

GOVERNO DO ESTADO DE MATO GROSSO  
SECRETARIA DO ESTADO DE CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DO ESTADO DE MATO GROSSO  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO STRICTO SENSU EM ENSINO DE CIÊNCIAS  
E MATEMÁTICA – PPGECM

ANTONIO MARCOS ALVES DA COSTA

**O USO DO *SOFTWARE* SCRATCH NA PRODUÇÃO DE CONHECIMENTOS  
MATEMÁTICOS E DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL**

BARRA DO BUGRES – MT  
2021

ANTONIO MARCOS ALVES DA COSTA

**O USO DO SOFTWARE SCRATCH NA PRODUÇÃO DE CONHECIMENTOS  
MATEMÁTICOS E DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL**

Dissertação Apresentada ao Programa de Pós-Graduação Stricto Sensu em Ensino de Ciências e Matemática – Nível Mestrado Acadêmico - Universidade do Estado de Mato Grosso– UNEMAT, na linha de pesquisa em Tecnologias Digitais no Ensino de Ciências e Matemática.

Orientador: Prof. Dr. Diego Piasson

Coorientador: Everaldo Fernandes Barbosa

BARRA DO BUGRES – MT  
2021

Luiz Kenji Umeno Alencar CRB 1/2037

C837o COSTA, Antonio Marcos Alves da.  
O Uso do Software Scratch na Produção de Conhecimentos Matemáticos e Desenvolvimento do Pensamento Computacional / Antonio Marcos Alves da Costa - Barra do Bugres, 2021.  
191 f.; 30 cm.

Trabalho de Conclusão de Curso  
(Dissertação/Mestrado) - Curso de Pós-graduação Stricto Sensu (Mestrado Acadêmico) Ensino de Ciências e Matemática, Faculdade de Ciências Exatas e Tecnológicas, Câmpus de Barra do Bugres, Universidade do Estado de Mato Grosso, 2021.

Orientador: Diego Piasson

Coorientador: Everaldo Fernandes Barbosa

1. Pensamento Computacional. 2. Matemática. 3. Scratch. I. Antonio Marcos Alves da Costa. II. O Uso do Software Scratch na Produção de Conhecimentos Matemáticos e Desenvolvimento do Pensamento Computacional: .

CDU 51:004.4

ANTONIO MARCOS ALVES DA COSTA

**O USO DO SOFTWARE SCRATCH NA PRODUÇÃO DE  
CONHECIMENTOS MATEMÁTICOS E DESENVOLVIMENTO DO  
PENSAMENTO COMPUTACIONAL.**

Dissertação apresentada ao Programa de Pós-Graduação *Stricto Sensu* em Ensino de Ciências e Matemática – PPGECM - da Universidade do Estado de Mato Grosso CARLOS ALBERTO REYES MALDONADO - UNEMAT, *Câmpus* Univ. Dep. Est. “Renê Barbours” – Barra do Bugres - MT, como requisito obrigatório para a obtenção do título de Mestre em Ensino de Ciências e Matemática.

Aprovado em: 08 de setembro de 2021.

BANCA EXAMINADORA



---

Prof. Dr. Diego Piasson (PPGECM/UNEMAT)  
Orientador



---

Prof. Dr. William Vieira Gonçalves (PPGECM/UNEMAT)  
Examinador Interno



---

Prof. Dr. Sérgio Carrazedo Dantas (UNESPAR)  
Examinador Externo

## Agradecimentos

Em um momento tão complicado da história, agradeço primeiramente a Deus por estar vivo. Agradeço a minha esposa Luzinete pela compreensão durante o tempo de estudos, também minha mãe Florinda e meu pai Augusto, por terem me dado a vida e meus irmãos Cleusa, Neusa, Nilce, Nelson e Fátima, por juntos formarmos uma família em que nos apoiamos nas dificuldades.

Ao meu orientador Diego Piasson e o coorientador Everaldo Barbosa por terem me ajudado a concluir esse sonho.

A todos os 14 cursistas que participaram da pesquisa, sem eles não seria possível a realização desse trabalho, eles foram incríveis, produziram muito em pouco tempo.

Aos amigos que me ajudaram durante o mestrado, em especial a toda turma 2019.

A UNEMAT Barra do Bugres por ter proporcionado essa oportunidade, pois, graças a essa Universidade tantas pessoas estão no mercado de trabalho se formando, se profissionalizando, é uma instituição que vem para democratizar o ensino e o mercado de trabalho, e o mais importante é do povo de Mato Grosso e toda luta por ela é pouco, obrigado a cada profissional que trabalha na UNEMAT.

## RESUMO

O pensamento computacional vem sendo, cada vez mais, discutido nos sistemas educacionais de muitos países. Desde a aprovação da Base Nacional Comum Curricular (BNCC), em 2017, o primeiro documento oficial que versa sobre o tema, têm crescido o número de estudos sobre como desenvolver as habilidades e competências relacionadas ao pensamento computacional no âmbito escolar. Neste contexto, uma pesquisa de cunho qualitativo foi realizada com o objetivo de analisar as vantagens e desvantagens do uso do *software* Scratch para o ensino da Matemática e desenvolvimento do pensamento computacional, na educação básica, com base na avaliação de grupo professores de Matemática de Tangará da Serra-MT. Participaram do estudo 14 professores que atuam no ensino fundamental e médio. A produção dos dados, para análise desta pesquisa, ocorreu por meio de um curso de formação com carga horária total de 80 horas. Como metodologia do curso foi utilizada a Aprendizagem Baseada em Problemas (ABP). Para a análise dos dados foi utilizada a Análise Textual Discursiva (ATD), a qual permite o pesquisador movimentar-se entre os dados produzidos. O corpus da pesquisa constitui-se de: relatórios elaborados pelos cursistas ao longo do curso; questionários abertos e fechados, aplicados antes do início do curso, produções dos cursistas e; entrevistas semiestruturadas, realizadas ao término do curso. A fundamentação teórica partiu de textos sobre as relações da Matemática e do pensamento computacional com o uso do *software* Scratch. A análise foi realizada por 2 categorias, sendo elas: Características favoráveis e não favoráveis, habilidades mobilizadas da matemática e do PC com o Scratch. Os resultados nos revelam que a utilização do Scratch para o ensino da Matemática e desenvolvimento do pensamento computacional apontados pelos professores apresentam vantagens significativas para o seu uso na educação básica, com os professores projetando essa utilização. As desvantagens nos apontam alguns caminhos para que outros trabalhos possam utilizar, principalmente a questão do trabalho com iniciantes.

**Palavras-chave:** Pensamento computacional; Matemática; Scratch.

## Abstract

Computational thinking has been increasingly discussed in educational systems in many countries. Since the approval of the Common National Curriculum Base (BNCC), in 2017, the first official document on the subject, the number of studies on how to develop skills and competences related to computational thinking in the school environment has grown. In this context, a qualitative research was carried out in order to analyze the advantages and disadvantages of using Scratch *software* for teaching Mathematics and developing computational thinking in basic education, based on the evaluation of a group of Mathematics teachers from Tangará of Serra-MT. 14 teachers who work in elementary and high school participated in the study. The production of data, for analysis of this research, took place through a training course with a total workload of 80 hours. Problem-Based Learning (PBL) was used as the course methodology. For data analysis, Discursive Textual Analysis (ATD) was used, which allows the researcher to move between the data produced. The research corpus consists of: reports prepared by students throughout the course; open and closed questionnaires, applied before the start of the course, students' productions and; semi-structured interviews, carried out at the end of the course. The theoretical foundation came from texts on the relationships of Mathematics and computational thinking with the use of Scratch *software*. The analysis was performed by 3 categories, namely: Favorable and unfavorable characteristics, mobilized math and PC skills with Scratch and Productions and artifacts. The results reveal that the use of Scratch for teaching Mathematics and development of computational thinking pointed out by teachers present significant advantages for its use in basic education, with teachers designing this use. The disadvantages show us some paths for other jobs to use, especially the issue of working with beginners.

Keywords: Computational thinking; mathematics; scratch.

## ABREVIATURAS

ABP – Aprendizagem Baseada em Problemas.

ATD – Análise Textual Discursiva.

BNCC – Base Nacional Comum Curricular.

CIEB – Centro de Inovação para a Educação Brasileira.

EUA – Estados Unidos da América.

NA – Notas de Aula

OCDE – Organização para a Cooperação e Desenvolvimento Econômico.

P – Professor Cursista.

PC – Pensamento Computacional.

PCNs – Parâmetros Curriculares Nacionais

STEAM – *Science, Technology, Engineering, Arts and Mathematics* (Ciências, Tecnologia, Engenharia, Artes e Matemática).

STEM – *Science, Technology, Engineering and Mathematics* (Ciências, Tecnologia, Engenharia e Matemática).

TDICs – Tecnologias Digitais da Informação e Comunicação.

## LISTA DE FIGURAS

Figura 1 - Relação entre as áreas e seus contextos.....	25
Figura 2 - Quatro pilares do Pensamento Computacional .....	26
Figura 3 - Integração do pensamento computacional no currículo de vários países.....	31
Figura 4 – Relação das competências estabelecidas pela BNCC.....	35
Figura 5 - Tela principal do Scratch 3.6 .....	49
Figura 6 - <i>Strings</i> de busca artigos .....	55
Figura 7 - Tela de avaliação do Dr. Scratch.....	64
Figura 8 - Estrutura do Curso de Formação – Google Sala de Aula .....	72
Figura 9 - Mapa Conceitual da ATD.....	77
Figura 10 - Simulador de Isolamento Social .....	84
Figura 11 - Função exponencial – empresário Murilo realizando simulação de um caminhão novo, para observar a desvalorização desse caminhão. ....	91
Figura 12 – Frames do projeto chamado função exponencial: Figura 12 (a) apresenta números de desvalorização do caminhão durante 5 anos, enquanto a Figura 12 (b) apresenta a figura do caminhão e seu valor com taxa de desvalorização mensal e, logo em seguida, uma fórmula de função exponencial.....	92
Figura 13 - Na Figura 13 (a) é apresentado a tela inicial do jogo, a Figura 13 (b) são apresentados hiperlinks e explicações transformações matemáticas. ....	93
Figura 14 - Brincando com a simetria, parte (a) da figura representa o que foi produzido pelo usuário, e a parte (b) representa a reflexão dessa produção. ....	93
Figura 15 - Acerte o Pato – frame do jogo no momento em que o usuário estava nos critérios de divisibilidade por 5, se ele mirasse e acertasse o pato com o número 10, ganharia 1 ponto, mas, se mirasse e acertasse o pato com o número 33, perderia um ponto.....	94
Figura 16 - História do livro Homem que calculava – trabalho com frações – Nesse <i>frame</i> , o matemático fez a divisão dos 35 camelos, que os três irmãos tinham de herança, dividindo os 35 camelos deixados conforme testamento. Arredondando 17,5 para 18; 11,67 para 12, e 3,89 para 4, em que a soma resultaria em 34, sobrando um camelo para o matemático. ....	95
Figura 17 - Merilu Bica – cenário inicial do jogo e apresentação da personagem principal, a galinha Merilu.....	96
Figura 18 - Merilu Bica – galinha capturando os grilos (grilos verdes somam um ponto e grilos vermelhos perdem 2) – na parte superior esquerda, estão os pontos e o tempo para a captura do grilo. ....	96
Figura 19 - Jogo do Miko 3.1 – Miko se deslocando até as figuras geométricas que são pedidas pelos balões (comandos criados). Na parte superior esquerda, estão os pontos; e, na superior direita, o tempo.....	97
Figura 20 - Base dez e base sexagesimal – medidas: légua (a) e pés (b). ....	98
Figura 21 - Simulação do amadurecimento de frutos e nascimento de pássaros. ....	100
Figura 22 – Projeto Base dez e base sexagesimal - <i>frame</i> .....	101
Figura 23 - Diferentes objetos construídos .....	104
Figura 24 - Coordenadas no Scratch .....	106
Figura 25 - Criatividade no Scratch.....	108
Figura 26 - Interatividade do usuário com o Scratch. A Figura 28 (a) apresenta o Miko pedindo para o usuário levá-lo aos triângulos; a Figura 28 (b) pede para o usuário colocar a unidade para calcular a área e o volume; já a Figura 28 (c) informa os usuários as regras do jogo Acerte o Pato, enquanto o <i>frame</i> (d) apresenta o usuário jogando e tentando acertar o pato correto. ....	110
Figura 27 - Alguns comandos do Jogo do Miko 3.1 .....	113
Figura 28 – Projeto função exponencial – matemática, PC com programação envolvidos no trabalho – Figura 28 (a) representa a habilidade 301, a Figura 28 (b) representa a habilidade 303, a Figura 28 (c) a habilidade 304 e a Figura 28 (d) a habilidade 406 de matemática prevista na BNCC. ....	118
Figura 29 - História dos 35 camelos – representação da habilidade matemática do Quadro 12 trabalhada no projeto – Imagens extraídas do projeto digital. ....	119

