA
MATEMÁTICA E SEUS FUNDAMENTOS FILOSÓFICO-CIENTÍFICOS

**Pensamento computacional, Scratch e Matemática: possíveis
relações**

PEDRO HENRIQUE GIRALDI DE SOUZA

INSTITUTO DE GEOCIÊNCIAS E CIÊNCIAS EXATAS

Rio Claro – SP
2021

Pedro Henrique Giraldi de Souza

Pensamento computacional, Scratch e Matemática: possíveis relações

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de Rio Claro, como parte dos requisitos para a obtenção do título de Mestre em Educação Matemática.

Orientadora: Prof. Dra. Sueli Liberatti Javaroni

Rio Claro – SP

2021

S729p

Souza, Pedro Henrique Giraldi de

Pensamento computacional, Scratch e Matemática : possíveis relações / Pedro Henrique Giraldi de Souza. -- Rio Claro, 2021
162 f.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp),
Instituto de Geociências e Ciências Exatas, Rio Claro

Orientadora: Sueli Liberatti Javaroni

1. Matemática (Ensino fundamental). 2. Matemática Estudo e ensino. 3. Tecnologia educacional. 4. Scratch. 5. Pensamento computacional. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Geociências e Ciências Exatas, Rio Claro. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Interessado(a): **PEDRO HENRIQUE GIRALDI DE SOUZA**

Programa de Pós-Graduação: **Educação Matemática**

Área de: **Ensino e Aprendizagem da Matemática e seus Fundamentos Filosófico-Científicos**

Assunto: **Encaminhamento da Dissertação de Mestrado**

DELIBERAÇÃO

Título: "Pensamento computacional, Scratch e Matemática: possíveis relações"

Comissão Examinadora

TITULARES

01. Prof(a). Dr(a). Sueli Liberatti Javaroni - Orientador(a)

FC/UNESP/Bauru (SP)

02. Prof(a). Dr(a). José Armando Valente

IA/UNICAMP/Campinas (SP)

03. Prof(a). Dr(a). Marcus Vinicius Maltempi

IGCE/UNESP/Rio Claro (SP)

SUPLENTES

01. Prof(a). Dr(a). Silvana Claudia dos Santos

UFV/Viçosa (MG)

02. Prof(a). Dr(a). Ricardo Scucuglia Rodrigues da Silva

IBILCE/UNESP/São José do Rio Preto (SP)

() Aprovação do Conselho do Programa em ____/____/____

(X) Aprovação "ad referendum" do Conselho do Programa em 13/09/2021



Prof^a Dr^a Heloisa da Silva
Coordenadora

Dedicado a Bruno Coury, o *Doug* (*in memoriam*).

AGRADECIMENTOS

Primeiro, gostaria de agradecer a cada pessoa que contribuiu direta ou indiretamente com a minha jornada, da Graduação até a conclusão do Mestrado.

À minha orientadora, Profa. Dra. Sueli Liberatti Javaroni, pelos ensinamentos, colaborações e paciência. Meu muito obrigado pelos momentos dedicados a mim e a essa pesquisa.

Aos meus irmãos e irmãs de orientação: Maitê, Eliel, Tiago, Daniel e Silvana. Nossas reuniões e suas contribuições foram muito importantes para a escrita dessa dissertação.

Ao professor Marcelo Borba, coordenador do GPIMEM (Grupo de Pesquisa em Informática, outras Mídias e Educação Matemática), pelo aprendizado não apenas nas reuniões do grupo, mas também nas disciplinas cursadas, que colaboraram com o meu desenvolvimento tanto pessoal quanto profissional, como professor e pesquisador.

Aos pesquisadores do GPIMEM, pela aprendizagem e acolhimento não só nas reuniões, mas também nas confraternizações.

A todos os amigos, colegas e funcionários do Programa de Pós-Graduação em Educação Matemática (PPGEM). Agradeço, em especial, à Jeimy, Liara, Kaoma, Daniela, Fran, Lara, Arthur, Jonson e Tiago.

Aos membros da banca, professores Marcus Vinicius Maltempo e José Armando Valente, agradeço pela leitura e pelas contribuições que me possibilitaram concluir esse trabalho.

A todas as pessoas da Escola Estadual Professora Carolina Augusta Seraphim, em especial, à Soráia Pessoa, diretora da instituição, por permitir o desenvolvimento do trabalho; à professora Ana Maria Soares, por assentir que suas aulas fossem palco para minha pesquisa de campo; e aos estudantes do nono ano do Ensino Fundamental de 2018, por participarem da pesquisa.

Aos queridos amigos e agregados das Repúblicas Magrelos, Atoas e 7 Anões, obrigado pelos momentos juntos, virtual ou pessoalmente. Um grande abraço a todos.

À Kit (Marília), Kaique, Bomba (Jhon), Maloca (Alécio), Preto (Gabriel) e Eduardo, obrigado pelos incríveis momentos e pela amizade que espero levar para a vida toda.

Aos velhos amigos de São Paulo, Lucas, Caio, Rafael e Bruna, obrigado pelas viradas de ano e outros momentos inesquecíveis. A companhia de vocês faz falta demais.

Ao meu pai, Otávio, e à sua esposa, Priscila, por me acolherem e ajudarem quando precisei.

Ao Pedro Dantas, irmão, obrigado pelos momentos que pudemos compartilhar nesses anos.

À Leandra dos Santos, pois sem você eu não teria finalizado essa dissertação. Não tenho palavras para dizer o quanto sou grato pela sua amizade, companhia e colaboração.

À minha mãe, por ser a pessoa com quem eu posso contar sempre. Também não tenho palavras para dizer o quanto sou grato pela senhora.

O presente trabalho foi realizado com apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – Código de Financiamento 001.

RESUMO

Esta dissertação apresenta os resultados de uma pesquisa cujo objetivo foi identificar os conceitos do pensamento computacional que emergiram de estudantes do nono ano do Ensino Fundamental ao desenvolverem atividades com conteúdo matemático no *Scratch*. Este estudo seguiu uma abordagem metodológica qualitativa, a partir da qual foram realizadas observações participantes na disciplina de Práticas de Matemática, gerando, por sua vez, cinco atividades com duas turmas do nono ano da Escola Estadual Professora Carolina Augusta Seraphim, pertencente ao Programa de Ensino Integral, na cidade de Rio Claro, interior paulista. Além disso, para a produção e análise dos dados, foi feito um diário com anotações escritas e gravações referentes aos acontecimentos no campo de pesquisa. A dinâmica das aulas em que tais atividades ocorreram era dividida em dois momentos: no primeiro, eu ensinava alguma funcionalidade do *software* por meio de uma tarefa, tentando relacionar com determinado conteúdo matemático; no segundo, a partir do conteúdo que havia sido ensinado, eles realizavam outra tarefa proposta, dedicando-se à manipulação e exploração do programa, com o intuito de generalizarem tais funcionalidades aprendidas para atividades futuras. A pesquisa também investigou e fundamentou as relações entre o pensamento computacional, a Matemática e o *Scratch*. A análise dos dados evidencia os conceitos, oriundos das relações dos temas, que emergiram dos estudantes ao realizarem tais atividades, além de mostrar as relações de tais temas nas atividades desenvolvidas por eles com o referido *software*.

Palavras-Chave: Anos Finais do Ensino Fundamental. Conceitos computacionais. Educação Básica. Educação Matemática. Tecnologias Digitais.

ABSTRACT

This dissertation presents a research that aimed to identify the concepts of computational thinking that emerged from ninth graders when developing activities with mathematical content in Scratch. This research followed a qualitative methodological approach, in which participant observations were carried out in the Mathematics Practices discipline, which generated five activities with two ninth-grade classes at the Professora Carolina Augusta Seraphim State School, which belongs to the Integral Education Program. In addition, a field diary was made with written notes and recordings referring to events in the research field for the production and analysis of data. The dynamics of the classes in which such activities occurred took place in two moments, in the first I taught some software functionality through a task, trying to relate some mathematical content, so that, in the second moment, from the content that had been taught, they performed another proposed task dedicating themselves to the manipulation and exploration of the software in order to generalize such features taught for future activities. The research also investigated the relationships between the themes computational thinking, Mathematics and Scratch. Data analysis highlights the concepts, arising from the relationships of the themes, that emerged from the students when carrying out such activities, in addition to showing the relationships of the themes in the activities they developed with Scratch.

Keywords: Final Years of Elementary School. Computational concepts. K-12 Education. Mathematical Education. Digital Technologies.

LISTA DE FIGURAS

Figura 1 – Cronologia dos acontecimentos da pesquisa	16
Figura 2 - Matriz Curricular dos Anos Finais do Ensino Fundamental.....	26
Figura 3 – Ator e respectiva programação com o uso do conceito de <i>sequência</i>	50
Figura 4 – Ator e respectiva programação com o uso do conceito de <i>ciclos</i>	51
Figura 5 - Programação com o uso do conceito de <i>eventos</i>	52
Figura 6 – Ator e respectiva programação com o uso do conceito de <i>paralelismo</i> ...	53
Figura 7 - Programação com o uso do conceito de <i>condicionais</i>	53
Figura 8 – Blocos da Paleta Operadores.....	54
Figura 9 – Captura de tela da interface de um projeto <i>on-line</i> do Scratch.....	56
Figura 10 - Representação do desenho da casa.....	68
Figura 11 - Programação para desenhar um traço, com 10 e 60 passos, no palco .	69
Figura 12 – Exemplo das tentativas de programação de grupos de estudantes da primeira situação	69
Figura 13 - Exemplo do equívoco de apertar a bandeira três vezes	70
Figura 14 - Desenho e programação dos estudantes da segunda característica.....	70
Figura 15 - Exemplo de desenho e programação da terceira característica	71
Figura 16 – Programação que pode considerada ambígua para traçar 150 passos	72
Figura 17 - Processo equivocado para se desenhar um traço de 150 passos	73
Figura 18 – Desenho e programação, elaborada por José e Caio, da casa <i>sem</i> o uso dos blocos “levante a caneta” e “use a caneta	75
Figura 19 - Desenho e programação, elaborada por José e Caio, da casa <i>com</i> o uso dos blocos “levante a caneta” e “use a caneta	77
Figura 20 - Desenho final da dupla José e Caio.....	79
Figura 21 – Programação da dupla José e Caio para desenhar a casa da Atividade 1	80
Figura 22 – Representação de um triângulo equilátero, um hexágono formado por seis triângulos equiláteros e uma “teia de aranha” formada por triângulos equiláteros de comprimento crescentes	83
Figura 23 - Programação da dupla João e Leonardo para o triângulo equilátero.....	85

Figura 24 – Desenho e procedimento “triângulo equilátero” da dupla João e Leonardo	87
Figura 25 – Desenho e procedimento “Hexágono regular” elaborado pela dupla João e Leonardo	89
Figura 26 - Procedimento “teia” realizado pela dupla João e Leonardo	89
Figura 27 - Teias A e B desenvolvidas com a programação de João e Leonardo....	90
Figura 28 – Área de programação da dupla João e Leonardo ao fim da Atividade 4	90
Figura 29 – Programação para calcular o Índice de Massa Corpórea.....	93
Figura 30 – A estrutura do bloco <i>if (se)</i>	95
Figura 31 – Programação para exemplificar o uso do bloco “se”	96
Figura 32 – A estrutura do bloco <i>if/else (se/senão)</i>	96
Figura 33 - Programação para exemplificar o uso do bloco “se/senão”	98
Figura 34 – Palco do programa que calcula as áreas de um retângulo, de um triângulo e de uma circunferência (parte 1)	99
Figura 35 - Palco do programa que calcula as áreas de um retângulo, de um triângulo e de uma circunferência (parte 2)	100
Figura 36 - Palco do programa que calcula as áreas de um retângulo, de um triângulo e de uma circunferência (parte 3)	101
Figura 37 - Palco do programa que calcula as áreas de um retângulo, de um triângulo e de uma circunferência (parte 4)	101
Figura 38 - Palco do programa que calcula as áreas de um retângulo, de um triângulo e de uma circunferência (parte 5)	102
Figura 39 – Programação inicial da segunda parte da Atividade 5 desenvolvida pela estudante Rhaylla.....	106
Figura 40 – Programação da estudante Rhaylla para calcular a área de um retângulo	107
Figura 41 – Diferença entre a montagem dos blocos para calcular a área de um triângulo	110
Figura 42 – Diferença de valores entre modos de compor os blocos operadores..	110
Figura 43 – Procedimento “TRIÂNGULO” desenvolvido pelo estudante Pedro	111
Figura 44 – Procedimento “triangulo” desenvolvido pela estudante Rhaylla	111
Figura 45 – Procedimento “circunferencia” realizado pela estudante Rhaylla	114

Figura 46 – Palco da segunda parte da Atividade 5 realizado pelo estudante Pedro	114
Figura 47 – Área de programação da segunda parte da Atividade 5 realizado pelo estudante Pedro	115
Figura 48 - Palco da segunda parte da Atividade 5 realizado pela estudante Rhaylla	115
Figura 49 - Área de programação da segunda parte da Atividade 5 realizada pela estudante Rhaylla.....	116
Figura 50 – Programação desenvolvida pela estudante Rhaylla na segunda parte da Atividade 5.....	118
Figura 51 – Programação desenvolvida pelo pesquisador para exemplificar como deveria ser a construção do bloco “se/senão” no desenvolvimento da segunda parte da Atividade 5.....	119

LISTA DE SIGLAS E ABREVIATURAS

BDTD	Biblioteca Digital Brasileira de Teses e Dissertações
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CSTA	Computer Science Teachers Association
EM	Educação Matemática
GPIMEM	Grupo de Pesquisa em Informática, outras Mídias e Educação Matemática
ISTE	International Society for Technology in Education
MIT	Massachusetts Institute of Technology
MM	Modelagem Matemática
NSF	National Science Foundation
PC	Pensamento Computacional
PEI	Programa de Ensino Integral
PIBID	Programa Institucional de Bolsas de Iniciação à Docência
PPGEM	Programa de Pós-Graduação em Educação Matemática
S4A	Scratch for Arduino
UNESP	Universidade Estadual Paulista “Júlio de Mesquita Filho”
USP	Universidade de São Paulo

SUMÁRIO

1 INTRODUÇÃO	21
1.1 TRAJETÓRIA DO TEMA DA PESQUISA E DO PESQUISADOR.....	21
1.2 REVISÃO DE LITERATURA.....	17
1.3 ESTRUTURA DA DISSERTAÇÃO.....	23
2 ASPECTOS METODOLÓGICOS DA PESQUISA	25
2.1 CENÁRIO DE INVESTIGAÇÃO	26
2.2 ATIVIDADES DESENVOLVIDAS COM OS ESTUDANTES	29
3 COMPUTADOR E PENSAMENTO: REORGANIZAÇÃO DE IDEIAS	33
4 CONEXÕES ENTRE PENSAMENTO COMPUTACIONAL, SCRATCH e MATEMÁTICA	42
4.1 PENSAMENTO COMPUTACIONAL.....	42
4.2 A RELAÇÃO ENTRE PENSAMENTO COMPUTACIONAL E MATEMÁTICA .	45
4.3 A RELAÇÃO ENTRE PENSAMENTO COMPUTACIONAL E SCRATCH.....	49
4.4 A RELAÇÃO ENTRE SCRATCH E MATEMÁTICA	59
4.5 SÍNTESE DOS CONCEITOS DO PENSAMENTO COMPUTACIONAL	64
5 APRESENTAÇÃO E ANÁLISE DOS DADOS	66
5.1 ATIVIDADE 1 COM A TURMA 9º2-T2	67
5.2 ATIVIDADE 4 COM A TURMA 9º1-T2	82
5.3 ATIVIDADE 5 COM A TURMA 9º1-T2	91
6 REFLEXÕES E APONTAMENTOS	121
REFERÊNCIAS	125
APÊNDICES	129
Apêndice A - O software Scratch	129
Apêndice B – Carta de apresentação do pesquisador	144
Apêndice C – Carta de apresentação de projeto de pesquisa	146

Apêndice D – Termo de consentimento livre e esclarecido	148
Apêndice E – Descrição das Atividades 2 e 3 realizadas pelos estudantes na produção de dados	150

1 INTRODUÇÃO

1.1 TRAJETÓRIA DO TEMA DA PESQUISA E DO PESQUISADOR

Nos últimos dois anos da minha graduação em Licenciatura em Matemática na Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP), *campus* Rio Claro, fui bolsista do Programa Institucional de Bolsas de Iniciação à Docência (PIBID) na Escola Estadual Professor Marciano de Toledo Piza, que pertence ao Programa de Ensino Integral (PEI) do Estado de São Paulo e é localizada no município de Rio Claro – SP. Tive a oportunidade de observar que, além das disciplinas da Base Nacional Comum Curricular, as instituições que integram o PEI possuem matérias como: Eletivas, Projeto de Vida, Protagonismo Juvenil, Orientações de Estudo, Atividades Experimentais e Laboratórios.

No último semestre letivo do curso de Licenciatura em Matemática, no ano de 2016, ainda não sabia qual caminho trilharia na pós-graduação. Inicialmente, fiquei em dúvida entre as áreas de Educação Matemática (EM) e Matemática Aplicada. Fazendo algumas pesquisas, descobri a subárea Modelagem Matemática (doravante MM), ligada a ambas, porém distinta delas. Assim, minha primeira leitura para a escrita do pré-projeto apresentado no exame de seleção do Programa de Pós-Graduação em Educação Matemática (PPGEM) foi o livro “Modelagem em Educação Matemática”, de Meyer, Caldeira e Malheiros (2013). Após a leitura, vi certa sinergia entre as disciplinas Eletivas do PEI e a MM, já que, nessas disciplinas, professores de campos diferentes trabalham juntos e, por essa razão, há uma integração das distintas áreas de ensino para formar o tema a ser abordado nelas no decorrer do respectivo ano letivo. Além disso, os estudantes escolhem de qual Eletiva vão participar e trabalham em grupos. No ano de 2016, em específico, dentre as diversas opções de disciplina que havia na Escola Estadual Professor Marciano de Toledo Piza, acompanhei uma cujos professores responsáveis eram de Matemática e de Inglês e na qual os grupos simularam a administração de um restaurante, de uma casa de câmbio, de uma agência de turismo e de um hotel. Tudo isso me motivou a buscar a Educação Matemática como uma área para desenvolver uma pesquisa de mestrado.

Decidi, então, no ano de 2017, ingressar como aluno especial no PPGEM e, ao participar das disciplinas, percebi que meu interesse recaía sobre as tecnologias na Educação Matemática. Desse modo, participei e fui aprovado no processo seletivo do referido Programa de Pós-Graduação, para ingresso em 2018, sob a orientação da Profa. Dra. Sueli Liberatti Javaroni, que pertence ao Grupo de Pesquisa em Informática, outras Mídias e Educação Matemática (GPIMEM). Na ocasião, em reunião para definir o projeto de pesquisa, fui informado de que outro pesquisador orientado por ela estava desenvolvendo um estudo cujo cenário de investigação para a produção dos dados havia sido a Escola Estadual Professora Carolina Augusta Seraphim, localizada em Rio Claro, no Estado de São Paulo. Esta escola pertence ao PEI e tinha, naquele momento, mostrado interesse na continuidade da parceria com a Universidade para o desenvolvimento de pesquisas. Além disso, a orientadora me informou que o principal tema a que estava se dedicando era o desenvolvimento do pensamento computacional (PC). Assim, chegamos ao consenso de que meu projeto versaria sobre a Modelagem Matemática e o desenvolvimento do pensamento computacional, tendo como cenário de investigação a escola supracitada.

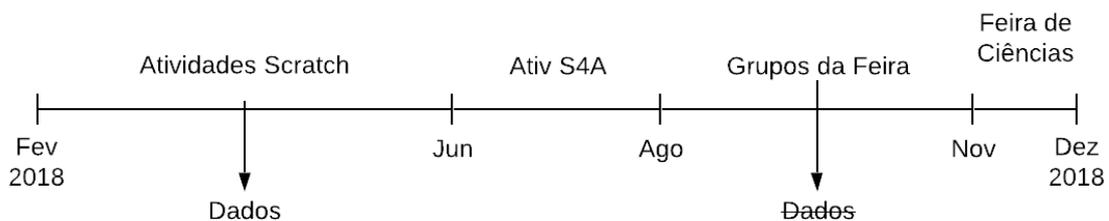
Tal parceria foi formalmente fixada com a entrega da “Carta de apresentação do pesquisador” e da “Carta de apresentação de projeto de pesquisa”, constantes nos apêndices B e C, respectivamente. Tal parceria foi estabelecida mais especificamente com a professora Ana Maria Freitas Soares, que, naquela ocasião, era a docente responsável pela disciplina “Práticas de Matemática”. Dessa forma, o cenário de investigação da pesquisa aqui relatada foi constituído por duas turmas de estudantes do nono ano do Ensino Fundamental que cursavam a referida disciplina, sob a responsabilidade da professora Ana Maria Freitas Soares.

Em fevereiro de 2018, a diretora da escola, Soráia Pessoa Vieira, e a professora Ana Maria Freitas Soares informaram que o tema da Feira de Ciências daquele ano seria robótica. Assim, os grupos de estudantes do nono ano deveriam desenvolver seus projetos usando o *kit* de robótica da escola, que contém uma placa Arduino Uno, e sua programação é feita por meio de um *software* homônimo, mas também pode ser realizada por um *software* denominado *Scratch for Arduino* (S4A). Portanto, o objetivo inicialmente proposto foi o de buscar possível relação entre o desenvolvimento do pensamento computacional e a Modelagem Matemática, por

meio da observação e análise da realização dos trabalhos que os estudantes participantes da pesquisa fariam para a Feira de Ciências.

A cronologia dos acontecimentos da pesquisa no ano de 2018 pode ser visualizada na Figura 1. Inicialmente, entre os meses de fevereiro e o fim de maio, desenvolvi as atividades com o *software Scratch*. De junho até meados de agosto, avancei para práticas com o *software S4A*. Depois disso, isto é, de meados de agosto até fim de outubro, os estudantes formaram os grupos para desenvolverem seus projetos para a Feira de Ciências, que ocorreu no mês de novembro. Esses trabalhos desenvolvidos por eles seriam os dados a serem analisados na pesquisa, porém não foi o que ocorreu, como será explicitado a seguir, desse modo as atividades desenvolvidas entre fevereiro e maio resultaram no objeto de investigação dessa pesquisa. A dinâmica dessas atividades com as duas turmas de nono ano será descrita na seção “Aspectos Metodológicos da pesquisa”.

Figura 1 – Cronologia dos acontecimentos da pesquisa



Fonte: O Autor.

Havia combinado com os estudantes que observaria todos os trabalhos que eles desenvolvessem para apresentação na Feira de Ciências. No entanto, esse processo de observação total não foi possível de ser realizado de forma a contribuir com a pesquisa. Isso ocorreu por diversos motivos. O primeiro foi o fato de que os eles já sabiam que o tema da Feira de Ciências seria robótica e, em função disso, alguns deles iniciaram seus trabalhos antes de agosto, ou seja, no período de agosto a novembro, tinham finalizado ou parcialmente finalizado seu desenvolvimento, não vendo necessidade de trazê-los para a escola. Conseqüentemente, não consegui acompanhar a elaboração desses nem de outros projetos de discentes que, durante esses meses, faziam-nos fora do ambiente escolar, com receio de não terem tempo hábil para sua finalização. Em segundo lugar, existiram também alguns deles que não quiseram que eu participasse do desenvolvimento das atividades. Por fim, foi possível observar e acompanhar três

grupos de estudantes, cujos trabalhos foram: um cubo de LED; dois jogos no *Scratch* e um sensor de proximidade.

Nesses três trabalhos observados, a Modelagem Matemática não foi evidenciada, de modo que o objetivo proposto a princípio, qual seja o de buscar uma possível relação entre o desenvolvimento do pensamento computacional e a Modelagem Matemática, não pôde ser concretizado. Assim, as atividades iniciais, realizadas pelos estudantes no *Scratch* entre fevereiro e fim de maio, tornaram-se objeto de análise da pesquisa, visto que delas emergiram os temas pensamento computacional, *Scratch* e Matemática. Dessa forma, a pesquisa de mestrado relatada nessa dissertação foi conduzida pela pergunta: “*Quais conceitos do pensamento computacional emergem de estudantes do nono ano ao realizarem atividades com conteúdo matemático no Scratch?*” e seu objetivo foi investigar as possíveis relações entre PC, *Scratch* e Matemática. O processo para tal propósito foi buscar o referencial teórico de cada tema e fundamentar tais relações na literatura, para que, na análise, fosse possível procurar indícios de quais conceitos do PC manifestaram-se nos estudantes do nono ano ao realizarem as atividades no *software*.

Com a mudança da pesquisa, houve necessidade de encontrar outros trabalhos que tratassem de temas similares. Desse modo, a seguir, discorrerei acerca dos estudos realizados dentro do PPGEM, e mais especificamente no interior do GPIMEM, para, posteriormente, apresentar investigações que, de alguma forma, aproximam-se dos assuntos supracitados, mas que, ainda assim, distinguem-se da minha pesquisa e, portanto, evidenciam sua importância.

1.2 REVISÃO DE LITERATURA

O intuito dessa subseção é apresentar estudos que se aproximam dos temas por mim investigados, indicar os motivos de terem sido escolhidos e evidenciar como se distinguem da minha pesquisa. Como dito anteriormente, iniciarei a revisão de literatura discorrendo sobre os trabalhos dos pesquisadores do GPIMEM, que são: Silva (2018), Barbosa (2019) e Bessa (2020), mostrando que, apesar das similaridades, há distinções nas investigações.

Silva (2018) buscou analisar o processo de formação de conceitos matemáticos de estudantes do nono ano do Ensino Fundamental sobre o significado

do resto da divisão euclidiana e da congruência de números inteiros (módulo n) a partir do desenvolvimento de seu pensamento computacional no decurso da realização de atividades com kit de robótica e programação. Durante a observação participante, o pesquisador notou que surgiram dúvidas nos discentes tanto relativas à compreensão de período quanto sobre a interpretação de cada componente da divisão euclidiana. Desse modo, ele propôs planos de aula que abordassem tais conceitos com os alunos e, em um deles, com o intuito de investigar se os estudantes conseguiriam chegar à definição de congruência entre números inteiros (módulo n), Silva (2018) trabalhou esses conteúdos matemáticos por meio da montagem e da programação de um semáforo, usando, para tanto, kit de robótica Arduino, programado pelo *Scratch for Arduino* (S4A). Os dados produzidos contaram com entrevistas com os participantes da pesquisa, com suas produções, escritas ou salvas em arquivos, além da captura da tela, que permitiu monitorar as ações realizadas no computador. A análise foi fundamentada nos estudos cognitivos de Vygotsky, bem como no desenvolvimento do pensamento computacional e, segundo o pesquisador, se estiver presente no planejamento pedagógico e nas ações docentes com a finalidade de permitir que o estudante seja ativo no processo de aprendizagem, então, o trabalho com o pensamento computacional em atividades de robótica pode contribuir para a formação de conceitos matemáticos, por meio do reconhecimento de padrões, da decomposição, do raciocínio algoritmo e da abstração.

Barbosa (2019) teve por objetivo investigar os aspectos do pensamento computacional emergentes em um grupo de estudantes de graduação em Matemática ao explorarem a Geometria Fractal por meio do *software GeoGebra*. Para tanto, foram realizados quatro experimentos de ensino com seis estudantes de graduação em Matemática, divididos em três duplas. Tais experimentos contaram, respectivamente, com atividades explorando características e construção dos fractais Triângulo, Tetraedro, Tapete de Sierpinski e Esponja de Menger, de modo que todas as duplas realizassem quatro delas, cada uma envolvendo um fractal supracitado diferente. A escolha por esses fractais foi devido à possibilidade de sua construção ser realizada a partir de elementos da Geometria Euclidiana, como pontos, segmentos, polígonos e poliedros e também à possibilidade de serem explorados e construídos em ambientes 2D (segunda dimensão) e 3D (terceira dimensão). Os procedimentos metodológicos para a produção dos dados foram: as

filmagens das sessões de ensino, a captura da tela do computador, o roteiro impresso das atividades e as entrevistas efetuadas com as duplas após o final de cada sessão. A análise da gravação dos experimentos contou com a observação e descrição dos dados, com a identificação de eventos críticos, com a transcrição, codificação e criação do enredo e com a composição da narrativa. Com isso, a pesquisadora elaborou diagramas dos momentos de construção dos fractais explorados em cada atividade, por meio do *GeoGebra*, os quais ofereceram evidências em relação aos conceitos associados ao pensamento computacional (pensamento algorítmico; decomposição e generalização; padrões e abstração; representação e automação; e avaliação). Além disso, Barbosa (2019, p. 148) conclui que esses conceitos “não aparecem de forma isolada, eles são miscíveis e um pode complementar e/ou fortalecer o outro”.

O propósito da pesquisa de Bessa (2020), por sua vez, foi investigar o processo de construção de jogos digitais em um ambiente que viabiliza o pensamento computacional, visando ao desenvolvimento de habilidades matemáticas pautadas pela Base Nacional Comum Curricular (BNCC). Para realizar a investigação, foram realizados oito encontros, ocorridos no contraturno escolar, com 12 estudantes dos Anos Finais do Ensino Fundamental. As atividades foram desenvolvidas na sala de leitura da escola, onde os estudantes se dispunham em duplas ou trios para programarem no *Scratch* e efetuarem a construção dos jogos digitais nos *notebooks* que a pesquisadora levou à escola. Os dados da pesquisa são oriundos das filmagens dos encontros, tanto da *webcam* quanto da captura de tela do computador, dos cadernos de anotações dos estudantes e da pesquisadora, das construções no *Scratch* e das entrevistas com os estudantes. No primeiro encontro, eles construíram um jogo seguindo as instruções contidas em cartões. Nos demais, eles programaram jogos digitais inspirados em temas provenientes de enunciados de problemas matemáticos, em leituras de livros didáticos e paradidáticos e em vídeos voltados à Matemática. Ao término das programações, os discentes trocavam os jogos entre si para sugerirem mudanças que julgassem pertinentes. Para analisar as ações dos participantes da pesquisa na produção dos jogos digitais, Bessa (2020) tomou os pressupostos do Construcionismo e do Turbilhão de Aprendizagem, fundamentados, respectivamente, em Papert (1985) e em Rosa (2008), tendo percebido que as construções no *Scratch* apresentaram

indícios tanto do desenvolvimento de habilidades matemáticas quanto de habilidades do pensamento computacional.

Como é possível verificar, o tema comum entre as três pesquisas e esta dissertação é o PC. No entanto, suas distinções são acerca de como o mesmo foi investigado, ou seja, as diferenças se iniciam no referencial teórico em que cada pesquisa se fundamentou para analisar seus dados. Além disso, Silva (2018) e Barbosa (2019) utilizaram *softwares* diferentes, quais sejam S4A e *GeoGebra*, respectivamente, e, apesar de Bessa (2020) também utilizar o *Scratch*, o desenvolvimento de suas atividades foi baseado em jogos.

Para encontrar outras pesquisas que apresentam os temas relacionados, o levantamento bibliográfico foi realizado na Biblioteca Digital Brasileira de Teses e Dissertações (BDTD), no Catálogo de Teses e Dissertações da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), na Biblioteca Digital de Teses e Dissertações da Universidade de São Paulo (USP) e no Repositório Institucional da UNESP. Assim, a seguir, apresentarei outras três pesquisas cujos temas são comuns à minha, mas, assim como nos estudos anteriormente comentados, há uma distinção que parte do referencial teórico em que os pesquisadores se basearam, ou seja, as investigações contêm os assuntos PC, *Scratch* e Matemática, contudo, a forma como são abordados e analisados é diferente, como, por exemplo, o fato de que elas não focam em mostrar a relação entre eles.

Lummertz (2016) investigou as potencialidades do uso do *software* de programação *Scratch* na constituição de aspectos relacionados à Literacia Digital e ao pensamento computacional, por meio da construção de jogos eletrônicos. O referencial teórico da pesquisa contou com as ideias construcionistas, com a Literacia Digital¹ (JENKIS et al., 2006), com o pensamento computacional e com a Lógica de Programação. Os sujeitos de seu estudo foram oito alunos do quarto ano do Ensino Fundamental e a produção de dados ocorreu em seis encontros no contraturno das aulas e visaram à construção de jogos eletrônicos. Os dados foram gerados por meio de gravações de vídeo, áudio e captura da tela dos computadores utilizados na pesquisa. Os resultados apontam três aspectos relacionados à

¹ “O conjunto de habilidades e competências em que o letramento auditivo, visual e digital se sobrepõem. Isso inclui a capacidade de compreender o poder das imagens e sons, de reconhecer e usar esse poder, de manipular e transformar a mídia digital, de distribuí-la generalizadamente e de adaptá-la facilmente a novas formas” (JENKINS et al., 2006, p.30).

Literacia Digital e ao pensamento computacional: possibilidades frente à Literacia Digital, potencialidades frente à construção do pensamento computacional e potencialidades frente à associação com conteúdos programáticos de Matemática propostos nos Parâmetros Curriculares Nacionais. Além disso, a investigação mostra indícios de que as habilidades da Literacia Digital são miscíveis e uma habilidade complementa ou fortalece a outra. Desse modo, o conjunto delas não só permite, mas também pode potencializar a capacidade que as crianças têm de articular a sua compreensão sobre a maneira de interagir com as Tecnologias Digitais que as cercam. Por fim, Lummerz (2016) ressalta que a Literacia Digital pode ser estimulada pelos conceitos e práticas do pensamento computacional, que, por sua vez, também intensificam a construção da Literacia Digital.

Corrêa (2021) se propôs a analisar os impactos da programação no desenvolvimento de conceitos matemáticos e do pensamento computacional em âmbito escolar com estudantes de 8º e 9º anos do Ensino Fundamental. A pesquisa aconteceu no contraturno escolar, em dez encontros, por meio de uma oficina de programação em *Scratch*, cuja proposta foi o desenvolvimento de jogos, utilizando o *software*. Os cinco primeiros encontros foram destinados à realização de atividades iniciais por parte dos estudantes, com o propósito de conhecerem e de explorarem alguns recursos do programa. Os demais foram dedicados ao planejamento e à programação dos jogos. Os dados foram produzidos pelas gravações dos áudios das intervenções e dos diálogos entre os estudantes, pela captura de telas e pelos registros dos estudantes, elaborados no *Scratch online*. A fundamentação teórica para análise das informações coletadas foi a Teoria do Desenvolvimento Cognitivo, de Jean Piaget (1978), para compreender o desenvolvimento e a construção de conceitos matemáticos dos estudantes, e a teoria Construcionista de Seymour Papert (1980), buscando aproximações com a ideia de formação do pensamento computacional de Janette Wing (2006, 2008). Os resultados mostraram que o uso da programação com o *Scratch* permitiu a utilização e a construção de conceitos matemáticos a partir de relações elaboradas pelos estudantes na atividade de programação, assim como desenvolveu elementos do pensamento computacional na resolução de problemas que emergiram da operação de desenvolver um jogo, como a decomposição de problemas, a generalização de situações semelhantes, a abstração de informações e o uso de algoritmos como forma de representação.

O interesse da pesquisa de Carvalho (2018) residiu nas possibilidades e desafios que emergem da associação da Modelagem Matemática com Tecnologias Digitais da Informação e Comunicação, mais especificamente, buscou-se compreender o que se revela quando se desenvolve uma tarefa de Modelagem Matemática aliada à programação de computadores. A abordagem da pesquisa foi qualitativa segundo uma visão fenomenológica, na qual o pesquisador elaborou uma tarefa de Modelagem Matemática, realizada por estudantes do Ensino Médio por meio da programação do *software Scratch*. O desenvolvimento da tarefa foi registrado por gravações de vídeos do ambiente, a partir das *webcams* e da captura de tela dos computadores, que permitiram a construção de unidades de significado, as quais culminaram em quatro categorias, a saber: C1: Sobre os debates, discussões e falas entre alunos e entre alunos e professores no desenvolvimento da tarefa; C2: Sobre as ações dos alunos no contexto da tarefa de Modelagem; C3: Sobre as relações com a matemática escolar que emergiram no desenvolvimento da tarefa de Modelagem; C4: Sobre os modos que os alunos viram o projeto. Tais categorias foram interpretadas hermeneuticamente e permitiram observar que a tarefa de Modelagem norteou o desenvolvimento do pensamento computacional, por meio de uma atividade de programação de computadores, potencializando as discussões e reflexões sobre o contexto envolvido pela atividade, abrindo espaço para um ambiente dialógico e dinâmico, construído pela sinergia da Modelagem com a programação de computadores.

De modo geral, as distinções em relação à minha pesquisa são que, em relação a Lummerz (2016), primeiro, o autor traz os mesmos temas, porém os relaciona com a Literacia Digital, fundamentando-se nas ideias construcionistas; segundo, os sujeitos de sua pesquisa são estudantes do quarto ano do Ensino Fundamental e, terceiro, as atividades desenvolvidas foram construções de jogos eletrônicos no *Scratch*. Essa distinção também ocorre no tocante a Corrêa (2021), que, além dos jogos, também se utiliza do Construcionismo e da teoria do desenvolvimento cognitivo de Jean Piaget para sua análise. Por fim, Carvalho (2018) apresenta os mesmos temas, relacionando-os e, conseqüentemente, fundamentando-os com a Modelagem Matemática Já na minha investigação, os participantes foram estudantes do nono ano do Ensino Fundamental, o referencial teórico versa sobre uso de computadores e cognição humana, sobre a interação de distintas tecnologias e estilos de pensamentos diferentes, bem como sobre as

possíveis conexões entre os temas pensamento computacional, *software Scratch* e Matemática. Além disso, no meu estudo, as atividades desenvolvidas pelos estudantes na produção de dados não tiveram o intuito de serem jogos, mas sim programações envolvendo conteúdo matemático. Para finalizar essa primeira seção, a seguir, apresento a estrutura da dissertação.

1.3 ESTRUTURA DA DISSERTAÇÃO

O presente trabalho está dividido em 6 seções, sendo esta a primeira, em que é realizada a apresentação da trajetória do tema e do pesquisador, a revisão de literatura e o esquema de sua composição.

Na segunda seção, “Aspectos Metodológicos da pesquisa”, é evidenciada a metodologia do estudo e seu caráter qualitativo e, em seguida, a pergunta de pesquisa. Além disso, a seção mostra o cenário de investigação, apontando as características de uma escola pertencente ao PEI, os participantes da pesquisa e finaliza com a descrição da dinâmica e do desenvolvimento das atividades realizadas na produção dos dados.

A terceira e quarta seções compõem o referencial teórico, sendo que ressaltam questões distintas. Isso porque a terceira trata das seguintes ideias: de como o uso do computador afeta a cognição humana (TIKHOMIROV, 1981); da forma como a interação com diferentes tecnologias intelectuais promove estilos de pensamento distintos (LEVY, 1997), e da abordagem experimental-com-tecnologia (BORBA; VILLARREAL, 2005). Já a quarta seção tem como foco consolidar ideias acerca do pensamento computacional (PAPERT, 1980; WING, 2006), bem como sua relação com a Matemática (ISTE; CSTA, 2011; BARR; STEPHENSON, 2011) e com o *Scratch* (BRENNAN; RESNICK, 2012), além de apresentar a ligação entre *Scratch* e Matemática (BENTON et al., 2016).

Na quinta seção, “Apresentação e análise dos dados”, mostro detalhadamente algumas das atividades realizadas pelos estudantes na produção dos dados, bem como suas análises, dialogando com o referencial apresentado na terceira e quarta seções. Finalizo fazendo algumas considerações gerais acerca das tarefas desenvolvidas.

Na sexta seção, “Reflexões e apontamentos”, assinalo as reflexões que foram feitas ao longo da pesquisa, trazendo possíveis respostas à pergunta diretriz e

ponderando sobre outros questionamentos que surgiram de toda essa investigação, além das possibilidades de estudos futuros.

2 ASPECTOS METODOLÓGICOS DA PESQUISA

Essa seção discorrerá sobre as características qualitativas da pesquisa, apresentará seu cenário de investigação, seus participantes e as atividades desenvolvidas na produção de dados, bem como descreverá a dinâmica em que foram ocorridas.

Para Creswell (2014), alguns dos motivos para se fazer pesquisa qualitativa são: um problema ou questão precisar ser explorado; a necessidade de uma compreensão complexa e detalhada dessa questão; a compreensão dos contextos ou ambientes em que os participantes de um estudo abordam esse problema ou questão. Assim, considero que o estudo que desenvolvi se deu na modalidade qualitativa, visto que a pergunta que norteou minhas ações, a saber: “*Quais conceitos do pensamento computacional emergem de estudantes do nono ano ao realizarem atividades com conteúdo matemático no Scratch?*”, precisou de uma compreensão complexa e detalhada, sendo necessário investigar as possíveis ligações entre *Scratch*, Matemática e PC, isto é, buscar o referencial teórico de cada tema e fundamentar as relações encontradas na literatura. Além disso, foi preciso procurar indícios de resposta à pergunta ao analisar as atividades desenvolvidas pelos estudantes.

Como já apresentado, o cenário de investigação foi o acompanhamento de duas turmas do nono ano do Ensino Fundamental na disciplina Práticas de Matemática, cujos estudantes realizaram um total de cinco atividades que exploraram funcionalidades do *Scratch* e conteúdos matemáticos. A dinâmica das aulas em que tais atividades ocorreram se dava da seguinte forma: em um primeiro momento, eu ensinava uma funcionalidade do *software*, por meio de uma tarefa, tentando relacioná-la a algum conteúdo matemático, para que, em um segundo momento, a partir do conteúdo que havia sido ensinado, eles realizassem outra tarefa proposta, dedicando-se à manipulação e à exploração do programa, com o intuito de generalizarem tais funcionalidades aprendidas para atividades futuras.

Ao desenvolver a pesquisa de campo dentro da escola, no laboratório de informática, reunindo as informações junto aos estudantes, falando diretamente com eles, vendo seus comportamentos e suas ações dentro desse contexto, tive interações pessoais dentro de seu ambiente habitual (BOGDAN; BIKLEN, 1994; CRESWELL, 2014). A essa descrição, também cabe adicionar o quadro em que a

pesquisa foi realizada, bem como seus participantes. Desse modo, na próxima subseção, são expostas as características do cenário de investigação da pesquisa.

2.1 CENÁRIO DE INVESTIGAÇÃO

A pesquisa de campo foi desenvolvida com duas turmas de nono ano, no período de fevereiro a novembro de 2018, na Escola Estadual Professora Carolina Augusta Seraphim (EECAS), localizada no município de Rio Claro – SP. Essa instituição atende estudantes do 6º ao 9º ano do Ensino Fundamental e pertence ao Programa Ensino Integral² (PEI), que, além das disciplinas da Base Nacional Comum Curricular, possui matérias da Parte Diversificada, tais como: Eletivas, Projeto de Vida, Protagonismo Juvenil, Orientações de Estudo e Práticas Experimentais. Na Figura 2, pode-se observar a Matriz Curricular dos Anos Finais do Ensino Fundamental de uma escola pertencente ao PEI.

Figura 2 – Matriz Curricular dos Anos Finais do Ensino Fundamental

Fundamentação Legal: LDBEN - Lei 9.394/96 e Lei Complementar nº 1.164/2012, alterada pela Lei Complementar nº 1.191/2012							
DISCIPLINAS/ COMPONENTES CURRICULARES	ANO	ANO	ANO	ANO	Carga horária		
	6º	7º	8º	9º			
	Nº DE AULAS	Nº DE AULAS	Nº DE AULAS	Nº DE AULAS			
BASE NACIONAL COMUM	Língua Portuguesa	6	6	6	6	960	
	Arte	2	2	2	2	320	
	Educação Física	2	2	2	2	320	
	Matemática	6	6	6	5	920	
	Ciências Físicas e Biológicas	4	4	4	4	640	
	História	4	4	4	4	640	
	Geografia	4	4	4	4	640	
	Ensino Religioso *	0	0	0	1	40	
	TOTAL DA BASE NACIONAL COMUM	28	28	28	28	4.480	
PARTE DIVERSIFICADA	Língua Estrang. Moderna – Inglês	2	2	2	2	320	
	Disciplinas Eletivas	2	2	2	2	320	
	ATIVIDADES COMPLEMENTARES	Práticas Experimentais	0	0	2	2	160
		Orientação de Estudos	4	4	2	2	480
		Protagonismo Juvenil	1	1	1	1	160
Projeto de Vida: valores para a vida cidadã	2	2	2	2	320		
Total da Parte Diversificada	11	11	11	11	1.760		
Total Geral	39	39	39	39	6.240		

(*) Caso não haja demanda para Ensino Religioso, acrescentar uma aula para Matemática

Fonte: São Paulo (2014b).