Figura 30 - Jogo do Miko 3.1 – Reconhecendo figuras geométricas, imagem extraída do projeto digital. ....	120
Figura 31 - Jogo: Acerte o pato – frame: divisibilidade por 2; nele, o usuário teria de acertar os patos que são múltiplos de 2 para ganhar pontos: imagem extraída do projeto digital.....	121
Figura 32 - Brincando com a simetria – imagem inicial da simulação extraída do projeto digital .....	122
Figura 33 – Representação da simetria de reflexão extraída do projeto digital em que a Figura 33 (a) representa a produção do usuário enquanto a Figura 33 (b) representa a reflexão da produção. ....	122
Figura 34 – Merilu Bica – Figura 34 (a) cenário do jogo, captura dos grilos e a marcação de pontos e tempo, na Figura (b), são apresentados alguns comandos.....	123
Figura 35 - História da base 10 e sexagesimal – imagens extraídas do projeto digital.....	125
Figura 36 – Exemplos de decomposição do Projeto Função Exponencial na imagem (a) apresenta o ator simulando a compra de um caminhão, e a imagem (b) apresenta a análise da Tabela Fipe e os juros ao mês – imagens extraídas do projeto digital.....	129
Figura 37 – Decomposição projeto Função Exponencial – Na Figura 38 (a), Ana pergunta qual o nome do usuário; na Figura 38 (b), após os cálculos, chega-se à conclusão de que o melhor caminhão é o B, já que o caminhão A teve maior desvalorização - extraídas do projeto digital.....	130
Figura 38 - Decomposição - Função Exponencial – Figura 38 (a) cálculo da taxa de desvalorização; Figura 38 (b) apresenta a porcentagem de desvalorização mensal juntamente com a Tabela Fipe; a Figura (c) cálculo da desvalorização do caminhão em 5 anos, e a Figura 38 (d) representa a porcentagem de desvalorização mensal juntamente com a tabela Fipe .....	130
Figura 39 - Decomposição - História dos 35 camelos – tela inicial – imagens extraídas do projeto digital .....	131
Figura 40 - Decomposição - História dos 35 camelos – sincronização de comandos – imagens extraídas do projeto original .....	132
Figura 41 – Decomposição – Jogo do Miko 3.1 – imagens extraídas do projeto original. ....	133
Figura 42 – Decomposição–2 - Jogo do Miko 3.1 – Imagem (a) o Miko pede para o usuário leva-lo a figura geométrica pedida, na imagem (b) é apresentado o cenário do jogo e as figuras criadas pelos cursistas – imagens extraídas do projeto digital .....	133
Figura 43 - Decomposição–3 - Jogo do Miko 3.1 – sincronização de comandos conforme programação – imagens extraídas do projeto digital .....	134
Figura 44 - Decomposição - Jogo do Miko 3.1 – comandos para realizar a lateralidade – imagens extraídas do projeto digital .....	135
Figura 45 - Decomposição - Jogo do Miko 3.1 – comandos para pontos e direção – imagens extraídas do projeto digital .....	135
Figura 46 - Decomposição – Acerte o pato – imagem (a) apresentação do jogo por Ana, personagem principal da narração; a imagem (b) critérios de divisibilidade por 2 – imagens extraídas do projeto digital. ....	137
Figura 47 - Comandos para Ana - imagens extraídas do projeto digital.....	138
Figura 48 - Comandos arma 1 e arma 2 - imagens extraídas do projeto digital.....	138
Figura 49 - Decomposição - Comandos mira 1 e mira 2 - imagens extraídas do projeto digital .....	139
Figura 50 - Decomposição - Comandos do tiro - imagens extraídas do projeto digital .....	140
Figura 51 - Decomposição Brincando com a Simetria – tela inicial - imagens extraídas do projeto digital .....	141
Figura 52 - Transformações matemáticas - imagens extraídas do projeto digital .....	141
Figura 53 - Simulação horizontal e vertical - imagens extraídas do projeto digital.....	142
Figura 54 - Comando do cenário de apresentação - imagens extraídas do projeto digital .....	142
Figura 55 - Comandos da borboleta - imagens extraídas do projeto digital .....	143
Figura 56 - Comandos para horizontal e vertical - - imagens extraídas do projeto digital.....	144
Figura 57 - Comando Pincel - - imagens extraídas do projeto digital .....	145
Figura 58 - Limpa tela - imagens extraídas do projeto digital .....	145
Figura 59 - Cenário Inicial Merilu bica - imagens extraídas do projeto digital .....	146
Figura 60 - Setas de comandos - - imagens extraídas do projeto digital.....	147

Figura 61 - Comando da contagem de pontos – na imagem (a), se o clone do grilo verde tocar no bico amarelo da galinha, é somado um ponto; na imagem (b), se o bico amarelo da galinha tocar no clone do grilo vermelho, perde-se dois pontos – imagens extraídas do projeto digital .....	147
Figura 62 - Contagem de pontos – imagem (a) sensor para encontrar a cor correta do bico da galinha na imagem (b) - imagens extraídas do projeto digital .....	148
Figura 63 - Sala de aula e pista de atletismo - - imagens extraídas do projeto digital .....	148
Figura 64 – Sincronização de comandos - - imagens extraídas do projeto digital .....	149
Figura 65 - Encostar nas figuras geométricas – na imagem (a), ao tocar o quadrado no Miko, ganharia 10 pontos; na imagem (b), ao tocar nos círculos, ganharia 10 pontos; na imagem (c), comando do ator (quadrado) ao tocar no Miko; na imagem (d), ao encostar o Miko na cor vermelha, o usuário ganharia 10 pontos; na imagem (e), ao encostar nas figuras geométricas (cores), o usuário ganharia 10 pontos - imagens extraídas do projeto digital.....	151
Figura 66 - Acertar pato errado perderia vida - imagens extraídas do projeto digital.....	152
Figura 67 - Comando da contagem de pontos - - imagens extraídas do projeto digital .....	153
Figura 68 - Contagem de pontos – imagem (a) conta-gotas para marcar a cor correta do bico da galinha Merilu na imagem (b). Imagens extraídas do projeto digital.....	153
Figura 69 - Jogo do Miko 3.1 – imagem (a) localização e variável eixos x e y; na imagem (b), a lógica matemática em “se; então”; e, na imagem (c), o comando “espere até que tocando repita” - imagens extraídas do projeto digital. ....	155
Figura 70 - Merilu Bica – Abstração – na imagem (a), a transposição está na lógica, “se, então” “sempre”, no tempo, na contagem de pontos; já na imagem (b), temos as variáveis, blocos lógicos- imagens extraídas do projeto digital.....	156
Figura 71 - Acerte o pato – blocos de comando na imagem (a) são para criar as ondas, que servem de base para os patos saírem e para a arma atirar; já na imagem (b), temos a probabilidade no comando deslize - imagens extraídas do projeto digital .....	156
Figura 72 – Produção matemática - imagens extraídas do projeto digital .....	157
Figura 73 – Nota da história Função exponencial.....	159
Figura 74 – Nota do Problema dos 35 camelos .....	160
Figura 75 – Nota da história da Base dez e base sexagesimal .....	161
Figura 76 – Nota do jogo Acerte o pato .....	162
Figura 77 – Nota do Jogo do Miko 3.1 .....	162
Figura 78 - Comando interno com o bloco lógico "ou". .....	163
Figura 79 - Comando repita até.....	163
Figura 80 - Nota no Dr. Scratch Merilu Bica.....	164
Figura 81 - Nota da simulação Brincando com a simetria .....	165
Figura 82 - Comando clone projeto Brincando com a simetria .....	166
Figura 83 - Figura 83 -Estrutura de organização das atividades .....	184

## LISTA DE QUADROS

Quadro 1 - Dimensões e subdimensões da competência Cultura Digital.....	36
Quadro 2 - Habilidades do PC e Matemática - CIEB e BNCC.....	38
Quadro 3 - Versões do Scratch.....	48
Quadro 4- Resultados da seleção de artigos do Google Acadêmico, no período de 2017-2020.....	56
Quadro 5 - Nível de desenvolvimento para cada conceito/habilidade do pensamento computacional. .....	65
Quadro 6 - Orientações e definições dos conceitos do PC .....	65
Quadro 7- Organização do curso de formação ofertado, com as datas de cada módulo, os temas e a distribuição da carga horária das atividades síncronas (S) e assíncronas (AS).....	70
Quadro 8 - Quadro resumo dos artefatos produzidos.....	90
Quadro 9 – Função exponencial – habilidades mobilizadas.....	117
Quadro 10 – Projeto função exponencial - habilidades matemáticas da BNCC mobilizadas com o uso Scratch.....	118
Quadro 11- Problema dos camelos - habilidades matemáticas da BNCC mobilizadas com o uso Scratch.....	119
Quadro 12 - Problema dos camelos - habilidades matemáticas da BNCC mobilizadas com o uso Scratch.....	119
Quadro 13 - Jogo do Miko 3.1 - habilidades matemáticas da BNCC mobilizadas com o uso Scratch .....	120
Quadro 14 - Jogo do Miko 3.1 - habilidades matemáticas da BNCC mobilizadas com o uso Scratch .....	120
Quadro 15 - Acerte o Pato - habilidades matemáticas da BNCC mobilizadas com o uso Scratch. ...	121
Quadro 16 - Acerte o Pato - habilidade matemática da BNCC mobilizada com o uso Scratch .....	121
Quadro 17 - Brincando com a simetria - habilidades matemáticas da BNCC mobilizadas com o uso Scratch.....	123
Quadro 18 - Brincando com a simetria - habilidades matemáticas da BNCC mobilizadas com o uso Scratch.....	123
Quadro 19 - Merilu bica - habilidades matemáticas da BNCC mobilizadas com o uso Scratch .....	124
Quadro 20 - Merilu bica - habilidades matemáticas da BNCC mobilizadas com o uso Scratch .....	124
Quadro 21 - História da base 10 e sexagesimal - habilidades matemáticas da BNCC mobilizadas com o uso Scratch.....	125
Quadro 22 - História da base 10 e sexagesimal - habilidades matemáticas da BNCC mobilizadas com o uso Scratch.....	125

## SUMÁRIO

1 INTRODUÇÃO .....	15
1.1 OBJETIVO GERAL .....	17
1.2 OBJETIVOS ESPECÍFICOS .....	17
1.3 ORGANIZAÇÃO DO TRABALHO .....	17
2 PENSAMENTO COMPUTACIONAL E SUA RELAÇÃO COM A MATEMÁTICA .....	19
2.1 INTRODUÇÃO AO PENSAMENTO COMPUTACIONAL.....	19
2.2 LINGUAGEM DE PROGRAMAÇÃO LOGO .....	20
2.3 PENSAMENTO COMPUTACIONAL - DEFINIÇÕES .....	22
2.4 PILARES DO PENSAMENTO COMPUTACIONAL .....	25
2.4.1 Decomposição.....	26
2.4.2 Reconhecimento de padrões .....	27
2.4.3 Abstração .....	27
2.4.4 Algoritmo .....	27
2.5 PANORAMA DA INSERÇÃO DO PENSAMENTO COMPUTACIONAL NO CONTEXTO ESCOLAR DO BRASIL E FORA DELE.....	29
2.6 RELAÇÕES ENTRE PENSAMENTO COMPUTACIONAL E MATEMÁTICA .....	40
3. O USO DO SCRATCH PARA O ENSINO DA MATEMÁTICA E O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL .....	46
3.1 BREVE APRESENTAÇÃO DO SCRATCH .....	46
3.2 SCRATCH NO ENSINO DA MATEMÁTICA E DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL.....	50
3.3 ABORDAGENS DO DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL, ALIADO AO ENSINO DA MATEMÁTICA, UTILIZANDO O SCRATCH: UMA REVISÃO SISTEMÁTICA DA LITERATURA.....	53
3.4 AVALIANDO CONCEITOS DO PENSAMENTO COMPUTACIONAL NO SCRATCH.....	63
4. METODOLOGIA E PROCEDIMENTOS DA PESQUISA .....	67
4.1 CARACTERIZAÇÃO DA PESQUISA .....	67
4.2 PERFIL DOS SUJEITOS .....	68
4.3 A PRODUÇÃO DE DADOS.....	70
4.3.1 Questionário .....	73
4.3.2 Entrevistas.....	74
4.3.3 Artefatos .....	74
4.4 ANÁLISE DOS DADOS .....	75
5. APRESENTAÇÃO E ANÁLISE DOS DADOS.....	79
5.1. O CURSO DE FORMAÇÃO .....	79
5.1.1 Módulo 0 - Momentos que antecederam o curso .....	80
5.1.2 Módulo 1 – Início do curso - estudos teóricos sobre pensamento computacional e suas possíveis relações com a Matemática .....	80

5.1.3 Módulo 2 - Estudos teóricos sobre Scratch e suas relações com pensamento computacional e a Matemática .....	81
5.1.4 Módulo 3 e Módulo 5 – A construção de um simulador no Scratch .....	83
5.1.5 Módulo 4 – Histórias.....	86
5.1.6 Módulo 6 - Jogos.....	88
5.1.7 Módulo 7 - Jogos, Discussão sobre os Artefatos Construídos .....	89
5.1.8 Módulo 8 - Desenvolvimento do plano de aula.....	89
5.1.9 Módulo 9 - Apresentação dos objetos produzidos no Scratch .....	90
5.1.10 Módulo 10 - Avaliação e encerramento do curso .....	98
<b>5.2 CATEGORIA 1: CARACTERÍSTICAS FAVORÁVEIS E DESFAVORÁVEIS DO SCRATCH PARA O ENSINO E APRENDIZAGEM DA MATEMÁTICA E DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL .....</b>	<b>99</b>
5.2.1 Subcategoria 1: características favoráveis do Scratch .....	99
5.2.1.1 Interdisciplinaridade .....	99
5.2.1.2 Trabalho colaborativo.....	101
5.2.1.3 Produções de diferentes objetos.....	102
5.2.1.4 Ludicidade .....	104
5.2.1.5 Criatividade .....	106
5.2.1.6 Interatividade.....	108
5.2.1.7 Protagonismo .....	110
5.2.2 – Subcategoria 2: características desfavoráveis/desvantagens do Scratch .....	111
5.2.2.1 – Tempo necessário para as construções .....	111
5.2.2.2 – Pouco espaço para visualizar os comandos.....	112
5.2.2.3 – Pause para visualização .....	114
5.2.2.4 – Outras desvantagens .....	114
<b>5.3 – CATEGORIA 2: HABILIDADES MOBILIZADAS DA MATEMÁTICA E DO PC COM O USO DO SCRATCH.....</b>	<b>116</b>
5.3.1 – Subcategoria 2.1: habilidades mobilizadas da Matemática com o uso do Scratch.....	116
5.3.1.1. Projeto Função Exponencial .....	116
5.3.1.2. Problema dos 35 camelos.....	118
5.3.1.3. Jogo do Miko 3.1 .....	120
5.3.1.4. Acerte o Pato.....	121
5.3.1.5. Brincando com a simetria.....	122
5.3.1.6. Merilu Bica.....	123
5.3.1.7 História da base decimal e sexagesimal.....	124
5.3.2 Metatexto relativo à subcategoria 2.1 .....	125
5.3.3 Subcategoria 2.2: habilidades do pensamento computacional mobilizadas com o uso do Scratch .....	128
5.3.3.1 Decomposição.....	128
5.3.3.1.1 Função Exponencial.....	128

5.3.3.1.2 Problema dos 35 Camelos.....	131
5.3.3.1.3 Jogo do Miko 3.1.....	132
5.3.3.1.4. Acerte o pato.....	136
5.3.3.1.5. Brincando com a simetria.....	140
5.3.3.1.6 Merilu Bica.....	146
5.3.3.1.7 Base dez e base sexagesimal.....	148
5.3.3.2 Reconhecimento de Padrões.....	150
5.3.3.2.1 – Jogo do Miko 3.1.....	150
5.3.3.2.2 – Acerte o pato.....	151
5.3.3.2.3 Merilu Bica.....	152
5.3.3.3 – Abstração.....	153
5.3.3.3.1 – Jogo do Miko 3.1 – Merilu bica – Acerto o Pato.....	154
5.3.3.3.2 – Brincando com simetria.....	157
5.3.3.4 – Algoritmo.....	158
5.3.2.4.1 – Função exponencial.....	158
5.3.3.4.2 – Problema dos 35 camelos.....	159
5.3.3.4.3 – Base dez e base sexagesimal.....	160
5.3.3.4.4 – Acerte o Pato.....	161
5.3.3.4.5 Jogo do Miko 3.1.....	162
5.3.3.4.6 – Merilu Bica.....	163
5.3.3.4.7 – Brincando com a simetria.....	164
5.3.4 Metatexto relativo à subcategoria 2.2.....	166
6. CONSIDERAÇÕES FINAIS.....	171
7. REFERÊNCIAS BIBLIOGRÁFICAS.....	173
8. APÊNDICES.....	183
8.1 Apêndice A – Planos de Aula.....	183
8.2 Apêndice B -Excertos.....	183
8.3 Apêndice C - Questionário.....	183
8.4 Apêndice D - Aprendizagem baseada em problemas (ABP).....	183

## 1 INTRODUÇÃO

Na contemporaneidade, a utilização de tecnologias digitais vem crescendo e modificando a forma como as pessoas vivem, comunicam-se e aprendem. Dessa forma, é de se esperar que os processos de ensino e aprendizagem também sejam modificados permitindo que certas atividades, antes imagináveis, sejam possíveis de serem realizadas.

Nesse contexto, a escola não pode restringir esse processo de transformações a um mero “movimento de atualização da configuração tecnológica” (CONFORTO et al., 2018, p. 101). Sob esse prisma, os autores defendem que a escola deva ir além do ensinar seus alunos a utilizarem tecnologias, o que apenas reforça o papel de consumidores, imprimido por aqueles que detém os conhecimentos tecnológicos, mas de estabelecer condições para torná-los produtores de tecnologias e conhecimentos tecnológicos, quebrando paradigmas.

Para que isso ocorra, devemos nos questionar se o conjunto das habilidades que são desenvolvidas nas escolas é suficiente para proporcionar essa quebra de paradigma. Ainda de acordo com Conforto et al. (2018, p. 101), essa resposta é não. Segundo essas autoras, “é preciso que a ‘caixa de ferramenta’ cognitiva dos estudantes seja acrescida de uma nova habilidade, a do raciocínio ou do pensamento computacional (PC)”.

Wing (2006) também sugere que, além da leitura, escrita e aritmética, que são as habilidades básicas trabalhadas nas escolas, deveria ser incluído o pensamento computacional, como uma habilidade analítica para todas as crianças.

Para Wing (2006, p.2), “Pensamento computacional é uma habilidade fundamental para todos, não somente para cientistas da computação.

Essa necessidade de desenvolver essa nova habilidade de pensamento, que surge primeiramente para atender às necessidades de um contexto socioeconômico e cultural, impostas pelo avanço das tecnologias, chegou às escolas e têm gerado novas demandas aos professores.

Com relação ao pensamento computacional, essa demanda está atualmente explicitada na forma de diretrizes educacionais, apresentadas pela Base Nacional Curricular Comum (BNCC), que é o documento oficial que norteia os currículos das escolas de educação básica de todo Brasil, da educação infantil ao ensino médio.

A BNCC estabelece 10 competências gerais para serem desenvolvidas ao longo da educação básica e, dentre elas, está a Cultura Digital, a qual abrange três dimensões: a Computação e Programação; a Cultura e Mundo Digital; e o Pensamento Computacional (habilidade).

Assim, como já vem ocorrendo em outros países há mais tempo (BRACKMANN, 2017), o pensamento computacional passa a ser inserido no contexto educacional brasileiro, cabendo às escolas definir estratégias metodológicas para seu desenvolvimento dentro de cada Componente Curricular<sup>1</sup>. Mas será que os professores, de um modo geral, estão preparados para desempenhar essa tarefa? Será que os professores da educação básica estão preparados para desempenhar esta tarefa?

Abar e Alencar (2013) salientam que os professores, na sua maioria, têm dificuldades para dominar tecnologias e é um grande desafio inseri-las em suas práticas. Segundo os autores, os professores buscam fazer o seu melhor, mas diante de uma prática que não contempla esse conhecimento, mantêm uma estrutura controladora que repete certos hábitos.

Nesse contexto de necessidade do desenvolvimento do pensamento computacional no âmbito escolar, e de limitações quanto ao uso de tecnologias por parte de uma grande parcela de professores, chama-nos a atenção as possibilidades apresentadas pelo *software* Scratch.

Inspirado na linguagem LOGO, criada por Papert na década de 1960, o Scratch é um *software* de programação em blocos, desenvolvido pelo grupo *Lifelong Kindergarten* no *Media Lab* do *Massachusetts Institute of Technology* (MIT), liderados por Mitchel Resnick (CASTRO, KOSCIANSKI, 2017) e disponibilizado, gratuitamente, desde 2007. Sua principal característica é o uso de blocos de programação para criar histórias, jogos e animações interativas, através de uma interface gráfica bastante intuitiva e de fácil manipulação (MAJED, 2014).

Mesmo o Scratch sendo útil para as diversas áreas da educação, Ramalho e Ventura (2018) entendem que o “Scratch pode contribuir de forma mais significativa na educação Matemática, nomeadamente na capacidade de resolução de problemas, cálculo mental e a capacidade de comunicar matematicamente” (RAMALHO; VENTURA, 2018, p. 237)

---

<sup>1</sup> Pela BNCC, 2017, as disciplinas passam a se chamar componentes curriculares.

Ainda sobre a relação Matemática com o PC, Barcelos e Silveira (2012, p. 8) mencionam que “a formação de sequências numéricas é um conteúdo matemático frequentemente explorado em experiências didáticas envolvendo a identificação de padrões associada a recursos computacionais”.

Já Hu (2011) afirma que:

O pensamento computacional também é percebido em algumas pesquisas como a forma para fazemos Matemática computacionalmente. É visto, por exemplo, como um auxílio para modelar, representar e resolver problemas matemáticos e efetivamente usar a Matemática em todas as outras disciplinas (Hu, 2011, p. 224, tradução nossa).

Considerando esses aspectos, propusemos um trabalho de pesquisa para analisar as contribuições do *software* Scratch em processos de ensino da Matemática e desenvolvimento do pensamento computacional na educação básica.

### 1.1 OBJETIVO GERAL

Analisar as vantagens e desvantagens, apontadas por professores de Matemática de Tangará da Serra-MT, com relação ao uso do *software* Scratch para o ensino da Matemática e desenvolvimento do pensamento computacional na educação básica.