² Instituído pela Lei Complementar nº 1.164, de 4 de janeiro de 2012, alterada pela Lei Complementar nº 1.191, de 28 de dezembro de 2012.

A Parte Diversificada, instituída pela Lei nº 9.394, de 20 de dezembro de 1996, em seu art. 26, tem a seguinte proposta:

Os currículos da educação infantil, do ensino fundamental e do ensino médio devem ter base nacional comum, a ser complementada, em cada sistema de ensino e em cada estabelecimento escolar, por uma parte diversificada, exigida pelas características regionais e locais da sociedade, da cultura, da economia e dos educandos (BRASIL, 1996, n.p).

Como mencionado acima, uma escola pertencente ao PEI possui algumas disciplinas específicas, sendo uma delas a de Práticas Experimentais, cuja definição, constante nas Diretrizes do Programa Ensino Integral (2014a), especificamente na Concepção do Modelo Pedagógico, é explicitada abaixo:

[...] contribuem para a melhoria do desempenho dos estudantes proporcionando-lhes a oportunidade de manipular materiais e equipamentos especializados no ambiente de laboratório, comparar, estabelecer relação, ler e interpretar gráficos, construir tabelas dentre outras habilidades e, desta forma, construir seu conhecimento a partir da investigação com práticas eficientes [...] as atividades investigativas podem contribuir para o desenvolvimento de competências e habilidades tais como: formular hipóteses, elaborar procedimentos, conduzir investigações, formular explicações, apresentar e defender argumentos científicos (SÃO PAULO, 2014a, p. 31-32).

Na escola em que realizei a pesquisa, as Práticas Experimentais foram divididas em Práticas de Matemática e Práticas de Ciências, ambas oferecidas para o oitavo e nono anos. As aulas para o nono ano eram realizadas às sextas-feiras e ocorriam simultaneamente para cada um deles, pois os laboratórios de informática e de ciências não comportavam uma turma completa de 34 estudantes. Assim, o 9º1 e o 9º2 precisaram ser divididos em duas turmas menores, com 17 estudantes cada, resultando em quatro turmas, que denominarei de 9º1-T1, 9º1-T2, 9º2-T1 e 9º2-T2. Essa divisão fez com que as aulas de cada disciplina ocorressem quinzenalmente para cada uma das quatro turmas. Tal divisão pode ser observada na Tabela 1.

Tabela 1 – Dinâmica das aulas de Práticas de Matemática e Práticas de Ciências

Sexta-feira				
Horário	Semana 1		Semana 2	
	Práticas de Matemática	Práticas de Ciências	Práticas de Matemática	Práticas de Ciências
10h35 às 11h25	9°1-T1	9°1-T2	9°1-T2	9°1-T1
12h30 às 13h20	9°1-T1	9°1-T2	9°1-T2	9°1-T1
13h20 às 14h10	9°2-T1	9°2-T2	9°2-T2	9°2-T1
14h10 às 15h	9°2-T1	9°2-T2	9°2-T2	9°2-T1

Fonte: O Autor.

Assim, o cenário de investigação dessa pesquisa foi constituído pelas aulas da disciplina de Práticas de Matemática para duas turmas de nono ano, 9°1 e 9°2, com 34 estudantes cada, que foram divididas em duas menores, resultando em quatro grupos: 9°1-T1, 9°1-T2, 9°2-T1 e 9°2-T2. A produção de dados ocorreu quinzenalmente no laboratório de informática, a partir das ações dessas quatro turmas que se agrupavam em duplas ou trios, dependendo da quantidade de presentes no dia da atividade. A docente responsável pela disciplina foi a professora Ana Maria Freitas Soares, graduada em Ciências Físicas, com habilitação em Biologia, pela Faculdade de Filosofia, Ciências e Letras de Guaxupé – MG; em Pedagogia, pela Universidade de São Paulo (USP), e Especializada em Psicopedagogia pela mesma universidade. Ela é professora na Escola Carolina Augusta Seraphim (EECAS) desde 2012, lecionando nas disciplinas de Ciências,

Matemática, Práticas de Matemática e Eletivas. Juntamente com ela, propus as atividades que foram desenvolvidas com o *software Scratch* e tiveram o intuito de instruir os estudantes em seu uso, bem como de incluir conteúdos matemáticos. A seguir, apresento as práticas e o seu desenvolvimento junto aos estudantes da disciplina Práticas de Matemática, que constituiu o cenário de investigação da pesquisa relatada nessa dissertação.

2.2 ATIVIDADES DESENVOLVIDAS COM OS ESTUDANTES

Antes de começar a desenvolver as atividades, entreguei aos estudantes o “Termo de consentimento livre e esclarecido”, como consta no Apêndice D, informando-os que os responsáveis deveriam assiná-lo. Junto com o termo, encaminhei também a “Carta de apresentação da pesquisa” (Apêndice C).

As atividades tinham em comum a intenção de explorar algum conteúdo matemático, buscando utilizar o *Scratch* como alternativa ao tradicional livro didático/caderno do aluno para apresentá-lo aos estudantes. Para tal, usava meu próprio *notebook* ou um dispositivo fornecido pela escola para projetar o que executava em uma TV contida no laboratório de informática. Às vezes, usava também o quadro branco, com o intuito de tirar algumas dúvidas dos discentes ou representar um desenho de forma mais rápida e eficaz. As cinco atividades exploradas com os participantes da pesquisa foram: Atividade 1 – Introdução ao *Scratch* e desenho casa; Atividade 2 – Jogo da maçã e do labirinto; Atividade 3 – Introdução dos Procedimentos; Atividade 4 – Variáveis e polígonos regulares, e Atividade 5 – Blocos “se”, “se/senão” e calculadora de áreas. A escolha das tarefas, bem como sua ordem de realização, foi pensada com base na evolução da programação com o *Scratch* e na afinidade do conteúdo matemático com as funcionalidades do *software*. O Quadro 1 traz a descrição teórica e prática dessas atividades.

Quadro 1 – Estrutura das atividades desenvolvidas na produção de dados

Atividade	Tema	Conteúdo do <i>Scratch</i>	Prática
1	Introdução ao <i>Scratch</i>	Uso da paleta Caneta e Movimento e suas funcionalidades.	Desenvolver programação para desenhar uma casa.
2	Jogo da maçã e do labirinto	Entendimento das mecânicas de funcionamento do jogo da maçã, como o movimento do ator por	Por meio da programação do jogo da maçã, desenvolver uma

		meio das teclas e o movimento de queda da maçã.	programação para criar um jogo de labirinto.
3	Introdução aos procedimentos	Uso da paleta Mais Blocos e suas funcionalidades.	Por meio de um procedimento já estabelecido, desenvolver uma programação que desenhasse uma flor e uma flor com um ramo.
4	Variáveis e polígonos regulares	Uso dos blocos de comando e de função da paleta Variáveis com conteúdo matemático sobre polígonos regulares.	Programação para desenhar um triângulo equilátero, um hexágono regular e uma teia de aranha.
5	Blocos “se” e “se/senão”	Estudo das estruturas dos blocos na tomada de decisão de uma programação.	Desenvolver uma programação que calculasse a área de um retângulo, de um triângulo e de uma circunferência.

Fonte: O Autor.

Cabe ressaltar que as principais funcionalidades do *software Scratch* são apresentadas no Apêndice A desta dissertação. Nele, constam imagens da interface e dos blocos, além de haver informações sobre as diferenças, tipos e funções deles. Como consta no Quadro 1, a progressão do conteúdo referente ao *software* foi seguindo o mesmo avanço exposto em tal Apêndice, ou seja, inicialmente, mostrei aos estudantes a interface do *Scratch*, projetando meu *notebook* na TV do laboratório, exibindo o palco, a área de programação e os blocos. Apresentei também algumas funcionalidades, como criar e editar um novo ator e um novo palco, as abas de fantasias e sons, bem como expliquei que o palco possui 480 passos de largura e 360 passos de altura, além de que, ao passar o ponteiro do mouse sobre ele, em seu canto inferior direito, são exibidas as coordenadas x e y desse ponteiro, sendo que o valor da coordenada x varia de -240 a 240 e o da coordenada y, de -180 a 180. Exibi que, na área de programação, aparece um ícone referente ao ator selecionado, mostrando as coordenadas atuais dele.

A primeira atividade foi a única cuja dinâmica não se deu em dois momentos. Fiz o desenho de uma casa na lousa e pedi aos estudantes que explorassem o *Scratch*, a fim de desenharem-na também, a partir do meu esboço, usando os

recursos associados à paleta caneta e ao movimento do ator. Auxiliei alguns deles, indicando quais blocos deveriam usar para que o ator “riscasse” o palco.

Na segunda atividade, inicialmente, mostrei aos estudantes o jogo da maçã, que foi baseado em Marji (2014) e funciona da seguinte maneira: um dos atores é uma maçã e o outro um pote; assim que o jogo inicia, as maçãs começam a cair pela tela e o objetivo é coletar o máximo possível delas com o pote, que é controlado pelas setas do teclado em um tempo ou pontuação determinados. Sugeri que jogassem por um tempo e, logo após, expliquei a eles a programação elaborada para este jogo. Em seguida, questionei o que poderia ser alterado para tornar o jogo mais difícil e, então, pedi que o recriassem e adicionassem tais alterações na programação. Na segunda parte da atividade, solicitei aos estudantes que elaborassem um jogo de labirinto a partir do que tinham visto no da maçã. Para tanto, primeiramente, projetei na TV do laboratório o jogo que eu havia desenvolvido, mostrando apenas que o palco era uma imagem de labirinto, sem exibir a programação. Assim, eles poderiam procurar na *web* algo semelhante para dar início à tarefa. Posteriormente, auxiliava-os com a programação quando pediam.

A primeira parte da terceira atividade foi referente à criação de um bloco/procedimento da paleta Mais Blocos e suas funcionalidades. Assim, como apresentado no Apêndice A, comecei dando o exemplo de como desenhar um quadrado com e sem o uso do bloco “repita”, evidenciando o que isso acarreta, por exemplo, se quisermos alterar o tamanho do quadrado ou desenhar quadrados de tamanhos diferentes. Por fim, partimos para a criação do bloco “Quadrado_passos”, o qual continha uma entrada numérica. Na segunda parte da atividade, defini para os estudantes um bloco denominado “folha”, o qual desenhava uma folha e, em seguida, a partir dele, pedi que desenvolvessem duas programações: uma que produzisse uma flor e outra que gerasse essa mesma flor, mas com um ramo.

A atividade 4, por sua vez, abrangeu o conteúdo de variáveis e polígonos regulares. Assim, na primeira parte, instruí os estudantes a criarem uma variável e mostrei as funcionalidades dos novos blocos que aparecem quando ela é feita, e o fato de que podemos atribuí-la a um ou mais atores, bem como adicioná-la a outros blocos. Na segunda parte da atividade, comentei a definição e alguns conceitos de polígonos regulares, como a soma dos ângulos internos, dos ângulos internos e externos, colocando algumas fórmulas na lousa. O intuito desse momento da atividade era fazer com que os discentes elaborassem uma programação que

desenhasse um triângulo equilátero, um hexágono regular, formado por seis desses triângulos, e a representação de uma teia de aranha, composta por esses triângulos equiláteros com o tamanho dos lados crescentes.

Finalmente, a atividade 5 tinha por objetivo desenvolver com os estudantes um programa no qual fosse possível calcular as áreas das figuras planas, quais sejam um retângulo, um triângulo e uma circunferência. Para tanto, eles usaram os conceitos de variáveis, de criar um bloco, de obter dados de entrada do usuário e de estruturas de decisão. Inicialmente, expliquei a funcionalidade dos blocos “pergunte_e espere a resposta”, “se” e “se/senão”, mostrando que podemos tomar decisões e controlar as ações dos programas de acordo com os resultados desses blocos de controle, que são determinados por condições avaliadas pelas expressões lógicas postas nos parâmetros deles. Além disso, também expliquei a eles que esses blocos podem ser encaixados um no outro.

Desta forma, essas foram as atividades desenvolvidas com os estudantes que fizeram parte do cenário de investigação dessa pesquisa, cujo interesse está no processo pelo qual esses sujeitos passaram ao desenvolverem as atividades e não somente no resultado final, que, no caso, corresponderia ao arquivo final com a programação de cada atividade. Essa importância dada ao processo evidencia mais uma característica da pesquisa qualitativa, isto é, a de ser uma investigação descritiva (BOGDAN; BIKLEN, 1994), em que os dados não são números, mas sim imagens, palavras oriundas de gravações, fala dos estudantes e notas de campo. O processo lógico indutivo-dedutivo (BOGDAN; BIKLEN, 1994; CRESWELL, 2014), apontado por Creswell (2014) como uma das características comuns da pesquisa qualitativa, constituiu-se quando avancei e retrocedi entre os temas da pesquisa e os dados, a fim de estabelecer as relações que ambos poderiam ter, ou seja, a busca pelo *como* as relações, de fato, dão-se, constituiu-se quando procurei, investiguei e explorei tanto a literatura dos temas quanto as atividades dos estudantes. Logo, nas próximas duas seções, será apresentado o referencial teórico que embasou o desenvolvimento dessa investigação.

3 COMPUTADOR E PENSAMENTO: REORGANIZAÇÃO DE IDEIAS

Essa seção tem o intuito de discorrer sobre ideias como: de que modo o uso do computador afeta a cognição humana (TIKHOMIROV, 1981); a interação com diferentes tecnologias intelectuais promove estilos de pensamento distintos (LEVY, 1997) e abordagem experimental-com-tecnologia (BORBA; VILLARREAL, 2005).

No final do século XX, houve um grande avanço tecnológico mundial, de modo que, a partir dos anos 1970, os computadores pessoais foram criados e tornaram-se mais comuns em sociedades desenvolvidas. Com isso, caminhou-se para uma integração da tecnologia, em particular da informática, à vida das pessoas, resultando em uma maior dinamicidade das comunicações entre elas. Levando em consideração o papel das técnicas de comunicação dentro da evolução humana, Lévy (1997) denomina tecnologias da inteligência todas as técnicas de transmissão e de tratamento de mensagens que modificam diretamente a interlocução, que alteram nossa representação do conhecimento e, conseqüentemente, o modo como transmitimos essas representações por meio da linguagem, e que transformam nosso modo de conhecer o mundo não apenas no meio externo, mas também em nosso próprio interior, como a escrita, a linguagem e a informática.

Segundo o autor, uma determinada configuração de tecnologias intelectuais em um intervalo de tempo abre certos campos de possibilidades a uma cultura. Desse modo, ele se questiona sobre “como” e “por quê” diferentes tecnologias da inteligência podem gerar modos de pensamentos distintos. Assim, Lévy (1997) faz uma analogia entre os estilos de pensamento das sociedades estritamente orais e das sociedades letradas.

Em uma sociedade que se comunica, exclusivamente, de forma oral, a palavra não está ligada apenas à expressão das pessoas ou à sua comunicação cotidiana, mas também à estrutura cultural delas. A palavra tem como função carregar a memória social desses indivíduos, de forma que “a inteligência, nestas sociedades, encontra-se muitas vezes identificada com a memória, sobretudo com a auditiva” (LÉVY, 1997, p. 77). O autor, fundamentado nos pressupostos da psicologia cognitiva, busca caracterizar a memória declarativa, fazendo uma distinção entre memória de curto e de longo prazo. A primeira está vinculada à atenção instantânea, por exemplo, quando lemos um número de telefone e o repetimos mental ou oralmente até que o disquemos.

Já a memória de longo prazo, por exemplo, é quando lembramos de um número de telefone em um momento oportuno, estando vinculada a uma única e enorme rede associativa do cérebro, de forma que “quando uma nova informação ou um novo fato surgem diante de nós, devemos, para gravá-lo, construir uma representação dele” (LÉVY, 1997, p. 79). Para o autor, ao buscarmos uma lembrança ou uma informação, devemos ter uma representação do que procuramos já conservada em nossa mente e construir um caminho de associações que nos leve até ela. Além disso, quanto mais conexões mentais e quanto mais complexas e numerosas forem as associações elaboradas nesse caminho, melhor será a capacidade posterior de lembrarmos de um fato.

Nas culturas orais, qualquer asserção, conto ou narrativa que não seja periodicamente retomada e repetida em voz alta está destinada ao esquecimento e, desse modo, a passagem temporal supõe um incessante movimento de recomeço, reiteração (LÉVY, 1997). O estilo de pensamento nessas sociedades se manifesta na importância atribuída aos tipos de representação em ritos e narrativas, que carregam características emocionais e dramáticas.

Nas sociedades letradas, segundo Lévy (1997, p. 87), a escrita “reproduz, no domínio da comunicação, a relação com o tempo e o espaço”. Isso porque ela permitiu que os discursos pudessem ser separados das circunstâncias em que foram produzidos, tornando possível que emissor e receptor estivessem distantes no tempo e no espaço. Nas palavras de Lévy (1997, p. 89), “de geração em geração, a distância entre o mundo do autor e o do leitor não para de crescer”. Ao separar a mensagem de um contexto, há uma impossibilidade de o emissor e de o receptor interagirem na construção do discurso e, com isso, a atribuição de sentido tem um papel fundamental no processo de comunicação, fazendo com que a escrita busque tornar o texto autossuficiente.

De acordo com o autor, “a notação escrita torna muito mais cômoda a conservação e a transmissão de representações modulares separadas, independentes de ritos ou narrativas” (LÉVY, 1997, p. 91). Tal fato fez com que, a partir dela, surgissem teorias nos mais diversos campos do saber, tornando-a um gênero teórico e um modo de difusão e de organização do conhecimento. Além disso, com a escrita, as representações ocorrem de outras formas, visto que estão inscritas no papel, perduram e não precisam mais de diversas conexões, como na memória de longo prazo, podendo agora ser transmitidas e durar de modo autônomo

(LÉVY, 1997). Assim, o exercício da memória não se refere mais às lembranças humanas, ou seja, “a memória separa-se do sujeito ou da comunidade tomada como um todo [...] A objetivação da memória separa o conhecimento da identidade pessoal ou coletiva” (LÉVY, 1997, p. 95). Significa dizer que a memória, em uma sociedade letrada, repousa em seus manuscritos, textos impressos, etc.

O estilo de pensamento das sociedades letradas é sistemático, lógico, de modo que tal “pensamento lógico’ corresponde a um estrato cultural recente ligado ao alfabeto e ao tipo de aprendizagem (escolar) que corresponde a ele” (LÉVY, 1997, p. 93). Segundo o filósofo, a escrita constitui os domínios de conhecimento e, sem ela, “não há datas nem arquivos, não há listas de observações, tabelas de números, não há *códigos* legislativos, nem *sistemas* filosóficos e muito menos crítica destes sistemas” (LÉVY, 1997, p. 96, grifos do autor). Com a escrita e o advento da impressão, o tempo se torna mais linear, histórico, com calendários, datas e arquivos que instauram referências fixas. Isso porque, antes dela, os debates eram verbais, mas, com ela, passaram a uma caracterização e a uma demonstração visual, como no meio científico, com publicações, artigos e práticas de laboratório. O desenvolvimento dessas atividades continua e se amplia com a chegada do computador, uma tecnologia intelectual que favorece a visualização (LÉVY, 1997).

O autor afirma que o computador se tornou um dispositivo técnico por meio do qual percebemos o mundo “não apenas em um plano empírico [...] mas também em um plano transcendental, pois hoje, cada vez mais concebemos o social, os seres vivos ou os processos cognitivos através de uma matriz de leitura informática” (LÉVY, 1997, p. 15). O uso frequente desse dispositivo o tornou uma ferramenta de informação e de interlocução, só que, diferente da oralidade e da escrita, sua interface permite a não linearidade no processo de comunicação, que “visa mais diretamente ao plano das representações” (LEVY, 1997, p. 21). Dito isso, Lévy (1997) cria a metáfora do hipertexto, ou seja, o entendimento de que o *layout* da comunicação possui uma complexidade em sua exposição/reprodução, como uma rede com nós e conexões.

Com o computador se tornando mais comum na vida das pessoas, vieram questionamentos acerca da influência e das consequências que ele poderia causar em termos cognitivos, isto é, dúvidas sobre como esse dispositivo poderia impactar nos processos da mente envolvidos na percepção, na representação, no pensamento, nas associações e lembranças, na solução de problemas, etc. Para

responder a essa questão, foi necessário comparar como seres humanos e computadores resolvem um mesmo problema (TIKHOMIROV, 1981) e, com isso, surgiram duas teorias: a teoria informacional do pensamento e a teoria psicológica do pensamento.

A primeira é, segundo Tikhomirov (1981), formulada como a descrição do pensamento de forma análoga ao funcionamento do computador, ou seja, nela, os processos complexos do pensar são entendidos como sendo constituídos por processos elementares, associados à informação, a qual, por sua vez, é processada pelo computador. Então, ao considerar tal teoria verdadeira, compreende-se que o computador suplementa o pensamento humano no processo da informação, aumentando o volume e a velocidade desse processo, levando em conta que o computador só afeta quantitativamente o desenvolvimento da atividade mental (TIKHOMIROV, 1981). Porém, ao comparar a forma como um humano e um computador resolvem um problema, a teoria informacional não se atenta para os procedimentos heurísticos que utilizamos nesse processo, nem leva em consideração as diferenças físico-biológicas entre o cérebro e o computador, ignorando, por exemplo, que os processos mentais (na resolução de uma problemática) podem mudar de pessoa para pessoa, além de o cérebro estar submetido a questões neurofisiológicas e bioquímicas que também modificam as atividades mentais (TIKHOMIROV, 1981).

Diante desse fato, deixou-se de lado a ideia da abordagem informacional na discussão da influência dos computadores no desenvolvimento da atividade intelectual humana e procurou-se compreendê-la psicologicamente, por meio da análise experimental e teórica dos processos de pensamento, na busca por expressar a real estrutura da inteligência humana (TIKHOMIROV, 1981). Inicialmente, o pensamento foi visto como a ação de resolver um problema, o qual, em geral, é definido em termos de um objetivo e esse, muitas vezes, não é concebido a princípio na resolução da questão, o que pode torná-la indefinida e, assim, gerar complexas interpretações (TIKHOMIROV, 1981). Portanto, formular e obter um objetivo é uma manifestação importante na atividade mental e, conseqüentemente, pensar não é apenas resolver um problema, mas também formulá-lo (TIKHOMIROV, 1981).

Ao se aprofundar nas condições de um problema, bem como no que um humano se preocupa ao tentar resolvê-lo, pode-se dizer que, pensando em um objeto (coisa e/ou pessoa), atribuímos valores a ele e às ações que temos com ele

(TIKHOMIROV, 1981). Além disso, as condições do problema podem incluir a correspondência de valores dos objetos e a intenção da pessoa que o resolve/formula, ou seja, não há neutralidade no contexto da resolução de uma problemática por parte do ser humano, o que cria uma distinção qualitativa de sua atividade mental quando comparada ao processamento da informação (TIKHOMIROV, 1981).

Antes de considerarmos o efeito do computador nos processos mentais, precisamos entender o uso das ferramentas nas ações humanas, ferramentas essas que, segundo a psicologia soviética, são o mais importante componente de nosso comportamento. Isso porque elas criam uma singularidade qualitativa em nossa ação, ou seja, não só são adicionadas, mas também transformam nossas ações (TIKHOMIROV, 1981). Vê-se que a importância não está, isoladamente, na atividade humana de pensar ou na atividade de um computador na resolução de um problema, mas no sistema humano-computador, interpretando este último como uma ferramenta da atividade intelectual humana que tem sua própria estrutura mediada, porém atribui-lhe novos significados, surgindo, assim, novos questionamentos tanto acerca da mediação dos processos mentais pelo uso do computador quanto sobre a introdução de mudanças na estrutura do pensamento a partir de tais significados. Nesse sistema, há um recebimento imediato (pela atividade humana) de informações sobre os resultados das ações, denominado *feedback* (TIKHOMIROV, 1981), mecanismo passível de adquirir informação e que pode ser regulado pela atividade intelectual humana, tornando seus processos mais controlados. A mediação do computador nas relações humanas possibilita que a atividade humana tenha, por sua vez, uma estrutura mais complexa e que a comunicação entre as pessoas também mude, já que a memória, o armazenamento de informações e a rapidez do *feedback* reorganizam-na. Por isso, para Tikhomirov (1981), a influência dos computadores deve ser examinada em termos ontológicos e funcionais na atividade mental.

O computador se desenvolveu de tal modo que sua interface se tornou mais amigável e seus programas começaram a desempenhar um papel de tecnologia intelectual, isto é, “eles reorganizam, de uma forma ou de outra, a visão de mundo de seus usuários e modificam seus reflexos mentais. As redes informáticas modificam os circuitos de comunicação e de decisão das organizações” (LÉVY, 1997, p. 54). O referido autor não busca definir a informática, pois ela está sempre

se atualizando, de modo que seu aspecto “mais determinante para a evolução cultural e [para] as atividades cognitivas é sempre o mais recente, [visto que] relaciona-se com o último envoltório científico, a última conexão possível [...]” (LÉVY, 1997, p. 102). Para ele, a principal característica relacionada à informática é a digitalização, a qual alcança tanto as técnicas de comunicação quanto as de processamento das informações e possibilita, por exemplo, que imagem e som sejam transformados. Na imagem digitalizada, é possível mudar a cor, o tamanho, a textura, criar animações, comentários, relacioná-la a algum som, sem precisar de grande recurso financeiro ou de conhecimento especializado.

A simulação de modelos é outro aspecto relevante destacado por Lévy (1997) como específico do digital. Um modelo digital é avaliado pela sua utilidade e eficácia em relação a um objetivo específico, e não por critérios de verdadeiro ou falso, já que há fatores distantes dessa ideia de dualidade que intervêm na avaliação, como “a facilidade de simulação, a velocidade de realização e modificação, as conexões possíveis com programas de visualização, de auxílio à decisão ou ao ensino...” (LÉVY, 1997, p. 120). Além disso, a forma interativa do modelo digital, devido à sua plasticidade, dinamicidade e autonomia de ação e reação, permite que ele seja explorado. O conhecimento por meio da simulação é diferente dos demais, pois possibilita que o usuário aja diretamente no objeto de seu interesse dentro do modelo, manipulando os parâmetros e simulando as circunstâncias possíveis. Desse modo, para o autor, a simulação digital, quando comparada à estrutura do alfabeto, é auxiliar à imaginação e potencializa o raciocínio, uma vez que, enquanto a escrita representa a formalidade e a estaticidade do pensamento no papel, ela está relacionada às etapas anteriores, as quais envolvem imaginação, tentativas e erros.

Conseguimos antecipar as consequências de nossas ações devido à capacidade que temos de simular mentalmente os possíveis movimentos e reações do mundo exterior. Assim:

Parece mais plausível que as pessoas construam *modelos mentais* das situações ou dos objetos sobre os quais estão raciocinando e depois explorem as diferentes possibilidades dentro destas construções imaginárias. A simulação, que podemos considerar como uma imaginação auxiliada por computador, é portanto ao mesmo tempo uma ferramenta de ajuda ao raciocínio muito mais potente que a velha lógica formal que se baseava no alfabeto (LÉVY, 1997, p. 124, grifos do autor).

Mesmo que tentemos estimular ao máximo nossa imaginação e memória a fim de construir tais modelos, a simulação permite que exploremos apenas aqueles mais complexos e em maior número, posto que ela remete a um aumento do poder da imaginação e da intuição. Porém, deve ficar claro que o conhecimento por simulação é somente válido em um quadro epistemológico, pois um “modelo é construído para um determinado uso de determinado sujeito em um momento dado” (LÉVY, 1997 p. 125).

Dessa produção da inteligência humana por meio do uso das tecnologias da inteligência, o hipertexto e a cultura informático-midiática formam uma ecologia cognitiva que pode ser entendida como um sistema/organismo envolvendo as técnicas de comunicação, as instituições sociais e o pensamento individual, de modo a refletir como esses elementos se articulam e estruturam um “coletivo pensante homem-coisas” (LÉVY, 1997), em que cada um desses elementos é ator no processo de desenvolvimento da inteligência coletiva.

Mas o que é inteligência? E por que coletiva? Lévy (1997) afirma que a inteligência é o resultado de uma rede complexa de interações entre muitos atores humanos, biológicos e técnicos. “Não sou ‘eu’ que sou inteligente, mas ‘eu’ com o grupo humano do qual sou membro, com minha língua, com toda uma herança de métodos e tecnologias intelectuais” (LÉVY, 1997, p. 135). Um ator dentro dessa coletividade é aquilo/algo considerado “capaz de produzir uma diferença em uma rede [...] e todo ator definirá a si mesmo pela diferença que ele produz. Esta concepção [...] nos leva [...] a pensar de forma simétrica os homens e os dispositivos técnicos” (LÉVY, 1997, p. 137). As máquinas desenvolvidas pelos humanos contribuem para formar e estruturar o funcionamento das sociedades e também as atividades e funções exercidas pelas pessoas e, por isso, tais dispositivos são atores em uma sociedade que não é mais exclusivamente humana. As técnicas agem direta e indiretamente sobre a ecologia cognitiva, de modo que uma modificação técnica é uma modificação da coletividade cognitiva.

Essa simetria entre seres humanos e dispositivos técnicos pode ser estendida às instituições, ou seja, há uma certa forma de analogia entre a atividade das instituições de uma coletividade e as operações cognitivas de um organismo, em que ambas mantêm uma ordem no ambiente que existem. A cultura, segundo Lévy (1997, p.143), fornece um grande aparato aos indivíduos, como:

línguas, sistemas de classificação, conceitos, analogias, metáforas, imagens, evitando que tenhamos que inventá-las por conta própria. As regras jurídicas ou administrativas, a divisão do trabalho, a estrutura hierárquica das grandes organizações e suas normas de ação são tipos de memória, de raciocínio e de tomada de decisão automáticas, incorporadas à máquina social e que economizam certa quantidade de atividade intelectual dos indivíduos.

Dessa maneira, as instituições são tecnologias intelectuais, já que fundamentam, classificam, ordenam e constituem um poder de ação no indivíduo, fazendo-o proceder as fundações culturais que comandam sua compreensão do real.

Além disso, Lévy (1997) defende o uso das tecnologias informáticas para usos educativos, pois a dinamicidade dos hipertextos faz com que suas representações e interfaces favoreçam a manipulação, diferentemente de um recurso estático, como lápis e papel. Para o autor, o envolvimento do estudante no processo de aprendizagem tem um papel fundamental e a interatividade com as tecnologias digitais gera uma navegação em seu conteúdo devido à sua dimensão reticular, o que “favorece uma atitude exploratória, ou mesmo lúdica, face ao material a ser assimilado” (LÉVY, 1997, p. 40).

No âmbito da Educação Matemática, Borba e Villarreal (2005), fundamentados em Tikhomirov (1981), Lévy (1997), entre outros, apresentam argumentos sobre a importância da experimentação para os processos de ensino e de aprendizagem de Matemática. Os autores realçam o papel das tecnologias da inteligência na produção de conhecimento, argumentando que antes de formalizar uma demonstração com raciocínios, há etapas de testes e experimentações vinculadas às investigações de descobertas, na busca por validar ou refutar uma hipótese. Partindo da premissa de que o ser humano está cercado por tecnologias, a ideia é que, ao se produzir conhecimento, está-se interagindo com humanos e mídias, sejam elas a oralidade, a escrita ou a informática. Borba e Villarreal (2005) ainda afirmam que essas tecnologias não são coadjuvantes ou mediadoras no processo de produção de conhecimento, ficando externas aos sujeitos, mas permeando-os.

Assim, ao pensar em um *software* para uso educacional, por exemplo, pode-se considerá-lo como coator no processo de conhecimento produzido pelo ser humano, já que seu uso reorganiza o pensamento de seu usuário. A tecnologia tem um componente humano, pois é criada e desenvolvida por ele, bem como o ser

humano tem um forte componente técnico. Portanto, não se deve implicar uma disjunção entre humanos e tecnologias, principalmente se quisermos entender o papel do computador na sociedade e, em particular, na educação (BORBA; VILLARREAL, 2005).

4 CONEXÕES ENTRE PENSAMENTO COMPUTACIONAL, SCRATCH E MATEMÁTICA

Essa seção tem a finalidade de conceituar o que é pensamento computacional e evidenciar sua relação com o *software Scratch* e com a Matemática. Nos diversos textos da literatura, foram encontrados termos como: características, habilidades (*skills, abilities*), vocabulário do PC (CT vocabulary), *concepts*, todos eles com o intuito de identificar ou de classificar as particularidades do pensamento computacional. Porém, neste trabalho, não os distinguirei, tratando-os apenas como *conceitos* computacionais ou *conceitos* do pensamento computacional.

4.1 PENSAMENTO COMPUTACIONAL

O termo pensamento computacional (*computational thinking*) foi mencionado pela primeira vez em 1980, por Seymour Papert, na obra *Mindstorms: children, computers, and powerful ideas*. Tal vocábulo também aparece em outro de seus livros, intitulado *The Children's Machine: Rethinking School in the Age of the Computer* (A Máquina das Crianças: Repensando a Escola na Era da Informática), de 1993.

O referido autor era professor do *Massachusetts Institute of Technology* (MIT), do *Media Lab*, do Laboratório de Inteligência Artificial e foi um dos responsáveis pela criação da linguagem de computador Logo, na qual há um subconjunto contendo comandos para programar uma tartaruga. A ideia era que a criança se comunicasse com o computador, dando-lhe coordenadas, a fim de que a tartaruga se movimentasse. Dessa forma, criava-se um ambiente de aprendizado diferente daquele da sala de aula tradicional.

Isso porque o educador vê que, com o advento do computador pessoal, este seria, cedo ou tarde, introduzido nas escolas. Assim, seria necessário entender qual o seu papel nesse ambiente, além de buscar compreender sua função no processo de ensino e de aprendizagem.

Nesse sentido, discute como o computador pode afetar o modo como as pessoas pensam e aprendem:

a presença do computador pode contribuir para os processos mentais não apenas instrumentalmente, mas de maneiras conceituais mais essenciais, influenciando a maneira como as pessoas pensam mesmo quando estão muito distantes do contato físico com um computador (PAPERT, 1980, p. 4, tradução nossa)³.

Papert (1980) entende que programar um computador é comunicá-lo em uma linguagem que ambos, usuário e máquina, possam entender, afirmando ainda que “aprender idiomas é uma das coisas que as crianças fazem melhor. Toda criança normal aprende a falar. Por que, então, uma criança não deveria aprender a ‘conversar’ com um computador?⁴” (PAPERT, 1980, p. 6, tradução nossa). Nessa analogia do programar com o aprender uma nova língua, o autor alega que os pequenos também têm dificuldades em aprender sua própria língua escrita e, assim, não é necessário temer nem a aquisição de conhecimentos em programação, nem a utilização do computador.

Ainda segundo o autor, o uso dessa máquina traz mudanças tanto para o interior da escola quanto para fora dela. Atualmente, podemos dizer que a utilização dos *smartphones* também impacta os espaços por onde circula, mas de uma forma diferente, já que são dispositivos móveis, de fácil transporte. Além disso, alguns dos *softwares* que antes só pertenciam ao computador, hoje são aplicativos de celulares e *tablets* (ROMANELLO, 2016). Desse modo, os estudantes não precisam mais ir ao laboratório da escola para utilizar o *GeoGebra*, por exemplo, visto que podem, seja dentro da sala de aula seja fora dela, fazer atividades que anteriormente só seriam exequíveis na frente de um computador. Apesar das mudanças que os computadores trouxeram e ainda trazem para o ambiente escolar, Papert (1980) estava preocupado com a mente ou, mais especificamente, em buscar compreender como os movimentos e culturas intelectuais se definem e crescem.

O papel atribuído por Papert (1980, p. 9, grifos do autor, tradução nossa) ao computador era o “de um *portador* de ‘germes’ ou ‘sementes’ culturais cujos produtos intelectuais não precisarão de suporte tecnológico depois que se enraizarem em uma mente em crescimento ativo”⁵, ou seja, a partir do uso da

³ The computer presence could contribute to mental processes not only instrumentally but in more essential, conceptual ways, influencing how people think even when they are far removed from physical contact with a computer.

⁴ And learning languages is one of the things children do best. Every normal child learns to talk. Why then should a child not learn to “talk” to a computer?

⁵ A carrier of cultural “germs” or “seeds” whose intellectual products will not need technological support once they take root in an actively growing mind.

máquina, há uma modificação do produto intelectual humano, lembrando que tal mudança prescinde da presença física do computador, ideia que vai ao encontro das proposições de Tikhomirov (1981). Além disso, a pessoa (adulto) que utiliza o computador germina essas transformações em outras pessoas (crianças).

Ao programar a tartaruga do Logo⁶, isto é, ao ensiná-la a executar, ou, ainda, ao ensinar o computador a pensar, “as crianças embarcam em uma exploração de como elas mesmas pensam [...] Quando uma criança aprende a programar, o processo de aprendizado é transformado. Torna-se mais ativo e autodirigido⁷” (PAPERT, 1980, p. 19, tradução nossa). Esse processo torna-se epistemológico, ou seja, a criança pensa sobre o pensamento para fazer a tartaruga executar algo.

Papert (1980) se vale da crítica de que o uso do computador gera um pensamento mecânico para afirmar que, em algumas situações, tal pensamento pode ser considerado apropriado e útil. Ao entendê-lo passo a passo, o estudante é capaz de distingui-lo de outros tipos.

O exercício pode levar a uma maior confiança sobre a capacidade de escolher um estilo cognitivo adequado ao problema. A análise do “pensamento mecânico” e como ele é diferente de outros tipos e práticas com a análise de problemas pode resultar em um novo grau de sofisticação intelectual (PAPERT, 1980, p. 27, tradução nossa)⁸.

Adquirindo a capacidade de distinguir “estilos de pensamento”, o estudante, ao enfrentar um problema, pode escolher aquele que considera mais eficaz (PAPERT, 1980). A partir do uso do computador, a criança reorganiza seu pensamento à medida que escolhe como enfrentar um desafio/problema educacional ou cotidiano. Essa reorganização fornece um aspecto qualitativo muito superior aos aspectos quantitativos que o computador (ou outro dispositivo, como o celular) pode fornecer, ou seja, as ideias de Papert (1980) corroboram a teoria psicológica do pensamento trazida por Tikhomirov (1981).

Além disso, quando Papert (1980) apresenta o conceito do pensar sobre o pensar, referindo-se ao movimento mental que a criança faz para programar a

⁶ Logo é uma linguagem de programação voltada ao ambiente educacional, usada para comandar um cursor, normalmente representado por uma tartaruga, com o propósito de ensinar-lhe novos procedimentos além daqueles que já conhece, a fim de criar desenhos ou programas.

⁷ Children embark on an exploration about how they themselves think [...] when a child learns to program, the process of learning is transformed. It becomes more active and self-directed.

⁸ The exercise can lead to greater confidence about the ability to choose a cognitive style that suits the problem. Analysis of “mechanical thinking” and how it is different from other kinds and practice with problem analysis can result in a new degree of intellectual sophistication.

tartaruga do Logo, ele enfatiza o processo de transformação da linguagem, isto é, o processo por meio do qual a criança, com sua linguagem, ensina a tartaruga a se mover na linguagem do *software*. Tais ideias se assemelham às de Lévy (1997), pois tal processo pode ser compreendido como uma técnica de transmissão e tratamento de mensagem que modifica a comunicação entre a criança e o computador, alterando suas representações mentais e, conseqüentemente, a forma como as transmite para outra criança ou pessoa, por meio de sua linguagem.

De fato, Seymour Papert (1980) não define o termo pensamento computacional, ele cria uma ideia do que seria o uso do computador educacionalmente e, a partir disso, como mudaríamos nossa forma de pensar, não necessariamente na presença física da máquina. Foi somente após um artigo publicado em 2006 na revista *Communications of the ACM*, por Jeannette Wing, à época, diretora em pesquisas computacionais do *National Science Foundation* (NSF), que o termo foi amplamente divulgado e reconhecido. Neste trabalho, a autora traz vários significados para o que compreende como pensamento computacional, ressaltando algumas de suas características, como resolução de problemas; pensamento recursivo; interpretação dos dados; abstração e decomposição; raciocínio heurístico.

Wing (2006) deixa claro que PC é um conceito, isto é, que ele é mais do que programação, tratando-se do modo como os humanos pensam e não os computadores. Significa dizer que “pensamento computacional é uma forma para seres humanos resolverem problemas; não é tentar fazer com que seres humanos pensem como computadores” (WING, 2006, p. 4). Para ela, PC é, portanto, uma ideia e não uma ferramenta, é a junção dos fundamentos da lógica da Computação com o pensamento lógico para resolver problemas.

A subseção a seguir ainda traz a conceituação de PC, porém, focalizando o momento em que autores e instituições trazem ao debate a Educação Básica, sendo possível, então, evidenciar sua relação com a Matemática.

4.2 A RELAÇÃO ENTRE PENSAMENTO COMPUTACIONAL E MATEMÁTICA

Encontrar uma definição unânime para pensamento computacional no meio científico não tem sido uma tarefa simples, já que, à medida que mais pessoas,

pesquisadores e organizações se dedicam ao assunto, tal definição tende, naturalmente, a se modificar.

A *International Society for Technology in Education* (ISTE) e a *American Computer Science Teachers Association* (CSTA) fornecem aos professores da Educação Básica (K-12) recursos que mostram uma definição operacional para pensamento computacional, identificam conceitos relacionados a ele, além de trazerem experiências de aprendizagem em PC. Essa definição oferece uma estrutura e um vocabulário que se assemelham às atividades relacionadas à Educação Básica e suas características são:

formular problemas de forma que seja possível o uso de computadores e ferramentas para ajudar a resolvê-los; organização lógica e análise dos dados; representar os dados através de abstrações como modelos e simulações; automatizar soluções através do pensamento algorítmico (uma série de etapas ordenadas); identificar, analisar e implementar possíveis soluções com o objetivo de alcançar a combinação mais eficiente e eficaz de etapas e recursos; generalizar e transferir esse processo de solução de problemas para uma ampla variedade de problemas (ISTE; CSTA, 2011, p. 7).

Tais conceitos “são apoiados e aprimorados por várias disposições ou atitudes que são dimensões essenciais do pensamento computacional” (ISTE; CSTA, 2011, p. 7). São elas:

confiança ao lidar com a complexidade; persistência no trabalho com problemas difíceis; tolerância à ambiguidade; capacidade de lidar com problemas em aberto e de se comunicar e trabalhar com outras pessoas para alcançar um objetivo ou solução comum (ISTE; CSTA, 2011, p. 7).

Além dos conceitos, as organizações supracitadas apresentam as experiências de aprendizagem do pensamento computacional, com o intuito de mostrar como ele pode aparecer em vários níveis e áreas de ensino. Desse modo, elas fornecem orientação e compreensão aos professores, para que possam examinar seu próprio trabalho diário.

Outros autores também têm procurado identificar os conceitos do pensamento computacional. Barr e Stephenson (2011), por exemplo, buscam entender o que está envolvido nele e qual é o papel da comunidade educacional ao trazê-lo para a Educação Básica (K-12). As autoras corroboram a ideia de que os estudantes vivem e viverão sobre influência da computação e que os avanços dela influenciarão seu

futuro trabalho. Desse modo, exigir-se-ão indivíduos mais instruídos, não havendo por que ensinar-lhes esses conceitos somente na faculdade.