### 1.2 OBJETIVOS ESPECÍFICOS

Identificar as principais características do Scratch que os professores consideram pedagogicamente viáveis nos processos de ensino e aprendizagem da Matemática e do pensamento computacional;

Identificar as principais vantagens e desvantagens apresentadas pelos professores ao utilizarem o Scratch para o ensino da Matemática e desenvolvimento do Pensamento Computacional;

Evidenciar as habilidades do pensamento computacional mobilizadas pelos professores com o auxílio do *software* Scratch;

### 1.3 ORGANIZAÇÃO DO TRABALHO

Neste capítulo 1, é feita a introdução geral do trabalho, na qual contextualizamos e justificamos a pesquisa realizada, além de apresentar os objetivos perseguidos durante o trabalho de pesquisa realizado.

No capítulo 2, relataremos o pensamento computacional, a partir de uma breve introdução, retornaremos ao seu precursor que é o LOGO, apresentaremos definições do PC, os 4 pilares e os conceitos introduzidos pelo Dr. Scratch, aos quais chamamos de habilidades de programação. Ainda apresentaremos um panorama global da forma que está sendo trabalhado o PC em vários países e no Brasil, e as relações entre Matemática e pensamento computacional. Por fim, realizaremos uma revisão sistemática de literatura das relações entre Matemática, pensamento computacional e Scratch.

O capítulo 3 é destinado à apresentação de uma discussão sobre o *software* Scratch e sua relação com o ensino da Matemática e o pensamento computacional. Nele, são trazidos, à luz da discussão, diversos trabalhos publicados, os quais são considerados como referências para sustentar este trabalho de pesquisa.

No capítulo 4, apresentaremos as metodologias e os procedimentos da pesquisa. Nele, serão feitas a categorização da pesquisa, a apresentação e caracterização dos sujeitos pesquisados, a apresentação dos instrumentos de produção de dados e discutido o método de análise.

No capítulo 5, será feita a apresentação do curso de formação, especificando, módulo a módulo, as ações realizadas e o que se pretendia com cada ação. Na sequência do capítulo, apresentaremos as análises preliminares dos dados produzidos considerando duas categorias macros: Características do Scratch (vantagens e desvantagens); e habilidades mobilizadas.

Por fim, apresentaremos nossas considerações finais, com o intuito de responder ao objetivo geral da pesquisa e também vislumbrando a possibilidade de embasar outros estudos sobre o tema.

## 2 PENSAMENTO COMPUTACIONAL E SUA RELAÇÃO COM A MATEMÁTICA

### 2.1 INTRODUÇÃO AO PENSAMENTO COMPUTACIONAL

O conceito de pensamento computacional vem-se desenvolvendo desde a metade do século passado, quando os computadores começaram a surgir. No entanto, a expressão pensamento computacional surgiu em 1980, com Papert<sup>2</sup>, em seu livro *Mindstorms, Children, Computers and Powerful Ideas*, no qual relata sobre a cultura dos computadores e o papel da tecnologia nos processos de ensino com crianças (PAPERT, 1980).

Contudo, segundo Brackmann (2017, p. 26), “naquele período não houve uma mobilização para a difusão de seus princípios”. Uma definição do termo só veio a ser apresentada, em 2006, por Jeanette Wing, diretora de pesquisas computacionais do *National Science Foundation* (NSF), por meio da publicação de um artigo no periódico “*Communications of the ACM*”.

Para Wing (2006), o pensamento computacional é uma forma de investigação dos fatos para se chegar a uma solução, é aprender a planejar e se estruturar perante uma incerteza, tentando dividir o problema para que se chegue a um veredito, por meio da linguagem de programação, buscando aproximar homem e máquina no que cada um tem de melhor, que é a utilização de diferentes representações e abordagens para visualizar e analisar dados.

Segundo Carvalho, Netto e Almeida (2017), a intenção do trabalho de Jeanette Wing era ajudar profissionais de outras áreas a entenderem como o ferramental das Ciências da Computação poderia ser útil para todos, inclusive para a formação escolar. A partir do trabalho de Wing (2006), os autores definiram o pensamento computacional como sendo “a aplicação de técnicas de Ciências da Computação para

---

<sup>2</sup> Seymour Papert (Pretória, 1 de março de 1928 – Blue Hill, Maine, 31 de julho de 2016) foi um matemático e educador estadunidense nascido na África do Sul. Lecionava no Massachusetts Institute of Technology (MIT). Ele foi o teórico mais conhecido sobre o uso de computadores na educação, um dos pioneiros da inteligência artificial e criador da linguagem de programação LOGO (em 1967). Na educação, Papert cunhou o termo construcionismo como sendo a abordagem do construtivismo que permite ao educando construir o seu próprio conhecimento por intermédio de alguma ferramenta, como o computador, por exemplo.

resolução de problemas nas mais diversas áreas do conhecimento” (CARVALHO, NETTO e ALMEIDA, 2017, p. 223).

Conforme supracitado, hoje a abordagem do Pensamento Computacional é mais ampla do foi preconizada por Papert, embora o autor já tivesse previsto que a introdução de computadores na educação seria de grande abrangência. Papert entendia que “o ato de educar consiste em criar situações para que os aprendizes se engajem em atividades que alimentem este processo construtivo” (MALTEMPI, 2005, p. 3).

Assim, buscando validar suas concepções, Papert criou a linguagem de programação LOGO, voltada para ajudar os profissionais da educação a desenvolver a lateralidade e conceitos geométricos, principalmente de alunos dos anos iniciais do ensino fundamental, ao mesmo tempo em que se desenvolviam os conceitos básicos de programação computacional (PAPERT, 1980).

A iniciativa de Papert pode ser interpretada como a primeira proposta de desenvolvimento do pensamento computacional em sala de aula. Por essa razão, na seção seguinte, serão apresentados mais detalhes sobre a linguagem de programação desenvolvida por Papert.

## 2.2 LINGUAGEM DE PROGRAMAÇÃO LOGO

A linguagem de programação LOGO e o *software* de interface foram criados por Papert, na década de 60, em conjunto com outros pesquisadores do MIT (*Massachusetts Institute of Technology*), Instituto de Tecnologia de Massachusetts (VICK-VIEIRA et. al, 2017).

A interface do programa desenvolvida por Papert continha uma tartaruga centrada na origem de um sistema de coordenadas cartesianas, que ficava no centro da tela do computador e que, a partir de instruções dadas pelo programador/usuário, movia-se pela tela (sistema cartesiano) desenhando figuras geométricas. Também era possível fazer aparecer e desaparecer objetos, girar e deslocar personagens etc., com a intenção de desenvolver nas crianças um conjunto de habilidades da programação computacional.

A escolha do nome da linguagem se deu devido ao significado da palavra LOGO na língua grega, a saber: pensamento, raciocínio, discurso (SCHFFER, 2004). Segundo Urruth (2015. p. 48), “suas raízes estão na inteligência artificial, na lógica

Matemática e no psicológico”. A ideia da linguagem LOGO era favorecer, de forma simples e divertida, o desenvolvimento de um conjunto de habilidades voltadas à programação de computadores em um público infanto-juvenil.

Para Valente (1998), a linguagem de programação LOGO traz características computacionais, tais como atividades espaciais e capacidade de criar novos termos ou procedimentos, que o aprendiz pode se utilizar em várias outras atividades, criando assim o reconhecimento de padrões.

[...] no processo de comandar a tartaruga para ir de um ponto a outro, estes conceitos devem ser explicitados. Isto fornece as condições para o desenvolvimento de conceitos espaciais, numéricos, geométricos, uma vez que a criança pode exercitá-los, depurá-los e utilizá-los em diferentes situações (VALENTE, 1998, p. 19).

O LOGO também contribuiu para que o estudante aprendesse com seus erros e acertos, conforme destacam Silva e Moro (1998):

Fica demonstrado, tanto pela progressão nas categorias de verbalização sobre os ‘erros’, quanto naquelas sobre os ‘acertos’, que a linguagem Logo é uma via de construção de conhecimentos pela qual os sujeitos constroem representações positivas sobre as suas produções e sobre si mesmos (SILVA E MORO, 1998, p. 58).

Considerando o que relatam Valente (1998) e Silva e Moro (1998), é possível dizer que o LOGO contribuiu nos processos de aprendizagem, uma vez que os usuários aprenderam a realizar, de forma sistemática, uma avaliação sobre suas construções.

MIRANDA et al. (2016) corroboram essa afirmação quando relatam que:

Com a utilização da linguagem LOGO, se desenvolveu a alfabetização em informática, melhorando e auxiliando os participantes a exercitarem a organização do pensamento. Ou seja, o uso da linguagem fez com que o estudante utilizasse seu pensamento para resolver problemas, planejando e sintetizando suas ideias (MIRANDA et al., 2016, p. 10).

Mesmo sendo uma plataforma bem simples e datada da década de 60, o LOGO possui um cabedal de aplicações na iniciação à programação, principalmente ao estudante que está aprendendo as primeiras figuras geométricas, como também noções de lateralidade, ângulos, explorando conceitos de forma lúdica e possibilitando, assim, a integração das partes corporais com as intelectuais.

No Brasil, o LOGO só foi difundido durante os anos 80 e 90, por meio das Universidades Estadual de Campinas e Federal do Rio Grande do Sul (ALMEIDA, 1999). Foi, inclusive, conteúdo de cursos de Licenciaturas em Matemática na década de 2000. No entanto, com a evolução da tecnologia e o aparecimento de outras

linguagens e *softwares* educativos, como o Scratch, por exemplo, o LOGO tem sido pouco utilizado pelos profissionais da educação, embora ainda possa ser considerado um importante recurso para a introdução à linguagem de programação e o desenvolvimento de habilidades do pensamento computacional.

O LOGO pode ser considerado o precursor das plataformas de programação voltadas ao desenvolvimento do pensamento computacional, cuja evolução proporcionou o surgimento de outros sistemas de programação, como por exemplo, o de programação em blocos visuais, o qual é utilizado pelo Scratch, que é nosso objeto de pesquisa.

### 2.3 PENSAMENTO COMPUTACIONAL - DEFINIÇÕES

Como destacamos, Papert foi o proponente da linguagem LOGO e também da expressão pensamento computacional, no entanto, em seu tempo, não apresentou uma definição desse termo, que, na contemporaneidade da era da digital, é de fundamental importância.

Na verdade, a definição dessa expressão não está pacificada. Muitos autores e documentos têm buscado apresentar uma definição capaz de expressar todas as dimensões do pensamento computacional.

Se há um consenso sobre o assunto é que não tem um consenso sobre o que é o pensamento computacional, como ficou muito claro após as discussões sobre o assunto durante um *workshop* sobre PC, em 2009, promovido pela Academia Nacional de Ciências dos Estados Unidos, em que os pesquisadores chegaram ao seguinte entendimento:

Os debates realizados no workshop de fevereiro 2009 não chegaram a um acordo geral entre os participantes sobre o conteúdo preciso de pensamento computacional, e muito menos a sua estrutura. No entanto, a falta de desacordo explícito sobre seus membros poderia ser entendida como refletindo uma intuição compartilhada entre os participantes do workshop que o pensamento computacional, como um modo de pensamento, tem o seu próprio caráter distintivo (USA NATIONAL RESEARCH COUNCIL, 2010, p. 65).

Mesmo Jeanette Wing, em 2006, que tornou o termo amplamente conhecido, parece ter apresentado dificuldades de exprimir uma definição precisa, visto que, ao invés de uma definição fechada, a autora apresentou um conjunto de ideias para tentar descrever do que se tratava o pensamento computacional, abaixo segue

definições trazidas por Wing em sua obra PENSAMENTO COMPUTACIONAL – Um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar.

Pensamento computacional é pensar recursivamente. É processamento paralelo. É interpretar código como dado e dado como código. É verificação de tipo como uma generalização da análise dimensional. É reconhecer ambas as virtudes e os perigos da utilização de aliás, ou dar a alguém ou alguma coisa mais de um nome (WING, 2006, p. 2).

Pensamento computacional é usar abstração e decomposição ao atacar uma tarefa grande e complexa ou projetar um sistema complexo e grande. É a separação de interesses. É escolher uma representação apropriada para um problema ou modelagem dos aspectos relevantes de um problema para torná-lo tratável. É usar invariantes para descrever o comportamento de um sistema de forma sucinta e declarativa. É ter a confiança de que podemos usar, modificar e influenciar um sistema grande e complexo sem entender todos os seus detalhes com segurança, modularizar algo em antecipação de múltiplos usuários (WING, 2006, p. 2).

Pensamento computacional é pensar em termos de prevenção, proteção, e recuperação em cenários de pior caso através da redundância, contenção de danos e correção de erros. É chamar impasse de deadlock e contratos de interfaces. É aprender a evitar condições de corrida quando sincronizações se encontram (WING, 2006, p. 3).

Pensamento computacional é usar raciocínio heurístico na descoberta de uma solução. É planejar, aprender e agendar na presença da incerteza. É pesquisar, pesquisar e pesquisar mais, resultando em uma lista de páginas da web, uma estratégia para vencer um jogo ou um contraexemplo. Pensamento computacional é usar quantidades imensas de dados para aumentar a velocidade da computação. É fazer concessões entre tempo e espaço e entre poder de processamento e capacidade de armazenamento (WING, 2006, p. 3).

Wing ressalta a importância do PC, destacando que essa habilidade é tão importante quanto ler, escrever e aprender aritmética, segundo a autora, essa habilidade deve ser incluída na habilidade analítica de todas as crianças (WING, 2006, p. 2).

Wing (2006) nos remete ao conjunto de competências e habilidades das Ciências da Computação que podem ser desenvolvidas por alunos desde os primeiros anos na escola; a esse conjunto ela chama de pensamento computacional.

Como já dissemos anteriormente, a autora classifica a expressão pensamento computacional de várias formas, um conceito que incorpora os quatro pilares do PC é “reformular um problema aparentemente difícil em um problema que sabemos como resolver, talvez por redução, incorporação, transformação ou redução, incorporação, transformação” (WING, 2006, p. 3). Essas habilidades são de extrema relevância para entender, como também para se desenvolver o PC.

Blikstein (2008) afirma que pensamento computacional (PC) é saber usar o computador como instrumento do poder cognitivo e operacional humano, a fim de aumentar nossa produtividade, inventividade e criatividade, já que a máquina é uma estrutura enfadonha por si só, sendo o ser humano que vai moldá-la para realizar suas atividades, utilizando-a para produzir o que mais lhe interessa no momento das suas produções.

Para Barr e Stephenson (2011, p. 115), o pensamento computacional é “uma metodologia de resolução de problemas, que pode ser automatizada, transferida e aplicada de forma subjetiva”. Ainda segundo os autores, o PC “permite que todos os tipos de coisas sejam feitas: física quântica, biologia avançada, computador humano, sistemas, desenvolvimento de ferramentas computacionais úteis” (BARR e STEPHENSON, 2011, p. 115, tradução nossa).

Brackmann (2017), após estudos de vários autores, definiu pensamento computacional como:

Uma distinta capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da Computação, nas mais diversas áreas do conhecimento, com a finalidade de identificar e resolver problemas, de maneira individual ou colaborativa, através de passos claros, de tal forma que uma pessoa ou uma máquina possam executá-los eficazmente. (BRACKMANN, 2017, p. 29).

Nessa definição, pode-se observar que o autor procura abranger e compilar várias definições (supracitadas) de outros autores, mas deixa claro que “mesmo após diversos estudos e quase uma década de esforços para definir o pensamento computacional, ainda existem críticas que sugerem que não sabemos o que o pensamento computacional significa” de fato (BRACKMANN, 2017, p. 29). Contudo, ainda segundo esse autor, o PC pode ser considerado um ponto macro que engloba as áreas da Computação, Programação e Codificação, não estando necessariamente dependendo destas. Esse entendimento do autor é ilustrado na Figura 1.

Como se pode perceber, com as várias redações elaboradas por estudiosos da publicação de Jeannette Wing, ainda não existe uma definição fechada sobre o que é pensamento computacional, o que permite surgir concepções equivocadas. Nesse bojo, consideramos adequado deixar claro que nosso entendimento sobre o tema está alinhado ao da Society for Technology in Education, que definiu pensamento computacional como “uma forma de desenvolver e empregar estratégias para entender e resolver problemas de forma a aproveitar o poder dos métodos

tecnológicos para desenvolver e testar soluções” (*International Society for Technology in Education*, 2016). Tal definição se mostra alinhada à de Wing (2006) e de Conforto et al. (2018) quando consideram o PC como uma habilidade.

Figura 1 - Relação entre as áreas e seus contextos



Fonte: Brackmann (2017).

Embora ainda não haja um consenso para a definição de pensamento computacional, estudos liderados por Code.Org (2016), Liukas (2015) e BBC Learning (2015) estabeleceram que essa forma de pensamento pode ser sistematizada sobre quatro principais pilares: decomposição, reconhecimento de padrões, algoritmos e abstração. Na seção seguinte, esses pilares serão apresentados e discutidos.

## 2.4 PILARES DO PENSAMENTO COMPUTACIONAL

Lee et al. (2011) realizaram uma análise sobre o pensamento computacional para os jovens. Nesse estudo, eles sugerem que “automação” e “análise” são úteis para os estudantes se aprofundarem nos estudos do pensamento computacional.

Posteriormente, Manilla et al. (2014) realizaram outro estudo, partindo do pressuposto de que os principais conceitos do PC são a coleta de dados, a análise de dados, a representação de dados, a decomposição, os algoritmos, a abstração, a simulação, a automação e o paralelismo. O resultado deste estudo revelou que:

[...] conceitos relacionados aos dados (coleta, análise e representação) formam um cluster, enquanto decomposição de problemas, algoritmos e

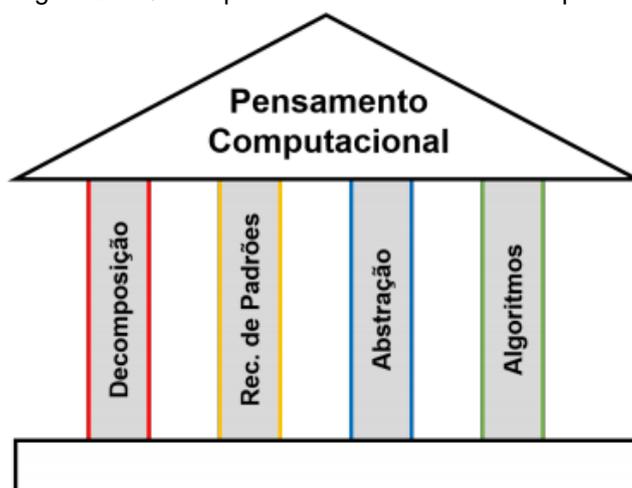
abstração formam outro, e finalmente simulação, automação e paralelização formam um terceiro grupo (MARQUES, 2019, p. 27).

Observando essa ligação entre Lee (2011) e Manilla (2014), a BBC Learning identificou um conjunto de quatro habilidades distintas e as considerou como sendo os pilares do PC, sendo elas: a decomposição, abstração, reconhecimento de padrões e algoritmo (Figura 2).

[...] o PC envolve tomar o problema complexo e dividi-lo em uma série de problemas pequenos e mais gerenciáveis (decomposição). Cada um desses problemas menores pode ser examinado individualmente, considerando como problemas semelhantes que foram resolvidos anteriormente (reconhecimento de padrões) e focalizando apenas os detalhes importantes, ignorando a informação irrelevante (abstração). Em seguida, etapas simples ou regras para resolver cada um dos problemas menores podem ser projetadas (algoritmos) (BBC LEARNING, 2015, s/p.).

Na Figura 2 são apresentados os quatro pilares, desenho apresentado por Brackmann (2017)

Figura 2 - Quatro pilares do Pensamento Computacional



Fonte: Brackmann (2017).

Para uma melhor compreensão de cada uma dessas quatro habilidades fundamentais (pilares), que são as bases do pensamento computacional (BRACKMANN, 2017), consideramos a situação-problema de realizar a troca do pneu de um carro, fato que pode acontecer com qualquer pessoa que possui um automóvel.

#### 2.4.1 Decomposição

A decomposição é o processo pelo qual os problemas são divididos em partes menores e mais fáceis de resolver. Compreende também a prática de analisar problemas a fim de identificar quais partes podem ser separadas e também de que forma podem ser reconstituídas para solução de um problema global. Essa prática também possibilita aumentar a atenção aos detalhes (CIEB, 2018).

### **2.4.2 Reconhecimento de padrões**

O reconhecimento de padrões está relacionado à identificação de características comuns entre os problemas como também em suas soluções.

“Resulta do fato de realizar a decomposição de um problema complexo para encontrar padrões entre os subproblemas gerados” (CIEB, 2018, p. 19).

Os padrões encontrados são similaridades ou têm características que alguns dos problemas têm em comum e que podem ser explorados para que sejam solucionados de forma mais eficiente (CIEB, 2018).

### **2.4.3 Abstração**

A abstração é entendida como o conceito que envolve a filtragem dos dados e sua classificação, ignorando elementos que não são necessários e visando os que são relevantes. Envolve também formas de organizar informações em estruturas que possam auxiliar na resolução de problemas (CIEB, 2018). Para o nosso trabalho vamos entender abstração como os conhecimentos matemáticos que os professores já possuem, sendo transpostos para a linguagem computacional. Analisaremos os projetos, considerando as particularidades do curso, e observando neles como foi realizada a transformação dos conhecimentos matemáticos em linguagem computacional.

### **2.4.4 Algoritmo**

Algoritmo é um conceito que agrega todos os demais, o algoritmo é um plano, uma estratégia, com um conjunto de instruções claras e necessárias para solução de um problema. Em um algoritmo as instruções são descritas e ordenadas para que o objetivo seja atingido e podem ser descritas em formato de diagramas, pseudocódigo (linguagem humana) ou descritos em códigos por meio de uma linguagem de programação (CIEB, 2018).

Para tornar mais claro o entendimento desses quatro pilares, consideramos uma situação-problema do cotidiano envolvendo a troca de um pneu furado em um carro.

Quando o motorista observa que um dos pneus de seu carro está furado, ele sabe que está diante de um problema e que precisará resolvê-lo para dar continuidade

à sua viagem. Mesmo que ele nunca tenha feito a troca de um pneu em seu carro, ele reconhece que há um padrão na resolução desse tipo de problema, o qual se aplica à grande maioria dos veículos. Reconhecer essa similaridade está relacionado ao princípio/pilar do reconhecimento de padrões. Segundo Brackmann (2017, p. 36), “O Reconhecimento de Padrões é uma forma de resolver problemas rapidamente fazendo uso de soluções previamente definidas em outros problemas e com base em experiências anteriores”.

Uma vez reconhecido esse padrão, o motorista procura visualizar o problema em partes menores a fim de facilitar sua resolução e poder se concentrar em cada uma separadamente, como, por exemplo: parar o carro em um local no acostamento da via que seja plano e firme; suspender o carro para conseguir trocar o pneu furado pelo estepe; e, colocar o triângulo de sinalização para avisar outros motoristas que ele está parado na via. Dividir o problema maior em partes menores está relacionado ao princípio da decomposição.

Agora, direcionando sua total atenção sobre um problema por vez, o motorista poderá abstrair as informações relevantes para a resolução de cada um. Por exemplo, na hora de suspender o carro para a troca do pneu, poderá se concentrar no uso adequado do “macaco”, na localização das ferramentas apropriadas, na força necessária para afrouxar e, depois, para apertar os parafusos da roda. Entretanto, deverá desconsiderar informações menos importantes, como o quanto alto o carro foi suspenso pelo “macaco” e se deve ou não colocar a calota na roda do estepe. Esse processo de filtragem dos dados/informações relevantes para a resolução do problema está relacionado ao princípio da abstração (BRACKMANN, 2017).