Porém, segundo Barr e Stephenson (2011), pensar em incorporá-los ao currículo da Educação Básica exige esforços em duas direções: a primeira delas é referente à política educacional, superando obstáculos relativos à infraestrutura e incluindo, por exemplo, o PC na formação inicial e continuada de professores; já a segunda direção seria no sentido de fornecer recursos aos docentes, tais como materiais escolares, uma definição clara do que vem a ser o conceito de pensamento computacional, bem como de trazer exemplos relevantes e apropriados que demonstrem como integrar “[...] esses novos conceitos, inicialmente em sua própria área de conteúdo e conhecimento pedagógico, e depois nos conteúdos e práticas da sala de aula”⁹ (BARR; STEPHENSON, 2011, p. 53, tradução nossa). Isso poderia auxiliar o desenvolvimento desses profissionais, agindo diretamente em suas atividades diárias e, conseqüentemente, impactando no ambiente escolar e, portanto, na sala de aula.

Ao identificar os conceitos relacionados ao PC, a ISTE e a CSTA enfatizam que suas atividades podem ser praticadas e desenvolvidas em todas as disciplinas e com todas as idades, corroborando as ideias de Barr e Stephenson (2011), que exemplificam a incorporação de tais conceitos nas tarefas de várias disciplinas, como nas de Ciência da Computação, de Matemática, de Ciência, de Estudos Sociais e de Artes da Linguagem¹⁰. O Quadro 2 sintetiza os nove conceitos identificados e definidos pela ISTE e CSTA (2011), bem como evidencia a relação que as autoras fizeram com Ciência da Computação e Matemática.

Quadro 2 – Síntese dos conceitos do pensamento computacional

	CSTA e ISTE (2011)	BARR e STEPHENSON (2011)	
Conceitos do PC	Definição	Ciência da Computação	Matemática
Coleção de dados	Reunir informação apropriada.	Encontrar uma fonte de dados para uma área com problemas.	Encontrar uma fonte de dados para uma área problemática, como, por exemplo, jogar moedas ou dados.

⁹ [...] these new concepts, first into their own sphere of content and pedagogical knowledge, and then into their classroom content and practice.

¹⁰ *Language Arts* – o estudo da gramática, da composição, da ortografia e da oratória é normalmente ensinado como disciplina única nos Ensinos Fundamental e Médio.

Análise de dados	Compreender os dados, encontrar padrões e tirar conclusões.	Escrever um programa para fazer cálculos estatísticos básicos em um conjunto de dados.	Contar ocorrências de lançamentos, como os de dados, e analisar os resultados.
Representação de dados	Representar e organizar dados em gráficos, tabelas, palavras ou imagens apropriadas.	Usar estruturas de dados como matriz, lista vinculada, pilha, fila, gráfico, tabela de dispersão, etc.	Usar histograma, gráfico de setores, gráfico de barras para representar dados; usar conjuntos, listas, gráficos, etc. para conter dados.
Decomposição do problema	Dividir tarefas em partes menores e gerenciáveis.	Definir objetos e métodos; definir a parte principal do programa e as funções.	Aplicar ordem de operações em uma expressão.
Abstração	Reduzir a complexidade para definir a ideia principal.	Usar procedimentos para encapsular um conjunto de comandos frequentemente repetidos que executam uma função; usar condicionais, <i>loops</i> , recursão, etc.	Usar variáveis em álgebra; identificar fatos essenciais em um problema de palavras; estudar funções em álgebra em comparação com funções em programação; usar a iteração para resolver problemas de palavras.
Algoritmos e Procedimentos	Resolver um problema ou alcançar algum objetivo a partir da tomada de uma série de medidas ordenadas.	Estudar algoritmos clássicos; implementar um algoritmo para uma área problemática.	Fazer divisão longa, fatorando; carregar adição ou subtração.
Automação	Ter computadores ou máquinas para executar tarefas repetitivas ou tediosas.	-	Usar ferramentas como: <i>Geometer's Sketchpad</i> ¹¹ ; <i>StarLogo</i> ¹² ; trechos de código <i>Python</i> ¹³ .
Simulação	Representar ou modelar um processo para executar experimentos utilizando-o.	Animar um algoritmo e fazer a varredura dos parâmetros.	Resolver sistemas lineares; fazer multiplicação de matrizes.
Paralelização	Organizar recursos para realizar tarefas simultaneamente, de modo a atingir um objetivo comum.	<i>Threading</i> ¹⁴ , <i>pipelining</i> ¹⁵ , divisão de dados ou tarefas de forma a serem processados em paralelo.	Representar graficamente uma função em um plano cartesiano e modificar os valores das variáveis.

Fonte: Adaptado de ISTE e CSTA (2011) e Barr e Stephenson (2011).

¹¹ Disponível em: <https://www.keycurriculum.com/>. Acesso em: jul. 2020.

¹² Disponível em: <https://education.mit.edu/project/starlogo-tng/>. Acesso em: jul. 2020.

¹³ Disponível em: <https://www.python.org/>. Acesso em: jul. 2020.

¹⁴ Recurso usado para organizar uma série de funções relacionadas ao mesmo assunto que uma sequência vinculada em um quadro de mensagens.

¹⁵ Uma forma de organização de computadores na qual etapas sucessivas de uma sequência de instruções são executadas por uma sequência de módulos capazes de operar simultaneamente, de modo que outra instrução possa ser iniciada antes que a anterior seja concluída.

No Quadro 2, nas colunas referentes às autoras Barr e Stephenson (2011), há o conteúdo das disciplinas de Ciência da Computação e Matemática do modo como elas imaginaram que seria possível relacioná-lo, em sala de aula, com os nove conceitos acerca do pensamento computacional. Já na coluna das organizações CSTA e ISTE (2011), os conceitos acerca do PC são definidos com base na ideia de que ele é um processo de solução de problemas, como mencionado anteriormente. Com as atividades relativas a esses conceitos, os estudantes podem adquirir vocabulário e disposições para programar, além de reconhecerem quando um computador é capaz de ajudá-los na resolução de um problema (ISTE; CSTA, 2011). Por fim, os cientistas da computação podem ajudar os professores de outras áreas a identificarem os processos dos estudantes na resolução de problemas, tais como algorítmicos, e a verificarem em que medida a computação é passível de se encaixar nas atividades das disciplinas, bem como na interseção delas (BARR; STEPHENSON, 2011).

Além da relação do PC com outras áreas de ensino/outras disciplinas, ele pode ser explorado na Educação com o uso de *software*, mais especificamente do *Scratch*.

4.3 A RELAÇÃO ENTRE PENSAMENTO COMPUTACIONAL E SCRATCH

“O Scratch é uma linguagem de programação *visual*. Ele foi desenvolvido no *Massachusetts Institute of Technology* (MIT) *Media Lab* para tornar o aprendizado de programação mais fácil e divertido” (MARJI, 2014, p. 22, grifo do autor). Ele substitui a programação baseada na digitação de códigos por blocos visuais que podem ser escolhidos, arrastados e encaixados em outros, de modo a fornecer instruções para o computador. Além disso, é um *software* gratuito, que conta com duas versões: a versão *on-line*, acessada por meio de seu *site*, e a versão *off-line* (versão para *desktop*), também baixada através dele, na qual há uma comunidade em que são criados, compartilhados e explorados projetos desenvolvidos por diversas pessoas ao redor do mundo. A interface do *Scratch* e seu funcionamento estão explicados no Apêndice A dessa dissertação.

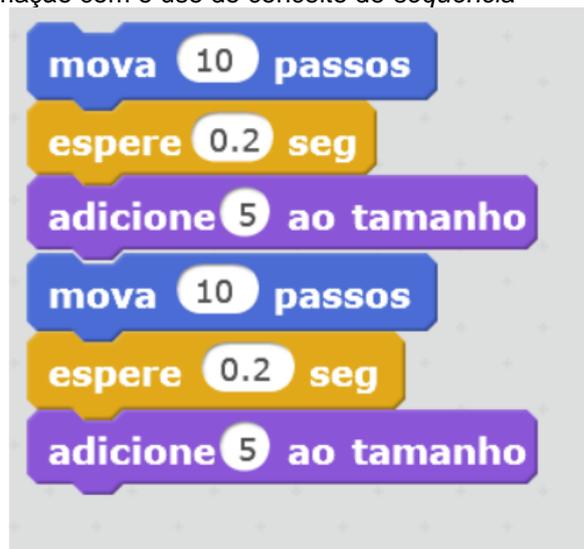
Durante muitos anos, Brennan e Resnick (2012) estudaram as atividades tanto das oficinas do *Scratch* quanto da comunidade *on-line* do *software*. Tal estudo contou com observações e entrevistas com jovens participantes, os quais foram

fundamentais para a compreensão do desenvolvimento e das práticas desses envolvidos (*scratchers*). Assim, elaboraram uma definição de pensamento computacional abarcando três dimensões principais: conceitos, práticas e perspectivas computacionais.

Os conceitos são empregados à medida que os jovens criam com o *Scratch*, desenvolvendo uma mídia interativa no *software* (BRENNAN; RESNICK, 2012). São eles: sequências, ciclos, paralelismo, eventos, condicionais, operadores e dados.

Na programação computacional, uma atividade pode ser compreendida como uma *sequência* de etapas executáveis pelo computador, em que cada uma delas tem sua função e fornece um comportamento específico ou uma ação a ser executada. Por exemplo, na Figura 3, tem-se ilustrada uma sequência de comandos que o ator (o gato) executará. Ele se moverá dez passos para frente, esperará 0,2 segundos, aumentará 5% de seu tamanho e repetirá esses três passos novamente.

Figura 3 – Ator e sua respectiva programação com o uso do conceito de *sequência*



Fonte: O Autor.

Se, no exemplo da Figura 3, quisermos um total de 20 repetições, precisaremos adicionar mais blocos sucessivamente. Desse modo, a sequência de ações (mover, esperar e aumentar de tamanho) será executada diversas vezes. Para fazê-lo, um dos mecanismos de programação utilizados é denominado *ciclo*. Na Figura 4, a seguir, é possível observar uma programação com o uso do bloco “repita”, em que basta definir a sequência que queremos repetir, no caso, mover, esperar e aumentar, e a quantidade de vezes a ser feita, que, no exemplo, são 20.

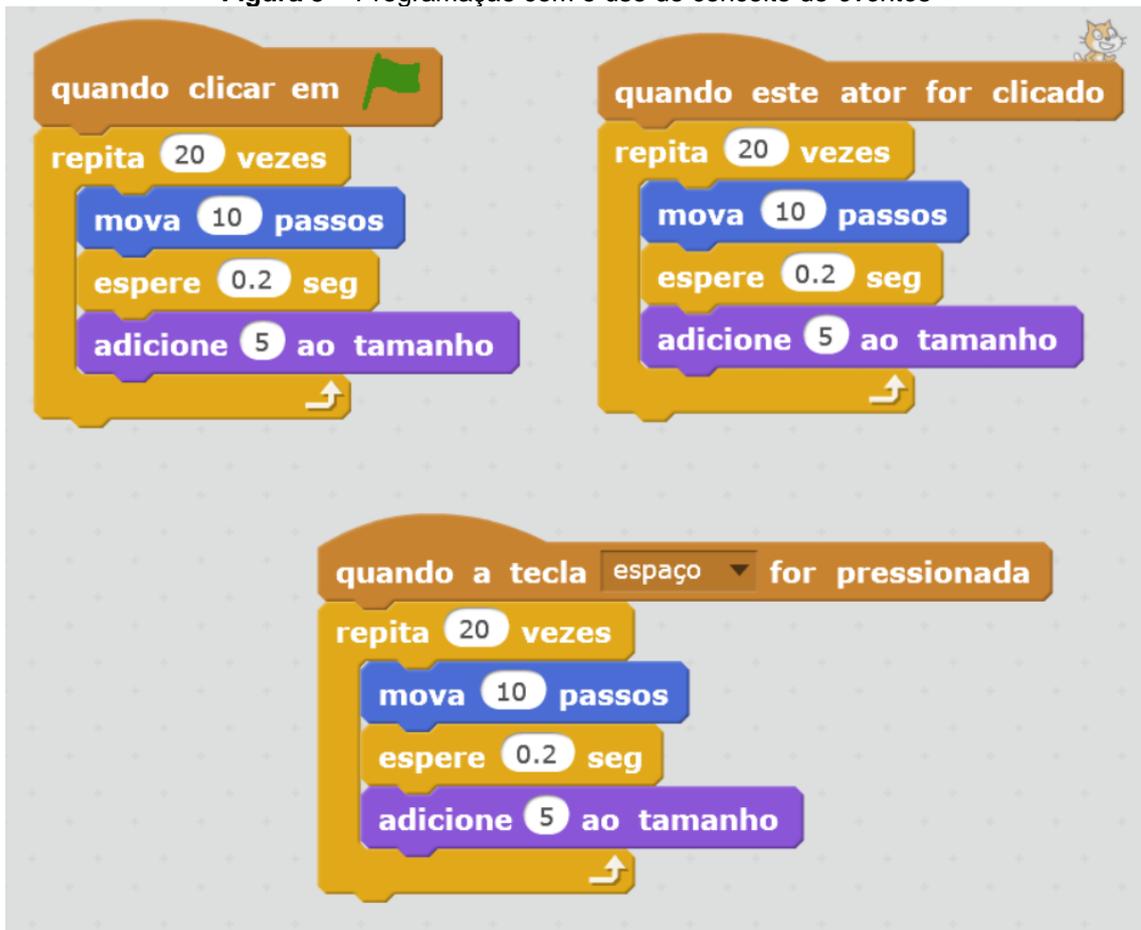
Figura 4 – Ator e sua respectiva programação com o uso do conceito de *ciclos*



Fonte: O Autor.

Os *eventos* são comandos que iniciam acontecimentos de outros blocos. Mais especificamente, dentro do *Scratch*, temos a Paleta Eventos e, nela, há blocos do tipo *trigger*, os quais são colocados no começo de uma pilha de outros blocos e executam aqueles que se encontram abaixo deles. Para mais detalhes, pode ser consultado o Apêndice A desta dissertação. Na Figura 5, são ilustradas três programações com o uso dos eventos, que executam a mesma programação exemplificada na Figura 4, cuja diferença reside no fato de que, para que se desencadeiem as ações, é preciso ocorrer situações diferentes, como clicar na bandeira verde, clicar no ator ou pressionar a barra de espaço.

Figura 5 – Programação com o uso do conceito de eventos



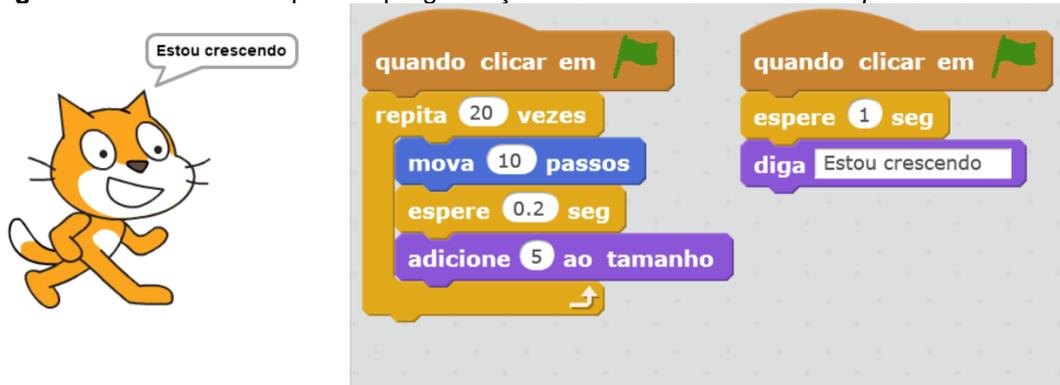
Fonte: O Autor.

O conceito de *paralelismo*, por sua vez, é a execução de várias ações (sequências) ao mesmo tempo. O *Scratch* suporta paralelismo entre atores e em um mesmo ator. Por exemplo, o jogo Pac-Man¹⁶ pode ser simulado no *Scratch*¹⁷. Nele, temos o personagem principal que percorre um labirinto, comendo pontos, pontos energizantes e frutas, fugindo de quatro fantasmas. Enquanto o jogo acontece, cada ator está executando uma série de ações ao mesmo tempo, representando um paralelismo entre atores. Já o paralelismo dentro de um único ator pode ser observado na Figura 6, cuja programação é a mesma da Figura 4, adicionando-se a ela uma sequência em que o ator diz que está crescendo.

¹⁶ PAC-MAN™ & ©1980 BANDAI NAMCO Entertainment Inc. Disponível em: <https://pt.wikipedia.org/wiki/Pac-Man#Personagens>. Acesso em: jul. 2020.

¹⁷ Disponível em: <https://scratch.mit.edu/search/projects?q=pac%20man>. Acesso em: jul. 2020.

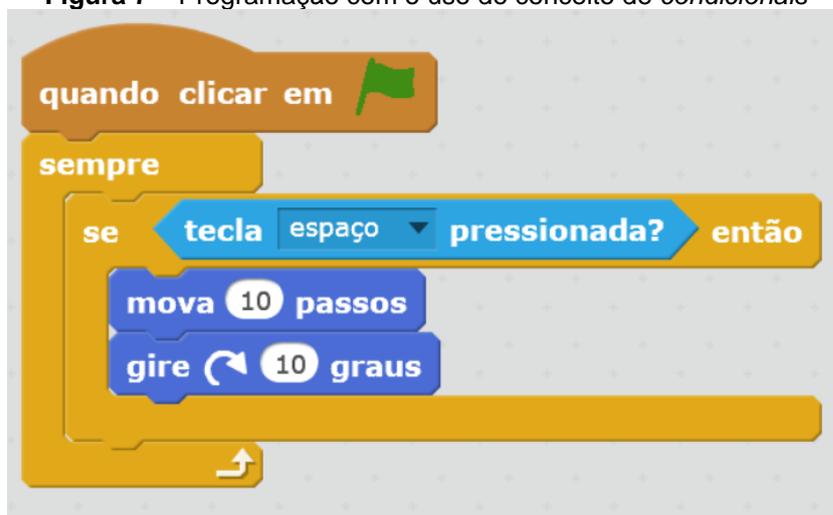
Figura 6 – Ator e sua respectiva programação com o uso do conceito de *paralelismo*



Fonte: O Autor.

O conceito de *condicionais* é referente à “capacidade de tomar decisões com base em determinadas condições, que auxilia a expressão de vários resultados”¹⁸ (BRENNAN; RESNICK, 2012, p. 5, tradução nossa). Um exemplo desse conceito pode ser observado na Figura 7, com o uso do bloco “se” para determinar o movimento do ator¹⁹. Cada vez que a tecla espaço é pressionada, ele se move dez passos e gira dez graus no sentido horário.

Figura 7 – Programação com o uso do conceito de *condicionais*



Fonte: O Autor.

O conceito de *operadores* é referente aos blocos de função, que “fornecem suporte para expressões matemáticas, lógicas e de *string*, permitindo que o programador execute manipulações numéricas e de *string*”²⁰ (BRENNAN; RESNICK, 2012, p. 5, grifo nosso, tradução nossa). A Figura 8 ilustra os blocos da Paleta

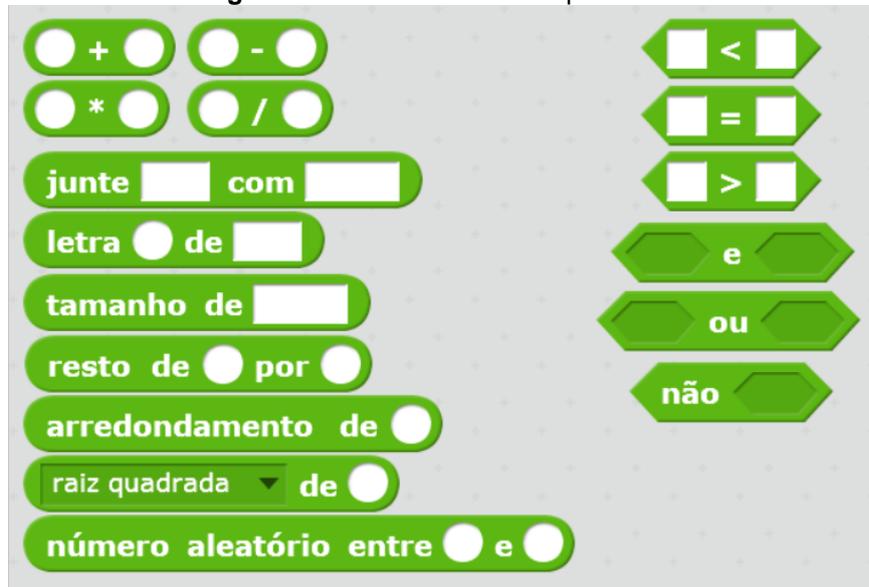
¹⁸ The ability to make decisions based on certain conditions, which supports the expression of multiple outcomes.

¹⁹ A funcionalidade do bloco “se” é descrita no Apêndice A.

²⁰ [...] provide support for mathematical, logical, and string expressions, enabling the programmer to perform numeric and string manipulations.

Operadores, estando, à sua esquerda, os blocos numéricos/*string* e, à sua direita, os booleanos (vide Apêndice A).

Figura 8 – Blocos da Paleta Operadores



Fonte: O Autor.

O conceito computacional de *dados* envolve armazenar, recuperar e atualizar valores, sendo definido em dois conjuntos: variáveis e listas (BRENNAN; RESNICK, 2012). No jogo Pac-Man, por exemplo, a pontuação pode ser vista como uma variável.

Além dos *conceitos*, Brennan e Resnick (2012) definiram as *práticas computacionais*, apoiando-se no que foi visto na participação e na criação dos projetos. A estrutura do PC foi descrita nos processos de construção e nas ações de desenvolver programação. Assim, “as práticas computacionais concentram-se no processo de pensar e aprender, indo além de *o que* você está aprendendo para *como* está aprendendo” (BRENNAN; RESNICK, 2012, p. 7, grifo dos autores, tradução nossa)²¹. Os autores observaram quatro conjuntos principais de práticas: ser “*incremental*” e iterativo, testar e depurar, reutilizar e remixar, abstrair e modularizar.

Ser incremental e iterativo significa que produzir uma mídia interativa no *Scratch* não é algo simples, mas um processo adaptativo. Significa dizer que, conforme surgem problemas na produção da programação, pode-se mudar e alterar

²¹ Computational practices focus on the process of thinking and learning, moving beyond *what* you are learning to *how* you are learning.

o código a qualquer momento. Dessa forma, a solução acaba sendo um procedimento de pequenas etapas e repetições.

A prática de testar e depurar decorre diretamente do fato de podermos alterar os blocos da programação a qualquer momento e testá-los, ou seja, é possível rodar essa programação e ver se está como desejado. Caso não esteja, basta rever os blocos até o problema ser encontrado e, então, fazer a depuração e um novo teste.

Brennan e Resnick (2012) afirmam que reutilizar códigos prontos é uma prática comum em programação e tem aumentado cada vez mais, devido ao fato de a internet fornecer acesso aos trabalhos de outras pessoas. Segundo os autores:

Um objetivo da comunidade on-line do Scratch é apoiar os jovens designers na reutilização e remixagem, ajudando-os a encontrar ideias e códigos a serem construídos, permitindo-lhes criar coisas muito mais complexas do que poderiam criar por conta própria. A reutilização e a remixagem dão suporte ao desenvolvimento de capacidades críticas de leitura de código e provocam questões importantes sobre propriedade e autoria (BRENNAN; RESNICK, 2012, p. 8, tradução nossa)²².

Quando se está logado na comunidade *on-line* do *Scratch* e se acessa um projeto, há a possibilidade de remixá-lo, ou seja, de salvar uma cópia dele e adicioná-la às suas criações para fazer ou não alterações. A Figura 9, a seguir, é constituída pela captura da tela do *site* do *Scratch*, em que se pode observar a interface²³ de um projeto selecionado da comunidade e, ainda, a quantidade de vezes que ele foi remixado, a saber, 805 vezes, tal como dado indicado ao lado do símbolo de uma espiral.

²² One goal of the Scratch online community is to support Young designers in reusing and remixing, by helping them find ideas and code to build upon, enabling them to potentially create things much more complex than They could have created on their own. Reusing and remixing support the development of critical code-reading capacities and provoke important questions about ownership and authorship.

²³ Disponível em: <https://scratch.mit.edu/projects/125751986/>. Acesso em: jul. 2020.

Figura 9 – Captura de tela da interface de um projeto *on-line* do *Scratch*



Fonte: O Autor.

Brennan e Resnick (2012) caracterizam a prática de abstrair e modularizar como construir algo grande, reunindo coleções de peças menores, que são os módulos. “No Scratch, os designers empregam abstração e modularização em vários níveis, desde o trabalho inicial de conceituar o problema até traduzir o conceito em atores individuais e pilhas de código”²⁴ (BRENNAN; RESNICK, 2012, p. 9, tradução nossa). Cada módulo tem o intuito de se compor com outro ou com algum elemento da programação da qual não faz parte (a programação de um outro ator, por exemplo). A afinidade que as partes menores têm umas com as outras ou com o todo facilita a visualização do problema que, então, pode ser solucionado por partes.

A última dimensão definida por Brennan e Resnick (2012) é denominada perspectivas computacionais. Nela, os jovens descrevem suas compreensões de desenvolvimento próprio, a relação com os outros jovens e com o mundo tecnológico, criando, assim, as seguintes perspectivas: expressar, conectar e questionar.

²⁴ In Scratch, designers employ abstraction and modularization at multiple levels, from the initial work of conceptualizing the problem to translating the concept into individual sprites and stacks of code.

O *Scratch* permite que os jovens possam se *expressar* tecnologicamente, visto que “um pensador computacional vê a computação mais do que algo para consumir; computação é algo que eles podem usar para projetar e autoexpressão”²⁵ (BRENNAN; RESNICK, 2012, p. 10, tradução nossa). Para um pensador computacional, a computação é um meio de se comunicar.

A perspectiva de *conectar* mostra que o *Scratch* proporciona enriquecidas interações pessoais. Com as entrevistas e observações, Brennan e Resnick (2012) constataram uma variedade de maneiras pelas quais a prática criativa de um *scratcher* individual se beneficiava do acesso a outras pessoas, seja por meio de interações presenciais seja por redes *on-line*. Além disso, essa conexão é dada de dois modos: *com* o outro e *para* o outro. “Ao criar *com* outras pessoas, [...] podiam fazer mais do que poderiam fazer por conta própria [...]. Ao criar *para* outros, [...] eles apreciaram o fato de outras pessoas estarem se envolvendo e apreciando suas criações [...]”²⁶ (BRENNAN; RESNICK, 2012, p. 10, grifo dos autores, tradução nossa).

A vida cotidiana está cada vez mais permeada por tecnologia, cuja complexidade também vem aumentando, o que, para muitas pessoas, parece ser intangível, incompreensível. Porém, Brennan e Resnick (2012), com a perspectiva computacional do *questionamento*, buscaram indicadores de que os jovens não sentem uma desconexão entre as tecnologias que os cercam e as habilidades para lidar com o mundo tecnológico: “os jovens devem se sentir capacitados para fazer perguntas sobre e com a tecnologia – ‘Eu posso (usar a computação para) fazer perguntas para entender (coisas computacionais) o mundo’”²⁷ (BRENNAN; RESNICK, 2012, p. 10, tradução nossa).

O Quadro 3 sintetiza as dimensões computacionais definidas por Brennan e Resnick (2012).

Quadro 3 – Dimensões computacionais

²⁵ A computational thinker sees computation as more than something to consume; computation is something they can use for design and self-expression.

²⁶ By creating with others, [...] they were able to do more than they could have on their own, [...]. By creating for others, [...] they appreciated that others were engaging with and appreciating their creations [...].

²⁷ Young people should feel empowered to ask questions about and with technology – “I can (use computation to) ask questions to make sense of (computational things in) the world”.

Brennan e Resnick (2012) – Dimensões computacionais					
Conceitos		Práticas		Perspectivas	
Sequências	Expressar uma atividade em etapas.	Ser incremental e iterativo	Ao surgirem problemas em um processo de solução, pode-se mudar e alterar o código a qualquer momento. Assim, a solução acaba sendo um procedimento adaptativo de pequenas etapas e repetições.	Expressar	Para um pensador computacional, a computação é um meio de se expressar, é algo que eles podem usar para <i>design</i> e autoexpressão.
Loops	Mecanismo que executa a mesma sequência várias vezes.				
Paralelismo	Sequências de instruções acontecendo ao mesmo tempo.	Testar e depurar	Pode-se alterar os blocos da programação a qualquer momento e testá-los, ou seja, rodar a programação e ver se está como desejado. Caso não esteja, basta rever os blocos até o problema ser encontrado e, então, fazer a depuração e novo teste.	Conectar	O <i>Scratch</i> proporciona enriquecidas interações <i>com</i> e <i>para</i> o outro, que beneficiam a prática criativa dos jovens por meio de interações presenciais ou redes <i>on-line</i> .
Eventos	Um comando que faz alguma coisa acontecer.				
Condicionais	Tomada de decisões com base em determinadas condições.	Reutilizar e remixar	Reutilizar e reformular códigos já prontos, com o intuito de encontrar ideias e permitir criar coisas mais complexas, além de dar suporte ao desenvolvimento de capacidades críticas de leitura de código e provocar questões sobre propriedade e autoria.	Questionar	Os jovens não sentem a desconexão entre as tecnologias que os cercam e as habilidades para lidar com o mundo tecnológico. Eles são capacitados para fazer perguntas sobre e com a tecnologia, independente de sua complexidade.
Operadores	Fornecem suporte para expressões matemáticas, lógicas e de <i>string</i> .				
Dados	Envolvem armazenar, recuperar e atualizar valores por meio de variáveis e listas.	Abstrair e modularizar	Construir algo grande, reunindo coleções de peças menores, que são os módulos, cujo intuito é se compor com outro módulo ou com alguma parte da programação. Desse modo, o problema pode ser analisado por partes.		

Fonte: Adaptado de Brennan & Resnick (2012).

As dimensões computacionais não necessariamente aparecem em todos os projetos desenvolvidos no *Scratch*, mas servem como uma nova estrutura para estudar e avaliar o desenvolvimento do PC (BRENNAN; RESNICK, 2012). Esse estudo e avaliação podem ser feitos por meio de análises de portfólio de projetos, de entrevistas ou de desenvolvimento de cenários. Como cada uma dessas maneiras tem suas forças e limitações, o apropriado seria realizar as três, porém restrições como tempo e número de alunos tornam a prática inviável (BRENNAN; RESNICK, 2012).

Em geral, precisamos pensar sobre como o desenvolvimento de um pensador computacional ocorre em diferentes contextos, em diferentes escalas de tempo, com diferentes motivações e com diferentes estruturas e suportes - e como essas diferenças levam a diferentes abordagens de avaliação (BRENNAN; RESNICK, 2012, p. 22, tradução nossa)²⁸.

Assim, os autores dão algumas sugestões para avaliar o PC por meio da programação. Eles acreditam que a melhor forma de avaliação é aquela útil ao estudante, isto é, aquela feita com o intuito de ser contextualizada e significativa para ele, como investigar se utilizou a comunidade do *software* ou fez alguma pesquisa *on-line* a fim de descobrir novos blocos e suas funcionalidades. Os autores enfatizam que as entrevistas são oportunidades de reflexão para os estudantes, assim como um meio de fazê-los apresentar seus projetos aos demais colegas. Além disso, a avaliação pode ocorrer em vários momentos e deve incluir diversos pontos de vista, envolvendo o exame de si mesmo, de colegas, de pais, professores e pesquisadores, conforme possível e apropriado (BRENNAN; RESNICK, 2012).

4.4 A RELAÇÃO ENTRE SCRATCHE MATEMÁTICA

O *ScratchMaths*²⁹ (SM) foi um projeto de pesquisa realizado pelo UCL *Knowledge Lab*³⁰, um dos Centros de pesquisa e educação do Instituto de Educação³¹ da *University College London*. Tal projeto teve uma duração de três

²⁸ In general, we need to think about how developing as a computational thinker takes place in different contexts, on different timescales, with different motivations, and with different structures and supports – and how these differences lead to different approaches to assessment.

²⁹ *ScratchMaths: supporting computational and mathematical thinking through programming*.

³⁰ Disponível em: <https://www.ucl.ac.uk/ioe/departments-and-centres/centres/ucl-knowledge-lab>. Acesso em: jul. 2020.

³¹ Disponível em: <https://www.ucl.ac.uk/ioe/about-ioe>. Acesso em: jul. 2020.

anos, sendo o primeiro usado para projetar e planejar como seriam os outros dois, os quais envolveram ações em sala de aula com estudantes entre 9 e 11 anos, idades correspondentes, respectivamente, ao quinto (*Year 5*) e sexto ano (*Year 6*), do *Key Stage 2* (KS2). Em cada ano de intervenção, dois professores de cada escola foram treinados para o uso da programação *Scratch* e dos materiais curriculares.

O objetivo da pesquisa foi:

projetar e avaliar iterativamente, tanto quantitativa quanto qualitativamente, materiais para alunos e professores que abordem diretamente o aprendizado do pensamento computacional e sua exploração para aprimorar o engajamento e a realização matemática (BENTON et al., 2016, p. 3, tradução nossa)³².

A hipótese do SM foi que, dado um planejamento correto do currículo, da pedagogia e das ferramentas digitais, os estudantes poderiam se envolver e se expressar com importantes ideias matemáticas por meio da programação de computador (BENTON et al., 2018a).

A pesquisa foi realizada em mais de 50 escolas (aproximadamente 100 turmas), tendo a parte quantitativa ficado sob a responsabilidade da *Sheffield Hallam University*, que, primeiramente, comparou o desempenho dos estudantes na avaliação final do Y6 (no tocante ao pensamento matemático) entre aqueles que tiveram contato com o SM e aqueles que não tiveram. Em segundo lugar, foi realizada, no final do primeiro ano de intervenção da pesquisa, uma avaliação em relação aos conceitos do pensamento computacional adequados à faixa etária. Já a parte qualitativa ficou sob responsabilidade da equipe do UCL *Knowledge Lab*, que se concentrou em analisar como os materiais curriculares são utilizados pelos professores e como engajaram os estudantes.

O projeto contou com um currículo³³ de dois anos, fundamentado em Computação e Matemática, sendo o primeiro ano (Y5) focado em pensamento computacional e programação *Scratch* e o segundo (Y6) voltado à exploração dos principais conceitos matemáticos, usando ferramentas de programação. Tal currículo contém materiais para os professores, apresentações para as aulas, projetos iniciais

³² The overarching goal of the research is to iteratively design and evaluate, both quantitatively and qualitatively, materials for students and teachers that directly address the learning of computational thinking and its exploitation to enhance mathematical engagement and attainment.

³³ Disponível em: https://www.ucl.ac.uk/ioe/sites/ioe/files/ScratchMaths_Curriculum_Overview.pdf. Acesso em: jul. 2020.

do *Scratch* e cartazes de desafio, referência e vocabulário. A escolha do conteúdo foi consonante com o Currículo Nacional da Inglaterra³⁴, que, no fim do KS2, propõe um teste nacional e avaliações em Inglês, Matemática e Ciências. Desse modo, ele foi dividido em seis módulos, três a cada ano. O Quadro 4 exibe uma visão geral dos módulos e dos tópicos do *ScratchMaths*.

Quadro 4 – Visão geral dos módulos e dos tópicos do SM

Módulos do Y5 (9-10 anos)	Módulos do Y6 (10-11 anos)
Módulo 1 – Padrões de ladrilhos	Módulo 4 – Construindo com números
Concentra-se em repetir padrões e iniciar com programação básica em <i>Scratch</i> . Apresenta os principais conceitos computacionais de sequenciação, repetição, algoritmos, depuração e definições, bem como a ligação à simetria, ângulos e números negativos através da construção de padrões repetidos.	Concentra-se na exploração do valor posicional e requer que os estudantes usem seu conhecimento de <i>broadcasting</i> para construir modelos de valor posicional em vários contextos diferentes.
Módulo 2 – Geometria do besouro	Módulo 5 – Explorando relações matemáticas
Concentra-se na criação de diferentes desenhos usando a ferramenta caneta, como algarismos romanos e polígonos regulares, apresentando aos estudantes a inicialização, as expressões e a aleatoriedade, bem como consolidando conceitos anteriores.	Concentra-se na exploração de diferentes tipos de relações matemáticas, incluindo proporcionalidade e razão, bem como na introdução do conceito de variável.
Módulo 3 – Interação entre <i>sprites</i>	Módulo 6 – Coordenadas e Geometria
É baseado na construção de comportamentos interativos entre vários <i>sprites</i> , primeiro usando condicionais e depois <i>broadcasting</i> , introduzindo vários atores e comportamentos paralelos, bem como vinculando as coordenadas, multiplicação e fatores.	Explora as coordenadas em todos os quatro quadrantes e exige que os estudantes usem seu conhecimento de variável para explorar escala.

Fonte: Adaptado do currículo³¹ *ScratchMaths*.

Para desenvolver esses conteúdos curriculares, foi elaborada uma estrutura de ação para fornecer orientação para a abordagem pedagógica, denominada 5 E's, que correspondem a *Explore, Explain, Envisage, Exchange* e *bridgE*, cuja tradução é, respectivamente, explorar, explicar, prever, trocar e conectar. Essa estrutura foi pautada em diversos autores, como Brennan (2015), Harel e Papert (1990), Hoyles (1985), Papert (1980, 2000), entre outros.

A ação de **explorar** fornece aos estudantes a oportunidade de investigar ideias, depurar erros conceituais e técnicos quando necessário e, ao lidar com diferentes restrições e ambiguidades, utilizar o pensamento iterativo, a resolução de problemas e a criatividade (BENTON et al., 2016, 2017).

³⁴ Disponível em: <https://www.gov.uk/national-curriculum>. Acesso em: jul. 2020.

Ao **explicar** uma questão ou atividade, o estudante expressa verbalmente suas ideias, produzindo uma melhor compreensão delas, além de ter um benefício cognitivo ao fazê-lo (HAREL; PAPERT, 1990). Desse modo, o conteúdo curricular deve incorporar questões reflexivas e oportunizar a discussão dos estudantes com seus colegas, assim como deve promover interações em toda a aula administrada pelo professor (BENTON et al., 2016, 2017).

Outra intencionalidade das atividades é fazer com que os estudantes **prevejam** os resultados de suas programações antes de executá-las e reflitam sobre o resultado real obtido. As tarefas de previsão auxiliam no desenvolvimento de habilidades de raciocínio e de resolução de problemas matemáticos (BENTON et al., 2017). É importante que o discente tenha um objetivo em mente ao criar um programa, pois programar facilita a reflexão sobre expectativas e conhecimentos intuitivos, o que pode levar a uma melhor compreensão da ideia quando ele dedica um tempo para prever o resultado, isto é, quando o explicita de alguma forma antes de obter o *feedback* (PAPERT, 1980).

A estrutura do **trocar** tem o sentido de colaborar, compartilhar, sendo uma maneira poderosa de aprender, pois permite que o estudante veja um problema da perspectiva de outra pessoa, além de defender a sua abordagem e compará-la com a dos demais (BENTON et al., 2016). O compartilhamento de ideias pode resultar em modificações nos processos de pensamento de um indivíduo, sendo útil para esclarecer previsões ou explicar conceitos que ainda não estão totalmente formados (HOYLES, 1985).

A estrutura de ponte (*bridgE*) é uma metáfora para as **conexões** entre as disciplinas. Papert (1980) apresenta o conceito de *powerful ideas* (ideias poderosas) e afirma que sua força é devida a suas conexões com outras disciplinas e à linguagem na qual elas são expressas. No caso do *ScratchMaths*, a disciplina é a Matemática, a linguagem é o *Scratch* e a conexão é feita por meio da programação. Para desenvolver essas conexões, as ideias precisam ser recontextualizadas e reconstruídas na linguagem da outra disciplina (BENTON et al., 2016).

Apesar de ainda não ter sido divulgado um relatório do projeto, houve publicações referentes às intervenções realizadas nas salas de aula. Benton et al. (2018b) tratam da compreensão que alguns estudantes do Y6 tiveram sobre algoritmo. Para tanto, os discentes realizaram um conjunto de tarefas escritas, a fim de analisar suas interpretações e avaliações (em termos da dificuldade percebida,

da legibilidade, da avaliação do professor e da facilidade de reutilização) de diferentes algoritmos (desenvolvidos no *Scratch*) que resolvem um mesmo problema. Com vistas a um melhor entendimento das respostas, posteriormente, eles foram entrevistados. As respostas dos estudantes acerca do que é algoritmo foram agrupadas em: definição básica (usar palavras como um programa, *script*, conjunto ou sequência de instruções, código), definição avançada (relatar conceitos como múltiplas estratégias para resolver o mesmo problema ou generalização) e em categorias de exemplos corretos e incorretos. Os autores concluíram que, apesar de haver uma dificuldade na definição de algoritmo, estabelece-se uma ponte com o conceito de abstração quando os estudantes conseguem generalizar o algoritmo para outros contextos, o que confere um poder intelectual à atividade, poder esse que os autores acreditam ser uma alavanca para o aprendizado de outras disciplinas.

Benton et al. (2018b) apresentam uma pesquisa sobre o envolvimento dos estudantes com o tema do valor posicional de um dígito por meio do conceito de “transmissão de mensagem” (*broadcasting*) no *software Scratch*. A escolha do referido assunto como conteúdo curricular do SM se deveu às pesquisas indicarem que discentes têm dificuldade com esse tópico, bem como com a adição e subtração de números inteiros de vários dígitos (BENTON et al., 2018b).

Inicialmente, o estudo foi feito com duas turmas do Y5, com um professor para cada turma observando o desenvolvimento de uma atividade em que os estudantes construíram interações, primeiramente, com apenas um ator reagindo a uma transmissão e, depois, com vários atores reagindo à mesma transmissão. Segundo os autores, de uma perspectiva computacional, esses estudantes são expostos aos conceitos de paralelismo, eventos, múltiplos atores e interações entre eles.

Seis meses depois, essas duas turmas, que já tinham avançado para o Y6 e, por conseguinte, tinham dois novos professores, foram novamente observadas, mas agora em uma atividade que consistiu em considerar as ordens (unidade e dezena) como atores e os dígitos como suas fantasias, em que cada uma delas representava o dígito correspondente. A programação das unidades consistiu em mudar a fantasia (“somar” 1) cada vez que o ator fosse clicado e, ao atingir a décima fantasia (numeral 0), era enviada uma mensagem (*broadcasting*) ao ator das dezenas que, então, mudava sua fantasia. Os autores viram o potencial da atividade e estenderam

o conceito para os decimais e para o contexto de tempo, por meio da construção de um cronômetro.

Os resultados da pesquisa mostraram que, em uma análise de conteúdo dos projetos do *Scratch* dos estudantes, a maioria dos alunos do Y5 (46 dos 49), de ambas as classes, teve êxito na atividade. No Y6, o motivo mais comum para o cronômetro não funcionar foi o posicionamento incorreto do bloco de transmissão, o que causava um atraso no ator das dezenas, resultando, por exemplo, nas sequências 09, 00, 11. Outro motivo foi que alguns estudantes perderam o bloco de recebimento correspondente ao ator das dezenas por terem muitos blocos de transmissão. A análise das tarefas do Y6 revelou que, na turma cujo(a) professor(a) deu mais suporte/estrutura aos discentes, como seguir os *slides* da apresentação do SM ou fornecer algum reforço visual, eles obtiveram mais êxito. Os autores concluem que a maioria dos estudantes foi capaz de se envolver e implementar o conceito de comunicação de vários objetos uns com os outros, bem como de desenvolver tal conceito e usá-lo para explorar em um contexto matemático. Os resultados também mostraram que os alunos foram além do conhecimento procedimental de valor posicional, generalizando-o para outros contextos, como tempo e medida.

4.5 SÍNTESE DOS CONCEITOS DO PENSAMENTO COMPUTACIONAL

A intenção dessa subseção é trazer um sumário com os conceitos do PC identificados nas subseções anteriores, visto que na análise o intuito é identificar quais desses conceitos emergiram dos estudantes ao realizarem as atividades desenvolvidas na produção de dados.

- a) ISTE e CSTA (2011) e Barr e Stephenson (2011): coleção de dados, análise de dados, representação de dados, decomposição do problema, abstração, algoritmos e procedimentos, automação, simulação e paralelização;
- b) Brennan e Resnick (2012):
 - Conceitos computacionais: sequências, *loops*, paralelismo, eventos, condicionais, operadores e dados;
 - Práticas computacionais: ser incremental e iterativo, testar e depurar, reutilizar e remixar e abstrair e modularizar;

- Perspectivas computacionais: expressar, conectar e questionar;
- c) Benton et al. (2016, 2017): explorar, explicar, prever, trocar e conectar.

Desse modo, a seção seguinte apresenta e analisa as atividades que os estudantes produziram, bem como quais dos conceitos supracitados emergiram deles nesse processo.

5 APRESENTAÇÃO E ANÁLISE DOS DADOS

Procurando uma resposta à pergunta diretriz: “*Quais conceitos do pensamento computacional emergem de estudantes do nono ano ao realizarem atividades com conteúdo matemático no Scratch?*”, analisamos as tarefas desenvolvidas pelos discentes nesse programa a partir dos arquivos das práticas, das gravações efetuadas durante esse processo e das anotações do pesquisador.

Os participantes da pesquisa foram estudantes de duas turmas de nono ano do Ensino Fundamental, cada uma com 34 discentes, os quais precisaram ser divididos em grupos menores, já que o cenário de investigação foi constituído pelo acompanhamento e intervenção na disciplina Práticas de Matemática, ministrada pela professora Ana Maria de Freitas Soares e desenvolvida no laboratório de informática. Tal divisão resultou em quatro turmas, com 17 estudantes cada, denominadas: 9º1-T1, 9º1-T2, 9º2-T1 e 9º2-T2, as quais realizaram as tarefas em duplas ou trios, dependendo do número de presentes nos dias da produção de dados. No total, foram desenvolvidas 5 atividades, a saber, Atividade 1 – Introdução ao *Scratch* e desenho casa; Atividade 2 – Jogo da maçã e labirinto; Atividade 3 – Introdução dos Procedimentos; Atividade 4 – Variáveis e polígonos regulares e Atividade 5 – Blocos “se”, “se/senão” e calculadora de áreas, com o intuito de explorar as funcionalidades do *Scratch* e determinados conteúdos matemáticos.

Os dados produzidos e analisados foram organizados seguindo a ordem cronológica de realização das atividades, as quais, com exceção da primeira, ocorreram em duas partes. Na primeira, eu apresentei uma tarefa aos estudantes, com o objetivo tanto de ensiná-los as funcionalidades do *Scratch*, quanto de fazê-los relacioná-las a algum conteúdo matemático. Já na segunda, eu lhes propus uma prática, visando a fazê-los se dedicarem a explorar o *software*, de modo a generalizarem sua aplicabilidade para usos futuros.

Cabe ressaltar que as Atividades 2 e 3 não tiveram o desempenho esperado, pois grande parte dos participantes não se motivou com a proposta e, devido a isso, elas apresentaram um caráter instrucionista maior do que o esperado, ou seja, ainda que os estudantes tenham tentado se empenhar ao fazê-las, eles não tiveram autonomia para desenvolverem o que havia sido proposto.

Na Atividade 2, apesar de os discentes acompanharem a parte inicial, que foi desenvolver o Jogo da maçã, no segundo momento, isto é, na construção de um

labirinto, eles, de modo geral, frustraram-se, por duas razões: primeiro, por enfrentarem dificuldades para compreender se deveriam criar um novo ator (que seria o labirinto) ou apenas modificar o palco para que se parecesse com um labirinto; segundo, por não conseguirem desenvolver com exatidão a programação necessária para que o ator não sobrepusesse as “paredes do labirinto”, tanto no caso de criarem um novo ator quanto de modificarem apenas o palco. Assim, apesar de haver um caminho que o ator deveria percorrer, ele acabava atravessando-o, de modo que podia andar livremente pelo palco, chegando ao final do labirinto.

Já na Atividade 3, pelo fato de o conteúdo da primeira parte, que foi dedicada à construção de um bloco, ser teórico, ou seja, envolver o conceito de Procedimentos (*procedures*), os estudantes não se animaram a desenvolver a segunda parte, na qual deveriam elaborar, a partir de um bloco que desenhava uma pétala, um outro que desenhava uma flor e um terceiro que, por sua vez, gerava uma flor com um ramo. O início da segunda parte pareceu efetivo, isto é, pareceu que os estudantes conseguiriam construir um bloco denominado flor, no entanto, não foram capazes de desenvolver a programação necessária para fazer o seu desenho, tampouco o desenho da flor com um ramo. Isso fez com que, assim como na Atividade 2, eu os auxiliasse mais do que o esperado e, desse modo, eles não a realizaram com autonomia. Portanto, os dados gerados nas Atividades 2 e 3 não fazem parte da análise dessa dissertação. Ainda assim, suas descrições encontram-se no Apêndice E.

Assim, utilizando os referenciais das seções 3 e 4, a análise das Atividades 1, 4 e 5 buscou evidenciar os conceitos do pensamento computacional que emergiram quando os estudantes exploraram o *software Scratch* e desenvolveram tarefas com conteúdo matemático.

5.1 ATIVIDADE 1 COM A TURMA 9º2-T2

A primeira atividade teve o intuito de propiciar aos estudantes a exploração do *Scratch*, e, mais especificamente, do uso das paletas Caneta e Movimento e suas funcionalidades, a fim de que eles tivessem um primeiro contato com o programa ou, no caso de alguns deles, que já haviam explorado o *software* com outros professores da Escola Carolina Augusta Seraphim, de que relembrassem a linguagem de programação por blocos. Tal exploração se deu por meio da criação

do esboço de uma casa, elaborada pelos estudantes utilizando os recursos associados à caneta e ao movimento do ator a partir do modelo que desenhei na lousa. A Figura 10 ilustra a representação do desenho da casa proposta na lousa.

Figura 10 – Representação do desenho da casa



Fonte: O Autor.

A proposta de construir a casa a partir da exploração do *software* propiciou aos estudantes um momento no qual o *feedback* dado pela mídia na interação desses sujeitos com as ferramentas/recursos (TIKHOMIROV, 1981; BORBA; VILLARREAL, 2005) poderia culminar no ato de pensar de acordo com a perspectiva da teoria psicológica do pensamento (TIKHOMIROV, 1981). Apesar de o objetivo ter sido definido previamente, sendo este o de criar o desenho da casa no *Scratch*, os estudantes precisariam **explorar** (BENTON et al., 2016, 2017) as possibilidades de construção/criação com a programação, **testando e depurando** (BRENNAN; RESNICK, 2012) os erros, formulando seus próprios problemas, objetivos e soluções pela experimentação.

Para os que não conheciam o *Scratch*, dei dicas iniciais, como, por exemplo, sobre como fazer o ator traçar uma linha no palco, considerando que há mais de uma forma de fazer isso. Sugeri que eles usassem o bloco “mova 10 passos” e, no lugar do 10, eles deveriam colocar outro valor para observar o comprimento do traço desenhado no palco. O intuito era que os estudantes identificassem, por meio dessa exploração, as alterações provocadas no comprimento dos traços ao alterarem o número de passos.

Ao exemplificar para eles, primeiramente, posicionei o ator em um lugar aleatório do palco e fiz um traço com 10 passos, executando a programação que pode ser observada à esquerda da Figura 11. Depois, reposicionei o ator e executei a programação para traçar 60 passos, tal como exposto à direita da mesma figura.

Além disso, disse a eles que, no caso de não estarem satisfeitos com o desenho, na paleta Caneta, há o bloco “apague tudo”, que limpa o que foi executado no palco.

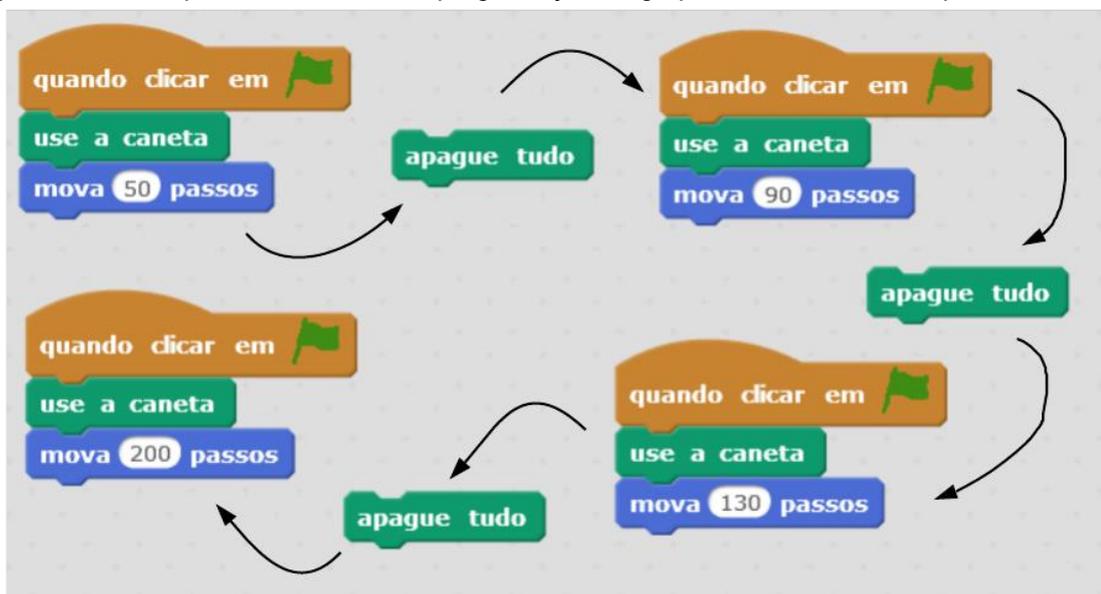
Figura 11 – Programação para desenhar um traço com 10 e 60 passos no palco



Fonte: O Autor.

Quando os estudantes começaram a fazer tal programação, pude notar quatro situações que emergiram da **experimentação** relacionada ao comprimento do traço. A primeira diz respeito aos estudantes que desenharam um traço, porém, não satisfeitos com seu comprimento, apagaram-no e alteraram o valor numérico do bloco de movimento, repetindo essa ação até que chegassem a um comprimento desejável. A Figura 12 representa esse processo de exploração.

Figura 12 – Exemplo das tentativas de programação de grupos de estudantes da primeira situação



Fonte: O Autor.

A segunda situação referente ao comprimento do traço é relacionada aos estudantes que colocavam um valor inicial dentro do bloco e executavam a programação consecutivas vezes, porém, faziam-no sem apagar o traço já desenhado, dando uma falsa percepção da extensão do novo traço. Significa dizer que colocavam, por exemplo, o valor 50 dentro do bloco de movimento e apertavam

o ícone da bandeira verde até chegarem a um traçado de comprimento desejável. Entretanto, não percebiam que o traço desenhado na terceira execução tinha, na verdade, 150 passos. Uma representação dessa situação pode ser visualizada na Figura 13, a seguir, em que, à esquerda, está representado um traço de 150 passos, feito com a programação esboçada à direita, cuja rotina funcionou clicando três vezes na bandeira verde.

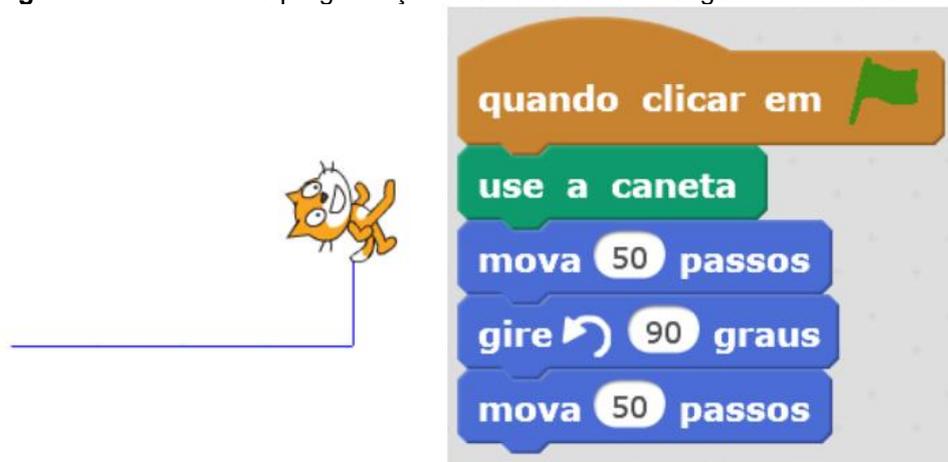
Figura 13 – Exemplo do equívoco de apertar a bandeira três vezes



Fonte: O Autor.

Os estudantes só perceberam tal equívoco quando seguiram com a programação para desenhar o lado da casa, adicionando o bloco “gire 90 graus” e novamente “mova 50 passos” (vide Figura 14), e os comprimentos estavam diferentes. Além disso, como era a quarta vez que clicavam na bandeira verde, a programação foi inteiramente executada mais uma vez, fazendo com que o traço que estava em 150 passos fosse para 200 e, depois, fosse desenhado o lado da casa com 50 passos.

Figura 14 – Desenho e programação dos estudantes da segunda característica

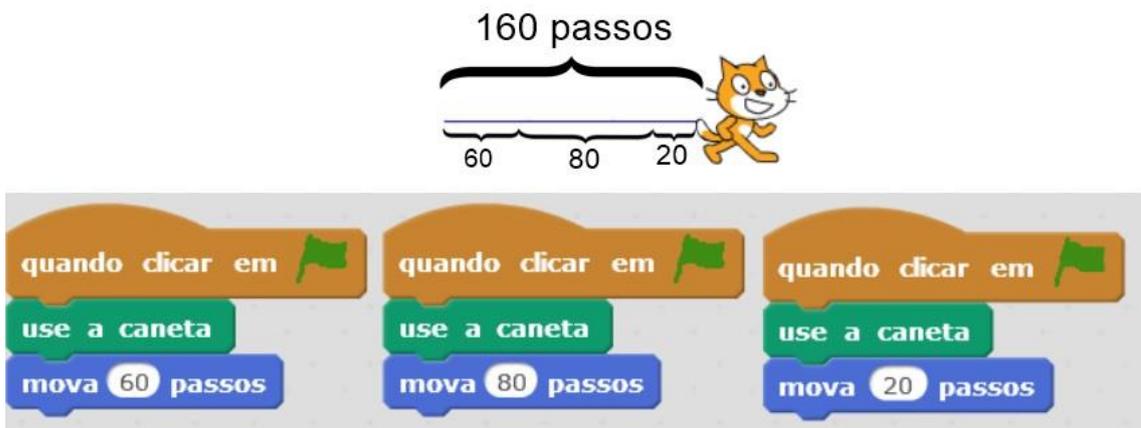


Fonte: O Autor.

A terceira situação observada referente à atividade do esboço da casa foi semelhante à primeira e à segunda, pois os estudantes utilizavam um único bloco de

movimento, sem apagar o traço, e executavam a programação diversas vezes, alterando o valor numérico do bloco até atingir o comprimento desejável, desconsiderando que o ator já havia traçado a linha, ou seja, que uma programação já havia sido executada por ele. A Figura 15 é um exemplo dessa situação. Nela, pode-se observar um traço de 160 passos, porém, ele foi construído em três momentos e sua última programação, na verdade, desenhava um traço de 20 passos e não de 160, como imaginado pelos estudantes.

Figura 15 – Exemplo de desenho e de programação da terceira característica

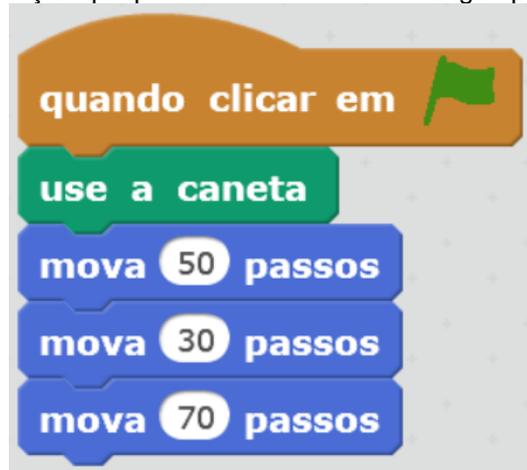


Fonte: O Autor.

Ao darem sequência à atividade, resolveram apagar o traço para recomeçarem o desenho da casa em uma posição melhor. Então, reposicionaram o ator e executaram a última programação, com a qual eles imaginavam desenhar um traço de 160 passos, porém, o que foi obtido foi um traço de 20 passos. Assim, perceberam que a programação, da forma como estava construída, na realidade, desenhava um traço com 20 passos, ou seja, menor do que aquele que eles tinham construído antes de apagar. Como não tinham anotado os valores, não sabiam qual era a extensão anterior e, dessa forma, testaram outros valores até ficarem satisfeitos com o comprimento do traço.

A quarta situação observada na construção do traçado é referente aos estudantes que adicionavam outro(s) bloco(s) de movimento abaixo dos que já tinham posto, não necessariamente com o mesmo valor, até que o traçado ficasse com o comprimento desejado, conforme pode ser observado na Figura 16.

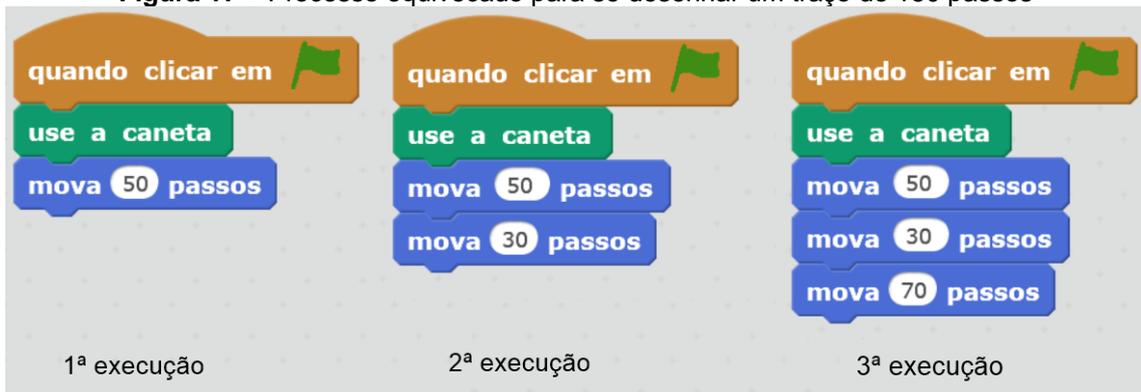
Figura 16 – Programação que pode ser considerada ambígua para traçar 150 passos



Fonte: O Autor.

Porém, assim como comentado na segunda situação, alguns estudantes adicionavam um bloco de movimento, sem, no entanto, apagarem o traço já feito, dando uma falsa percepção da extensão real dele. Por exemplo, a programação observada na Figura 16 faz o ator desenhar um traço de 150 passos, mas também pode criar um de 280. No primeiro caso, se não há nenhum traçado já delineado, ao clicar na bandeira verde, o ator move-se 150 passos para desenhá-lo. Já no segundo caso, retratado na Figura 17, pode ocorrer o seguinte: o estudante adiciona o bloco “mova 50 passos”, e, não satisfeito com o comprimento, inclui o bloco “mova 30 passos”, sem, no entanto, apagar o traço já realizado; na sequência, ele simplesmente clica na bandeira verde pela segunda vez, fazendo com que a programação realize todos os comandos novamente, ou seja, que esboce um traço de 80 passos, somados aos 50 anteriores, totalizando um traço de 130 passos. Se ele, então, continuar e adicionar o bloco “mova 70 passos”, tal como na execução anterior, o programa fará um traço de 150 passos, que se somarão aos 130 anteriores, finalizando o desenho com um traço de 280 passos.

Figura 17 – Processo equivocado para se desenhar um traço de 150 passos



Fonte: O Autor.

Ao notar essas ocorrências, percebi a necessidade de interromper a atividade para dar novas instruções aos estudantes. Então, com o *software* projetado na TV do laboratório, disse que eles deveriam escolher um ponto inicial para o desenho da casa. Para isso, havia duas maneiras: passar o ponteiro do mouse sobre o palco e escolher um ponto adequado, anotando as coordenadas e colocando-as no bloco “vá pra x:_ y:_”, ou arrastar o ator no local inicial determinado, o que faria com que, automaticamente, tal bloco já estivesse com as coordenadas desse ator. Em seguida, mostrei e expliquei a eles as características mencionadas acima, dando ênfase na situação em que eles não apagavam o traço e o que isso ocasionava. Para facilitar o processo de visualização, pedi que adicionassem o bloco de Controle “espere_seg”, pois, desse modo, seria possível entender o movimento que o ator fazia conforme a programação planejada por eles.

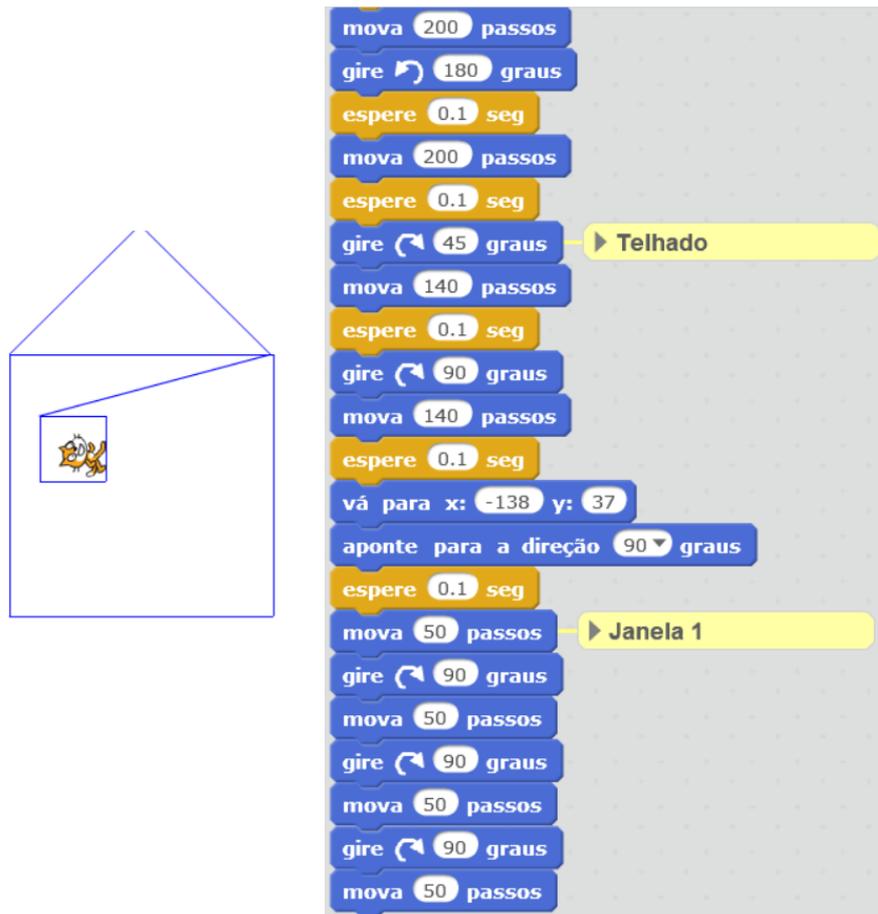
Ao inserirem um novo bloco para que o ator se movesse e ao simularem usando a bandeira verde, puderam visualizar que o comprimento do traçado foi maior. Nesse sistema de interação dos sujeitos com o computador, a informação que emergiu da ação realizada por eles, isto é, o *feedback* (TIKHOMIROV, 1981; PAPERT, 1980), propiciou que reconhecessem que a inserção de novos blocos resultaria em um traço mais extenso. Aproveitando o que havia surgido na última situação, questionei os estudantes se poderíamos colocar apenas um bloco de movimento ao invés de vários e alguns responderam que sim, bastava saber quantos passos, de fato, queríamos andar no total. Portanto, perguntei novamente a eles o que deveríamos fazer para saber isso e, então, responderam-me que deveríamos somar os valores dos blocos.

Considero que, naquele momento, o questionamento propiciou aos discentes pensarem de maneira lógica, à medida que, tendo por objetivo ampliar o tamanho do traço, otimizando a solução com o uso de menos blocos, esses sujeitos reconheceram que era necessário determinar o total de passos do ator, isto é, compreenderam os dados, encontraram um padrão e, conseqüentemente, chegaram à conclusão de que vários blocos de movimento poderiam ser substituídos por apenas um cujo valor numérico fosse a soma dos blocos anteriores. Assim, “identificaram, analisaram e implementaram possíveis soluções com o objetivo de alcançar a combinação mais eficiente e eficaz de etapas e recursos” (ISTE; CSTA, 2011, p. 7), o que evidencia o emprego dos conceitos de **coleção e análise de dados**, bem como do de **abstração** (ISTE; CSTA, 2011).

Isso porque, ao responderem à minha pergunta acerca da possibilidade de utilizarem um único bloco afirmando que bastava obter o total de passos desejados por meio da soma dos valores de cada bloco, os sujeitos deram indícios de que organizaram o problema de forma menos complexa se comparada à solução já apresentada no computador e, em particular, no *software Scratch*. Significa dizer que a resposta dos estudantes ao meu questionamento evidenciou uma solução que não foi baseada em um algoritmo (TIKHOMIROV, 1981), mas na mobilização de ideias matemáticas frente ao problema decomposto. Isso mostra que, a partir da atividade, a investigação não se deu de forma informacional, mas sim em um nível psicológico do pensamento, visto que eles traduziram a situação do acúmulo de blocos, identificando que cada etapa correspondia a valores que poderiam ser somados para obtenção de uma solução mais simples.

No decorrer da atividade, alguns estudantes começaram a levantar a mão e a dizer que havia algo errado com o desenho da casa. Tal erro era que os traços de suas partes (lados, janelas, porta, etc.) ficaram conectados. Isso ocorre quando não se usa o bloco “levante a caneta” entre a programação das partes da casa. Ao desenhar uma janela, por exemplo, deve-se usar o bloco “levante a caneta”, indicar onde o ator deve começar a desenhar a próxima parte e, então, selecionar o bloco “use a caneta” para continuar a programação. Assim, as partes não ficam conectadas. Observa-se, na Figura 18, a programação realizada pela dupla de estudantes José e Caio, sem o uso dos blocos “levante a caneta” e “use a caneta”, assim como o respectivo desenho da casa.

Figura 18 – Desenho e programação, elaborados por José e Caio, da casa *sem* o uso dos blocos “levante a caneta” e “use a caneta”



Fonte: Dados da pesquisa.

Quando surgiu o problema de as partes da casa estarem conectadas, perguntei à turma se, quando desenhavam uma casa com lápis e papel, tiravam o lápis do papel em algum momento e me disseram que sim. Nesse sentido, eles *avaliaram* que uma programação como a da Figura 18, por exemplo, não realizava aquilo que eles queriam, isto é, ter um desenho sem os traços conectados. Então, naquele momento, eles tiveram que *reorganizar* a ideia de desenhar uma casa com os recursos disponíveis no *Scratch*. Essa reorganização fez com que **decompusessem** (CSTA; ISTE, 2011) a figura da casa, haja vista que, mediante a observação, os estudantes identificaram, na programação, os momentos nos quais seria necessário usar os blocos “levante a caneta” e “use a caneta” para a construção de cada janela e da porta.

No Quadro 5, exponho o diálogo entre mim e a dupla José e Caio no momento em que eles observaram um risco no desenho, como expresso na Figura 18.

Quadro 5 – Diálogo entre mim, José e Caio sobre o risco no desenho da casa

Caio: Professor, nosso desenho tá [sic] com um risco. A gente já **tentou** umas coisas, mas não sai [...] Só piora se a gente continua o desenho.

Pesquisador: Vocês precisam usar o bloco “levante a caneta” quando forem desenhar as partes da casa. Vocês fizeram primeiro esse retângulo [referindo-me aos lados da casa] e depois o telhado?

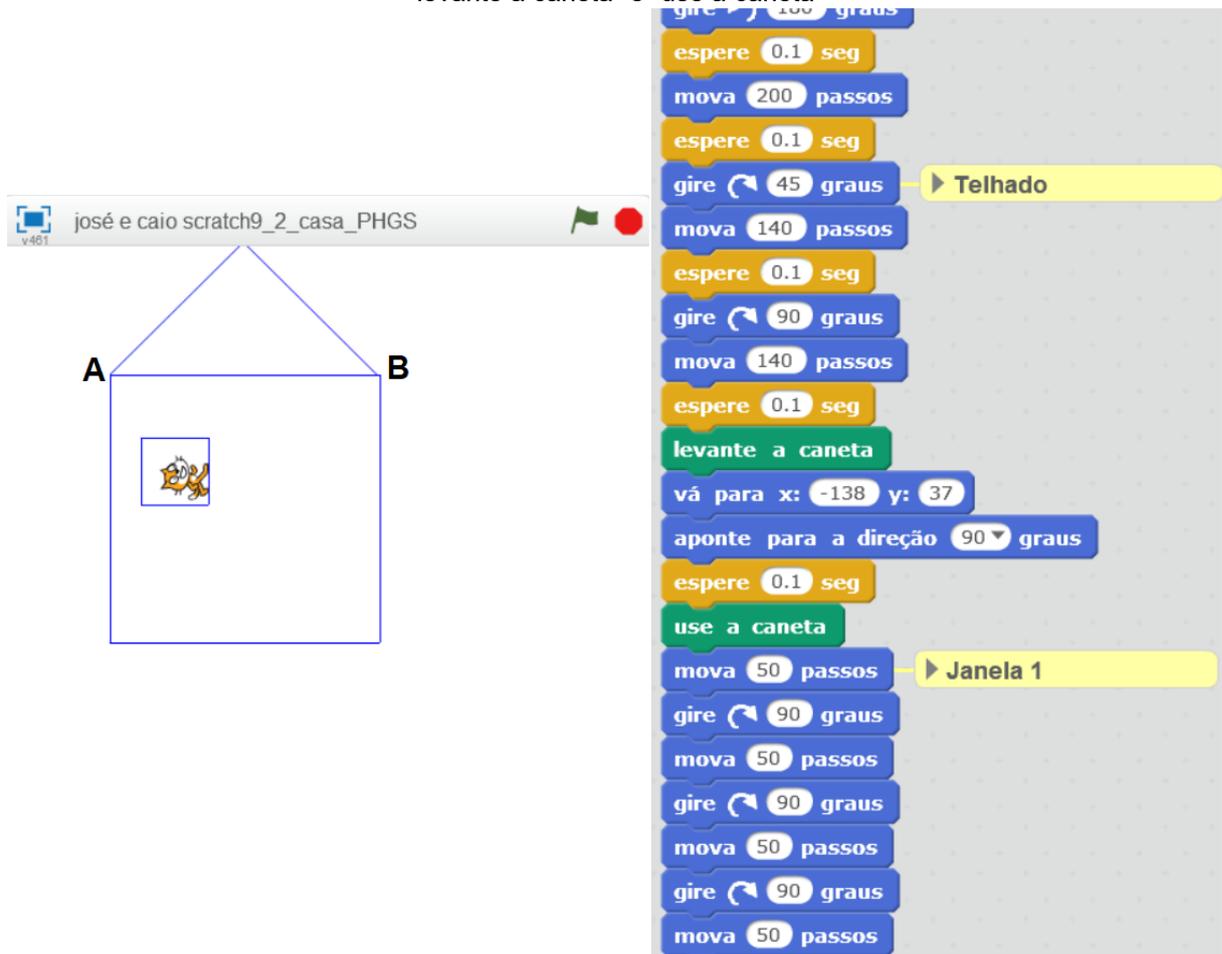
José: Sim.

Pesquisador: Então, aqui [apontando para a programação referente à respectiva parte do desenho], vocês não precisam usar o bloco [referindo-me ao bloco “levante a caneta”] porque vocês emendaram o lado da casa com o telhado. Mas, quando vocês forem fazer as janelas e as portas, têm que usar [novamente, referindo-me ao bloco “levante a caneta”].

Fonte: Dados da pesquisa.

Na Figura 18, pode-se observar que o desenho do triângulo que representa o telhado da casa está incompleto. Assim, logo na sequência do momento descrito no Quadro 5, comentei com a dupla que o desenho do triângulo não estava bom. Apontando para o palco do *Scratch*, disse que eles deveriam “descer” o desenho da casa. Então, eles me responderam que, primeiro, tentariam arrumar o risco na figura e, posteriormente, dedicar-se-iam a corrigir o triângulo. A nova programação feita por José e Caio, assim como o desenho correspondente, pode ser observada na Figura 19.

Figura 19 – Desenho e programação da casa, elaborados por José e Caio com o uso dos blocos “levante a caneta” e “use a caneta”



Fonte: Dados da pesquisa.

Tendo em vista os aspectos psicológicos do pensamento (TIKHOMIROV, 1981), a interação com a mídia no momento de usar os blocos “levante a caneta” e “use a caneta”, e, mais especificamente, com a programação criada, possibilitou aos estudantes avaliarem o que estavam desenvolvendo, com o objetivo de solucionar o problema dos traços conectados. Isso se deu a partir da reflexão sobre o ato de desenhar a casa com lápis e papel, juntamente com a possibilidade de modificarem a programação e visualizarem rapidamente os resultados, explorando a dinamicidade e experimentando a autonomia de ação e reação pela simulação (LÉVY, 1997). Significa dizer que os estudantes reconheceram que os blocos supracitados deveriam ser utilizados em associação com aquele que determinaria a mudança de coordenada “vá para x: _ y: _” do ator.

No entanto, mesmo corrigindo o problema do risco, ainda faltava solucionar o do telhado. Porém, os estudantes da dupla estavam convictos de que o desenho do

triângulo estava certo, apenas ultrapassando o limite superior do palco. O Quadro 6 apresenta o diálogo desse momento.

Quadro 6 – Diálogo entre mim, José e Caio sobre o desenho do telhado da casa

Caio: Então, professor, a gente fez um triângulo retângulo. Por isso, aqui [apontando para o bloco “gire 90 graus”], tá [sic] 90 graus. E esse lado é igual a esse [referindo-se aos catetos do triângulo].

Pesquisador: Tá [sic], mas vocês estão vendo que não tá [sic] encaixando aqui, né [sic]? [referindo-me ao ponto B da Figura 19, isto é, aos vértices do triângulo e do retângulo que representam os lados da casa]. E a ponta do triângulo está pra fora. Isso é fácil arrumar.

José: A gente fez um triângulo isósceles. Aí, aqui, é 45 [referindo-se ao ângulo da base do triângulo, ponto A da Figura 19] e, nesses dois aqui, é 140 [referindo-se aos blocos de movimento que representam os lados do telhado] e, ali em cima, tá [sic] 90 [referindo-se ao ângulo reto do triângulo].

Fonte: Dados da pesquisa.

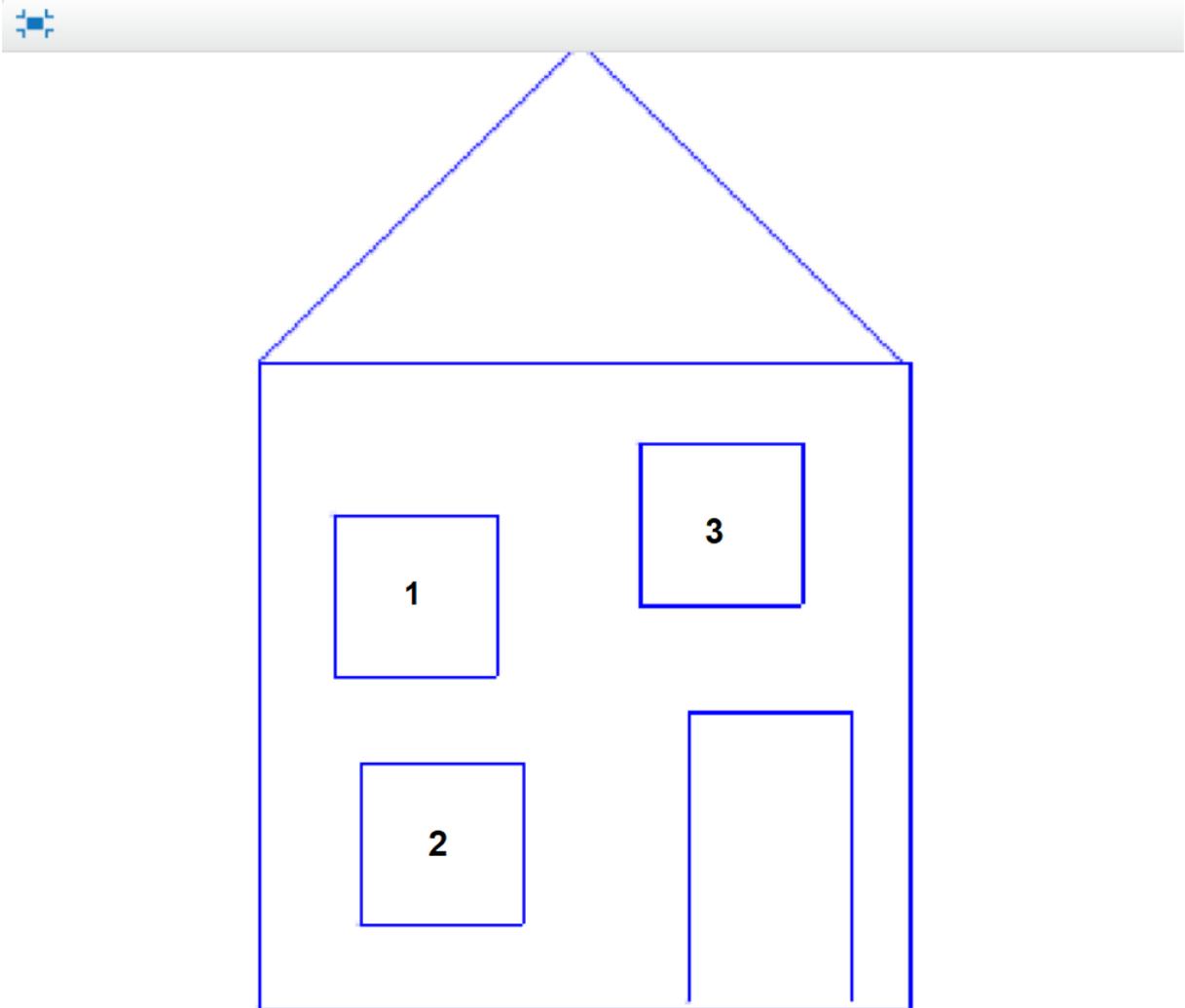
Disse que era fácil arrumar a ponta do triângulo, pois estava me referindo ao que já tinha comentado anteriormente sobre “descer” o desenho, ou seja, sobre mudar o ponto inicial de onde começava a programação. Além disso, outros estudantes cujos desenhos ultrapassaram o limite do palco já os tinham corrigido após minha observação.

A dupla José e Caio tentou fazer um triângulo retângulo e isósceles para representar o telhado no desenho da casa. Eles sabiam que os catetos deveriam ser iguais e, por isso, colocaram 140 nos dois blocos de movimento que simbolizam os lados do triângulo. Como o ângulo da parte superior do telhado era 90 graus, eles sabiam que os dois ângulos “de baixo” (ângulos da base do triângulo isósceles e retângulo) deveriam ser 45° . Porém, não se atentaram que o valor que escolheram para os catetos não formaria tal triângulo. Isso porque, tendo em vista que a hipotenusa é a base do telhado, ou seja, a largura da casa, sua medida é de 200 unidades de comprimento e, dessa forma, pela fórmula de Pitágoras, o valor dos catetos deveria ser de aproximadamente 141,42, ou, ainda, raiz quadrada de 20.000.

Após alguns minutos, os estudantes da dupla disseram ter finalizado a atividade e, então, dirigi-me ao computador no qual estavam para observar o que haviam feito. Na Figura 20, pode-se observar o desenho final e, na Figura 21, a

respectiva programação (exceto da janela 3), que dividi em três colunas para ficar legível ao leitor. O Quadro 7 apresenta nosso diálogo naquele momento.

Figura 20 – Desenho final da dupla José e Caio



Fonte: Dados da pesquisa.

Figura 21 – Programação da dupla José e Caio para desenhar a casa da Atividade 1



Fonte: Dados da pesquisa.

Quadro 7 – Diálogo entre mim, José e Caio, referente ao desenho e à programação final da casa (Parte 1)

José: Chega aí, professor, terminamos!

Pesquisador: Tá [sic] meio torto isso aí, não? As janelas não estão alinhadas. E vocês não arrumaram o telhado, né [sic]?

Caio: A gente tentou mudar o telhado, mas, se a gente muda os valores, fica diferente.

Fonte: Dados da pesquisa.

A dupla estava se referindo tanto ao valor do bloco de movimento, que representa os lados do triângulo, quanto ao valor do ângulo do vértice do telhado. Primeiramente, tal fato evidencia que eles não compreenderam que, para fazer o topo do telhado aparecer, era necessário mudar a coordenada y do início da programação, cujo valor escolhido foi -116, como pode ser observado na Figura 21 com o bloco “vá para x : -161, y : -116”. Em segundo lugar, como eles optaram por desenhar um triângulo retângulo e isósceles, deveriam usar a fórmula de Pitágoras para obter o valor dos catetos, já que um dos lados do triângulo era também um lado da casa, como citado anteriormente. O Quadro 8 apresenta a parte 2 desse diálogo.

Quadro 8 – Diálogo entre mim, José e Caio, referente ao desenho e à programação final da casa (Parte 2)

Pesquisador: E essas janelas?

Caio: A gente copiou a primeira, escolheu o lugar e fez as outras. Tipo, a gente fez a primeira janela, depois, a porta, né [sic]? Só que, aí, ficou muito vazio, então, a gente fez mais duas.

Pesquisador: E aí vocês não alinharam por quê?

José: Você fala de deixar aqui e aqui retinho? [referindo-se ao alinhamento dos lados das janelas 1 e 2 da Figura 20].

Pesquisador: Isso. E deixar essas duas na mesma altura [referindo-me às janelas 2 e 3 da Figura 20].

Fonte: Dados da pesquisa.

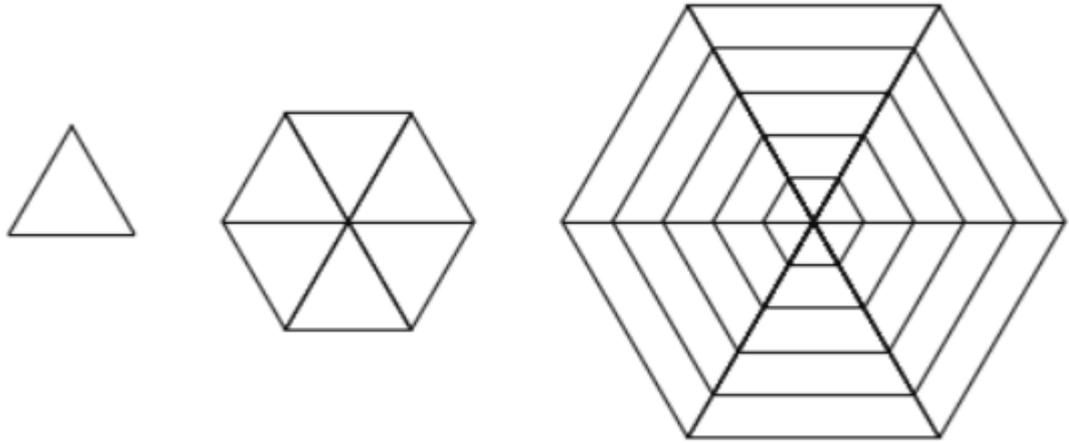
Depois disso, a dupla de estudantes não fez mais alterações na programação. Apesar de **reconhecerem o padrão** dos blocos que configuram o desenho da janela e usarem a ferramenta “duplicar” para fazer as demais, o que, no diálogo do Quadro 8, pode ser depreendido do excerto: “a gente **copiou a primeira, escolheu o lugar e fez as outras**”, os estudantes não aprimoraram nem o desenho e, conseqüentemente, nem a programação. Quando revi a atividade deles, notei que alguns traços estavam mais largos que outros, evidenciando que, em algum momento, eles clicaram no bloco “adicione _ ao tamanho da caneta” ou no bloco “mude o tamanho da caneta para _”. A diferença é sutil, mas, na Figura 21, pode-se observar que tanto o canto inferior direito das janelas quanto da casa parecem não se conectar inteiramente. Além disso, a parte inferior da porta não se liga à parte inferior da casa. Isso ocorre porque os estudantes iniciaram o desenho da porta na coordenada $y = -114$, enquanto que a parte inferior da casa é iniciada em $y = -116$.