Uma vez pensado e analisado todo o problema, o plano de ação, com o conjunto de instruções claras, necessárias e na ordem adequada, constitui o algoritmo de resolução do problema. Assim, para o problema em questão, uma proposta de algoritmo de resolução poderia ser: 1) parar o carro; 2) abrir o porta-malas; 3) retirar triângulo de sinalização e sinalizar a via; 4) pegar chave de rodas e pneu estepe e deixar ao lado do pneu que furou; 5) posicionar o macaco no local correto da estrutura do carro; 6) afrouxar os parafusos da roda; 7) suspender o carro usando o macaco; 7) retirar os parafusos da roda; 8) retirar e guardar o pneu furado; 9) colocar o pneu estepe; 10) colocar os parafusos com o carro ainda elevado; 11) descer o carro; 12) Apertar os parafusos da roda; 13) recolher e guardar a chave de roda; e 14) Recolher e guardar o triângulo de sinalização.

Ressaltamos que, embora o algoritmo agrega, sistematiza os resultados da aplicação dos outros pilares, não se pode estabelecer uma hierarquia entre eles.

O exemplo descrito nos permite evidenciar como essa forma estrutura de pensamento (o computacional) pode e, muitas vezes, é empregada na resolução dos mais diversos problemas do dia a dia. Por esse e outros motivos, seu desenvolvimento vem sendo incentivado/integrado aos sistemas de ensino.

Na próxima seção, será discutida a integração do pensamento computacional no contexto escolar de alguns países. Para tal, um dos documentos analisados foi o relatório produzido pela Google For Education<sup>3</sup> em parceria com a Canvas<sup>8</sup><sup>4</sup>, que traz o resultado de uma pesquisa realizada em 2019, em diversos países, a qual mapeou as tendências atuais e emergentes para a educação básica no mundo.

Embora o foco desse documento não se restrinja ao educacional, já que eles também visam o atendimento das demandas de mercado, a obra é uma boa referência para mostrar a abrangência do desenvolvimento do pensamento computacional no contexto escolar. Complementarmente, utilizamos outros documentos e obras como as de Christian Puhlmann Brackmann (2017), Maria Elizabeth Bianconcini Almeida e Jose Armando Valente.

## 2.5 PANORAMA DA INSERÇÃO DO PENSAMENTO COMPUTACIONAL NO CONTEXTO ESCOLAR DO BRASIL E FORA DELE

Recentemente no Brasil, com a implementação da BNCC, como o novo documento diretriz para o sistema educacional, o pensamento computacional, pela primeira vez na história da educação brasileira, é integrado ao currículo escolar como uma habilidade a ser desenvolvida ao longo dos anos da educação básica.

Essa integração, de certa forma, busca acompanhar o que já vem sendo realizado em outros países, principalmente entre aqueles com melhores índices educacionais e maior desenvolvimento socioeconômico.

---

<sup>3</sup> É uma plataforma educacional colaborativa que possibilita às escolas, professores e estudantes extrapolarem a sua criatividade no uso da tecnologia em sala de aula. Trata-se de uma solução tecnológica desenvolvida para facilitar a vida de professores e alunos dentro e fora das salas de aula, a qualquer hora e a partir de qualquer dispositivo móvel conectado à internet.

<sup>4</sup> Canvas8 trabalha na interseção da compreensão comercial, ciências comportamentais e cultura popular. Ao desenterrar os insights por trás dos anúncios, a ciência que sustenta os fenômenos culturais e a aplicação tangível de teorias baseadas na ciência, podemos acompanhar as mudanças em todas as três áreas, a fim de compreender e rastrear melhor o comportamento humano.

Em função disso, nesta seção, será apresentado um panorama geral da inserção do pensamento computacional em sistemas educacionais de vários países, a fim de traçar um paralelo cronológico e característicos das ações realizadas nesses países em comparação com o Brasil. Consideramos como principais referências para este texto, o relatório de uma pesquisa do *Google for Education*, o qual resalta oito tendências apontadas para a educação do presente e do futuro, sendo uma delas o PC; dados da OCDE<sup>5</sup>; e os trabalhos de Christian Puhlmann Brackmann (2017) e Maria Elizabeth Bianconcini Almeida e José Armando Valente (2019).

O estudo da *Google For Education*, realizado em 2019, embora tenha, a nosso ver, um certo viés mercadológico, resultou em um relatório que aponta para as tendências emergentes da educação mundial, a saber: responsabilidade digital; pensamento computacional; salas de aula colaborativas; pedagogias inovadoras; aprendizagem liderada pelo estudante; conexão entre os pais/responsáveis pelos alunos e as escolas; e tecnologias emergentes.

No que se refere ao PC, o documento destaca que esse tema tem apoio de pais e professores, que desejam que seus estudantes utilizem esse pensamento para resolver problemas da vida cotidiana. O estudo também revelou que a maioria dos países participantes da pesquisa adota o pensamento computacional em seus sistemas educacionais.

Os países que participaram da pesquisa são: Austrália, Brasil, Canadá, Japão, México, Holanda, Nova Zelândia, Países Nórdicos, Espanha, Reino Unido e Estados Unidos.

Iniciamos este panorama mostrando as diferentes fases de inserção do PC em diferentes países da Europa.

### **Vários países**

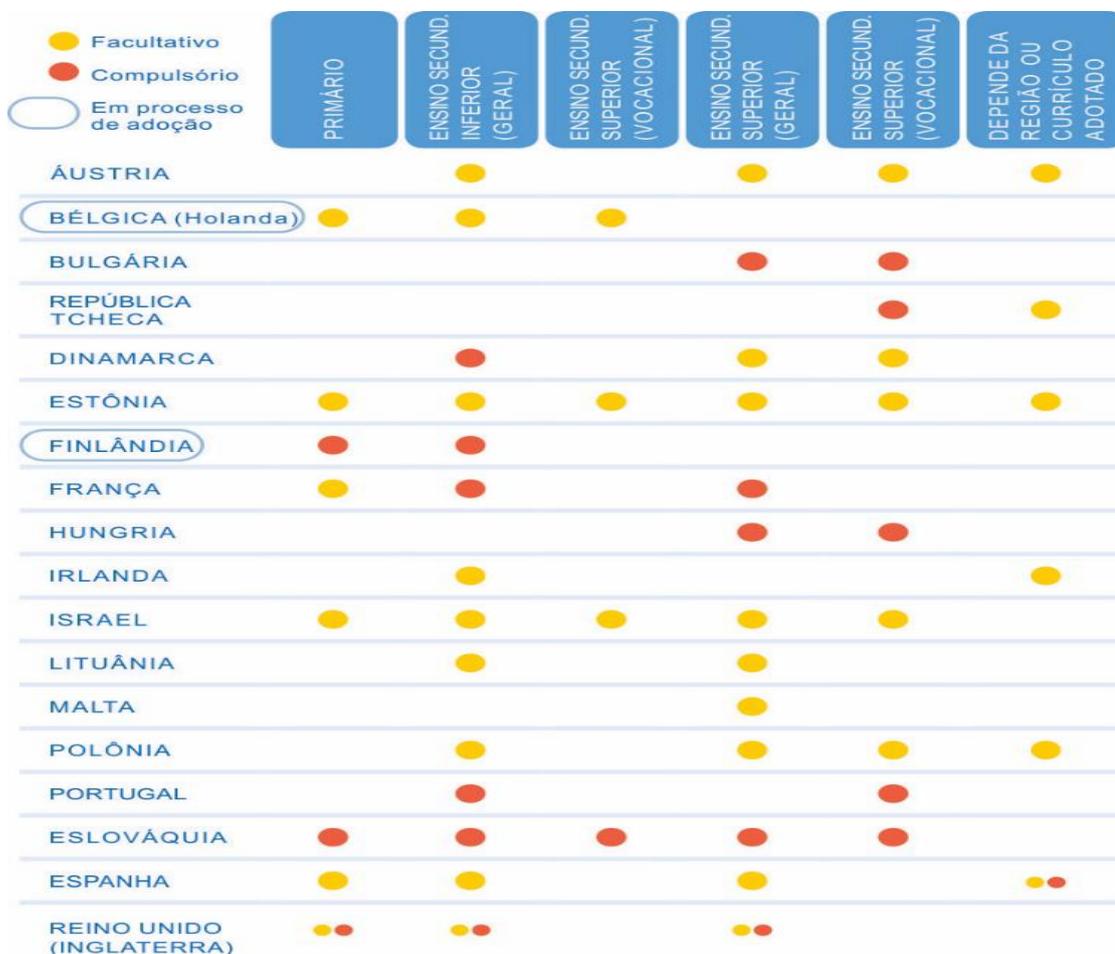
Na Figura 3, será apresentado o panorama da integração do pensamento computacional aos currículos educacionais dos vários países, com base no ano de 2015. Como se poderá notar, alguns deles já integraram o desenvolvimento do pensamento computacional, em seus currículos, desde os primeiros anos escolares, enquanto outros ainda estavam em processo de adoção. Mas, considerando que os

---

<sup>5</sup>Organização para Cooperação e Desenvolvimento Econômico. Constitui uma organização voltada ao desenvolvimento econômico e à busca do bem-estar social por meio da cooperação entre seus países-membros.

dados são de 2015, é de se esperar que a inserção do PC aos currículos desses países já deva estar em fase mais avançada.

Figura 3 - Integração do pensamento computacional no currículo de vários países.



Fonte: adaptado de Balanskat e Engelhardt (2015 apud Brackmann, 2017, p. 92).

### Inglaterra, Espanha e Portugal

Além do documento Google for Education (2019), existem outros estudos que mostram o panorama global do pensamento computacional, como o de Almeida e Valente (2019). Segundo esses autores, “a Inglaterra é o país que mais avançou nesse campo. A partir de setembro de 2014 entrou em vigor o Currículo Nacional Inglês (UK Department for Education, 2013), que implanta a disciplina de Computing” (Almeida e Valente (2019, p. 19). Eles excluíram a disciplina que focava no letramento digital e implementaram a disciplina *Computing* que foca no PC.

Na Espanha, “a comunidade autônoma de Cataluña também incorporou a programação ao seu currículo no ensino secundário obrigatório” (EUROPEAN SCHOOLNET, 2015, p. 23).

Segundo Almeida e Valente (2019), “Em 2015 a Espanha incluiu a programação no currículo como um objeto da disciplina ‘Tecnologías de la Información y la Comunicación I’ para todo o país como opcional a partir da educação secundária pós-obrigatória”, (ALMEIDA; VALENTE, 2019).

Em uma análise sobre a Espanha, são apontados alguns quesitos para o uso das TDICs (EUROPEAN SCHOOLNET, 2015), nesses estão presentes quesitos do pensamento computacional, que, mesmo corroborado por Almeida e Valente (2019), relatam que não há como afirmar que estão presentes todos os conceitos dele na proposta educacional do País.

A Espanha assim como a Inglaterra aplica o PC em âmbito nacional, desde o ano de 2015 (BRACKMANN, 2017), mas ainda está realizando essa implementação de forma optativa conforme Figura 3.

Em Portugal, o que se trabalha de forma ostensiva é o ensino de programação, mas uma programação voltada aos conceitos do pensamento computacional.

Nessa linha, o trabalho com a programação é justamente para alcançar os conceitos do pensamento computacional, e, assim, temos o PC como algo a ser atingido com a programação.

Nesse País, esse trabalho é obrigatório nos níveis de ensino secundário e superior com metas para atingir domínio das TDICs, deixando explícito que o objetivo é desenvolver o pensamento computacional (ALMEIDA e VALENTE, 2019).

Assim Almeida e Valente (2019, p. 32) nos relatam que “Desse modo, o pensamento computacional se tornou o foco de estudos no currículo desenvolvido na prática, em consonância com o previsto nas metas curriculares, que compõem o currículo forma”.

Portugal vem desenvolvendo o PC na educação básica em âmbito nacional desde o ano de 2012 (BRACKMANN, 2017) na educação básica nos anos finais do ensino fundamental e no ensino médio vocacional, conforme Figura 3.