Quando os estudantes da dupla dizem, no Quadro 5, que “*tentaram* algumas coisas”, como, por exemplo, corrigir o risco que estava no desenho da casa (Figuras 19 e 20) e se dedicar à construção do telhado, tornam-se evidentes os conceitos de **testar e depurar** e **ser incremental e iterativo** (BRENNAN; RESNICK, 2012), bem como o de **explorar** (BENTON et al., 2016, 2017), pois, ao se depararem com problemas, tais como um risco indesejado no desenho e a imperfeição do telhado, perceberam que seria algo na programação do *Scratch*. Assim, tentaram mudar os blocos (a programação), porém, não obtiveram êxito. Além disso, ao **explicar** (BENTON et al., 2016, 2017) o que estava ocorrendo na atividade, tanto no momento do risco quanto no do telhado, o conceito de **conectar** (BRENNAN; RESNICK, 2012) também emergiu, já que os estudantes estavam interagindo entre si, com o intuito de realizar a atividade, e, comigo, quando não conseguiram progredir nela. Por fim, há a **conexão** (BENTON et al., 2016) do conteúdo matemático com o *Scratch* por meio da linguagem de programação do *software*.

5.2 ATIVIDADE 4 COM A TURMA 9^o1-T2

O intuito da atividade foi o de trazer o tema de polígonos regulares e a função da Paleta Variáveis, que, assim como a Paleta Mais Blocos, não apresenta nenhum bloco ao ser selecionada. Nesse sentido, o usuário é responsável por criar uma ou mais variáveis e usá-las conforme o conteúdo ou o objetivo da programação. Sendo assim, a primeira parte da atividade consistiu em criar uma variável, observar os novos blocos que surgem ao fazê-lo e entender a função de cada um deles, tal como descrito no Apêndice A desta dissertação. Já a segunda parte foi composta por atividade na qual os estudantes, mediante os conhecimentos adquiridos na primeira, criavam programações que desenhavam um triângulo equilátero e, a partir dele, formavam tanto um hexágono regular quanto uma “teia de aranha”. Na Figura 22, a seguir, pode-se observar os respectivos desenhos de um triângulo equilátero, de um hexágono regular, formado por seis desses triângulos, e de uma representação de uma teia de aranha, formada por esses triângulos equiláteros com o tamanho dos lados crescentes.

Figura 22 – Representação de um triângulo equilátero, de um hexágono formado por seis triângulos equiláteros e de uma “teia de aranha” formada por triângulos equiláteros de comprimento crescente



Fonte: O Autor.

Iniciei a atividade explicando aos estudantes o que faríamos naquele dia, assim como citado no parágrafo anterior. Então, comentei que uma variável poderia ser numérica, isto é, que poderia ser um número, inteiro ou decimal, usado, por exemplo, para a pontuação de um jogo, para alterar o tamanho de um ator, para mostrar o tempo decorrido ou, ainda, para alterar os lados de uma figura geométrica. Além disso, disse que a variável poderia ser uma *string*, ou seja, uma sequência de caracteres incluindo letras, números e outros símbolos. Desse modo, pedi que criassem uma variável referente à medida do lado das figuras que desenvolveríamos. Em seguida, perguntei aos estudantes se eles sabiam o que é um polígono regular, cujo diálogo está reproduzido no Quadro 9.

Quadro 9 – Diálogo com a turma 9^o1-T2 acerca da definição de polígono regular

Pesquisador: Vocês lembram o que é um polígono regular?

Luiz: Tem que ter os ângulos iguais.

Laisa: Não é [sic] os lados?

Pesquisador: Isso, na verdade, são os dois juntos. O polígono regular tem que ter os lados iguais e os ângulos também.

Vocês lembram algum polígono regular?

Rafael: O triângulo, o quadrado.

Ryhan: Tem vários, não tem, professor?

Pesquisador: Então, o triângulo é um polígono de três lados [nesse momento, desenhei vários triângulos na lousa]. Mas vejam, pra [sic] ser regular, os lados têm que ser iguais [apontando para os desenhos na lousa].

Ana: Então, é um triângulo com todos os lados iguais.

Rhaylla: É o triângulo equilátero.

Pesquisador: Isso, o triângulo equilátero é o polígono regular de três lados. Vocês lembram qual o valor do ângulo interno desse triângulo?

Rhaylla: 60.

Pesquisador: Isso, o ângulo interno do triângulo equilátero é 60 graus.

Vejam que o que o Ryhan falou é verdade. Tem vários polígonos regulares. Qual o polígono regular de 5 lados?

Kayo: Pentágono.

Pesquisador: E de 6?

Ana: Hexágono.

Pesquisador: E assim por diante, certo? Agora, vocês vão fazer a programação para desenhar um triângulo equilátero.

Fonte: Dados da pesquisa.

Quando, no momento do diálogo descrito no Quadro 9, fiz desenhos de vários triângulos na lousa, pude ouvir um dos estudantes dizer “isósceles” para indicar o polígono regular de três lados. Assim, aproveitei a oportunidade e expliquei-lhes as classificações dos triângulos quanto aos ângulos e aos lados.

Na sequência, comentei a definição e alguns conceitos relativos a polígonos regulares, tais como ângulos internos, ângulos externos e soma dos ângulos internos, mostrando as seguintes fórmulas na lousa:

- a) $S_i = (n - 2) \cdot 180$, em que S_i é a soma dos ângulos internos e n é o número de lados do polígono;
- b) $A = \frac{(n-2) \cdot 180}{n}$, em que A é a medida do ângulo interno e n é o número de lados do polígono;
- c) $E = 180 - A$ ou $E = \frac{360}{n}$, em que E é a medida do ângulo externo, A é a medida do ângulo interno e n é o número de lados do polígono.

Seguindo a tarefa e aproveitando o desenho na lousa, perguntei a eles qual seria a programação para construirmos um triângulo equilátero. Então, um dos estudantes, Leonardo, apontou para a lousa e, gesticulando acerca do desenho, disse o que pode ser observado no Quadro 10, a seguir:

Quadro 10 – Fala do estudante Leonardo sobre a programação para desenhar um triângulo equilátero

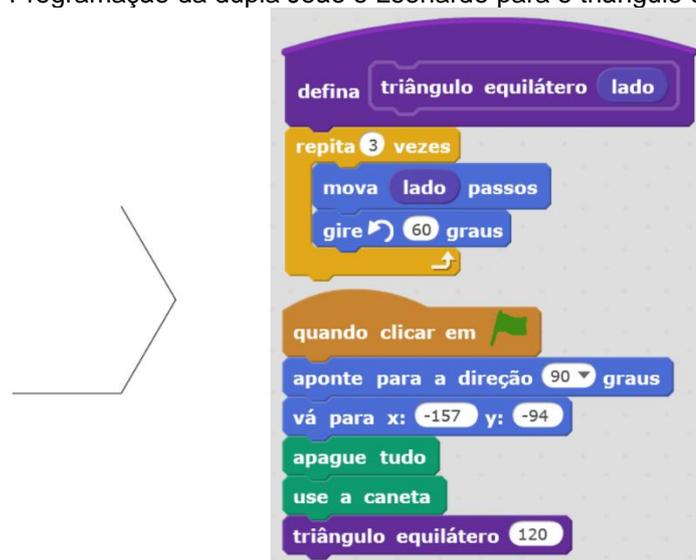
Leonardo: Tem que, tipo, andar o lado, aí virar, andar de novo, virar e andar o último lado.

Fonte: Dados da pesquisa.

Ao **explicar** (BENTON et al., 2016, 2017) como seria a programação, Leonardo se **expressa** (BRENNAN; RESNICK, 2012), fazendo associações entre o seu pensamento e a linguagem do *software*, ou seja, o estudante tem a ideia, na sua mente, de como desenhar um triângulo com lápis e papel, por exemplo, e a expressa por meio de uma linguagem “mista”, caracterizada pela soma da linguagem falada pelo estudante com a linguagem do *Scratch* ao dizer “andar o lado”. Essa linguagem fica evidente quando observamos a programação da dupla João e Leonardo (vide Figura 23), na qual eles usam o bloco “mova *lado* passos”, representando a fala “andar o lado”, e “gire 60 graus”, representando a fala “virar”.

Posteriormente, enquanto os alunos desenvolviam a programação, surgiram dúvidas referentes ao bloco para executar o comando de girar, visto que estavam colocando 60 graus para efetuar essa operação, valor correspondente ao ângulo interno do triângulo equilátero. Como exemplo do resultado obtido, na Figura 23, podemos observar o desenho gerado pela programação da dupla João e Leonardo, bem como a programação em si.

Figura 23 – Programação da dupla João e Leonardo para o triângulo equilátero



Fonte: Dados da pesquisa.

Por meio do *feedback* (TIKHOMIROV, 1981; PAPERT, 1980), puderam perceber que algo não estava bom no esboço do triângulo. Significa dizer que o

pensamento inicial de que o ator precisava girar a quantidade de graus referente ao ângulo interno estava equivocada, já que a figura esboçada não era a representação de um triângulo equilátero, conforme queriam desenhar com a rotina escrita por eles. A seguir, o diálogo mostra o questionamento do estudante:

Quadro 11 – Diálogo entre mim e Leonardo

Leonardo: Mas, professor, essa linha não está inclinada 60 graus pra [sic] lá? [inclinando o braço para representar o lado do triângulo desenhado na lousa].

Pesquisador: 60 graus em relação a quê?

Leonardo: Ao lado do triângulo [referindo-se a uma das bases do triângulo].

Fonte: Dados da pesquisa.

Na ação de **explorar** o *software*, apesar de a dupla ser **incremental e iterativa**, ao **testar e depurar** a programação que haviam criado, perceberam que o desenho não correspondia a um triângulo e muito menos equilátero. No diálogo, o que o estudante Leonardo estava querendo dizer é que, como os ângulos internos são 60 graus, então, um lado está inclinado 60 graus em relação ao outro. Porém, como a direção de seu ator era horizontal para a direita, representada pelo bloco “aponte para a direção 90 graus”, e, no plano cartesiano, a orientação é no sentido anti-horário, o lado do triângulo equilátero está inclinado 120 graus em relação ao eixo horizontal, ou seja, a quantidade de graus a ser girada é referente ao ângulo externo e não ao interno, como eles haviam pensado.

Então, depois que eles me disseram que o lado estava inclinado 60 graus em relação à base (eixo), fiz a observação do parágrafo anterior acerca de a orientação do plano ser no sentido anti-horário, afirmando que eles estavam olhando a inclinação da esquerda para a direita quando, na verdade, deveriam fazer o contrário. Depois disso, deixei a dupla por uns instantes para alterarem a programação com os ajustes necessários. Ao voltar, percebi que eles tinham conseguido corrigi-la e o nosso diálogo a esse respeito pode ser observado no Quadro 12.

Quadro 12 – Diálogo com João e Leonardo

Pesquisador: E aí, deu certo agora?

João: Deu, professor.

Pesquisador: O que vocês fizeram?

João: A gente arrumou o bloco que gira.

Leonardo: É, a gente mudou o lado que gira, mas não era isso. Como ele gira daqui pra lá [apontando para a tela do computador e indicando o giro no sentido anti-horário] e aqui é 60 [referindo-se ao ângulo interno do triângulo], então, tem que girar 120 graus.

Pesquisador: É isso aí! E isso tem alguma relação com o que eu passei na lousa?

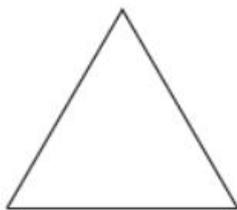
Leonardo: Ah, é o ângulo externo do triângulo.

João: Ah, entendi! É aquele “E” ali [referindo-se à fórmula do ângulo externo escrita na lousa], que é 180 menos o ângulo, que é 60.

Fonte: Dados da pesquisa.

Quando Leonardo afirma que mudaram o bloco que gira, ele queria dizer que escolheram o bloco para girar no sentido horário, mas mantendo os mesmos 60 graus, o que ainda fazia com que o desenho ficasse incorreto. Inicialmente, os alunos não associaram que a quantidade de graus que deveriam usar era o valor do ângulo externo, mas chegaram ao total de 120 graus com o raciocínio de que essa era a quantidade que o ator deveria girar em relação ao eixo horizontal, já que o ângulo interno do triângulo era de 60 graus. Isso fica evidente com fala de Leonardo: “como ele gira *daqui pra lá* [apontando para a tela do computador e indicando o giro no sentido anti-horário] e *aqui é 60* referindo-se ao ângulo interno do triângulo], *então, tem que girar 120 graus*”. Na Figura 24, pode-se observar a programação final da dupla para realizar um triângulo equilátero e seu respectivo desenho.

Figura 24 – Desenho e procedimento “triângulo equilátero” da dupla João e Leonardo



Fonte: Dados da pesquisa.

Disse aos estudantes, fazendo um desenho na lousa, que um hexágono regular é um polígono com seis lados iguais e seis ângulos internos congruentes,

passível de ser dividido em seis triângulos equiláteros. Assim, pedi para que criassem um procedimento de um hexágono regular a partir do bloco do triângulo equilátero já feito, usando também a ferramenta de entrada numérica, posto que o tamanho dos triângulos seria diferente para formar a teia de aranha. No Quadro 13, pode-se observar o diálogo da dupla João e Leonardo antes de iniciarem a programação do hexágono.

Quadro 13 – Diálogo entre mim, João e Leonardo antes de iniciarem a programação do hexágono

Leonardo: Então, a gente vai pegar esse bloco [referindo-se ao bloco do triângulo equilátero] e, com ele, a gente vai construir um hexágono?

Pesquisador: Isso, vocês vão criar um novo bloco chamado hexágono regular e, na programação desse novo bloco, tem que ter o bloco que vocês já criaram do triângulo.

João: Tá [sic]. Vai ter que ter 6 triângulos, então, como tá [sic] lá na lousa. A gente só precisa saber como, agora. Tipo, a gente faz um e vê onde tem que começar o outro.

Leonardo: É, a gente pode começar na origem pra [sic] saber aonde termina e começar o outro. Faz aí.

Fonte: Dados da pesquisa.

O intuito da dupla era fazer o ator iniciar a programação na origem para ver onde ele se encontraria ao fim do traçado, isto é, quais seriam suas coordenadas quando desenhasse o primeiro triângulo. Deste modo, eles usariam o bloco “vá para x: _ y: _” para fazer o segundo triângulo e assim sucessivamente, até formarem o hexágono. Porém, os estudantes ficaram surpresos quando perceberam que o ator finalizou o desenho na origem. Tal fato pode ser observado no Quadro 14 a seguir.

Quadro 14 – Diálogo da dupla João e Leonardo durante a programação do hexágono regular

João: Ué, terminou na origem.

Leonardo: Tá [sic] apontado pra [sic] direita.

João: Tem que fazer aqui, né? [sic] [apontando para a tela e se referindo ao fato de que o segundo triângulo começaria encostado no primeiro]. De ponta-cabeça, será?

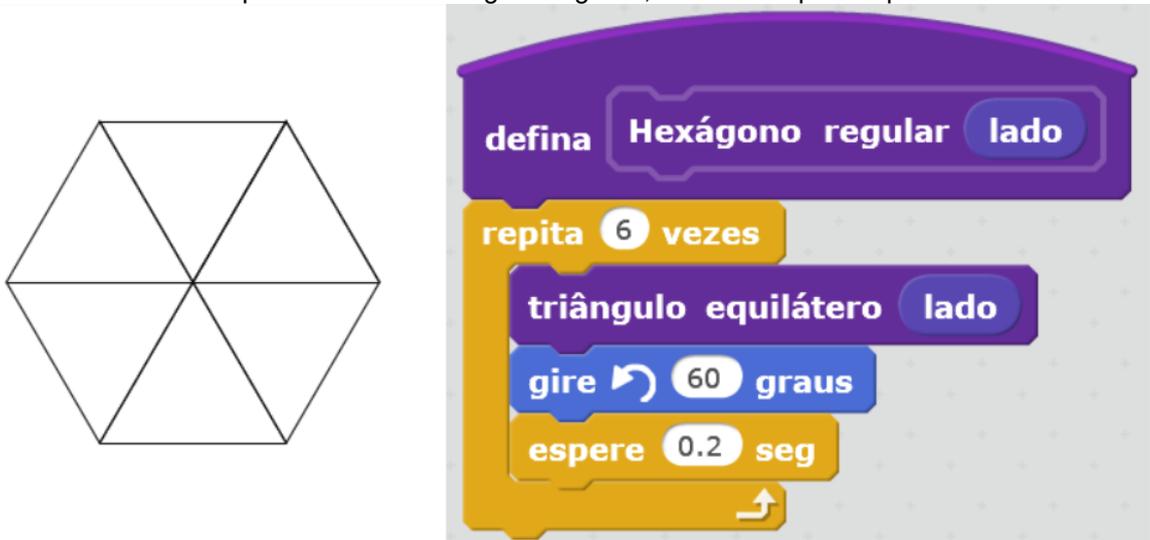
Leonardo: É, acompanha esse [indicando o lado do primeiro triângulo] e termina na origem também [usando os dedos para desenhar o segundo triângulo na tela].

Acho que é o mesmo esquema do outro, tem que girar pra fazer.

Fonte: Dados da pesquisa.

Usando a mesma lógica do bloco do triângulo equilátero, a dupla conseguiu realizar a programação para desenhar o hexágono regular, como pode ser observado na Figura 25.

Figura 25 – Desenho e procedimento “Hexágono regular”, elaborados pela dupla João e Leonardo



Fonte: Dados da pesquisa.

Na sequência, disse aos estudantes que deveriam elaborar um procedimento, cujo nome poderia ser “teia”, caracterizado por hexágonos regulares concêntricos, feitos a partir de um triângulo equilátero, mas, agora, com o tamanho dos lados crescente. Isso significa que a medida do lado dos triângulos aumenta, causando a impressão de que os hexágonos se afastam do centro. Logo, podemos dizer que o lado do hexágono é variável, o que levou a dupla João e Leonardo a criar uma variável denominada “tamanho do lado”. Portanto, ao idealizarem o procedimento “teia”, usaram o bloco “Hexágono regular” e, no espaço reservado ao parâmetro numérico, inseriram a variável “tamanho do lado”, como observado na Figura 26.

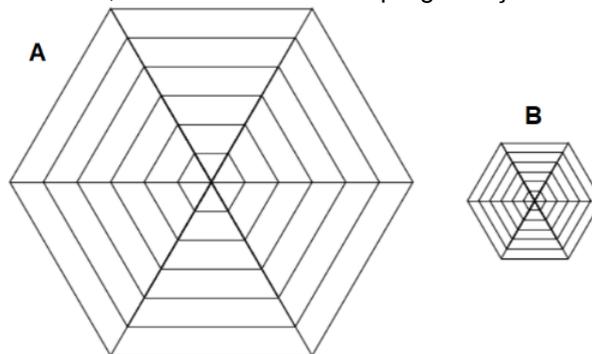
Figura 26 – Procedimento “teia”, realizado pela dupla João e Leonardo



Fonte: Dados da pesquisa.

A execução do procedimento “teia” (vide Figura 26) faz com que o tamanho do lado do hexágono seja zero (0) e, a cada repetição, adicione-se 20 unidades à medida do lado, resultando, ao final das oito vezes, na Figura 27. Em tal procedimento, o número dessas repetições representa a quantidade de “bordas” da teia, e o valor numérico adicionado a cada uma delas simboliza o seu tamanho. Conseqüentemente, o espaçamento das “bordas”, como pode ser observado na Figura 27, em que ambas as teias têm seis repetições, na teia A, foi de 30 unidades e, na B, de 10.

Figura 27 – Teias A e B, desenvolvidas com a programação de João e Leonardo



Fonte: Dados da pesquisa.

A Figura 28 apresenta a área de programação da dupla ao final da atividade.

Figura 28 – Área de programação da dupla João e Leonardo ao fim da Atividade 4



Fonte: Dados da pesquisa.

Ao analisarmos as programações que a dupla desenvolveu, podemos evidenciar conceitos relacionados ao PC, tais como: **sequências; algoritmos e procedimentos; ciclos**, já que utilizaram o bloco “repita” nas programações do triângulo, do hexágono e da teia; **eventos**, pois, para o ator começar os respectivos desenhos, foi necessário clicar na bandeira verde; **dados**, uma vez que usaram variáveis; **coleção e análise de dados**, visto que reuniram as informações acerca de polígonos regulares a fim de compreender e encontrar padrões na programação para desenhá-los; **decomposição do problema**, posto que avaliaram que o hexágono é formado por seis triângulos e, dessa forma, entenderam que deveriam, a partir do bloco do triângulo equilátero, desenhar o hexágono regular. Além disso, nos diálogos apresentados, vemos que João e Leonardo **testaram, depuraram e reutilizaram** os códigos de um polígono para o outro e, depois, para a programação da teia. Em todo esse processo, houve igualmente o conceito de **abstrair e modularizar**, uma vez que, para concluir sua programação, os estudantes reuniram as informações anteriores, as quais estavam divididas, e conseguiram compô-las em uma única que fizesse o desenho pedido. Houve também os conceitos de **explorar, expressar/explicar, questionar, prever, trocar e conectar**, dado que toda a atividade foi uma exploração de algumas funcionalidades do *software*, o que traz a oportunidade de emergir outros conceitos. Além disso, quando a dupla conversou comigo e com seus outros colegas, eles questionaram e expressaram suas ideias, proporcionando momentos de interações e reflexões, que, nas diversas vezes, colaborou positivamente para o coletivo da turma no desenvolvimento da atividade.

5.3 ATIVIDADE 5 COM A TURMA 9^o1-T2

Na última atividade, os estudantes desenvolveram uma programação que calculava a área de um quadrilátero, de um triângulo e de uma circunferência. Para tanto, eles usaram os conceitos de variáveis, de ler números e de caracteres e tomada de decisão. Toda a funcionalidade da Paleta Variáveis já havia sido explicitada durante a tarefa anterior (4) e, desse modo, a primeira parte da Atividade 5 foi dedicada exclusivamente à explicação dos blocos “pergunte _ e espere a resposta”, “resposta”, “se” e “se/senão”. Disse aos discentes que, para serem

executadas, algumas programações requerem a inserção de dados por parte do usuário, como aquela que calcula nosso Índice de Massa Corpórea (IMC), por exemplo. Disse também que esse cálculo era comum, havendo diversos sites que o faziam, e, em seguida, perguntei se sabiam o que significava IMC e como era mensurado. Devido ao fato de a maioria desconhecer-lo, expliquei o que era e, ao mesmo tempo, coloquei na lousa do laboratório a fórmula utilizada para realizar tal medição, a saber, $IMC = \frac{peso}{altura^2} = \frac{peso}{altura \times altura}$. Na sequência, esclareci que peso, altura e IMC seriam, então, as variáveis mobilizadas na programação e, tendo em vista que o valor das duas primeiras deveria ser informado pelo usuário, precisaríamos de um recurso no *Scratch* que armazenasse a informação. Foi quando lhes apresentei o bloco “pergunte _ e espere a resposta”, com o *Scratch* projetado na TV do laboratório. No Quadro 15 a seguir, reproduzimos o diálogo com alguns estudantes do 9º1-T2 naquela ocasião.

Quadro 15 – Diálogo entre mim, Leonardo, Pedro, Rafael e Rhaylla sobre os blocos “pergunte _ e espere a resposta” e “resposta”

Pesquisador: O *Scratch* tem essa função de armazenamento, tá [sic] aqui na Paleta Sensores. Estão vendo esse bloco aqui? [apontando para o bloco “pergunte _ e espere a resposta”]. Arrastem ele para a área de programação.

Rafael: E essa pergunta aí, professor? [O estudante estava se referindo ao que já vem originalmente escrito no bloco, ou seja, “*What’s your name?*”].

Pesquisador: Não se preocupem, é justamente aí que a gente vai mudar. Clica duas vezes no bloco e vê o que acontece.

Pedro: Apareceu uma barra embaixo.

Leonardo: E o boneco falou também. [quando o bloco “pergunte _ e espere a resposta” é executado, aparece um balão de fala no ator, com os dizeres da entrada do bloco e, na parte inferior do palco, aparece uma barra onde o usuário pode digitar números, letras ou símbolos].

Pesquisador: Exatamente. Estão vendo essa barra? Nela, a gente pode digitar letras, números ou símbolos que o *Scratch* armazena essa informação.

Rhaylla: Como assim?

Pesquisador: Vocês estão vendo que aqui ao lado do bloco “resposta” tem um quadradinho que pode ser selecionado? [apontando para a TV do laboratório, para indicar a parte dos blocos]. Clica nele.

Rhaylla: Apareceu no palco, tipo a variável.

Pesquisador: Isso mesmo! Agora, digitem o nome de vocês na barra que apareceu e aperta *enter* ou clica no botão azul no final da barra para ver o que acontece.

Rafael: Aparece o que a gente escreveu. No meu, foi “Rafa”.

Pesquisador: Beleza! Então, esse bloco pega a informação que a gente escreveu, que pode ser um número também, e armazena no bloco resposta. É isso que a gente vai usar na programação pra calcular o IMC. A pessoa digita o peso e a altura dela, o *Scratch* armazena essas informações e faz o cálculo.

Fonte: Dados da pesquisa.

Naquele momento, os estudantes Leonardo, Pedro, Rafael e Rhaylla expuseram suas compreensões do que estavam desenvolvendo, apresentando, dessa maneira, suas perspectivas computacionais (BRENNAN; RESNICK, 2012) sobre a atividade. Além disso, a interação entre eles, ou seja, o contato de um *com* o outro, evidencia a **conexão** (BRENNAN; RESNICK, 2012) que tiveram na elaboração da atividade.

Em seguida, ainda com o *software* projetado na TV do laboratório, mostrei a eles uma programação que fazia o cálculo do IMC, conforme ilustrado na Figura 29, para discutirmos seu funcionamento.

Figura 29 – Programação para calcular o Índice de Massa Corpórea



Fonte: Dados da pesquisa.

No Quadro 16, apresento o diálogo que tecemos no momento de construção.

Quadro 16 – Diálogo com Pedro, Rafael e Rhaylla sobre a programação que calcula o IMC

Pesquisador: E aí, alguém sabe me dizer como essa programação funciona? [referindo-me à programação para calcular o IMC, como na Figura 29].

Rhaylla: A gente vai precisar do peso e da altura. Por isso, tem as duas perguntas.

Pesquisador: Exato! Vamos por parte, então. Esses dois primeiros blocos aqui,

como funcionam? [apontando para os blocos das linhas 1 e 2 da Figura 29].

Pedro: Tipo, o ator vai perguntar a minha altura, aí, eu digito na barra.

Pesquisador: Isso, digita e depois aperta *enter*.

Rafael: Professor, depois é só digitar o peso e apertar *enter* que ele calcula tudo?

Pesquisador: Sim! Vejam que, além das variáveis peso e altura, eu criei uma variável chamada IMC [apontando para a área dos blocos]. E é nessa linha aqui que [se] faz o cálculo da programação, estão vendo? [apontando para a área de programação, tal como se vê na linha 5 da Figura 29]. Pera [sic] aí! Deixa eu dar um zoom. Como que tá [sic] essa segunda parte do bloco, alguém consegue explicar? [referindo-me ao que foi posto no bloco da variável IMC (linha 5 da Figura 29)].

Rhaylla: Pegou um bloco de divisão e um de multiplicação, aí, colocou as variáveis. Esse final é pra [sic] quê? [mencionando o bloco “arredondamento de _” na linha 6 da Figura 29].

Pesquisador: Vocês estão vendo esse bloco de arredondamento aqui? Sem ele, a resposta seria um número com muitas casas decimais. Por isso, eu optei por colocar. Quando vocês forem fazer, façam o teste com e sem esse bloco. Outra coisa: podem vir aqui perto da TV pra [sic] ver como eu montei os blocos de operação no bloco do IMC. Estão vendo a fórmula ali? [apontando para a fórmula escrita na lousa]. Embaixo da fração, tá [sic] altura x altura, que eu coloquei nesse bloco aqui [apontando para o bloco de operação na linha 5 da Figura 29].

Fonte: Dados da pesquisa.

Naquele momento, os estudantes começaram a **explicar** o que estavam entendendo da programação, **prevendo** seu funcionamento. Isso fez com que eles se **expressassem** e se **questionassem** acerca do mecanismo dos blocos para realizar o cálculo do IMC. Após a explicação da programação para esse cálculo, pedi para que eles a reproduzissem em seus computadores, de modo a verificarem se surgiam outras dúvidas e para que pudessem ver a diferença da resposta com e sem o uso do bloco “arredondamento de _”. Assim, na sequência da explicação dos blocos “pergunte _ e espere a resposta” e “resposta”, assim como de suas funcionalidades, partimos para os blocos de decisão (blocos condicionais) “se” e “se/senão”. A explicação que eu segui com os estudantes é a mesma apresentada no Apêndice A, a saber, colocando um desenho na lousa semelhante ao da Figura 30 a seguir.

Figura 30 – A estrutura do bloco *if* (se)



Fonte: Marji (2014).

Disse aos estudantes que o tipo de bloco que se encaixava no cabeçalho do bloco “se” tinha o formato hexagonal (bloco verde escrito *Condition=true* na Figura 30) e representava blocos de decisão, os quais forneciam respostas de sim ou não (verdadeiro ou falso) a uma pergunta. Caso a condição de teste do bloco hexagonal fosse verdadeira, o programa executaria os comandos listados “dentro” do bloco “se”, que vão do *Command 1* ao *Command N* para, então, seguir para aquele(s) que viesse(m) depois, denominado(s) *Command M*. Entretanto, se a condição de teste fosse falsa, o programa não executaria os comandos de dentro do bloco e seguiria direto para o *Command M*.

Com vistas a exemplificar o funcionamento do bloco “se”, apresentei aos discentes a programação dada a ver na Figura 31, a seguir, cujo intuito era avaliar o desempenho de algum estudante por meio do cálculo da média aritmética da nota de duas de suas provas. Para tanto, o usuário informava o valor numérico obtido em suas avaliações e, então, o ator dizia a média. Além disso, se a média fosse maior ou igual a cinco, o ator dizia: “Parabéns, aprovado!”. Caso contrário, isto é, se a média fosse menor que cinco, então, o ator dizia: “Infelizmente, você reprovou”.

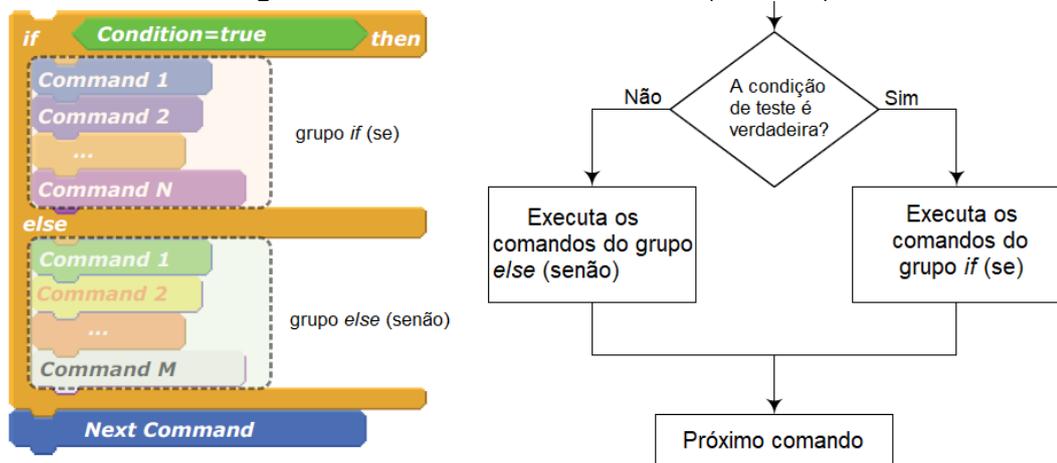
Figura 31 – Programação para exemplificar o uso do bloco “se”



Fonte: O Autor.

Em seguida, expliquei aos alunos o uso bloco “se/senão” de maneira semelhante, fazendo na lousa um desenho parecido com o da Figura 32 e explicando parte por parte cada funcionalidade da programação.

Figura 32 – A estrutura do bloco *if/else* (se/senão)



Fonte: Marji (2014).

Inicialmente, evidenciei aos estudantes que o bloco possuía dois grupos de comando: um referente ao “se” e outro ao “senão” (na Figura 32, identificados por *if* e *else*). Então, naquela situação, caso a condição de teste do bloco hexagonal fosse verdadeira, o programa executaria os comandos referentes ao grupo “se” (de *Command 1* a *Command N*). Em contrapartida, se a condição fosse falsa, seriam os comandos do grupo “senão” que seriam executados (de *Command 1* a *Command M*). A programação executaria apenas um dos dois grupos de comando para, posteriormente, seguir para o comando seguinte (*Next Command*). Para elucidar o funcionamento do bloco “se/senão”, mostrei-lhes uma programação semelhante à da Figura 31, mas, nela, aprovação contava, além da média aritmética, com a frequência dos alunos, ou seja, para o usuário ser aprovado, precisaria obter uma média acima ou igual a cinco e uma frequência maior ou igual a 70%. Tal programação pode ser observada na Figura 33.

Figura 33 – Programação para exemplificar o uso do bloco “se/senão”



Fonte: O Autor.

As cinco primeiras linhas da programação da Figura 33 eram iguais às da Figura 31. A mudança ocorria a partir da sexta linha, na qual se perguntava ao usuário a sua frequência para que esse valor fosse atribuído à variável “freq” (linha sete da Figura 33). O bloco “se/senão” pode ser visto na nona linha da Figura 33 e, em seu cabeçalho, havia a condição de a média aritmética das duas provas ser maior ou igual a cinco. Assim, caso essa condição fosse verdadeira, os comandos do grupo “se” seriam executados, porém, como havia duas condições para a aprovação, quais sejam ter média maior ou igual a cinco e frequência maior ou igual a 70%, esses comandos pertenciam a outro bloco, denominado “se/senão” (linha 10 da Figura 33), que correspondia à avaliação da frequência do estudante. Desse modo, se ela fosse maior ou igual a 70%, o estudante seria aprovado, sendo

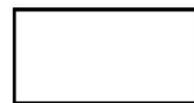
executado o bloco “diga *Parabéns, aprovado!* por 2 segundos”. Caso contrário, ou seja, se a frequência fosse menor que 70%, o estudante seria reprovado e, conseqüentemente, a programação executaria o bloco “diga *Reprovado(a) por frequência baixa* por 2 segundos”. Já se a condição do cabeçalho do bloco “se/senão” da linha nove da Figura 33 fosse falsa, isto é, se a média aritmética das provas fosse menor que cinco, então, seria executado o comando do grupo “senão”, com o bloco “diga *Reprovado(a) por média baixa* por 2 segundos”.

Após a explicação dos blocos de decisão, partimos para a segunda parte da Atividade 5, na qual os estudantes deveriam desenvolver uma programação que calculasse a área de três figuras geométricas planas, a saber: retângulo, triângulo e circunferência. Para que tivessem uma melhor compreensão do que foi solicitado, considerei adequado mostrar o que eu já havia feito, exibindo, no entanto, apenas o palco com o programa em execução. A seguir, farei o mesmo para que o leitor entenda como o *software* funciona e, dessa maneira, acompanhe a análise do desenvolvimento da atividade realizada pelos alunos.

Ao abrir a programação, o usuário encontra três opções à esquerda do palco: retângulo, triângulo e circunferência, associadas, respectivamente, aos números 1, 2 e 3. Além disso, à direita dele, há figuras geométricas referentes a essas opções. Por fim, próximo ao seu centro, encontra-se o ator. A Figura 34 ilustra o que foi supracitado.

Figura 34 – Palco do programa que calcula as áreas de um retângulo, de um triângulo e de uma circunferência (parte 1)

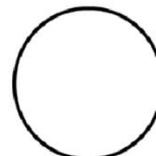
1) Retângulo



2) Triângulo



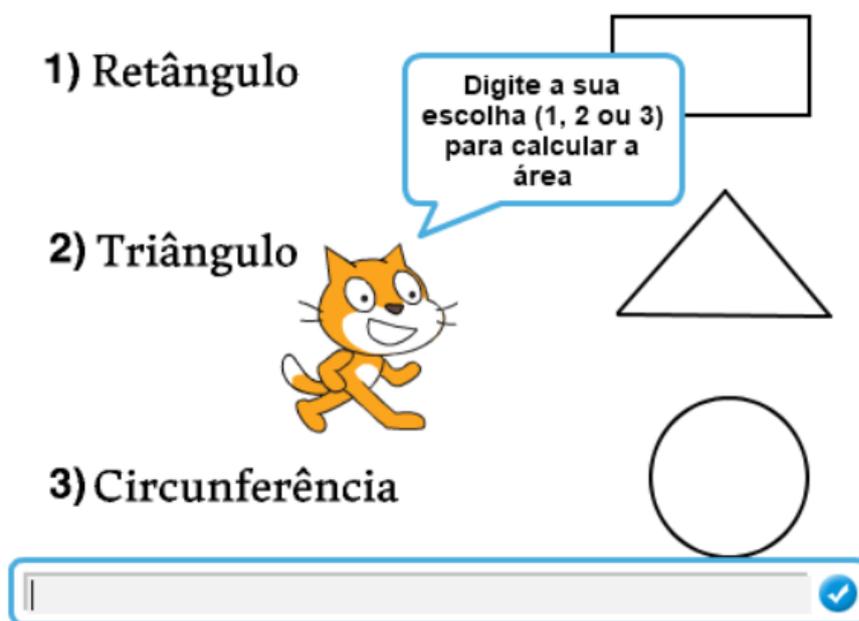
3) Circunferência



Fonte: O Autor.

Quando a programação é executada, o ator pede ao usuário que escolha de qual das figuras deseja calcular a área. Para isso, deve escolher o número correspondente a ela. Por exemplo, caso deseje calcular a área de um retângulo, deve digitar o número um e, em seguida, apertar *enter* ou clicar no ícone azul da barra inferior. Tal situação pode ser observada na Figura 35 a seguir.

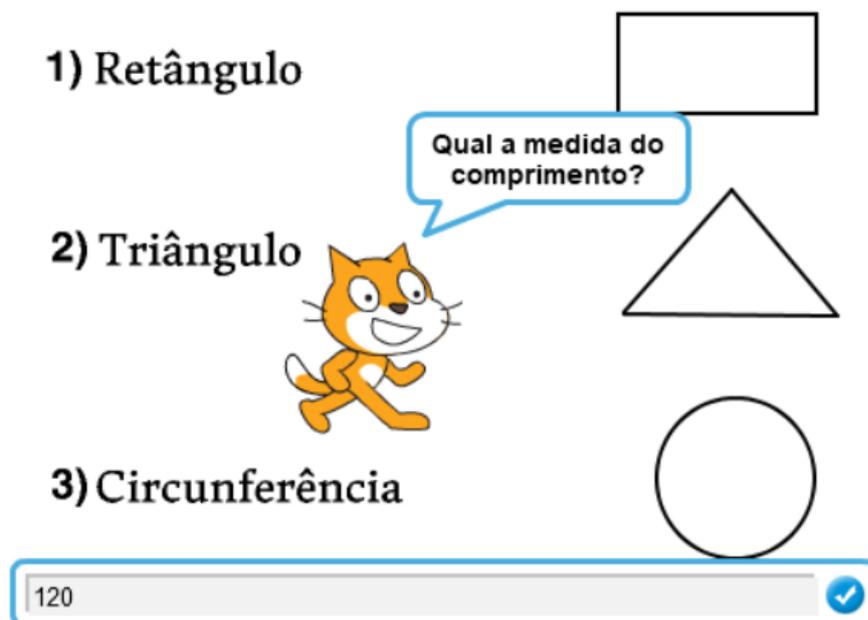
Figura 35 – Palco do programa que calcula as áreas de um retângulo, de um triângulo e de uma circunferência (parte 2)



Fonte: O Autor.

Supondo que a escolha tenha sido calcular a área de um retângulo, o ator pedirá ao usuário que forneça a medida do comprimento dessa forma geométrica; desse modo, deve-se colocar esse valor na barra inferior e apertar *enter* ou clicar no ícone azul da barra. Na Figura 36, é ilustrado que o valor informado para a medida do comprimento é 120.

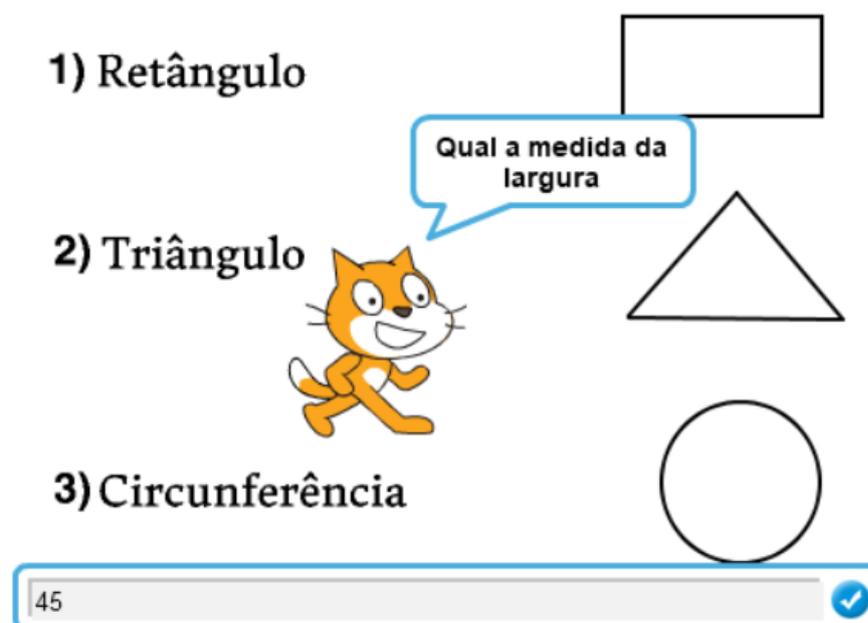
Figura 36 – Palco do programa que calcula as áreas de um retângulo, de um triângulo e de uma circunferência (parte 3)



Fonte: O Autor.

Na sequência, o ator pedirá ao usuário que forneça a medida da largura, seguindo o mesmo procedimento descrito anteriormente. Na Figura 37, pode-se observar esse momento da programação, sendo que o valor informado para a largura é 45.

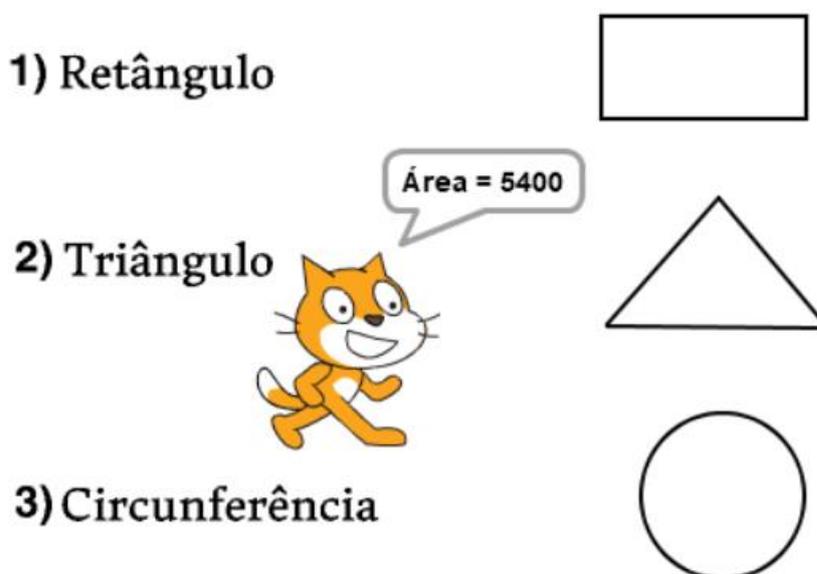
Figura 37 – Palco do programa que calcula as áreas de um retângulo, de um triângulo e de uma circunferência (parte 4)



Fonte: O Autor.

Por fim, mediante as informações das medidas do comprimento e da largura, o ator informa ao usuário quanto vale a área do retângulo, tal como pode ser observado na Figura 38. Nesse caso, sendo 120 e 45 as respectivas medidas, a área é 5400.

Figura 38 – Palco do programa que calcula as áreas de um retângulo, de um triângulo e de uma circunferência (parte 5)



Fonte: O Autor.

A programação funciona analogamente para o cálculo da área do triângulo e da circunferência. Então, assim como explicitado aqui, foi dessa maneira que expliquei aos estudantes como deveriam desenvolver a última atividade. Dessa forma, a seguir, volto a descrever o desenvolvimento da tarefa.

Assim, após a explicação, perguntei aos estudantes se eles se lembravam das fórmulas para calcular as referidas áreas. No Quadro 17, apresento o diálogo que tive com os discentes Pedro e Rhaylla acerca do cálculo das áreas das figuras planas analisadas nessa atividade.

Quadro 17 – Diálogo com Pedro e Rhaylla sobre as fórmulas das áreas das figuras planas

Pesquisador: Agora que eu expliquei o que deve ser feito, eu quero saber se vocês lembram de alguma fórmula para calcular a área dessas figuras.

Rhaylla: Tipo, o senhor tá [sic] falando do retângulo, essas coisas?

Pesquisador: Isso mesmo.

Rhaylla: Ah, então, do retângulo é lado vezes lado.

Pedro: O triângulo, eu sei que tem mais de uma.

Pesquisador: A Rhaylla disse que, pra [sic] calcular a área do retângulo, é lado vezes lado, certo? Como vamos criar variáveis para esses cálculos, a gente vai

definir que a área do retângulo é o comprimento vezes a largura, beleza?

Rhaylla: Então, é uma variável largura e outra comprimento?

Pesquisador: Isso. E a [fórmula para calcular a área] do triângulo, o Pedro disse que tem mais de uma, alguém lembra?

Pedro: Sei que tem mais de uma, mas pode ser base vezes altura, dividido por dois.

Pesquisador: Exatamente! Tem mais de uma fórmula para calcular a área dos triângulos, mas vamos usar essa que o Pedro falou, ok?

Faltou a fórmula para calcular a área de uma circunferência. Alguém sabe ou alguém lembra?

Fonte: Dados da pesquisa.

O conteúdo sobre o estudo das circunferências estava programado para o quarto bimestre e, por isso, no momento da Atividade 5, nenhum dos estudantes soube responder corretamente qual era a fórmula da área dessa forma geométrica. De modo resumido, escrevi na lousa do laboratório tal fórmula e expliquei a eles³⁵. Assim, após apresentar-lhes como seria o desenvolvimento da atividade que deveriam fazer, bem como comentar as fórmulas das áreas das figuras, eles partiram para a criação do programa. O Quadro 18, a seguir, apresenta o diálogo inicial entre mim e alguns estudantes.

Quadro 18 – Diálogo com Pedro, Rafael e Rhaylla sobre o início da segunda parte da Atividade 5 (parte 1)

Pesquisador: Agora, eu quero que vocês criem o seguinte programa: ao abrir o arquivo *Scratch*, aparecerá no palco as três figuras geométricas, cada uma com um número ou letra associado. Aí, o usuário vai escolher qual figura ele deseja calcular a área. Em seguida, o programa pergunta ao usuário sobre as medidas das figuras. Por exemplo, caso ele opte por calcular a área de um retângulo, o programa perguntará sobre as medidas dos lados dessa figura e fará o cálculo da área.

Rhaylla: Tá! [sic]. E por onde começa?

Pesquisador: Bem, são três figuras, certo? Então, a gente vai criar blocos para cada uma delas. Então, no bloco do retângulo, por exemplo, vai [sic] ter as variáveis largura e comprimento, como comentei antes, ou seja, quando o usuário executar o programa e optar por calcular a área de um retângulo, o programa irá perguntar sobre as medidas da largura e do comprimento.

³⁵ Mais adiante na análise, descreverei detalhadamente esse momento para o leitor.

Rafael: Eu vou começar a fazer os comandos da área, então.

Pedro: Mas vai ter mais coisa no bloco, né? [sic]. Tipo, no bloco do retângulo, vai ter o cálculo da área e mais programação?

Pesquisador: Isso, no bloco do retângulo, o programa de vocês precisa fazer duas perguntas, como eu mostrei lá no cálculo do IMC. Então, uma pergunta é para saber quanto mede a largura e a outra quanto mede o comprimento. Aí, vem a programação pra calcular a área, tudo isso no bloco do retângulo.

Rhaylla: Tá! [sic]. Então, eu crio um bloco quadrilátero e coloco tudo isso lá?

Pesquisador: É.

Rhaylla: Eu posso começar pelo cálculo da área, então?

Pesquisador: Vocês podem começar pelos blocos da área, mas depois... Seguinte, como eu não quero dar *spoiler*, comecem a fazer e qualquer dúvida me chama.

Fonte: Dados da pesquisa.

Naquele momento, não dei muitas explicações sobre o programa, afinal, eu gostaria de ver por onde os estudantes começariam suas programações. Ademais, Pedro e Rhaylla já estavam apresentando o conceito de **previsão**, como se verifica quando Pedro diz: “Mas vai ter mais coisa no bloco, né? [sic]. Tipo, no bloco do retângulo, vai ter o cálculo da área e mais programação?”, pois sabia que, além dos blocos que fariam o cálculo da área do retângulo, a programação exigiria mais coisas, e quando Rhaylla afirma: “Tá! [sic]. Então, eu crio um bloco quadrilátero e coloco *tudo isso lá?*”, mencionando que, ao criar o bloco que nomeou de “quadrilátero”, deveria colocar nele os blocos das variáveis, da área, das perguntas, etc. O Quadro 19 apresenta a sequência do nosso diálogo.

Quadro 19 – Diálogo com Pedro, Rafael e Rhaylla sobre o início da segunda parte da Atividade 5 (parte 2)

Rafael: Eu criei duas variáveis. E agora, como eu monto? [referindo-se a construir a programação do cálculo da área do retângulo].

Pedro: Tem que criar uma variável área, né [sic], pra [sic] fazer o cálculo?

Pesquisador: Sim, sim. Vejam que a fórmula é a seguinte: $\text{área} = \text{comprimento} \times \text{largura}$. Então, será atribuído um valor para a área. Por isso, temos que criar uma variável com esse nome. Além disso, essa vai ser a única variável que a gente vai usar nas três figuras, porque o restante [das variáveis] será de acordo com a nomenclatura da figura. Então, pro [sic] triângulo, será base e altura, mas, aí, na hora do cálculo da área, nós podemos usar essa variável aqui [apontando para a variável área escrita na lousa].

Pedro: Professor, vê se é assim [mostrando-me o bloco como o da figura a seguir]:



Pesquisador: A operação é essa, mas você tem que colocar esse valor na variável área.

Rhaylla: É só usar o “mude para” [fazendo alusão ao bloco “mude _ para _”].

Pesquisador: Isso, lá nas variáveis, tem um bloco que é “mude *variável* para *tal valor*”. É esse bloco que vocês vão usar para o cálculo da área.

Rhaylla: Professor, vem dar uma olhadinha no meu [nesse momento, a estudante mostrou o bloco como o da figura a seguir]:



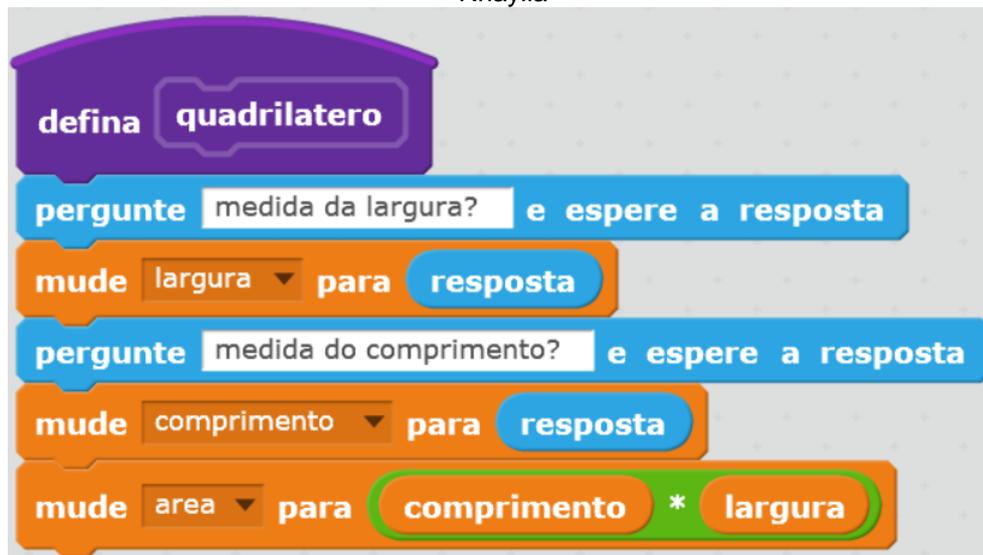
Pesquisador: Isso! Agora, precisa fazer os blocos das perguntas.

Rhaylla: Aí, cada valor que a pessoa colocar vai ser o valor do bloco? [referindo-se aos blocos “comprimento” e “largura”].

Fonte: Dados da pesquisa.

No momento desse diálogo, Rhaylla apresenta tanto o conceito de **explicação** quanto o de **previsão**. Primeiro, porque, ao explicar, logo após a fala de Pedro, que ele deveria colocar o bloco de operação no bloco da área, ela já teve a consciência da montagem correta do bloco da variável “área”. Em segundo lugar, porque a estudante estava prevendo a programação, isto é, ela já estava começando a compreender que as duas perguntas iniciais, referentes, respectivamente, às medidas do comprimento e da largura, quando respondidas corretamente, gerariam um valor atribuído às respectivas variáveis, e, então, o programa conseguiria fazer o cálculo da área do retângulo. Após alguns minutos, Rhaylla me chamou para apresentar a programação que já havia desenvolvido, dada a ver na Figura 39 a seguir.

Figura 39 – Programação inicial da segunda parte da Atividade 5, desenvolvida pela estudante Rhaylla



Fonte: Dados da pesquisa.

A estudante disse que criou o bloco “quadrilatero” porque o usuário poderia escolher calcular a área de um quadrado e não apenas de um retângulo. Como o foco da Atividade 5 era o cálculo da área e a programação para tal, considerei plausível não comentar com a estudante e com os demais colegas o conceito de que todo quadrado é um retângulo, pedindo apenas para que me explicasse como sua programação estava rodando. Nosso diálogo pode ser observado no Quadro 20 a seguir.

Quadro 20 – Diálogo com Rhaylla sobre sua programação inicial

Rhaylla: Professor, vem cá! Dá uma olhada [mostrando-me a programação apresentada na Figura 39].

Pesquisador: Beleza! Agora, clica duas vezes no bloco roxo pra [sic] gente ver.

[ao executar a programação, a estudante percebeu que o ator não falava a área no final].

Rhaylla: Então, não tá [sic] aparecendo a área.

Pesquisador: Você lembra que, para o boneco falar alguma coisa, tem que usar um bloco específico?

Rhaylla: Ah, é verdade! Peraí [sic], é esse aqui, né?[sic] [indicando o bloco “diga _ por _ segundos”]. Aí, eu coloco direto?

Pesquisador: O quê?

Rhaylla: A área aqui no bloco.

Pesquisador: Pode até colocar direto, só que, aí, o boneco vai falar só o valor.

Então, eu sugiro você usar esse aqui ó [sic] [indicando o bloco “junte _ com _”]. Aí,

na primeira parte, você pode colocar “a área é igual” e, no final, você põe a área [referindo-me ao bloco “área”].

Fonte: Dados da pesquisa.

Naquele momento, ao **testar** sua programação, a estudante apresentou um processo de **depuração**, pois teve a clareza de que ela não estava como pedido, visto que o ator precisaria falar a resposta do quanto vale a área. Além disso, pude perceber que a aluna tinha apenas se esquecido, posto que, quando eu disse que, no *Scratch*, havia um bloco específico para isso, ela rapidamente se lembrou e, em instantes, selecionou o bloco correto para essa função. A minha sugestão de usar o bloco “junte _ com _” foi com o intuito de o ator explicitar que, de fato, aquele valor se tratava da área da figura que o usuário escolheu calcular. A seguir, na Figura 40, pode-se observar a programação final da discente para calcular a área de um retângulo, que ela denominou procedimento “quadrilatero”. Nesse sentido, primeiro, a programação perguntava a medida da largura para atribuir esse valor à variável homônima; em seguida, perguntava a medida do comprimento para fazer o mesmo procedimento, mas, agora, na respectiva variável homônima. Na sequência, a programação fazia o cálculo da área e o ator dizia ao usuário quanto valia esse valor.

Figura 40 – Programação da estudante Rhaylla para calcular a área de um retângulo



Fonte: Dados da pesquisa.

Na sequência da construção do bloco que calculava a área de um retângulo, os estudantes partiram para a elaboração do que calculava a área de um triângulo. Por ser a última atividade, eles já sabiam algumas funcionalidades do *Scratch*, como duplicar os blocos, o que facilitou o processo inicial da programação. Dessa forma, copiaram aqueles que continham as perguntas do retângulo e passaram para o do triângulo, mudando apenas as variáveis das questões referentes à largura e ao comprimento para as relativas à base e à altura, o que claramente evidencia o conceito de **utilizar e remixar**.

Quando o estudante Rafael começou a desenvolver a programação do bloco triângulo, questionou-me acerca da fórmula da área, querendo saber se ela era válida para qualquer triângulo. Veja no Quadro 21 o diálogo desse momento.

Quadro 21 – Diálogo com Pedro, Rafael e Rhaylla acerca da fórmula da área de triângulos

Rafael: Mas, professor, esse cálculo dá pra [sic] qualquer triângulo?

Pesquisador: Você fala da base vezes altura? [dividido por dois].

Rafael: Isso, da área.

Pesquisador: Gente, o Rafael perguntou se essa fórmula para calcular a área do triângulo serve para qualquer triângulo, o que vocês acham?

[Nesse momento, alguns estudantes se olharam, mas nenhum respondeu convictamente, apesar de ter havido respostas “sim”].

Rhaylla: Pra [sic] alguns, com certeza dá, mas falar pra [sic] todos [os triângulos], eu já não sei.

Pesquisador: Bem, seguinte: quantas bases tem um triângulo? [nesse momento, desenhei vários triângulos na lousa e, novamente, os estudantes não afirmaram com convicção, porém houve as respostas “uma base” e “três bases”].

Pesquisador: Olhem aqui pra [sic] lousa. Estão vendo esses triângulos aqui? Eu conseguiria “apoiar” cada lado desses triângulos aqui embaixo? [apontando para a parte inferior da lousa]. Então, cada lado do triângulo é também considerado uma base do triângulo. E cada uma dessas bases, ou seja, [cada um] desses lados aqui, vai ter uma altura respectiva. Por isso, essa fórmula aqui [apontando para a fórmula escrita na lousa $A=(b.h)/2$] serve para qualquer triângulo. É que, em algumas vezes, nós não teremos explicitamente o valor da altura ou dos lados, por isso, existem outras maneiras de calcular a área de um triângulo, [como] usando a medida de um ângulo, por exemplo. Mas, aqui, como a gente vai ter a informação da medida da base e da altura respectiva a ela, fica válido para qualquer triângulo.

Pedro: Então, sempre que eu tiver a base, na verdade, sempre que eu souber o valor de um lado do triângulo e a altura, então, eu vou saber a área.

Rafael: Não pode esquecer que tem que ser a altura do lado que você sabe [a medida]. Então, na programação, é óbvio que a pessoa estaria falando da base e da altura “certa”, senão, não faria sentido.

Pesquisador: Exatamente! Estamos supondo que o usuário dará a informação de uma das bases e, conseqüentemente, da altura relacionada a ela.

Fonte: Dados da pesquisa.

Nesse contexto, é possível verificar que os estudantes **expressam** tanto algumas dúvidas, como, por exemplo, quando Rafael **questiona** sobre a validade da fórmula para qualquer triângulo, quanto algumas *tímidas* certezas, como no momento em que Rhaylla diz que a fórmula para calcular a área do triângulo é válida para alguns, mas não sabe se para todos. Significa dizer que o momento proporcionou, por meio da **explicação** e da **troca** de ideias entre eles, que os estudantes pudessem refletir sobre os conceitos de base e de altura de um triângulo, envolvidos na seguinte fórmula para calcular a área de um triângulo: $area = \frac{base \cdot altura}{2}$.

Quando os estudantes partiram para a construção do bloco do triângulo, houve uma diferença entre suas programações para o cálculo da área, visto que Rhaylla e Rafael fizeram de um modo enquanto Pedro fez de outro. O Quadro 22, a seguir, apresenta o diálogo desse momento.

Quadro 22 – Diálogo com Pedro, Rafael e Rhaylla sobre a construção do bloco da área do triângulo (parte 1)

Pesquisador: E aí, o que vocês fizeram?

Rhaylla: Então, professor, a gente não tá [sic] conseguindo fazer a área.

Rafael: É, na verdade, a gente fez, mas ficou diferente um do outro.

Pesquisador: Deixa eu [sic] ver o que vocês fizeram.

Pedro: É para montar assim: $área = base \cdot \left(\frac{altura}{2}\right)$? [indicando a fórmula escrita em seu caderno].

Rhaylla: Assim também daria certo: $área = \frac{base \cdot altura}{2}$? [também indicando a fórmula escrita em seu caderno].

Pesquisador: Deixa eu confirmar aqui, mas tô [sic] achando que esses dois jeitos dão certo.

Fonte: Dados da pesquisa.

Os estudantes estavam na dúvida sobre como seria o bloco da área do triângulo e, então, escreveram as fórmulas que julgavam corretas em seus cadernos

para poderem observar as diferenças entre elas. Naquele momento, questionaram-me acerca de como deveriam montá-las no *software*, isto é, se havia apenas um jeito ou se os modos que estavam considerando dariam certo. Dessa forma, fui ao meu computador para verificar se a montagem dos blocos, como na Figura 41, resultaria no mesmo valor e pude constatar que sim. Então, disse aos estudantes que ambas as construções do bloco da área estavam corretas. Ressalto que fiz a verificação, pois há casos, como o ilustrado na Figura 42, em que, dependendo da montagem, o resultado é diferente.

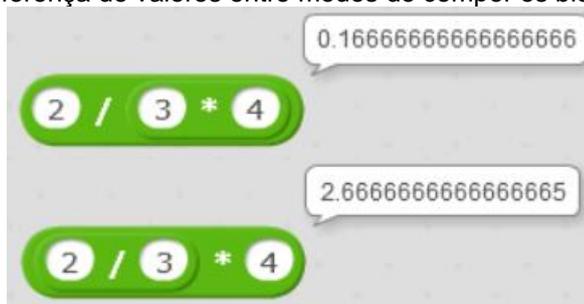
Figura 41 – Diferença de montagem dos blocos para calcular a área de um triângulo



Fonte: Dados da pesquisa.

Vale ressaltar que os blocos indicados com o número um na Figura 41 apontam para a fórmula escrita por Pedro, enquanto que os outros, evidenciados pelo número dois, denotam a fórmula escrita por Rhaylla e Rafael.

Figura 42 – Diferença de valores entre modos de compor os blocos operadores



Fonte: O Autor.

Como dito anteriormente, ao iniciarem a construção do procedimento para calcular a área de um triângulo, os estudantes duplicaram os blocos iniciais, cujas perguntas eram referentes às medidas do retângulo, e passaram-nas para o bloco do triângulo, efetuando alterações. Dessa maneira, a construção desse procedimento foi mais rápida e gerou menos dúvidas quando comparada à do anterior. A única observação válida é aquela relativa ao bloco da área do triângulo,

como supracitado. Assim, a seguir, pode-se observar, nas Figuras 43 e 44, respectivamente, as programações que os estudantes Pedro e Rhaylla realizaram.

Figura 43 – Procedimento “TRIÂNGULO”, desenvolvido pelo estudante Pedro



Fonte: Dados da pesquisa.

Figura 44 – Procedimento “triangulo”, desenvolvido pela estudante Rhaylla



Fonte: Dados da pesquisa.

Na continuidade da Atividade 5, os estudantes precisavam construir um bloco que calculasse a área de uma circunferência. Porém, como dito anteriormente, eles não sabiam a fórmula e, dessa maneira, para lhes dar maior suporte em sua elaboração, coloquei $área = \pi r^2$ na lousa. Li como a pronunciamos, a saber, “pi erre ao quadrado”, e, por fim, perguntei a eles quantas variáveis ela tinha. Naquele momento, pude ouvir “uma” e “duas”, porém, para mim, era inequívoco que quem disse duas estava se referindo a pi e a r e não a r^2 . Para deixar claro aos estudantes, expliquei-lhes que, como variável, tínhamos apenas a letra r, a qual simbolizava o raio da circunferência, e que consideraríamos pi (π) com um valor constante de 3,14. A partir disso, os discentes começaram a construção do bloco para calcular a área da circunferência, porém, ao longo do processo, surgiram algumas dúvidas, como pode ser observado no Quadro 23 a seguir.

Quadro 23 – Diálogo com Pedro, Rafael e Rhaylla sobre o início da construção do procedimento para calcular a área de uma circunferência

Rhaylla: Então, a gente só vai usar uma variável?

Pedro: Mas não tem a área também?

Rhaylla: Verdade! A gente vai usar as duas.

Rafael: Calma. Qual é a segunda variável?

Pedro: O raio.

Rafael: Ah, eu achei que era erre (r), mas, agora, eu entendi. Na fórmula, tá [sic] erre (r), mas a gente vai criar [a variável] com o nome de raio. E como a gente faz o raio ao quadrado?

Rhaylla: Professor, a gente só vai usar uma variável, certo? Então, vai ter apenas uma pergunta?

Pesquisador: Isso.

Rafael: Então, a pessoa só digita o raio e já calcula?

Pesquisador: É, como só tem uma variável, a pessoa coloca quanto mede esse raio e o programa já calcula, porque aquele pi é um valor fixo, então, é ele multiplicado pelo raio ao quadrado.

Rafael: Tá [sic], mas como faz o raio ao quadrado?

Fonte: Dados da pesquisa.

Os estudantes começaram a **prever** como seria a programação, primeiro, ao se **questionarem** sobre qual seria a quantidade de variáveis para realizar o cálculo da área, e, por conseguinte, quantas perguntas teriam no procedimento, e, segundo, ao **expressarem/explicarem** como fariam a construção desses blocos. Além disso,

essa **troca** de ideias fez com que Rafael esclarecesse sua dúvida acerca de qual seria a segunda variável a que Pedro e Rhaylla estavam se referindo.

Após o questionamento de Rafael, fui à lousa e comecei a dar exemplos do que significava um expoente associado ao número, de modo a tentar fazer com que eles se lembrassem de que tal conceito era referente à quantidade de vezes que um número se multiplicava. Como eles compreenderam a ideia, ou seja, como associaram que $r^2 = r.r$ a partir da fórmula $\text{área} = \pi r^2$, eu a reescrevi na lousa como $\text{área} = \pi.r.r = \pi.\text{raio.raio}$, com o intuito de que eles também a relacionassem ao bloco da variável que criariam, isto é, ao bloco da variável “raio”. Observe no Quadro 24 o diálogo desse momento.

Quadro 24 – Diálogo com Pedro, Rafael e Rhaylla sobre o bloco da área da circunferência

Rafael: Tá [sic], não vai precisar usar esse lance de expoente.

Rhaylla: É só raio vezes raio.

Pedro: Mas e o pi?

Pesquisador: Lembram que o pi a gente vai considerar um valor fixo? Então, lá no bloco da multiplicação, em um dos espaços, vocês vão colocar 3,14.

Rhaylla: Então, vai ser 3,14, depois, raio vezes raio?

Pesquisador: Isso! Vai ser 3,14 vezes o raio, vezes o raio de novo.

Rafael: São três espaços, então?

Pedro: Vai ter que colocar um dentro do outro [referindo-se a juntar dois blocos de multiplicação].

Fonte: Dados da pesquisa.

Naquele momento, os estudantes já demonstravam compreender como seria a construção dos blocos para calcular a área de uma circunferência, ou seja, já **previam** o funcionamento da programação. Quando Rafael disse que seriam três espaços e Pedro afirmou que deveriam colocar um bloco dentro do outro, eles conseguiram construir corretamente o procedimento para calcular tal área. A Figura 45, a seguir, exhibe a programação desenvolvida por Rhaylla para calcular a área de uma circunferência³⁶. Nela, era perguntado ao usuário qual a medida do raio da circunferência de que se desejava calcular a área, e, então, esse valor era atribuído ao raio para que fosse feito tal cálculo. No fim, o ator lhe dizia quanto valia a área.

³⁶ Como as programações de Pedro e de Rafael ficaram semelhantes, foram suprimidas.

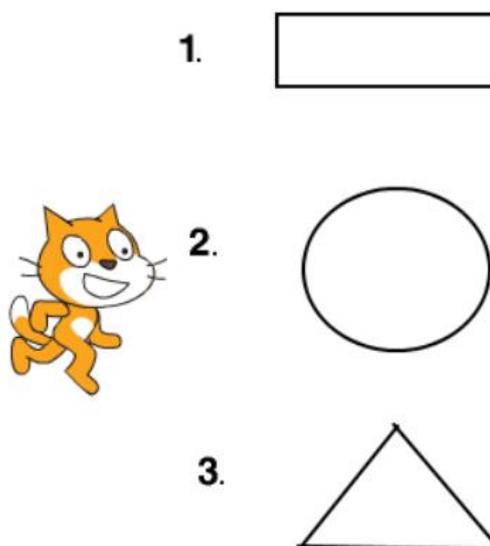
Figura 45 – Procedimento “circunferencia”, realizado pela estudante Rhaylla



Fonte: Dados da pesquisa.

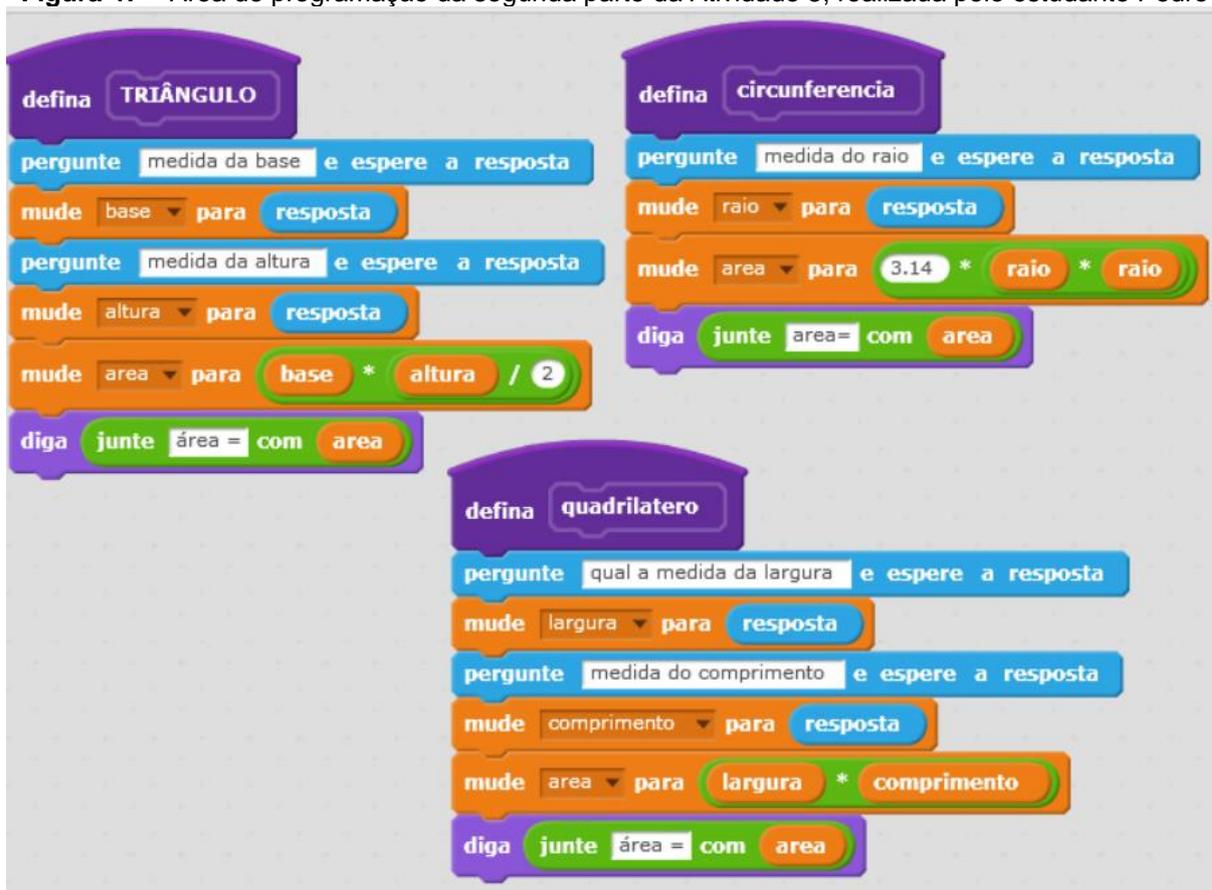
Após terem feito os três procedimentos para o cálculo das áreas, alguns alunos acharam que a programação estava finalizada. Nas Figuras 46, 47, 48 e 49, a seguir, apresento, respectivamente, o palco e a programação desenvolvidos pelos estudantes Pedro e Rhaylla até aquele momento da atividade.

Figura 46 – Palco da segunda parte da Atividade 5, realizado pelo estudante Pedro



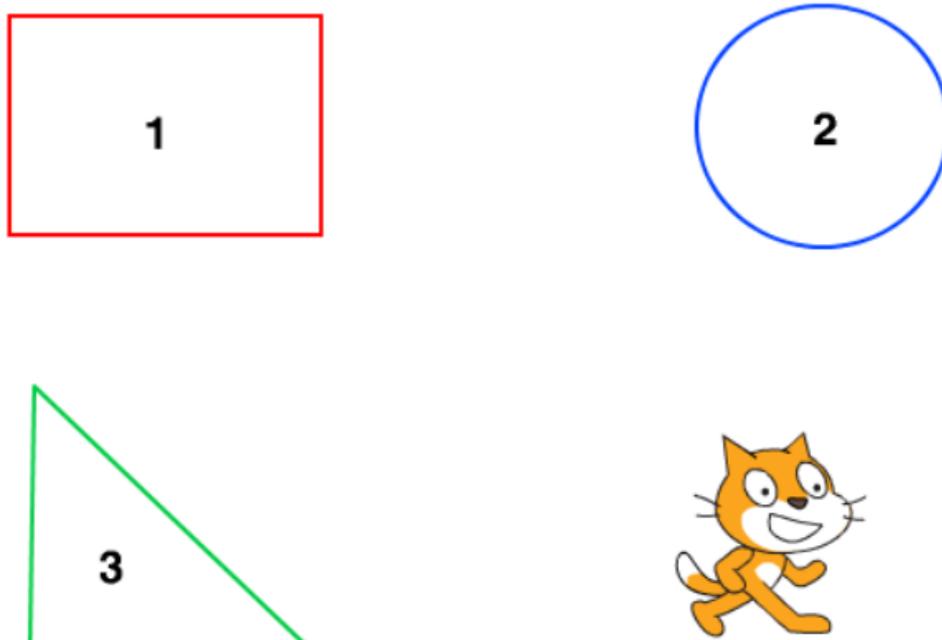
Fonte: Dados da pesquisa.

Figura 47 – Área de programação da segunda parte da Atividade 5, realizada pelo estudante Pedro



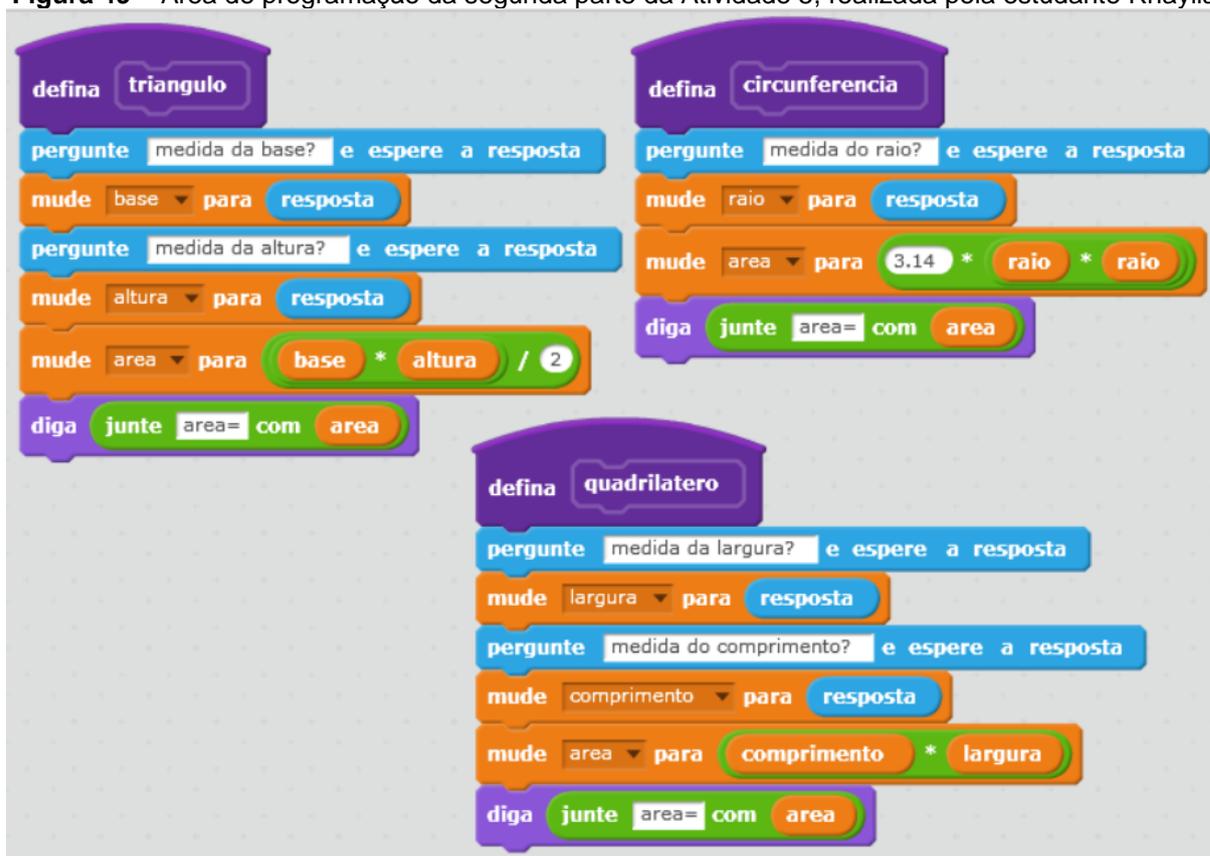
Fonte: Dados da pesquisa.

Figura 48 – Palco da segunda parte da Atividade 5, realizado pela estudante Rhaylla



Fonte: Dados da pesquisa.

Figura 49 – Área de programação da segunda parte da Atividade 5, realizada pela estudante Rhaylla



Fonte: Dados da pesquisa.

Apesar de os estudantes terem desenvolvido grande parte da atividade, ou seja, apesar de terem construído adequadamente o palco, assim como os blocos para calcular as áreas das figuras planas, eles esqueceram que faltava o que eu havia explicado no início da segunda parte da Atividade 5: a decisão de qual figura o usuário escolheria para calcular a área. Para determiná-la, empregariamos o bloco “se/senão”. Observe no Quadro 25, a seguir, o diálogo entre mim e os estudantes sobre essa questão.

Quadro 25 – Diálogo com Pedro, Rafael e Rhaylla sobre o desenvolvimento final da Atividade 5

Rhaylla: Professor, é só isso?

Pesquisador: O quê?

Rhaylla: A atividade. Tipo, a gente acabou?

Pesquisador: Ainda não! Vocês se lembram como o programa tem que funcionar? Quando a pessoa for abrir o programa, vocês precisam imaginar que ela não verá a parte dos blocos, apenas o palco que vocês criaram. Então, assim que o programa começa a rodar, o ator tem que perguntar para a pessoa qual figura ela deseja calcular a área e, aí, a pessoa digita o número respectivo à figura e é nesse momento que o ator executa os blocos que vocês criaram, entenderam?

Pedro: Ah, nosso programa não tá [sic] fazendo isso ainda, né [sic]?

Rafael: A gente só fez, então, os blocos da área.

Pesquisador: Vocês precisam lembrar que tem uma condição para o ator executar ou o bloco para calcular a área do retângulo ou do triângulo ou da circunferência. [nesse momento, mostrei a eles novamente a programação que eu já havia desenvolvido, como nas Figuras 34 a 38].

Rhaylla: Vai depender do número que ele [o usuário] digitar.

Pesquisador: Essa “dependência” é uma condição. Vocês se lembram de algo relacionado a isso?

Pedro: Já sei! É o bloco que você passou antes [referindo-se ao bloco “se/senão”], porque a gente não usou ainda.

Pesquisador: Exatamente! Agora, vocês têm que usar o bloco “se/senão”, porque a condição que tem que acontecer para o programa calcular a área de um retângulo é a pessoa digitar o número um, para o triângulo, o dois, e, para a circunferência, o três.

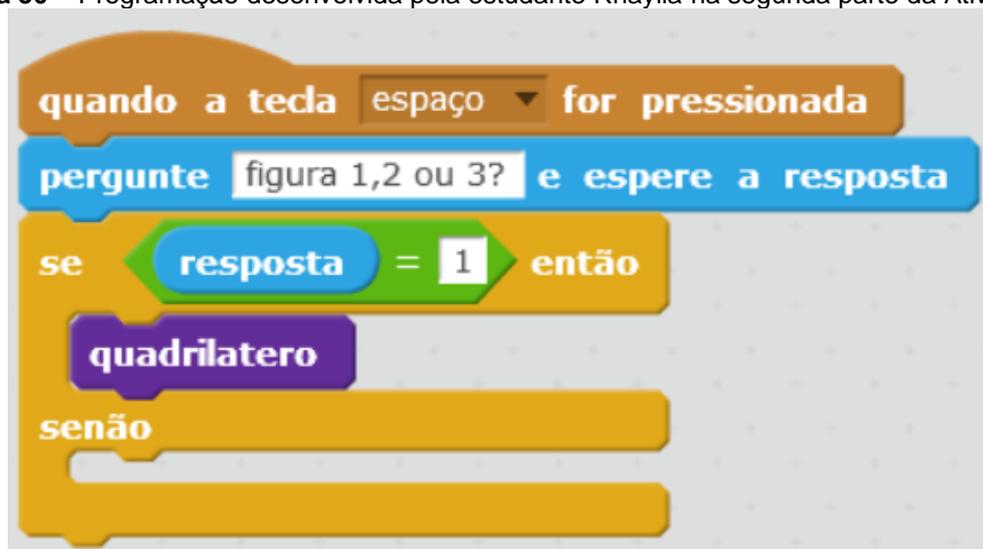
Rhaylla: Então, assim que começar [a programação], já tem que perguntar o que a pessoa quer calcular.

Pesquisador: Isso! E, assim, vai seguindo pro [sic] bloco respectivo ao número que ela [a pessoa/o usuário] escolheu.

Fonte: Dados da pesquisa.

O diálogo do Quadro 25 mostra os estudantes se **expressando/explicando** como deveria ser a programação final da Atividade 5, ou seja, tentando **prever** como seria a construção dos blocos. Além disso, algumas falas, como a de Pedro: “Ah, nosso programa não tá [sic] fazendo isso ainda, né [sic]?” ressaltam o conceito de **depuração**, pois o aluno compreende que o programa ainda não estava completo, já que não executava exatamente o que havia sido pedido. As falas de Rhaylla, como em: “Vai depender do número que ele [o usuário] digitar” e “Então, assim que começar [a programação], já tem que perguntar o que a pessoa quer calcular”, também evidenciam os conceitos de **depuração** e de **previsão**. Após nosso diálogo, percebi que apenas Rhaylla tinha compreendido, ainda que parcialmente, como deveria ser a construção final da Atividade 5. No entanto, ela não conseguiu montar inteiramente o bloco “se/senão”, a fim de executar integralmente a tarefa como solicitado. A seguir, na Figura 50, pode-se observar até que ponto a estudante atingiu o desenvolvimento da programação por conta própria.

Figura 50 – Programação desenvolvida pela estudante Rhaylla na segunda parte da Atividade 5



Fonte: Dados da pesquisa.

Como Rhaylla e os demais estudantes não estavam conseguindo finalizar a construção do bloco “se/senão”, o avanço em suas programações só foi possível mediante o meu auxílio, de modo que, a partir daquele momento, eles não tiveram mais autonomia para desenvolver a atividade, uma vez que tive que lhes mostrar como deveria ser, de fato, a programação. Em outras palavras, projetei o que eu havia desenvolvido na TV do laboratório para eles conseguirem concluir a Atividade 5. A Figura 51, a seguir, exhibe como deveria ser a elaboração do bloco “se/senão” para executar corretamente o que foi solicitado na tarefa.

Figura 51 – Programação desenvolvida pelo pesquisador para exemplificar como deveria ser a construção do bloco “se/senão” no desenvolvimento da segunda parte da Atividade 5



Fonte: O Autor.

Alguns conceitos ficaram evidentes durante o desenvolvimento da segunda parte da Atividade 5, já que os estudantes estavam realizando-a com o *software Scratch*. São eles: o uso de **seqüências**, **paralelismo**, **eventos**, **condicionais**, **operadores** e **dados**, afinal, essa era a última tarefa da produção de dados e teve o intuito de ser mais complexa. Ademais, conforme foram caminhando na elaboração dessa segunda parte da Atividade 5, tiveram que **analisar os dados**, **decompor o problema** para reduzir sua complexidade (**abstração**), **reutilizar** e alterar os blocos para ficarem como desejado. Dessa maneira, puderam **explorar** o *software*, investigar ideias (matemáticas), lidar com ambiguidades, **explicar** seus pensamentos, tornando o desenvolvimento da atividade um processo reflexivo, o que lhes proporcionou também a possibilidade de tentar **prever** os resultados do que estavam realizando. Além disso, **questionaram-se** acerca do que foi solicitado pela tarefa, tendo, a partir disso, momentos de **trocas** de ideias com prósperas

interações entre eles. Por fim, a atividade ressalta a **conexão** entre os conceitos relacionados ao pensamento computacional, à Matemática e à linguagem de programação *Scratch*.

Para finalizar a seção de análise, apresento os conceitos do pensamento computacional que emergiram dos estudantes do nono ano ao realizarem atividades com conteúdo matemático no Scratch:

- a) ISTE e CSTA (2011) e Barr e Stephenson (2011): coleção de dados, análise de dados, representação de dados, decomposição do problema, abstração, algoritmos e procedimentos e paralelização;
- b) Brennan e Resnick (2012):
 - Conceitos computacionais: sequências, *loops*, paralelismo, eventos, condicionais, operadores e dados;
 - Práticas computacionais: ser incremental e iterativo, testar e depurar, reutilizar e remixar e abstrair e modularizar;
 - Perspectivas computacionais: expressar, conectar e questionar;
- c) Benton et al. (2016, 2017): explorar, explicar, prever, trocar e conectar.

Apenas os conceitos de automação e simulação (ISTE; CSTA, 2011, BARR; STEPHENSON, 2011) não se manifestaram nas atividades analisadas. Além disso, vale ressaltar que os conceitos acima não apareceram integralmente em todas as atividades e dificilmente aparecerão, não havendo nenhum problema nisso, pois é um fator que dependerá do objetivo de cada atividade a ser realizada. Assim sendo, a próxima seção expõe as reflexões do pesquisador e os apontamentos para uma possível continuidade da pesquisa e/ou futuras pesquisas.