### **Austrália**

No continente da Oceania, o documento da Google For Education destaca a Austrália. Para atender à demanda do PC na educação básica, aliada as tecnologias

digitais, no ciclo de 2016-2026 o governo australiano reservou 46 milhões de dólares por ano em tecnologia STEM<sup>6</sup> e em formação de professores, inspirando medidas voltadas à alfabetização tecnológica (GOOGLE FOR EDUCATION, 2019, p. 19).

A OCDE estima que globalmente, 92% dos empregos futuros precisarão de habilidades digitais. Segundo essa organização, o futuro dos australianos vai se concentrar em duas vertentes: conhecimento e compreensão dos componentes do sistema de informação de dados e sistemas digitais; e processos e produção de habilidades digitais - usando sistemas digitais para criar ideias, projetar, implementar e avaliar soluções digitais (MCLEAN, 2018).

Além disso, “o programa Digital Technologies in Focus, na Austrália, apoia 160 escolas desfavorecidas, fornecendo expertise em tecnologias digitais para professores e líderes escolares” (GOOGLE FOR EDUCATION, 2019, p. 19).

Com a demanda existente, o investimento do governo australiano é uma forma de potencializar o desenvolvimento do PC aliado ao uso de tecnologias digitais.

### **Estados Unidos**

Analisando o cenário da inserção do PC em alguns países do continente americano, destacamos a situação dos Estados Unidos (2012) e México (ainda não concretizada) na América do Norte, e do Brasil na América do Sul.

Nos Estados Unidos, segundo Brackmann et al. (2016),

O país não possui ainda uma legislação específica para o ensino da Computação como obrigatória, porém existem iniciativas estaduais que permitem a concretização do ensino formal, substituindo disciplinas de Matemática, Ciências, Língua Estrangeira e outras, por Computação. Até início de 2016, 54% das escolas estadunidenses já adotaram essa equivalência de disciplinas (BRACKMANN et al., 2016, p. 198).

Os Estados Unidos também estão desenvolvendo a programação fora das salas de aula e organizações como a Code.org e Google estão ajudando nesse desenvolvimento. Segundo Almeida e Valente (2019, p. 200), a Code.org “tem contribuído para a sensibilização sobre a falta de disciplinas de Ciência da Computação na Educação Básica dos EUA”, e a Google

Montou uma página de lições prontas para serem usadas em sala de aula e exemplos de programas para educadores da Educação Básica. Eles também

---

<sup>6</sup> A educação STEM é um movimento que propõe um ensino baseado em quatro disciplinas específicas [(ciências (science), tecnologia (technology), engenharia (engineer) e Matemática (mathematics)], integradas em uma abordagem interdisciplinar inovadora.

estão financiando oficinas, para professores para implantação de atividades de Ciência da Computação em todos os níveis de ensino (ALMEIDA; VALENTE, 2019, p. 220-221).

Para esse País, a OCDE estima que haverá vagas para 3,5 milhões de empregos para STEM em 2025, e que 2 milhões delas não serão preenchidas por causa da falta de trabalhadores altamente qualificados.

Essa demanda por habilidades STEM está impulsionando o governo americano a apontar suas estratégias para fazer do “pensamento computacional, um elemento integral de toda educação, incluindo construir alfabetização computacional e desenvolver e enriquecer parcerias estratégicas” (GOOGLE FOR EDUCATION, 2019 p. 11), que se concentram em trazer junto: escolas, faculdades e recursos da comunidade (como museus para construir ecossistemas STEM).

### **México**

Nesse País, o pensamento computacional (PC) tem apoio de pais e professores. Eles desejam que os alunos resolvam problemas juntamente com as habilidades digitais, para que estejam mais bem preparados para os empregos futuros (Google For Education, 2019, p. 9).

No México, 51% dos professores afirmam que preparar os alunos para o uso da tecnologia no futuro é uma das principais vantagens do uso da tecnologia em sala de aula (Google For Education, 2019, p. 8). Entretanto, 84% deles acreditam que necessitam de uma formação profissional adequada para trabalhar com as habilidades STEM, já que, mesmo com as reformas educacionais propostas pelo governo, o País não tem uma estrutura de competências para o trabalho com o ecossistema digital, no qual está composto o pensamento computacional (Google For Education, p. 9).

### **Brasil**

O documento que inseriu o pensamento computacional no contexto escolar brasileiro foi a BNCC, em 2017. Os estudos para elaboração desse documento foram iniciados em 2013, com a finalidade de estabelecer um currículo base para todos os estudantes da educação básica do País, desde a educação infantil até o ensino médio, além de implementar outras mudanças, entre elas, o desenvolvimento do pensamento computacional.

Esse documento traz, em seu bojo, as competências<sup>7</sup> e habilidades consideradas essenciais para a educação básica (BNCC, 2017). São 10 competências gerais (ver Figura 4) e um conjunto de habilidades que os alunos necessitam desenvolver durante sua vida escolar.

Figura 4 – Relação das competências estabelecidas pela BNCC.

<p><b>10 competências Gerais</b></p> <ol style="list-style-type: none"><li>1. Conhecimento.</li><li>2. Pensamento científico, crítico e criativo.</li><li>3. Repertório cultural.</li><li>4. Comunicação.</li><li>5. Cultura digital.</li><li>6. Trabalho e projeto de vida.</li><li>7. Argumentação.</li><li>8. Autoconhecimento e autocuidado.</li><li>9. Empatia e cooperação</li><li>10. Responsabilidade e cidadania</li></ol>
---

Fonte: BNCC (2017).

Essas competências, porém, não devem ser vistas como um componente curricular, mas, sim, tratadas de forma interdisciplinar, trabalhadas em todas as etapas da educação básica, perpassando todos os componentes curriculares e as unidades temáticas.

Dentre as 10 competências gerais da BNCC, está a Cultura Digital. Rosi Rico (2018), em artigo publicado na Revista Nova Escola destaca que:

Essa competência reconhece o papel fundamental da tecnologia e estabelece que o estudante deve dominar o universo digital, sendo capaz, portanto, de fazer um uso qualificado e ético das diversas ferramentas existentes e de **compreender o pensamento computacional** e os impactos da tecnologia na vida das pessoas e da sociedade (RICO, 2018, n.p, grifo nosso).

Segundo a autora, a tecnologia está muito presente na sociedade atual e trabalhar a competência da Cultura Digital na escola perpassa o desenvolvimento do pensamento computacional, alinhando-se ao que é proposto no documento produzido pelo Movimento Pela Base (2018).

---

<sup>7</sup> É considerada a mobilização de conhecimentos, habilidades, atitudes e valores para resolver demandas da vida cotidiana, do exercício da cidadania e do mundo do trabalho. Isso significa que competência é aquilo que permite aos estudantes desenvolverem plenamente cada uma das habilidades e aprendizagens essenciais estipuladas pela Base (BNCC, 2017).

Para o Movimento Pela Base (2018), a competência da Cultura Digital perpassa três dimensões, a Computação e Programação, Cultura e Mundo Digital e o Pensamento computacional. Cada uma dessas dimensões traz consigo subdimensões associadas.

Para o PC, as subdimensões são domínios de algoritmos e visualização e análise de dados, conforme apresentado no Quadro 1:

Quadro 1 - Dimensões e subdimensões da competência Cultura Digital.

<b>5ª Competência: Cultura Digital</b>	Dimensões	Subdimensões	
<p><b>O que:</b> Compreender, utilizar e criar tecnologias digitais de forma crítica, significativa e ética;</p> <p><b>Para:</b> Comunicar-se, acessar e produzir informações e conhecimentos, resolver problemas e exercer protagonismo e autoria.</p>	<b>COMPUTAÇÃO e PROGRAMAÇÃO</b>	Utilização de ferramentas digitais	Utilização de ferramentas multimídia e periféricos para aprender e produzir.
		Produção multimídia	Utilização de recursos tecnológicos para desenhar, desenvolver, publicar, testar e apresentar produtos para demonstrar conhecimento e resolver problemas.
		Linguagens de programação	Utilização de linguagens de programação para solucionar problemas.
	<b>PENSAMENTO COMPUTACIONAL</b>	<b>Domínio de algoritmos</b>	<b>Compreensão e escrita de algoritmos. Avaliação de vantagens e desvantagens de diferentes algoritmos. Utilização de classes, métodos, funções e parâmetros para dividir e resolver problemas.</b>
		<b>Visualização e análise de dados</b>	<b>Utilização de diferentes representações e abordagens para visualizar e analisar dados.</b>
	<b>CULTURA e MUNDO DIGITAL</b>	Mundo digital	Compreensão do impacto das tecnologias na vida das pessoas e na sociedade, incluindo nas relações sociais, culturais e comerciais.
Uso ético		Utilização das tecnologias, mídias e dispositivos de comunicação modernos de forma ética, comparando comportamentos adequados e inadequados.	

Fonte: Movimento pela base (2018).

Mesmo com alguns pontos controversos em torno da BNCC, é inegável que esse é o primeiro documento oficial que menciona e insere o desenvolvimento do

pensamento computacional no sistema educacional do Brasil. Contudo, não estabelece claramente o que deve ser desenvolvido em relação ao tema. Assim, documentos extraoficiais, como o supracitado, vêm realizando interpretações e propondo um conjunto de habilidades e metas para o desenvolvimento do pensamento computacional na educação básica.

Mesmo assim, o documento elaborado pelo Movimento Pela Base (2018), também não estabelece o conjunto de habilidades do pensamento computacional que devem ser desenvolvidas ao longo de cada ciclo escolar, dificultando aos profissionais da educação estabelecerem um currículo adequado para o cumprimento da BNCC no que tange a esse quesito.

Outro documento que busca suprir essa carência da BNCC sobre o tema é o Currículo de Referência em Tecnologia e Educação, produzido pelo Centro de Inovação para Educação Brasileira (CIEB), o qual especifica um conjunto de habilidades sobre pensamento computacional, contemplando a Educação Infantil, Ensino Fundamental e Ensino Médio.

Esse documento foi elaborado a partir do estudo encomendado pelo CIEB aos seus consultores, o Professor da Universidade do Vale do Itajaí (UNIVALI), Doutor André Luís Alice Raabe; o Professor do Instituto Federal Farroupilha (IFFAR), Doutor Christian Puhlmann Brackmann; e o Professor colaborador da Escola de Formação de Professores do Governo do Estado de São Paulo (EFAP) e Consultor Pedagógico do Senac São Paulo (CIEB, 2018), Doutor Flávio Rodrigues Campos.

O Currículo de Referência em Tecnologia e Educação define um conjunto de habilidades do PC, para cada ano do ensino fundamental e também as fases da educação infantil, alinhadas às competências e habilidades estabelecidas na BNCC.

Nesse documento, o pensamento computacional não é tratado como uma dimensão da competência da Cultura Digital, mas, sim, por eixos estruturantes, compreendidos de forma horizontal, não existindo hierarquia conforme estabelecido no documento do Movimento pela Base.

No documento do Movimento Todos Pela Base, a Cultura Digital é uma competência e o PC é uma dimensão dela. Nesse documento, a Cultura Digital é hierarquicamente superior ao PC. Enquanto no documento construído pelo CIEB a Cultura Digital e o PC estão num eixo estruturante.

O documento ressalta que “Os eixos estruturantes são entendidos como os grandes temas que este currículo compreende” e que “estão contemplados três eixos

estruturantes: Cultura Digital, Tecnologia e Sociedade e Pensamento computacional” (CIEB, 2018, p. 18).

No Quadro 2, será apresentado parte desse documento para destacar como está organizado por níveis de ensino e como são relacionadas as habilidades do pensamento computacional com as habilidades ligadas aos componentes curriculares na BNCC. No caso em particular, é destacada a relação com habilidades relacionadas à Matemática.