6 REFLEXÕES E APONTAMENTOS

A primeira reflexão que trago ao leitor é a de como foi se constituindo meu entendimento acerca do pensamento computacional. O que eu concordo sobre as ideias referentes ao PC é que o tema trata de resoluções de problemas, podendo ser, desde o planejamento de uma viagem, até o desenvolvimento de um programa de computador, por exemplo. Então, o pensamento computacional envolve processos mentais para a solução de questões, fundamentando-se em alguns conceitos, tais como o reconhecimento de padrões, a decomposição do problema em partes menores e mais “fáceis” de resolver, o raciocínio abstrato, etc. Porém, ele também abarca a interação tanto entre pessoas quanto entre pessoas e dispositivos tecnológicos, de modo que, à medida que vamos resolvendo o problema, ou sozinho ou com o outro, ou, ainda, com uma máquina, novas ideias vão surgindo nesse processo heurístico.

Essa noção sobre o tema foi trilhada desde o início da pesquisa até o presente momento e, como dito no início da dissertação, eu não tinha nenhum conhecimento sobre o assunto antes de entrar na pós-graduação. A ordem que eu li os referenciais teóricos é a mesma apresentada no texto, ou seja, iniciei minhas leituras com Pierre Lévy (1997), Tikhomirov (1981), Borba e Villareal (2005), para, posteriormente, travar contato com o pensamento computacional e, por isso, a palavra *pensamento*, para mim, já tinha um sentido atribuído por essas leituras.

No começo, eu procurei saber o significado de pensamento, bem como de computação/computacional, pois entendia o termo como um processo intrínseco ao ser humano, isto é, como uma atividade *individual* da mente humana, de modo que cheguei a considerar *raciocínio* computacional ao invés de *pensamento* computacional. Assim, a princípio, eu estava preocupado em buscar uma definição para o PC, fato que vinha me intrigando, já que esse conceito está em desenvolvimento e existem várias abordagens. No entanto, com as reflexões que estava realizando na ocasião da escrita do material para o exame de qualificação, bem como com as discussões que ocorreram naquele momento, identifiquei os preceitos ligados ao tema.

A partir disso, busquei relacionar o desenvolvimento do pensamento computacional à Matemática e ao *software Scratch*. Quero ressaltar ao leitor que a pesquisa de Brennan e Resnick (2012) e as pesquisas do projeto *ScratchMaths*

(BENTON et al., 2016, 2017, 2018a, 2018b) foram de extrema importância para a minha reflexão e compreensão do assunto, pois fizeram com que a ideia que eu tinha de que o PC era algo exclusivamente intrínseco ao ser humano fosse reconsiderada.

Trago essa primeira reflexão ao leitor, pois acredito que, apesar de a pergunta norteadora da pesquisa ser: “*Quais conceitos do pensamento computacional emergem de estudantes do nono ano ao realizarem atividades com conteúdo matemático no Scratch?*”, o que moveu minhas ações e inquietações durante o processo foi investigar as relações entre o pensamento computacional, a Matemática e o *software Scratch*. E foi mediante a investigação de tais relações que consegui elaborar a seção quatro dessa dissertação, em que trago o referencial teórico, mostrando quais são os conceitos envolvidos nessas conexões. A partir deles, faço a análise das atividades desenvolvidas pelos estudantes na produção de dados da pesquisa, as quais servem também para mostrar essas ligações. Portanto, a pergunta norteadora da pesquisa só é respondida, ou seja, os conceitos que emergem dos estudantes só podem ser evidenciados porque realizei todo esse processo descrito anteriormente.

Um aspecto importante, mas não visto na literatura é o processo de *composição*. Apesar dos referenciais trazerem o conceito de decomposição do problema, não faz sentido ela existir sem que haja uma série de ações para a sua composição, afinal estamos tratando de um processo de resolução de problemas, dessa maneira tão importante quanto decompor o problema é o método para compor essas partes do problema para resolvê-lo. Ainda em se tratando dos conceitos referentes ao PC, em minha concepção, os mais relevantes são aqueles relativos às perspectivas computacionais de Brennan e Resnick (2012), pois dizem respeito aos estudantes descreverem suas compreensões de desenvolvimento próprio, sua relação com os outros jovens e com o mundo tecnológico que os permeia, juntando o fato de que eles se expressam e se conectam por meio das tecnologias digitais. Os outros conceitos mais relevantes são os determinados pela abordagem pedagógica trazida pelo projeto de pesquisa *ScratchMaths* (BENTON et al., 2017), denominados 5 E's, que correspondem a *Explore, Explain, Envisage, Exchange* e *bridgE*, sendo, respectivamente, compreendidos como explorar, explicar, prever, trocar e conectar. Desses, o conceito de explicar pode ser análogo à perspectiva computacional de expressar, e, trocar, análogo à perspectiva de conectar. Esse

último “E”, de *bridgE*, traz uma metáfora de conexão entre as disciplinas, a qual está consolidada no conceito de *powerful ideas* (PAPERT, 1980), ou seja, no entendimento de que a força dessas ideias está nas conexões com outras disciplinas e é devida à linguagem com a qual são expressas. Assim, no caso das atividades retratadas nessa dissertação, a disciplina é a Matemática, a linguagem é a do *Scratch* e a conexão é estabelecida por meio da programação.

Essa questão acerca da linguagem começou a me causar interesse quando li *Mindstorms* (1980) e, por essa razão, a seguir, no Quadro 26, trarei alguns recortes do livro para fazer um outro movimento de reflexão.

Quadro 26 – Trechos do livro *Mindstorms* (1980)

Programar um computador significa nada mais nada menos do que comunicá-lo em uma linguagem que ele e o usuário humano possam “entender”. E aprender idiomas é uma das coisas que as crianças fazem melhor. Toda criança normal aprende a falar. Por que então uma criança não deveria aprender a “conversar” com um computador? [...] Duas ideias fundamentais percorrem este livro. A primeira é que é possível projetar computadores para que aprender a se comunicar com eles possa ser um processo natural [...]. A segunda, aprender a se comunicar com um computador pode mudar a maneira como ocorre outro aprendizado. O computador pode ser uma entidade de língua matemática e de língua alfabética [...] (PAPERT, 1980, p. 5).

E, ao ensinar o computador a pensar, as crianças embarcam em uma exploração sobre como elas mesmas pensam (PAPERT, 1980, p. 19).

Mesmo o trabalho mais simples da Tartaruga pode abrir novas oportunidades para aprimorar o pensamento sobre o pensamento: A programação da tartaruga começa fazendo-a refletir sobre como se faz o que gostaria que a tartaruga fizesse. Assim, ensinar a Tartaruga a agir ou a “pensar” pode levar a pessoa a refletir sobre suas próprias ações e pensamentos. E, à medida que avançam, as crianças programam o computador para tomar decisões mais complexas e se envolvem em refletir sobre aspectos mais complexos de seu próprio pensamento (PAPERT, 1980, p. 28).

Fonte: Adaptado de Papert (1980).

Há outros trechos em que o autor traz esse aspecto da linguagem, mas os do Quadro 26 já são suficientes para eu apresentar minha outra reflexão. O que me

instigou foi o processo de “tradução” da linguagem que falamos para a “linguagem que o computador fala”. Assim, tive os seguintes questionamentos: será que quem já fala inglês tem mais facilidade com programação, já que as linguagens de programação são em inglês?; quais são os aspectos envolvidos nesse processo de tradução da linguagem falada para a linguagem de programação? Portanto, era essa “tradução” que eu precisava compreender para saber sobre o quê, o como e o quando eu queria que a máquina executasse uma tarefa, ou seja, para entender quais eram os processos que deviam acontecer dentro da máquina para que houvesse a execução esperada. Necessitava entender igualmente quais eram os processos que precisavam se dar na minha mente, a fim de que eu pudesse refletir sobre os processos que tinham que ocorrer na máquina para ela executar exatamente aquilo que eu queria. Se partirmos da ideia que a linguagem determina o pensamento, então os processos mentais que os estudantes desenvolvem ao se expressarem e traduzirem sua linguagem para a do computador, realizando alguns dos conceitos computacionais, pode determinar o que se chama de desenvolver o pensamento computacional.

Nesse sentido, elaborar atividades com conteúdo matemático no *Scratch* mostrou ser uma possibilidade para que estudantes expressem raciocínio lógico-matemático e desenvolvam alguns dos conceitos que constituem o PC. Além disso, uma outra possibilidade de investigação seria fazer uma pesquisa com estudantes de outros anos de ensino ou mostrar a relação dos temas com distintas áreas de ensino, ou, ainda, buscar diferentes relações, como PC e Modelagem Matemática, PC e formação de professores, PC e História da Educação Matemática, etc. Por fim, este estudo traz uma possibilidade de se trabalhar com pensamento computacional, Matemática e *Scratch*, utilizando o computador, além de contribuir com discussões sobre a relação entre esses três temas.

REFERÊNCIAS

BARBOSA, L. M. **Aspectos do Pensamento Computacional na Construção de Fractais com o software GeoGebra**. 2019. 168f. Dissertação (Mestrado em Educação Matemática) – Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista (Unesp), Rio Claro, 2019.

BARR, V.; STEPHENSON, C. Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? **Acm Inroads**, New York, v. 2, n. 1, p. 48-54, 2011.

BENTON, L. et al. Building mathematical knowledge with programming: insights from the ScratchMaths project. In: CONFERENCE CONSTRUCTIONISM IN ACTION, s/n., 2016, Bangkok. **Proceedings...** Bangkok: Suksapattana Foundation, 2016. p. 26-33.

BENTON, L. et al. Bridging Primary Programming and Mathematics: Some Findings of Design Research in England. **Digit. Exp. Math Educ**, Inglaterra, n. 3, 115-138, 2017.

BENTON, L. et al., Designing for learning mathematics through programming: A case study of pupils engaging with place value. **International Journal of Child-Computer Interaction**, Londres, v. 16, p. 68-76, 2018a.

BENTON, L. et al. Beyond Jam Sandwiches and Cups of Tea: An Exploration of Primary Pupils' Algorithm-Evaluation Strategies. **Journal of Computer Assisted Learning**, Londres, v. 34, n. 5, p. 590-601, 2018b.

BESSA, K. F. **Pensamento Computacional e Matemática: uma abordagem com o Scratch**. 2020. 150f. Dissertação (Mestrado em Educação Matemática) – Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista (Unesp), Rio Claro, 2020.

BOGDAN, R. C.; BIKLEN, S. K. **Investigação qualitativa em Educação: uma introdução à teoria e aos métodos**. Porto: Porto Editora, 1994. 335 p.

BORBA, M. C.; VILLARREAL, M. E. **Humans-with-media and the reorganization of mathematical thinking: information and communication technologies, modeling, visualization and experimentation**. Boston: Springer, 2005. 232 p.

BRASIL. **Lei n. 9.394**, de 20 de dezembro de 1996. Estabelece as diretrizes e bases da educação nacional. Disponível em: <https://www2.camara.leg.br/legin/fed/lei/1996/lei-9394-20-dezembro-1996-362578-publicacaooriginal-1-pl.html>. Acesso em: 15 fev. 2020.

BRENNAN, K. Beyond Technocentrism: Supporting Constructionism in the Classroom. **Constructivist Foundations**, Bruxelas, v. 10, n. 3, p. 289-296, 2015.

BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. In: AMERICAN EDUCATIONAL RESEARCH ASSOCIATION MEETING, s/n., 2012, Vancouver. **Proceedings...** Vancouver, 2012. p. 1-25. Disponível em: https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf. Acesso em: jul. 2020.

CARVALHO, F. J. R. **Introdução à programação de computadores por meio de uma tarefa de Modelagem Matemática na Educação Matemática**. 2018. 133f. Dissertação (Mestrado em Ensino) – Centro de Educação, Letras e Saúde, Universidade Estadual do Oeste do Paraná, Foz do Iguaçu, 2018.

CORRÊA, B. S. **Programando com Scratch no Ensino Fundamental: uma possibilidade para a construção de conceitos matemáticos**. 2021. 174f. Dissertação (Mestrado em Ensino de Matemática) – Instituto de Matemática e Estatística, Universidade Federal do Rio Grande do Sul, Rio Grande do Sul, 2021.

CRESWELL, J. W. **Investigação qualitativa e projeto de pesquisa: escolhendo entre cinco abordagens**. 3. ed. Porto Alegre: Penso, 2014. 341 p.

HAREL, I.; PAPERT, S. Software design as a learning environment. **Interact Learn Environments**, Reino Unido, v. 1, n. 1, p. 1-32, 1990.

HOYLES, C. What is the point of group discussion in mathematics? **Educational studies in mathematics**, v. 16, n. 2, p. 205–214, 1985.

International Society for Technology in Education (ISTE); Computer Science Teachers Association (CSTA). **Computational Thinking Teachers Resource**. 2 ed. 2011.

JENKINS, H. et al. **Confronting the challenges of participatory culture: Media education for the 21st Century**. 2006. Disponível em:

https://www.macfound.org/media/article_pdfs/jenkins_white_paper.pdf. Acesso em: jul. 2020.

LÉVY, P. **As tecnologias da inteligência**. Rio de Janeiro: Editora 34, 1997. 203 p.

LUMMERTZ, R. S. **As potencialidades do uso do software scratch para a construção da literacia digital**. 2016. 131f. Dissertação (Mestrado em Ensino de Ciências e Matemática) – Programa de Pós-Graduação em Ensino de Ciências e Matemática, Universidade Luterana do Brasil, Canoas, 2016.

MARJI, M. **Aprende a programar com Scratch**: uma introdução visual à programação com jogos, arte, ciência e matemática. São Paulo: Novatec, 2014. 284 p.

MEYER, J. F. C. A.; CALDEIRA, A. D.; MALHEIROS, A. P. S. **Modelagem em Educação Matemática**. 3. ed. Belo Horizonte: Autêntica Editora, 2013. (Coleção Tendências em Educação Matemática).

PAPERT, S. **Mindstorms**: children, computers and powerful ideas. 1. ed. Nova Iorque: Basic Books, 1980. 230 p.

PAPERT, S. **Logo**: computadores e educação. São Paulo: Editora Brasiliense, 1985. 253 p.

PAPERT, S. **The Children's Machine**: Rethinking School in the Age of the Computer. Londres: Pearson Education Limited, 1993. 256 p.

PAPERT, S. What's the big idea? Toward a pedagogy of idea power. **IBM Syst. J.**, v. 39, n. 3.4, p. 720-729, 2000.

PIAGET, J. **Fazer e Compreender**. São Paulo: Melhoramentos, 1978. 186 p.

ROMANELLO, L. A. **Potencialidades do uso do celular na sala de aula**: atividades investigativas para o ensino de função. 2016. 135f. Dissertação (Mestrado em Educação Matemática) – Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista “Júlio de Mesquita Filho”, Rio Claro, 2016.

ROSA, M. A. **Construção de identidades on-line por meio do Role Playing Game**: relações com ensino e aprendizagem matemática em um curso a distância.

2008. 263f. Tese (Doutorado em Educação Matemática) – Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista, Rio Claro, 2008.

SÃO PAULO (Estado). Secretaria da Educação. **Diretrizes do Programa Ensino Integral**. Programa Ensino Integral. São Paulo, 2014a. 56 p.

SÃO PAULO (Estado). Secretaria da Educação. **Informações básicas. Programa Ensino Integral**. São Paulo, 2014b. 8 p.

SILVA, E. C. **Pensamento Computacional e a formação de conceitos matemáticos nos Anos Finais do Ensino Fundamental**: uma possibilidade com kits de robótica. 2018. 264f. Dissertação (Mestrado em Educação Matemática) – Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista (Unesp), Rio Claro, 2018.

TIKHOMIROV, O. K. The psychological consequences of computerization. In: WERTSCH, J. V. **The concept of activity in Soviet Psychology**. New York: M. E. Sharpe, 1981. p. 256-278.

WING, J. M. Computational thinking. **Communications of the ACM**, New York, v. 49, n. 3, p. 33-35, mar. 2006.

WING, J. M. Computational thinking and thinking about computing. **Phil. Trans. R. Soc. A**, Londres, v. 366, n. 1881, 3717–3725, 2008.

APÊNDICES

Apêndice A - O software Scratch

“O Scratch é uma linguagem de programação *visual*. Ele foi desenvolvido no Massachusetts Institute of Technology (MIT) Media Lab para tornar o aprendizado de programação mais fácil e divertido” (MARJI, 2014, p. 22, grifo do autor), visto que substitui a programação baseada na digitação de códigos pelo uso de blocos visuais, os quais podem ser escolhidos, arrastados e encaixados em outros para a formação de instruções dadas ao computador. Na Figura 1, pode-se observar a interface do *software*.

Figura 1 – Interface do *software* Scratch



Fonte: Adaptado de Marji (2014).

A interface apresenta quatro campos principais. São eles: o palco, a área de programação, a paleta de blocos e a lista de atores. O **palco** ou *stage* (primeiro elemento à esquerda) é onde o ator escolhido poderá se movimentar, atendendo à programação, que, por sua vez, acontece na “**área de programação**” ou área de *scripts*, localizada à direita. A região central é responsável pelos **blocos** do *software*, divididos em dez categorias (ou paletas): Movimento, Aparência, Som, Caneta, Variáveis, Eventos, Controle, Sensores, Operadores e Mais Blocos. Além disso, é

nessa área central que se pode alterar a fantasia dos atores ou adicionar novos sons à programação.

Na parte inferior ao palco, tem-se a “**Lista de atores**”, em que são mostrados os nomes e as miniaturas de todos os atores do projeto. Nela, existem ícones que permitem criar novos atores tanto a partir da biblioteca do *Scratch*, quanto do *Paint Editor*, de uma câmera conectada ao computador³⁷ ou do próprio computador. Além disso, cada ator tem suas próprias programações, fantasias e sons.

O Palco possui 480 passos de largura e 360 de altura. Ao passar o ponteiro do mouse sobre ele, em seu canto inferior direito, são exibidas as coordenadas x e y do ponteiro, cujo valor de x varia de -240 a 240 e o de y, de -180 a 180. Na área de programação, há um ícone referente ao ator selecionado, mostrando sua posição atual em relação a essas coordenadas.

Para que cada ator execute uma tarefa, é necessário que o usuário elabore uma programação, arrastando os blocos necessários para a área de programação, segundo a sua lógica. Segundo Marji (2014, p. 27):

Os blocos do Scratch somente se encaixam de determinadas maneiras, eliminando os erros de digitação que tendem a ocorrer quando as pessoas usam linguagens de programação baseadas em texto. Não é preciso completar os scripts para executá-los, o que significa que você pode testá-los à medida que os criar.

Dessa maneira, pode-se desmontar uma pilha de blocos e testar cada um deles separadamente, efetuando, assim, várias simulações. Esse recurso é muito útil quando se tem programações longas, além de permitir criar um projeto por partes.

Além de serem divididos em categorias (paletas), os blocos são separados por tipos, tendo nomes e usos específicos. Portanto, a seguir, será feita uma seção para descrevê-los, com o intuito de, posteriormente, mostrar suas funcionalidades.

1 BLOCOS DO SCRATCH

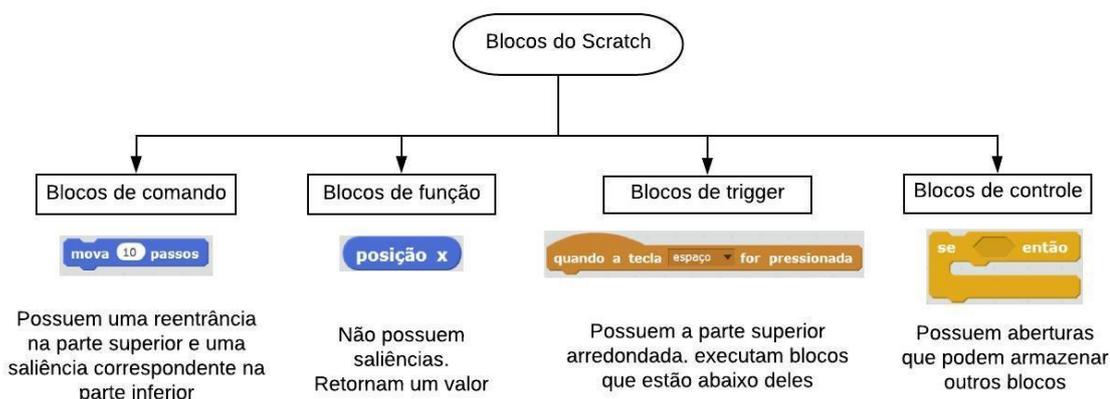
Segundo Marji (2014), o *Scratch* possui quatro tipos de blocos: os blocos de comando, os de função, os de trigger e os de controle, assim definidos:

³⁷ Uma *webcam*, por exemplo, com a qual se pode tirar uma foto do usuário para que ele vire um ator.

Os blocos de comando e os blocos de controle [...] tem saliências na parte inferior e/ou nas reentrâncias na parte superior. Esses blocos podem ser unidos na forma de pilhas. Os blocos trigger, também chamados de hats (chapéus), têm a parte superior arredondada porque são colocados no topo de uma pilha. Os blocos de trigger conectam eventos a scripts. [...] Os blocos de função não têm saliências nem reentrâncias. Eles não podem constituir uma camada de um script sozinhos; em vez disso, são usados como entradas para outros blocos (MARJI, 2014, p. 41, grifos do autor).

Na Figura 2, pode-se observar os tipos de blocos do *Scratch* e suas características.

Figura 2 – Os quatro tipos de blocos disponíveis no *Scratch*



Fonte: Marji (2014).

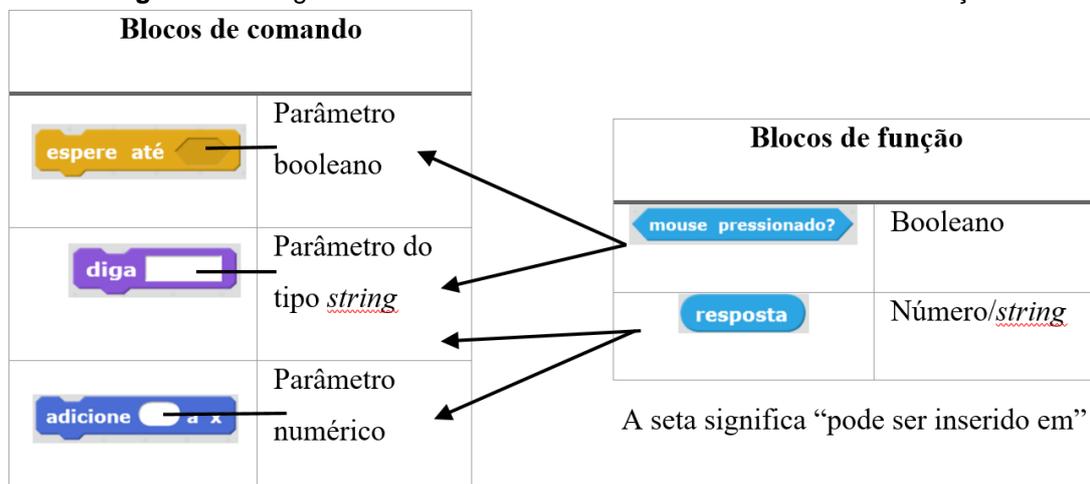
Além dos tipos de blocos, há também os tipos de dados que o *Scratch* processa. Isso porque os roteiros de programação “processam e manipulam diferentes tipos de dados durante a sua execução. Esses [...] podem ser de entrada para blocos de comando [...] ou de saída para blocos de função [...] ou os dados podem ser inseridos pelo usuário” (MARJI, 2014, p. 118), e, nesse sentido, os dados suportados pelos blocos do *Scratch* são booleanos, números e *strings*. Em relação a isso, Marji (2014, p. 119, grifos do autor) dá a seguinte definição:

Um *booleano* (Boolean) pode ter apenas um de dois valores: verdadeiro ou falso. Este tipo de dado pode ser usado para testar uma ou mais condições e, de acordo com o resultado, fazer o seu programa selecionar um caminho diferente de execução. [...] Uma variável *numérica* pode armazenar tanto valores inteiros quanto decimais. [...] ambos são classificados como “números”. [...] Uma *string* corresponde a uma sequência de caracteres, que pode incluir letras (tanto maiúsculas quanto minúsculas), números (de 0 a 9), além de outros símbolos que podem ser digitados a partir do teclado (+, -, &, @ e assim por diante).

Os parâmetros dos blocos de comando do *Scratch* apresentam espaços reservados para eles, os quais possuem formatos específicos (hexágono, retângulo

e retângulo arredondado), relacionados aos tipos de dados que aceitam. Já o formato dos blocos de função indica o tipo de dado que retornam. Na Figura 3, pode-se observar um resumo dessa relação.

Figura 3 – O significado dos formatos dos blocos de comando e de função



Fonte: Marji (2014).

Ao observar-se a Figura 3, pode-se perceber que os espaços com formatos hexagonais ou retangulares arredondados recebem apenas blocos de função com o mesmo formato, enquanto que o espaço de formato retangular aceita qualquer bloco de função. A vantagem disso é que a incompatibilidade dos blocos não permite que eles sejam encaixados, evitando-se, assim, erros na programação.

Para uma melhor visualização dos formatos dos blocos e de seus espaços, a Figura 4 traz os de função hexagonais (sendo os azuis-claros da paleta Sensores e os verdes da Operadores) e os de controle com espaço para o parâmetro hexagonal.

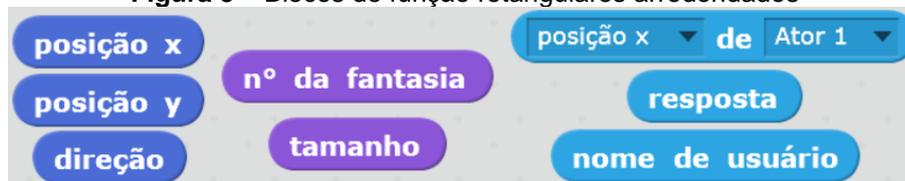
Figura 4 – Blocos hexagonais de função e de controle



Fonte: O Autor.

Já a Figura 5 mostra blocos de função retangulares arredondados, em que os azuis-escuros são de Movimento, os lilases de Aparência e os azuis-claros de Sensores.

Figura 5 – Blocos de função retangulares arredondados



Fonte: O Autor.

Por fim, a Figura 6 apresenta blocos de comando (os laranja pertencentes à paleta Variáveis; os azuis-claros à Sensores; os lilases à Aparência) e de função (o verde, da paleta Operadores), com espaços retangulares para os parâmetros.

Figura 6 – Blocos de comando e de função com espaços retangulares



Fonte: O Autor.

Um dos recursos que o *Scratch* oferece é a criação de novos blocos, recurso esse encontrado na paleta Mais Blocos. Com essa funcionalidade, podemos criar Procedimentos que tem algumas técnicas específicas e, portanto, a próxima subseção tratará sobre eles.

1.1 PROCEDIMENTOS (PROCEDURES)

Um dos recursos do *Scratch* é denominado Procedimentos (*procedures*). Nele, por meio dos blocos personalizados, pode-se estruturar programas maiores em partes menores e, assim, ter um poder de decisão mais amplo sobre a programação. Para tal recurso, usa-se a paleta Mais Blocos, cuja finalidade é definir um novo bloco, capaz de executar vários comandos. As *procedures* são, segundo Marji (2014, p. 92):

Uma sequência de comandos que executa uma função específica. Por exemplo, podemos criar *procedures* que façam os sprites desenhar formas, realizar cálculos complexos, processar dados de entrada de usuário, tocar notas musicais em sequência, administrar jogos e executar várias outras

tarefas. Após terem sido criadas, essas procedures podem servir como blocos de construção para a implementação de todo tipo de aplicação útil.

Esta divisão em partes menores pode tornar a programação mais compreensível e “contribui para manter uma visão clara do programa como um todo e dos relacionamentos entre as partes componentes” (MARJI, 2014, p. 96). A seguir, serão exemplificados a criação e o uso de um procedimento.

Observe a Figura 7. Nela, são apresentadas duas rotinas para se desenhar um quadrado com a medida do lado igual a 100 passos.

Figura 7 – Interface do *Scratch* com duas programações para desenhar um quadrado de lado 100



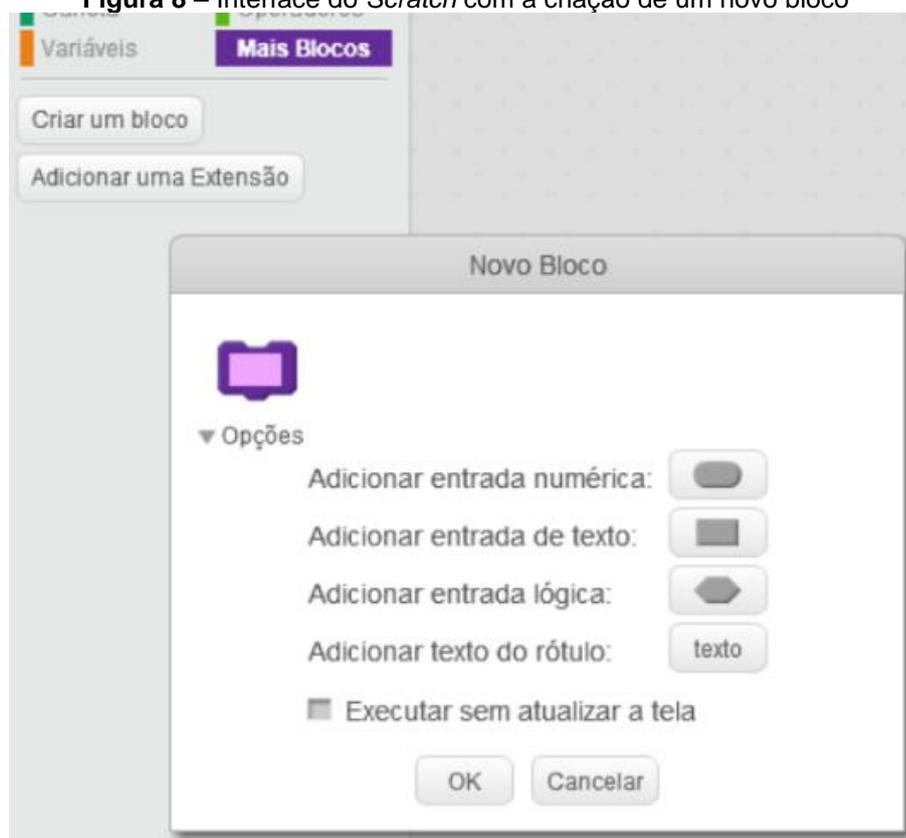
Fonte: O Autor.

A programação apresentada à esquerda da Figura 7 utiliza blocos iguais repetidas vezes e, caso queira-se desenhar um quadrado de lado maior ou menor, deve-se alterar o valor numérico de cada bloco de movimento (mova _ passos). Para evitar a repetição e a conseqüente alteração deles, à direita da Figura 7, usa-se o bloco “repita”. Assim, basta alterar apenas uma vez o valor numérico do bloco de movimento para que o comprimento de todos os lados do quadrado seja modificado. Porém, a programação à direita tem uma limitação. Isso porque, se a programação

requerer quadrados de tamanhos diferentes, é preciso utilizar o bloco “repita” diversas vezes, além de trocar o valor numérico do bloco de movimento, o que pode ser resolvido com a criação de um novo bloco, definido como “Quadrado _ passos”.

Partindo para a criação de um novo bloco, na paleta Mais Blocos, deve-se clicar no botão “criar um bloco” e, então, abre-se a janela observada na Figura 8:

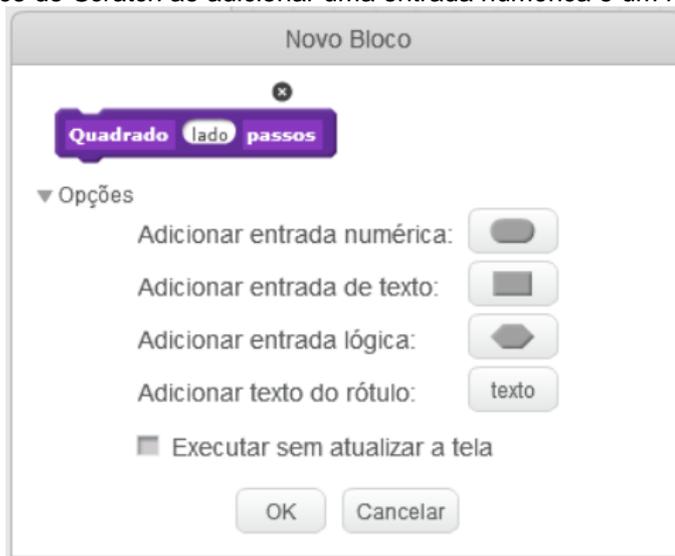
Figura 8 – Interface do Scratch com a criação de um novo bloco



Fonte: O Autor.

Adiciona-se um parâmetro a esse bloco, isto é, clica-se no ícone para *adicionar uma entrada numérica*, a qual é denominada *lado*. Em seguida, acrescenta-se um novo *rótulo* para informar que essa entrada numérica equivale à quantidade de passos que o ator dará para desenhar um quadrado. Desse modo, a criação do bloco fica de acordo com a Figura 9:

Figura 9 – Interface do *Scratch* ao adicionar uma entrada numérica e um rótulo ao novo bloco



Fonte: O Autor.

Ao clicar em “OK”, surge na área de programação o bloco observado na Figura 10, ao qual se denomina procedimento “Quadrado *lado* passos”:

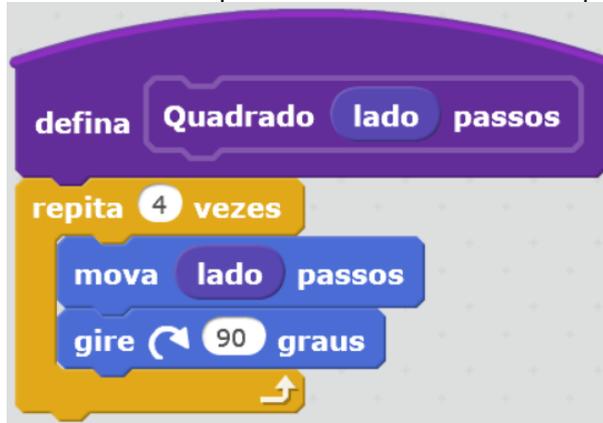
Figura 10 – Procedimento “Quadrado lado passos”



Fonte: O Autor.

Para montar esse procedimento, usa-se o princípio do bloco “repita”, como exposto na Figura 7, mas, no lugar do parâmetro numérico do bloco “mova _ passos”, adiciona-se o parâmetro “lado”, como apresentado na Figura 11:

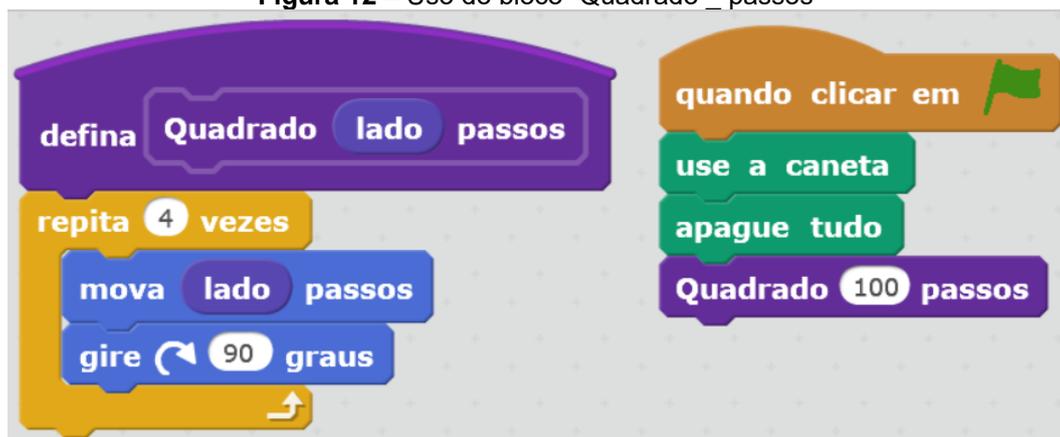
Figura 11 – Blocos do procedimento “Quadrado lado passos”



Fonte: O Autor.

Por fim, para desenhar um quadrado no palco, arrasta-se o bloco criado para a área de programação e escolhe-se o valor do tamanho do lado, como pode ser observado à direita da Figura 12, em que o tamanho do lado é 100. Portanto, quando se quer construir um quadrado, basta selecionar o bloco “Quadrado _ passos” e, na entrada numérica, colocar o valor respectivo ao tamanho do lado que se deseja gerar.

Figura 12 – Uso do bloco “Quadrado _ passos”



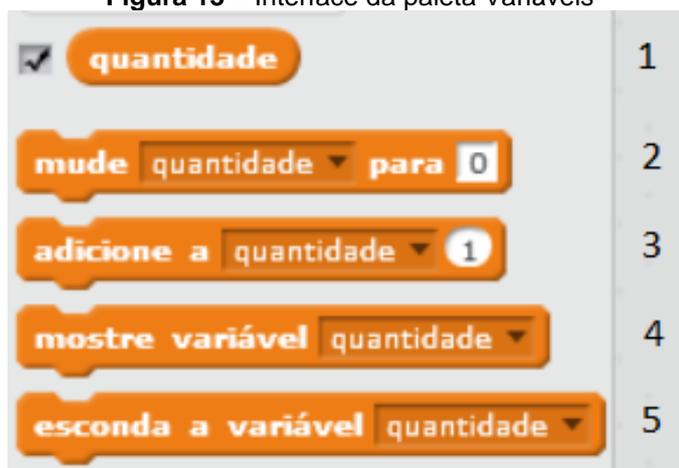
Fonte: O Autor.

A entrada numérica adicionada à criação de um bloco/procedimento (no caso da Figura 11, denominada “lado”) é passível de ser substituída por uma variável, originada na paleta Variáveis, podendo ser um dado numérico ou uma *string*, o que não é, de fato, um problema, pois o “[...] Scratch tenta fazer automaticamente a conversão entre os tipos de dados conforme for necessário” (MARJI, 2014, p. 120). Por essa razão, a seguir, será discutida a funcionalidade das variáveis.

1.2 VARIÁVEIS

Ao criar-se uma variável, pode-se tanto atribuí-la a um ou mais atores quanto adicioná-la a outros blocos, como o de movimento ou qualquer outro que receba um parâmetro numérico/*string*. Após ser gerada, aparecem novos blocos na paleta Variáveis, sendo aquele referente à nova variável um bloco de função no formato retângulo arredondado, como é possível observar na Figura 13. Nela, a variável é denominada “quantidade”:

Figura 13 – Interface da paleta Variáveis



Fonte: O Autor.

Na Figura 13, há cinco blocos: o primeiro representa o nome da variável e é um bloco de função do tipo número/string; o segundo permite definir o valor da variável; o terceiro possibilita alterar o valor da variável para um número fixo, positivo ou negativo, inteiro ou decimal; o quarto e quinto blocos permitem mostrar ou ocultar o monitor da variável no palco enquanto a programação é executada.

Como mencionado anteriormente, o *Scratch* não sabe o tipo de dado de uma variável, mas tenta automaticamente fazer uma conversão. Na Figura 14, são apresentados dois exemplos de programação: na parte A, a *string* “muitos” é convertida para um número (0) e passada para o comando “mova _ passos”; na parte B, a *string* “70” é convertida para um número (70) e passada para o comando “mova _ passos”. Como tal comando espera por um parâmetro numérico, na parte A, o *Scratch* não consegue converter “muitos” para um número e, como resultado, o ator da programação não se move. Por sua vez, na parte B, há a conversão de “70”, logo, o ator executa o movimento de 70 passos.

Figura 14 – Relação do tipo de dado em uma variável



Fonte: O Autor.

Ao se elaborar uma programação, pode-se desejar que o usuário insira algum dado para que o programa seja executado e, por meio desse tipo de informação, que tome alguma decisão no decorrer da programação. Isso envolve duas

funcionalidades do *Scratch*: obter dados do usuário e estruturas de decisão, as quais serão discutidas na subseção a seguir.

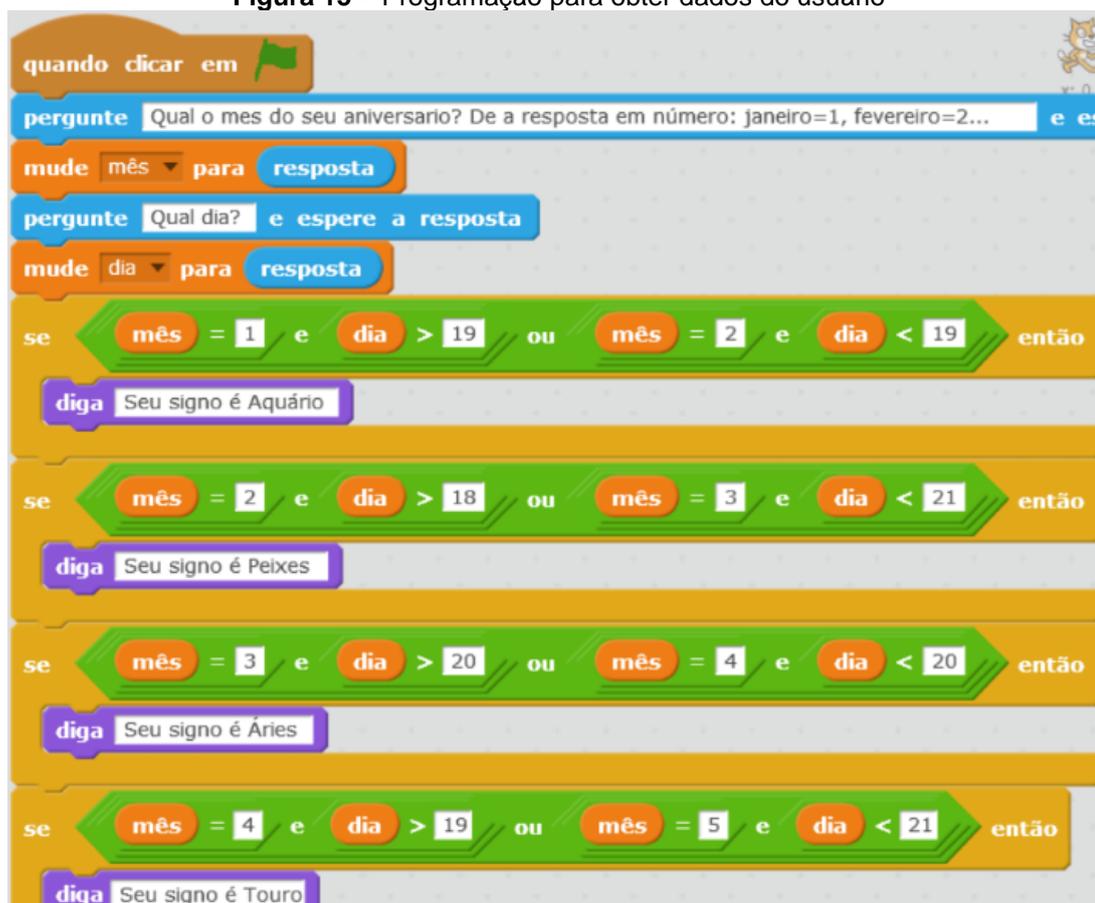
1.3 OBTER DADOS DE ENTRADA DO USUÁRIO E ESTRUTURAS DE DECISÃO

Algumas programações requerem que o usuário insira dados para “rodarem”. Por exemplo, para que um programa calcule o Índice de Massa Corpórea (IMC), ele pedirá ao usuário que digite sua altura e, posteriormente, seu peso.

Para se obter um dado do usuário, deve-se usar a paleta Sensores, e, mais especificamente, o bloco “pergunte _ e espere a resposta”. O *Scratch* armazena o que foi digitado pelo usuário (podendo ser letras, números ou símbolos), grava no bloco “resposta” e continua a execução no comando que estiver imediatamente após o bloco “pergunte _ e espere a resposta”.

Na Figura 15, é mostrado um exemplo de rotina para obter dados do usuário, cuja programação identifica o signo desse sujeito. Para tal cálculo, são feitas duas perguntas: a primeira é sobre o mês de seu aniversário, que deverá responder com o número correspondente, isto é, deve digitar um se seu aniversário é em janeiro, dois se é em fevereiro e assim por diante, até o número 12, caso faça aniversário em dezembro. Após a resposta, o *Scratch* armazena essa informação na variável “mês” e passa para a segunda pergunta, que é sobre em que dia do mês o usuário nasceu. Então, novamente, o *software* armazena essa informação na variável “dia” e, em seguida, o programa avalia tais informações a partir da utilização do bloco “se” da paleta Controle, e dos blocos “igual a”, “maior que”, “menor que”, “e” e “ou” da paleta Operadores.

Figura 15 – Programação para obter dados do usuário

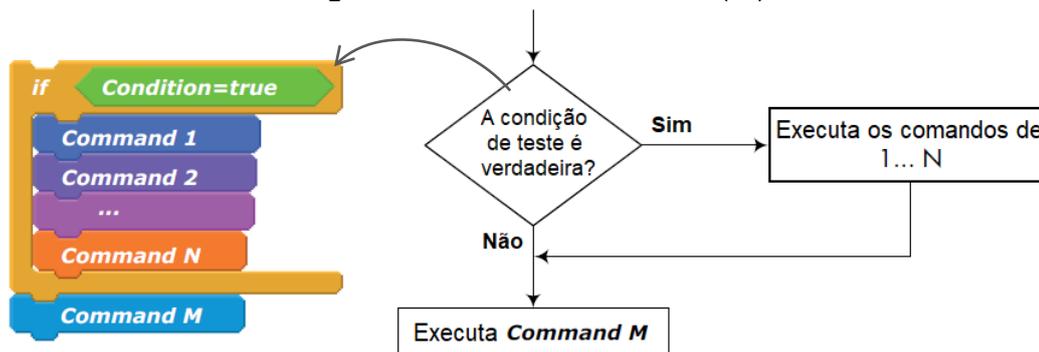


Fonte: O Autor.

Os blocos da paleta Controle nos permitem tomar decisões e controlar as ações de nossos programas de acordo com os resultados. Assim, os blocos “se” e “se/senão” são usados para fazer escolhas, determinadas por condições avaliadas pelas expressões lógicas.

Segundo Marji (2014, p. 157), “o bloco *if* (se) corresponde a uma estrutura de decisão que oferece a capacidade de especificar se um conjunto de comandos deve (ou não) ser executado, de acordo com o resultado da condição de teste”. A funcionalidade do bloco “se” pode ser observada na Figura 16.

Figura 16 – A estrutura do bloco if (se)

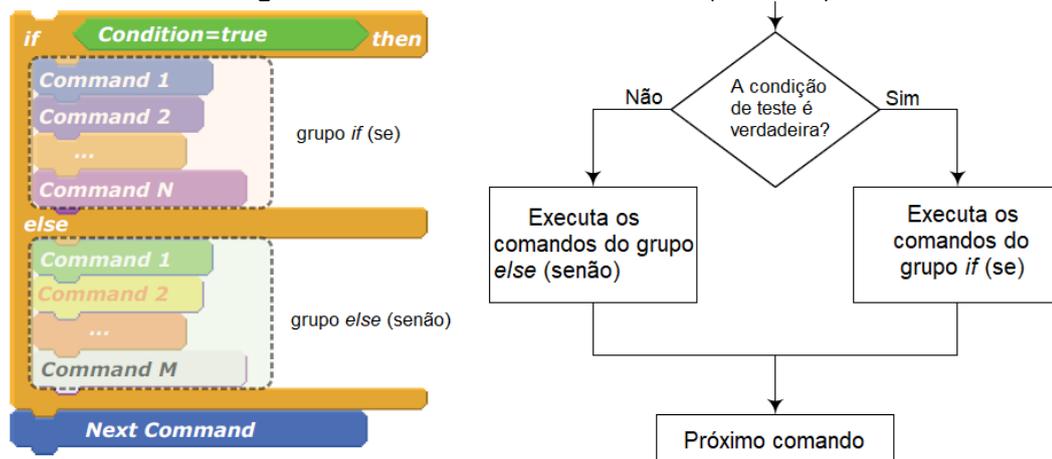


Fonte: Marji (2014).

Nela, “o formato de losango representa um bloco de decisão que fornece uma resposta sim/não (ou verdadeiro/falso) a uma pergunta” (MARJI, 2014, p. 157). Caso a condição de teste (hexágono verde do bloco “se”) seja verdadeira, o programa executará os comandos listados “dentro” do bloco (*Command 1* a *Command N*) antes de seguir para o próximo comando do bloco “se”, isto é, o *Command M*. Entretanto, se a condição de teste for falsa, o programa não executa os comandos de dentro do bloco e segue direto para o *Command M*.

A estrutura do bloco “se/senão” (trazida na Figura 17) possui dois grupos de comando: um referente ao “se” e outro ao “senão”. Caso a condição de teste (hexágono verde) seja verdadeira, o programa executará os comandos referentes ao grupo “se” (*Command 1* a *Command N*), mas, se a condição for falsa, os comandos do grupo “senão” é que serão executados (*Command 1* a *Command M*). A programação efetuará apenas um dos dois grupos de comando para, posteriormente, seguir para o comando seguinte (*Next Command*).

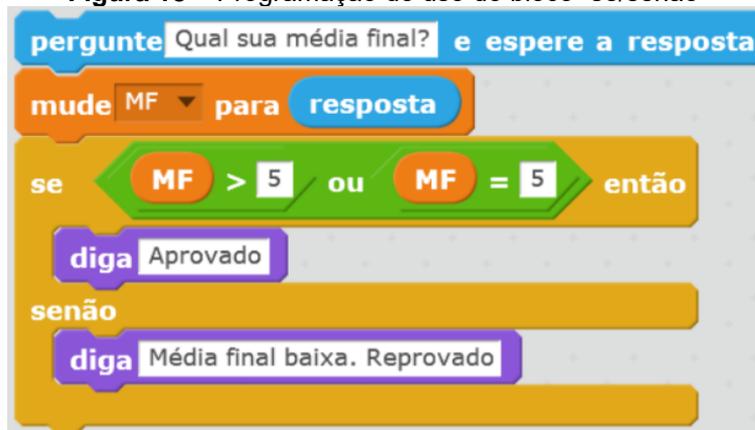
Figura 17 – A estrutura do bloco if/else (se/senão)



Fonte: Marji (2014).

Uma programação que mostra o uso do bloco “se/senão” pode ser observada na Figura 18. Nela, são utilizados os operadores “maior que”, “igual a” e “ou” para determinar se a média final fornecida pelo usuário é suficiente para que ele seja aprovado em uma disciplina.

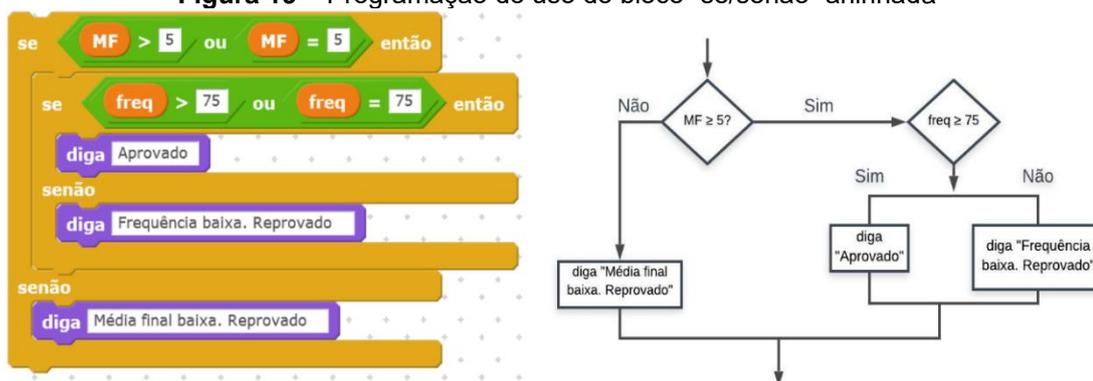
Figura 18 – Programação do uso do bloco “se/senão”



Fonte: O Autor.

Caso se queira testar mais de uma condição antes de realizar a ação, é possível aninhar vários blocos “se” ou “se/senão” uns dentro de outros, de modo a efetuar as análises necessárias. Considerando-se o exemplo da Figura 18 e adicionando nele mais uma condição para ser aprovado, por exemplo, a frequência, ter-se-ia a programação da Figura 19.

Figura 19 – Programação do uso do bloco “se/senão” aninhada



Fonte: O Autor.

Inicialmente, a expressão $MF \geq 5$ (representada no cabeçalho do bloco por: $MF > 5$ ou $MF = 5$) é testada. Se for falsa, não há necessidade de verificar a outra condição, e é dito “Média final baixa. Reprovado”. Caso contrário, se a expressão for verdadeira, a outra condição é testada. Isso é feito por meio do bloco “se/senão” aninhado, o qual verifica a frequência por meio da seguinte expressão: $freq \geq 75$ (representada no cabeçalho do bloco por: $freq > 75$ ou $freq = 75$). Se a segunda condição também for verdadeira, então, o usuário atendeu aos requisitos necessários para aprovação, afirmando-se: “Aprovado”. Do contrário, o usuário não está apto para a aprovação e a seguinte mensagem é enunciada: “Frequência baixa. Reprovado”.

Apêndice B – Carta de apresentação do pesquisador

À Senhora

SORÁIA PESSOA VIEIRA

Diretora da Escola Estadual Professora Carolina Augusta Seraphim

Endereço: Rua 5B, 1191, Vila Indaiá, CEP: 13506-734, Rio Claro/SP

Assunto: **Apresentação do Pesquisador**

Prezada Diretora,

O mestrando **Pedro Henrique Giral di de Souza**, do Programa de Pós-Graduação em **Educação Matemática** da Universidade Estadual Paulista (**UNESP**), Campus de Rio Claro, está desenvolvendo uma pesquisa de mestrado intitulada “Modelagem Matemática e pensamento computacional na Escola de Ensino Integral”, sob minha orientação. O objetivo de sua pesquisa é investigar as possíveis contribuições que a Modelagem Matemática e o desenvolvimento do pensamento computacional podem propiciar ao processo de aprendizagem de conceitos matemáticos dos alunos das séries finais do Ensino Fundamental. O desenvolvimento do pensamento computacional e da Modelagem será trabalhado ao investigar e interagir com atividades didáticas desenvolvidas com o uso da linguagem de programação Scratch e Arduino.

Assim, solicito autorização para que o mestrando possa frequentar as aulas da disciplina “Práticas de Matemática” do 9º ano, e em conjunto com o professor responsável pela disciplina, elaborar e propor atividades investigativas de conteúdos matemáticos compatíveis com o currículo desenvolvido na referida disciplina. Essas atividades comporão assim o cenário de investigação da pesquisa de mestrado de Pedro. Com o intuito de registrar para posterior análise e escrita de sua dissertação, o mestrando irá gravar as aulas. Informo ainda que esse é um procedimento usual em pesquisas dessa natureza e, para a preservação das pessoas, todo o material produzido será somente utilizado na dissertação e trabalhos científicos relacionados à pesquisa desenvolvida.

Agradeço vossa compreensão e colaboração no processo de desenvolvimento desta pesquisa de Mestrado e colocamo-nos à disposição para eventuais esclarecimentos através dos endereços eletrônicos suelilj@fc.unesp.br ou pe_gisouza@hotmail.com, bem como pelo celular (XX) XXXX-XXX.

Atenciosamente,

Pedro Henrique Giraldi de Souza

Mestrando do Programa de Pós-Graduação em Educação Matemática do Instituto de Geociências e Ciências Exatas (IGCE) da Universidade Estadual Paulista (UNESP), Campus Rio Claro

Profa. Dra. Sueli Liberatti Javaroni

Docente da Faculdade de Ciências (FC) da Universidade Estadual Paulista (UNESP), Campus Bauru e Docente/Pesquisadora do Programa de Pós-Graduação em Educação Matemática do Instituto de Geociências e Ciências Exatas (IGCE) da Universidade Estadual Paulista (UNESP), Campus Rio Claro

Apêndice C – Carta de apresentação de projeto de pesquisa

À Senhora

SORÁIA PESSOA VIEIRA

Diretora da Escola Estadual Professora Carolina Augusta Seraphim

Endereço: Rua 5B, 1191, Vila Indaiá, CEP: 13506-734, Rio Claro/SP

Assunto: **Apresentação de Projeto de Pesquisa e solicitação de autorização condicionada**

Prezada Senhora,

Apresentamos o Projeto de Pesquisa “Modelagem Matemática e pensamento computacional na Escola de Ensino Integral”, desenvolvido pelo mestrando Pedro Henrique Giraldi de Souza sob a orientação da Profa. Dra. Sueli Liberatti Javaroni.

A pesquisa tem como objetivo investigar as contribuições da Modelagem Matemática e do pensamento computacional, subjacente ao uso da linguagem de programação Scratch e Arduíno, por alunos da Educação Básica, para a aprendizagem de conceitos matemáticos.

Espera-se ainda contribuir com as aulas de Matemática da turma escolhida para a realização da pesquisa, seja auxiliando a professora e os alunos no processo de ensino e aprendizagem, seja proporcionando a utilização de recursos tecnológicos nas aulas de Matemática.

Ressaltamos que a Profa. Dra. Sueli Liberatti Javaroni já realizou parcerias com a Diretoria de Ensino – Região de Limeira através do Projeto intitulado “Mapeamento do uso de tecnologias da informação nas aulas de Matemática no Estado de São Paulo”, realizando cursos de formação continuada para professores de Matemática, utilizando o software GeoGebra.

No que tange a pesquisa atual, solicitamos autorização para que a pesquisa seja realizada na disciplina “Práticas de Matemática” do 9º ano, através de entrevistas e observação de aulas como instrumentos de produção de dados, utilizando gravações de áudio. Queremos informar que o caráter ético desta pesquisa assegura a preservação da identidade das pessoas participantes.

Uma das metas para a realização deste estudo é o comprometimento do pesquisador em possibilitar, aos participantes, um retorno dos resultados da pesquisa. Solicitamos ainda a permissão para a divulgação desses resultados e suas respectivas conclusões, em forma de pesquisa, preservando sigilo e ética,

conforme termo de consentimento livre que será assinado pelos participantes envolvidos. Esclarecemos que tal autorização é uma pré-condição.

As informações a serem oferecidas para o pesquisador serão guardadas pelo tempo que determinar a legislação e não serão utilizadas em prejuízo desta instituição e/ou das pessoas envolvidas, inclusive na forma de danos à estima, prestígio e/ou prejuízo econômico e/ou financeiro. Além disso, durante ou depois da pesquisa é garantido o anonimato de tais informações.

Agradecemos vossa compreensão e colaboração no processo de desenvolvimento desta pesquisa de Mestrado. Em caso de dúvida, você pode encontrar o pesquisador no telefone: (11) 99493-3960 ou pelo e-mail: pe_gisouza@hotmail.com

Atenciosamente,

.....
Pedro Henrique Giraldi de Souza

Mestrando do Programa de Pós-Graduação em Educação Matemática do Instituto de Geociências e Ciências Exatas (IGCE) da Universidade Estadual Paulista (UNESP), Campus Rio Claro

.....
Profa. Dra. Sueli Liberatti Javaroni

Docente da Faculdade de Ciências (FC) da Universidade Estadual Paulista (UNESP), Campus Bauru e Docente/Pesquisadora do Programa de Pós-Graduação em Educação Matemática do Instituto de Geociências e Ciências Exatas (IGCE) da Universidade Estadual Paulista (UNESP), Campus Rio Claro

Apêndice D – Termo de consentimento livre e esclarecido

Estamos convidando o aluno(a) que está sob sua responsabilidade, para participar de uma pesquisa a ser realizada na disciplina “**Práticas de Matemática**”, intitulada, inicialmente, “Modelagem Matemática, pensamento computacional e suas relações”. Para tanto, necessitamos do seu consentimento.

A pesquisa tem como objetivo investigar se o desenvolvimento do pensamento computacional articulado a atividades de Modelagem Matemática possibilitam a criação de um cenário que propicia a produção de conhecimento matemático elaborado pelos estudantes. Sendo essas atividades desenvolvidas por estudantes do nono ano do Ensino Fundamental de uma Escola Estadual de Ensino Integral de Rio Claro/São Paulo, ao realizarem atividades com kits de robótica. Serão utilizados como instrumentos de produção de dados: **gravação de áudio e vídeo das aulas e atividades desenvolvidas pelos alunos, como a Feira de Ciências**. A pesquisa será realizada nas dependências da **Escola Estadual Professora Carolina Augusta Seraphim**.

Os dados produzidos serão utilizados unicamente com finalidades acadêmicas e os sujeitos terão suas identidades preservadas caso assim desejarem e abaixo se manifestarem.

A pessoa que realizará a pesquisa será o mestrando Pedro Henrique Giraldo de Souza, do Programa de Pós-Graduação em **Educação Matemática** da Universidade Estadual Paulista (**UNESP**), Campus de Rio Claro e a Prof. Dra. Sueli Liberatti Javaroni da Universidade Estadual Paulista (**UNESP**), Campus de Bauru, orientadora da pesquisa. Solicitamos a sua autorização para a realização do estudo e para produção de artigos técnicos e científicos. Caso aceite, assine ao final deste documento, que está em duas vias. Uma delas é sua, a outra é do pesquisador responsável. **Em caso de recusa, o aluno continuará participando da aula normalmente, mas não participará da pesquisa, e não será prejudicado de forma alguma.**

**CONSENTIMENTO DA PARTICIPAÇÃO DA PESSOA COMO SUJEITO DA
PESQUISA**

Eu, _____
_____, RG/CPF _____,
_____, responsável pelo
aluno(a)

_____, autorizo que ele(a) participe da pesquisa de Mestrado mencionada, como
sujeito. Fui informado sobre a pesquisa e seus procedimentos e declaro (escolher
uma das opções abaixo):

- não ser necessário manter o sigilo do seu nome e imagem.
- ser necessário manter o sigilo do seu nome e imagem
- ser necessário manter o sigilo apenas do seu nome
- ser necessário manter o sigilo apenas da sua imagem

Foi-me garantido que posso retirar o consentimento a qualquer momento e que
todas as dúvidas que eu tiver, posso estar esclarecendo com o pesquisador e sua
orientadora.

Rio Claro de de 2018.

Apêndice E – Descrição das Atividades 2 e 3 realizadas pelos estudantes na produção de dados

Atividade 2

A segunda atividade foi composta por duas tarefas. A primeira, denominada “Pegando maçãs”, teve o intuito de fazer com que os estudantes usassem as setas da direita e da esquerda do teclado para moverem o carrinho horizontalmente, de modo a colher o máximo possível de maçãs que caíam verticalmente. Na Figura 1, observa-se o palco do jogo que projetei na TV do laboratório para os discentes.

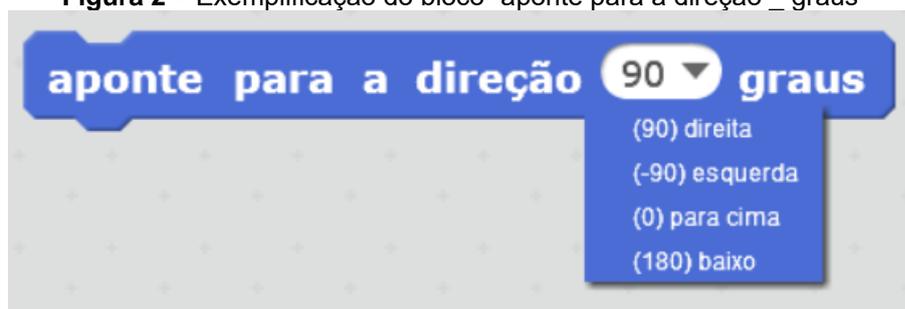
Figura 1 – Palco do *Scratch* representando o jogo “Pegando maçãs”



Fonte: O Autor.

O objetivo dessa primeira tarefa foi mostrar aos estudantes que a movimentação alteraria apenas a coordenada x do carrinho, já que sua posição y estava fixada. Expliquei que, ao selecionar um novo ator, automaticamente, sua direção é para a direita, direção essa considerada 90 graus no *Scratch*, como pode ser observado na Figura 2.

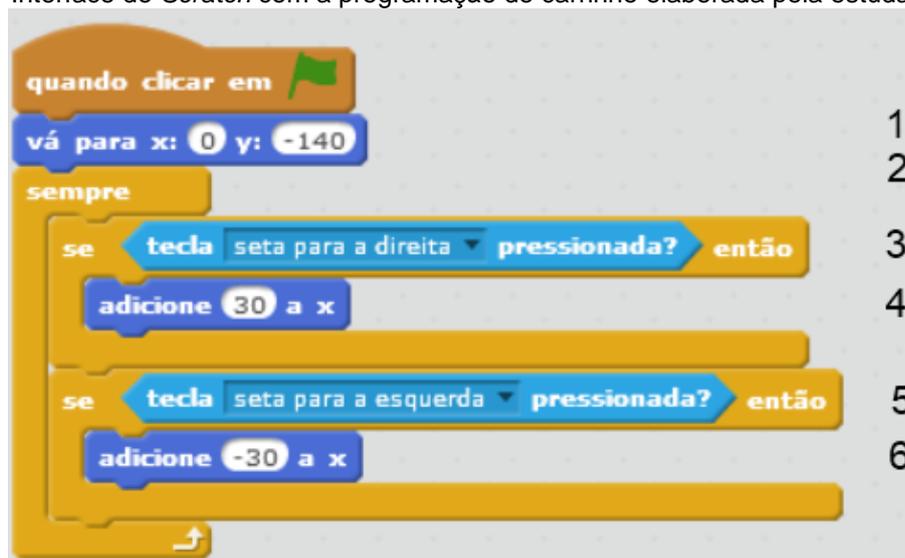
Figura 2 – Exemplificação do bloco “aponte para a direção _ graus”



Fonte: O Autor.

Assim, como se vê, devemos adicionar um valor positivo para que ele se desloque nesse sentido, porém, se quisermos que se movimente para a esquerda, ou seja, na direção contrária à orientação do carrinho, é preciso acrescentar um valor negativo a x. Por isso, o sinal de menos na sexta linha da Figura 3, cuja programação foi realizada pela estudante Estefani.

Figura 3 – Interface do *Scratch* com a programação do carrinho elaborada pela estudante Estefani



Fonte: Dados da pesquisa.

Feito isso, pedi que executassem a programação, isto é, que clicassem na bandeira verde para que pudessem observar, no canto superior direito da área de *script*, o que acontecia com os valores das coordenadas x e y do carrinho. Desse modo, os estudantes puderam constatar, por meio da simulação, que a coordenada y não se alterava, apenas a x.

Já com a programação da maçã (vide Figura 4), o movimento era vertical para baixo, ou seja, as maçãs apareciam aleatoriamente (linha 7) em uma altura fixa e com coordenada x aleatória (linha 5), e, então, caíam (linha 11). Explicando aos

discentes, relacionei o movimento da queda com o eixo y, pois somente ele se alterava enquanto a maçã caía, a coordenada x mantinha-se fixa. Assim, mostrei a eles que deveríamos adicionar um número negativo ao movimento relacionado ao eixo y, porque o valor da coordenada y da maçã ficaria cada vez menor conforme caísse no jogo. Para verificar tal fato, pedi que colocassem um valor positivo para observarem o que acontecia e, então, puderam constatar que, assim o fazendo, a maçã subia em vez de descer.

Figura 4 – Programação da maçã, elaborada pela estudante Estefani

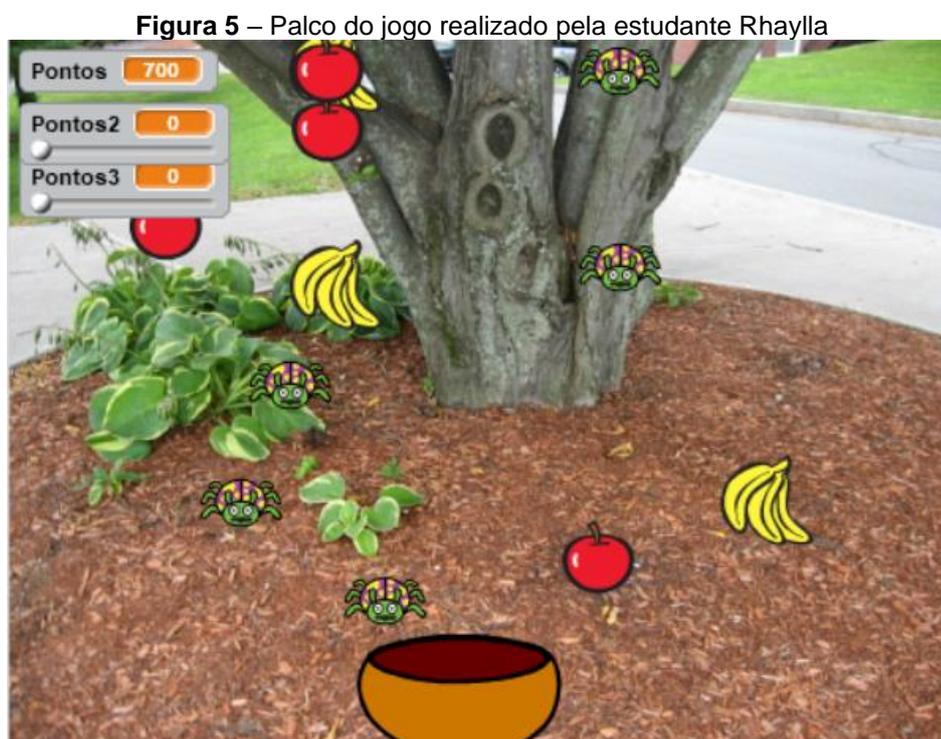


Fonte: Dados da pesquisa.

A pontuação dessa tarefa requeria a criação de uma variável. Por isso, expliquei aos alunos como era feita a programação, visto que, caso a maçã encostasse no carrinho (linha 12), aumentaria a pontuação (linha 13). Já se ela atingisse uma determinada altura, equivalente a encostar no chão, ou seja, se tivesse uma determinada coordenada y (linha 16), ela sumiria (linha 18) e a

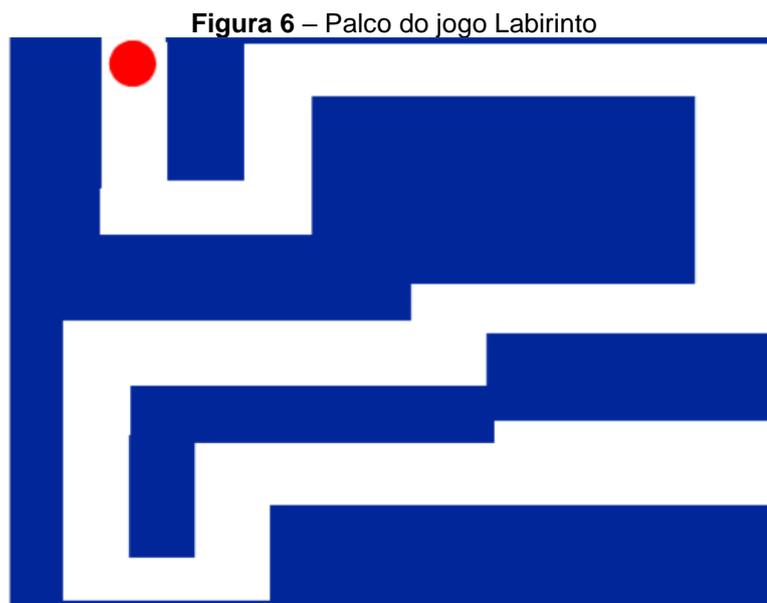
pontuação não se alteraria. Naquele momento, não aprofundi as funcionalidades da Paleta Variáveis, já que isso seria feito posteriormente em outra atividade.

Após concluírem a programação do jogo, pedi que explorassem os blocos, no sentido de alterarem os valores e verem o que acontecia quando os executavam novamente. Em seguida, perguntei o que poderia ser feito para deixar o jogo mais difícil, questionando-os sobre quais seriam os respectivos blocos que deveriam ser alterados para que essas mudanças acontecessem. Decidi, então, dar um tempo para que eles explorassem o jogo e fizessem eventuais mudanças. Na Figura 5, pode-se observar o palco do jogo de uma das estudantes. Nele, ela adicionou dois novos atores, a saber, uma banana e um inseto, sendo que o primeiro soma pontos e o segundo retira.



Fonte: Dados da pesquisa.

Na segunda tarefa, pedi para que os estudantes desenvolvessem um jogo de labirinto. Para tanto, precisavam fazer com que o ator se movesse em duas direções (horizontal e vertical) e em quatro sentidos (para a direita, para a esquerda, para cima e para baixo), podendo usar quaisquer teclas para realizar tal movimento. Porém, sugeri que usassem as setas do teclado e exibi um exemplo na TV do laboratório (vide Figura 6), mas sem revelar a programação necessária para tal execução.



Fonte: O Autor.

Semelhante à do jogo da maçã, a finalidade dessa tarefa era compreender se os estudantes associavam o movimento horizontal com o eixo x e o vertical com o y, além de entenderem que, para deslocar o ator para baixo e para a esquerda, precisariam adicionar um número negativo ao bloco de movimento. Isso se deve ao fato de que, ao selecionar um novo ator, automaticamente, sua direção é para a direita. Então, para que o objeto se mova para a esquerda, movimento oposto ao que o objeto naturalmente se encontra, é necessário inserir um sinal negativo na quantidade de passos que o ator deve se movimentar. O mesmo é válido para o movimento vertical para baixo³⁸.

Nesta tarefa, ocorreram duas situações: na primeira, observada na Figura 7, os estudantes duplicaram os blocos de movimento do carrinho da atividade da maçã (Figura 3) para utilizá-los também nesta, adicionando a eles a “seta para cima” e a “seta para baixo”. Porém, não se atentaram ao fato de que o seu deslocamento ainda estava relacionado ao eixo x, pois não mudaram o bloco de movimento vinculado às novas teclas, o que fez com que o ator se movesse para a direita ao pressionarem a seta para cima e para a esquerda quando apertavam a seta para baixo.

³⁸ Durante esta etapa da tarefa, a programação criada para o jogo “Pegando maçãs” foi exibida no display da TV.

Figura 7 – Representação da primeira situação observada na Atividade 2



Fonte: Dados da pesquisa.

Logo após esse fato, pedi para que observassem na programação todos os blocos referentes às teclas “seta para cima” e “seta para baixo” e procurassem onde poderia estar o erro ocorrido. A maioria percebeu e modificou o bloco de movimento “adicione _ a x” para “adicione _ a y”, alterando também seu valor numérico. Para aqueles que ainda não tinham conseguido corrigir a programação, ressaltéi que o movimento vertical, isto é, o deslocamento gerado ao usarem as setas para cima e para baixo, tem relação com o eixo y e, por isso, na programação, deveriam mudar os blocos de movimento relacionados a tais setas, lembrando-os também que, no movimento para baixo, o valor adicionado deveria ser negativo, assim como o da maçã.

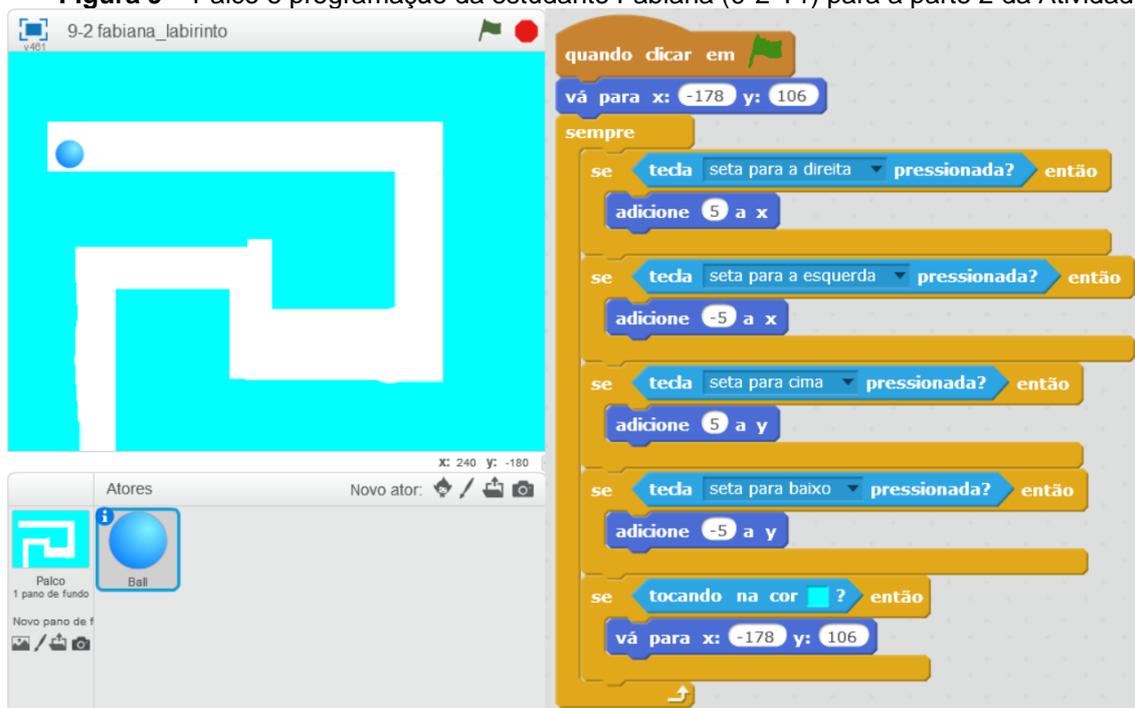
Na segunda situação, observada na Figura 8, os estudantes mudaram corretamente os blocos para o movimento vertical, ou seja, colocaram “adicione _ a y” nas setas para cima e para baixo, mas esqueceram de adicionar um valor negativo ao bloco da seta para baixo, resultando, assim, no deslocamento ascendente do ator quando pressionada a seta pra baixo. Fiz, então, a mesma ressalva da primeira situação.

Figura 8 – Representação da segunda situação observada na Atividade 2

Fonte: Dados da pesquisa.

Após programarem a movimentação do ator, os estudantes precisavam desenhar um labirinto no palco, no qual o personagem não poderia encostar, senão voltaria ao ponto de partida. Uma das maneiras para fazê-lo era adicionar o bloco de controle “se” e, no espaço para o parâmetro *booleano*, selecionar o bloco “tocando na cor _”, escolhendo a mesma coloração do labirinto. Além disso, dentro desse bloco, é preciso inserir um outro, referente à posição inicial do ator. A Figura 9 denota essa representação realizada pela estudante Fabiana (9^o2-T1).

Figura 9 – Palco e programação da estudante Fabiana (9º2-T1) para a parte 2 da Atividade 2



Fonte: Dados da pesquisa.

Na primeira parte da segunda atividade, os estudantes atingiram rapidamente o objetivo do jogo, qual seja o de criar um ator (segundo ator) responsável por “colher” frutas que caíam verticalmente na tela. Alguns até adicionaram novos personagens para somar/diminuir pontos, como pode ser observado na Figura 5. Uma das duplas de estudantes adicionou inclusive um outro elemento para também colher as frutas, assim, enquanto um deles jogava com as setas do teclado, o outro o fazia com as teclas A e D. Porém, infelizmente, eles não salvaram o *upgrade* do jogo inicial e, desse modo, não possui tal registro para apresentá-lo aqui. No entanto, esse desenvolvimento evidencia o potencial que os alunos podem ter a partir de uma atividade proposta.

Como dito anteriormente, a segunda parte da atividade dois consistia em fazer um labirinto. No que concerne a ela, consideramos que a generalização da programação da primeira parte para a segunda foi razoável no que diz respeito à movimentação. No entanto, houve dificuldade em fazer o labirinto, visto que foi quase unânime a opinião de que ele precisaria, necessariamente, ser um novo ator. Disse que poderiam desenhar o labirinto no palco e, a partir dele, elaborar a programação do ator que se moveria pelo jogo.

Atividade 3

Na primeira parte da Atividade 3, mostrei aos estudantes a função dos procedimentos (*procedures*), assim como descrito no Apêndice A, e realizamos a primeira parte da tarefa juntos. Para tanto, eu projetei o *Scratch* na TV do laboratório e eles acompanharam por seus computadores. Comecei mostrando duas programações que executavam o desenho de um quadrado de lado 100, como se pode observar na Figura 10.

Figura 10 – Programação para desenhar um quadrado



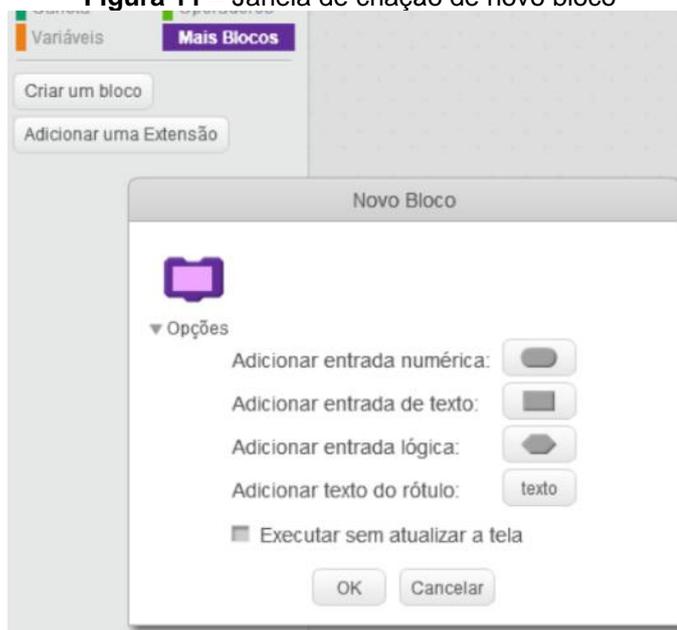
Fonte: O Autor.

Comentei com os estudantes que ambas faziam o desenho de um quadrado de lado 100, mas a programação à esquerda da Figura 10 utilizava blocos iguais repetidas vezes e, por isso, caso quisessem aumentar ou diminuir o tamanho deste quadrado, deveriam alterar o valor de todos os blocos de movimento (mova _ passos). Para evitar essa repetição e a consequente alteração dos blocos, deveriam usar o bloco “repita” (vide programação à direita da Figura 10). Assim, bastava alterarem uma vez o valor do bloco de movimento para que o comprimento de todos os lados do quadrado fosse alterado. Expliquei ainda que a programação à direita também apresentava uma limitação, visto que, se quiséssemos desenhar quadrados

de tamanhos diferentes, teríamos que alterar o valor do bloco de movimento ao utilizar o bloco “repita”. Disse a eles que esse entrave poderia ser resolvido com a criação de um novo bloco, definido como “Quadrado _ passos”.

Então, partimos para a criação de um bloco. Afirmei que, ao clicar no botão “criar um bloco”, abriria a janela observada na Figura 11.

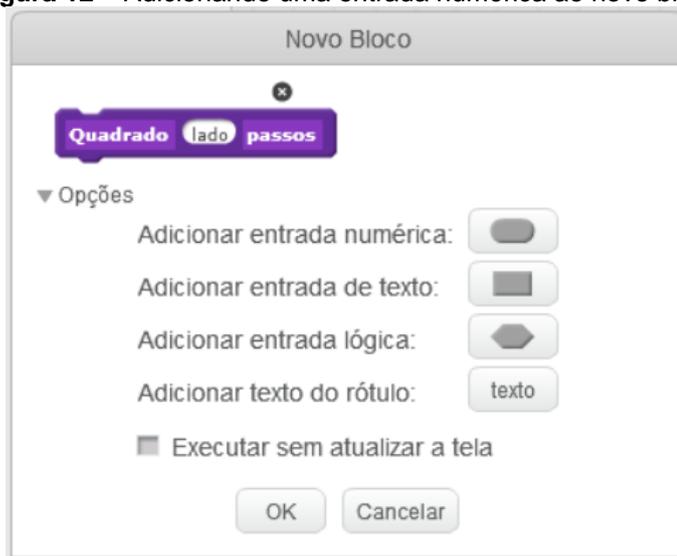
Figura 11 – Janela de criação de novo bloco



Fonte: O Autor.

Adicionamos um parâmetro a esse bloco, isto é, clicamos no ícone para incluir uma entrada numérica e a chamamos de lado; em seguida, inserimos um novo rótulo para informar que essa entrada numérica equivalia à quantidade de passos que nosso ator daria para desenhar um quadrado. Desse modo, a criação do bloco ficou da seguinte maneira (vide Figura 12):

Figura 12 – Adicionando uma entrada numérica ao novo bloco



Fonte: O Autor.

Ao clicar em “OK”, aparecia na área de programação o bloco mostrado na Figura 13, o qual denominamos de procedimento “Quadrado *lado* passos”:

Figura 13 – Procedimento “Quadrado lado passos”



Fonte: O Autor.

Para montarmos esse procedimento, usamos o princípio do bloco “repita”, como na Figura 10, mas, no lugar do parâmetro numérico do bloco “mova _ passos”, adicionamos o parâmetro “lado”, como observado na Figura 14.

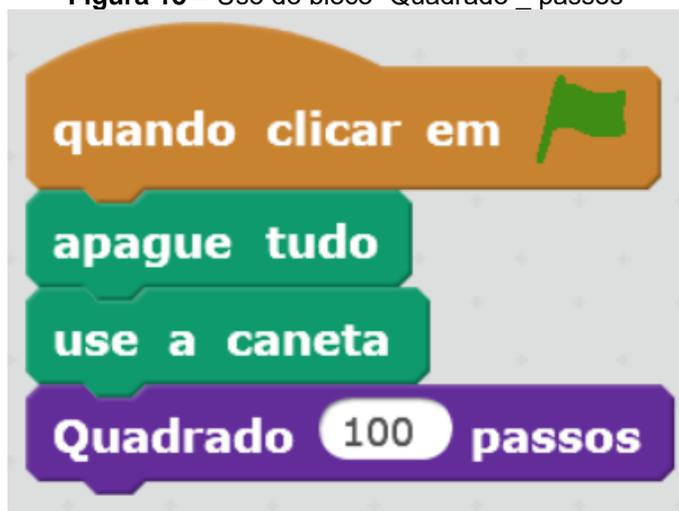
Figura 14 – Blocos do Procedimento Quadrado



Fonte: Dados da pesquisa.

Por fim, expliquei aos estudantes que, para desenharmos um quadrado no palco, seria preciso arrastar o bloco criado para a área de programação e escolher o valor do tamanho do lado, como na Figura 15, em que esse tamanho é 100. Portanto, quando quiséssemos construir um quadrado, bastava selecionar o bloco “Quadrado _ passos” e, na entrada numérica, colocar o valor respectivo ao tamanho do lado desejado.

Figura 15 – Uso do bloco “Quadrado _ passos”



Fonte: Dados da pesquisa.

Na parte dois da Atividade 3, apresentei aos estudantes o procedimento “Folha”, que, como o próprio nome diz, definia o desenho de uma folha, e cuja programação pode ser observada na Figura 16.

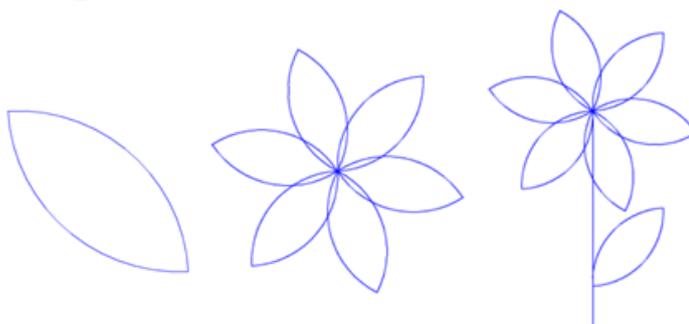
Figura 16 – Blocos do Procedimento “Folha”



Fonte: O Autor.

A partir desse bloco, pedi a eles que criassem um novo Procedimento que, dessa vez, desenhasse uma flor e, depois, um que desenhasse essa mesma flor com um ramo. Para melhor compreensão do que era pedido, fiz um esboço na lousa, mostrando tais desenhos, os quais também podem ser observados na Figura 17.

Figura 17 – Desenho da folha, flor e flor com ramo



Fonte: O Autor.

Disse aos alunos que o desenho da flor não precisaria necessariamente ter seis folhas como na Figura 17, mas deveria conter pelo menos três. Alguns deles se frustraram, pois não acertavam o ângulo/quantidade de graus que deveriam usar e, desse modo, as folhas ficavam sobrepostas. Devido ao tempo, ninguém conseguiu fazer o desenho da flor com o ramo.