Quadro 2 - Habilidades do PC e Matemática - CIEB e BNCC

<b>Educação Infantil</b>		
<b>Habilidade CIEB</b>	<b>Prática (como Desenvolver essa habilidade?)</b>	<b>Habilidades da BNCC</b>
PCEIDE01 - Compreender o conceito de decomposição por meio de agrupamentos de brinquedos.	Identificar similaridades e diferenças entre brinquedos, por exemplo, separando os brinquedos de acordo com seu peso, tamanho, cor e formato.	EI03ET01 Estabelecer relações de comparações entre objetos, observando suas propriedades.  EI03ET05 Classificar objetos e figuras de acordo com suas semelhanças e diferenças.
<b>Ensino Fundamental Anos Iniciais</b>		
PC02AL01  Compreender o uso de repetição com número fixo de iterações.	Criando algoritmos utilizando repetições - por exemplo, criando programas simples que utilizem pelo menos uma iteração com 4 repetições, encontrando e corrigindo os seus erros e ajudando os demais colegas.	EF02MA07 - Resolver e elaborar problemas de multiplicação (por 2, 3, 4 e 5) com a ideia de adição de parcelas iguais por meio de estratégias e formas de registro pessoais, utilizando ou não suporte de imagens ou material manipulável.
<b>Ensino Fundamental Anos Finais</b>		
PC07RP01  Identificar elementos que se repetem em diferentes <i>softwares</i> e compreender a modularização ou reuso de algoritmos.	Utilizando diferentes <i>softwares</i> e reconhecendo elementos ou rotinas em comum entre eles, por exemplo, os elementos de interface de um aplicativo.	EF07MA06 - Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura podem ser obtidas utilizando os mesmos procedimentos.

Fonte: CIEB (2018, ps. 31, 43 e 87)

Até o momento, esse foi o material que encontramos disponível, que apresenta de forma sistematizada e com certo alinhamento com a BNCC, uma proposta de desenvolvimento do pensamento computacional no âmbito escolar.

No entanto, a nosso ver, esse documento tem um viés muito forte para o desenvolvimento de competências e habilidades ligadas à Computação e à Programação. Isso torna sua utilização muito difícil no âmbito escolar, principalmente considerando as carências de infraestrutura tecnológica das escolas, a falta de formação dos professores na área da computação, e mesmo por se afastar do entendimento do que é pensamento computacional, segundo alguns autores, com os quais nos alinhamos, como Wing (2006), Conforto et al. (2018) e a *International Society for Technology in Education* (2016).

A pesquisa da Google For Education (2018, p. 7) também mencionou a implementação do PC no Brasil, mas deu destaque a um programa chamado STEM Brasil<sup>8</sup>, citando: “uma estratégia que tem sido considerada eficaz pela STEM Brasil, é o uso da aprendizagem baseada em projetos”. Nesses projetos

Alunos colaboram em projetos baseados em STEM e trabalham juntos para resolver problemas, enquanto o professor desempenha o papel de facilitador. A sessão prática visa promover a compreensão dos conceitos e da teoria, o que pode ser difícil de captar (GOOGLE FOR EDUCATION, 2018, p.7).

Observamos nessa seção que vários países do mundo já iniciaram o trabalho com o PC e alguns já estão avançados como é o caso da Espanha, Inglaterra, Portugal e Eslováquia que, em algumas etapas do ensino, já o utilizam compulsoriamente, passando da fase facultativa.

Na Oceania, a Austrália já desenvolve um planejamento para 10 anos de investimento. Na América do Norte, os Estados Unidos, mesmo não possuindo uma legislação específica, já atualiza seu currículo e organizações como a Code.Org faz trabalhos além dos muros da escola. Já o México, mesmo com apoio de pais e professores, o País ainda não possui uma estrutura para que o trabalho com o PC seja executado.

---

<sup>8</sup> É um programa completo de formação de 2 anos, de alta qualidade, para professores de matemática e ciências naturais. O programa possui abordagem prática baseada em técnicas inovadoras para elevar o nível do currículo de Física, Química, Biologia e Matemática das escolas públicas, com 304 atividades práticas, aplicáveis nos três níveis de ensino brasileiros (Ensino Fundamental I, Ensino Fundamental II e Ensino Médio). Desde sua implantação no Brasil no ano de 2009, o STEM Brasil já atendeu 1.096 escolas em 21 Estados (<https://educando.org/pt/stem-brasil>)

No Brasil, entendemos que PC tem um boom nas escolas pós BNCC, reforçando que esse documento apenas cita PC, não traz habilidades do PC e nem uma estrutura mínima para implantar esse trabalho do ponto de vista teórico. Entretanto, existem parceiros que estão construindo documentos interessantes, que é o caso do Movimento Todos Pela Base, que, dentro da competência da Cultura Digital, inclui a dimensão PC.

O documento mais completo para a Educação Básica é o documento construído pelos consultores do CIEB, eles trazem o PC como eixo estruturante e utilizam habilidades do PC para se utilizar nas disciplinas da Educação Básica, da Educação Infantil ao Ensino Médio.

Na seção seguinte, apresentaremos as relações entre Matemática e pensamento computacional.

## 2.6 RELAÇÕES ENTRE PENSAMENTO COMPUTACIONAL E MATEMÁTICA

A BNCC traz alguns recortes relacionando o pensamento computacional com a Matemática, conforme se pode notar nos trechos apresentados:

Outro aspecto a ser considerado é que **a aprendizagem de Álgebra pode contribuir para o desenvolvimento do pensamento computacional** dos alunos, tendo em vista que eles precisam ser capazes de traduzir uma situação dada em outras linguagens, como transformar situações-problema, apresentadas em língua materna, em fórmulas, tabelas e gráficos e vice-versa (BRASIL, 2017, p. 227, grifo nosso).

Outra habilidade relativa à álgebra que mantém estreita relação com o **pensamento computacional** é a identificação de padrões para se estabelecer generalizações, propriedades e algoritmos (BRASIL, 2017, p. 227, grifo nosso).

Associado ao **pensamento computacional**, cumpre salientar a importância dos algoritmos e de seus fluxogramas, que podem ser objetos de estudo nas aulas de Matemática (BRASIL, 2017, p. 227, grifo nosso).

Na Seção 2.3, foi apresentado o que entendemos por pensamento computacional, a saber: “uma forma de desenvolver e empregar estratégias para entender e resolver problemas de forma a aproveitar o poder dos métodos tecnológicos para desenvolver e testar soluções” (*International Society for Technology in Education*, 2016). Para discutirmos a relação com a Matemática, se faz necessário elucidar o entendimento sobre o pensamento matemático.

Nesse sentido, deve-se considerar o que afirmam Neves, Jung e Altmann (2020). Os autores alegam que, para se entender o que é pensamento matemático,

Se faz necessária a ampliação dos significados com relação aos números e operações, [...] buscar relações existentes entre eles, desenvolvendo uma capacidade de análise, raciocínio e na tomada de decisões frente a um novo desafio [...] exploração da abstração, propiciando a descoberta de regularidade, propriedades numéricas, geométricas e métricas, fazendo com que os discentes criem condições através de estímulos, curiosidade e investigação para resolver problemas (NEVES; JUNG; ALTMANN (2020, p. 3).

Os autores nos trazem que pensamento matemático vai além de números e operações, pois, para atingir pensamento matemático se faz necessário ampliar esse entendimento, buscando a capacidade de análise e tomadas de decisão, abstração, transformando esses saberes através da curiosidade e investigação para resolver problemas e explorar o pensar para se chegar na resolução..

Considera-se também o que afirma Wielewski (2005), citando Krutetskii (1968):

Para completar sua conclusão, Krutetskii (1968) enumerou alguns componentes que não são obrigatórios na estrutura, entretanto, determinam o pensar matemático, que são:

- (1). A velocidade do processo mental (característica temporária).
- (2). Habilidades computacionais.
- (3). Uma memória para símbolos, números e fórmulas.
- (4). Uma habilidade para conceitos espaciais.
- (5). Uma habilidade para visualizar relações matemáticas abstratas e dependências. (Krutetskii 1968 apud WIELEWSKI, 2005, p. 76).

E, além disso, Krutetskii (1968, 352) apud Wielewski (2005, p. 76), define que o pensar matemático: “É caracterizado por pensamento generalizado, abreviado e flexível no reino das relações matemáticas, símbolos numéricos e literais e por uma constituição matemática da mente”.

Assim, considerando o que foi exposto pelos autores citados, caracterizamos o pensamento matemático como relação entre números e operações, formas, grandezas, propriedades, postulados, teoremas, com desenvolvimento da capacidade de análise, raciocínio e tomadas de decisões, explorando a abstração e descoberta de regularidades para resolver problemas matemáticos, inclusive os ligados às habilidades computacionais.

Considerando os entendimentos sobre pensamento computacional e pensamento matemático, se percebe que há um conjunto de relações/características que se interlaçam, podendo ser mobilizadas simultaneamente. Considerando esta simbiose, Barcelos e Silveira (2012) sugerem, como estratégia para inserção do

pensamento computacional no Ensino Básico, inseri-lo a partir de disciplinas que já existem no currículo, como, por exemplo, a Matemática. Essa estratégia também se alinha, ao que parece, com o que propõe a BNCC.

Ademais, considerando os processos matemáticos de resolução de problemas, de investigação, de desenvolvimento de projetos e da modelagem, todos elencados na BNCC, eles são potencialmente ricos para o desenvolvimento de competências para o letramento matemático e também para o desenvolvimento do pensamento computacional.

Hu (2011) concorda que o pensamento computacional está diretamente relacionado ao pensamento matemático. Para a autora, “a computação é mais matemática do que muitos pensam. Alguns aspectos da computação, similarmente na matemática, são sobre reconhecer e manipular padrões” (HU, 2011, p. 225, tradução nossa), que também são habilidades relacionadas ao pensamento computacional.

Ainda segundo Hu (2011),

O pensamento computacional é muito semelhante ao fazer matemática. Pensar matematicamente se traduz diretamente em pensar recursivamente, abstratamente, logicamente e proceduralmente - as habilidades essenciais de raciocínio para qualquer um fazer computação efetivamente (HU, 2011, p. 225, tradução nossa).

Segundo Barcelos (2015, p. 1370), “é razoável inferir que o processo de ensino-aprendizagem de Matemática seria beneficiado de alguma forma pela incorporação do pensamento computacional ao currículo do ensino básico”. O autor também destaca que a simbiose entre Matemática e PC é extremamente válida para a melhoria de índices educacionais relativos à Matemática.

Barcelos (2015) observa que o Brasil, nas avaliações internacionais, está, em média, 100 pontos atrás de países que fazem parte da OECD (*Organization for Economic Co-operation Development*), os quais, em sua maioria, já vêm desenvolvendo o PC desde a educação infantil. Dessa forma, o autor considera que os índices dos países melhor pontuados, nessas avaliações, estão ligados, de alguma forma, ao desenvolvimento do pensamento computacional no âmbito escolar desses países.

Para Koscianski et al. (2017),

O pensamento computacional (do inglês, computational thinking) engloba métodos para solução de problemas baseado nos fundamentos e técnicas da Ciência da Computação, e é visto como uma das formas de desenvolver o raciocínio lógico (KOSCIANSKI et al., 2017, p. 2).

Segundo esses autores, o PC é uma forma de desenvolver o raciocínio lógico, o qual é fundamental para o desenvolvimento do pensamento matemático. Lima e Sousa (2015) reforçam essa relação e complementam:

As escolas do ensino fundamental são veículos relevantes para implantação do uso do PC (pensamento computacional), no sentido de incentivar habilidades para desenvolvimento pleno e efetivo do raciocínio lógico-dedutível, principalmente nos primeiros anos escolares (LIMA e SOUSA, 2015, p. 1381).

Para Liukas (2015, p.10), o pensamento computacional “inclui o pensamento lógico, a habilidade de reconhecimento de padrões, raciocinar através de algoritmos, decompor e abstrair um problema”. Essas habilidades do pensamento computacional se relacionam com a Matemática, como observa a Secretaria de Educação do Estado do Paraná, que vem desenvolvendo e divulgando atividades de ensino da Matemática aliadas ao desenvolvimento computacional desplugado:

Essa atividade, além de trabalhar conceitos matemáticos [...], auxilia no desenvolvimento do pensamento lógico e [...] Trabalha com os quatro pilares do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmos (SECRETARIA DA EDUCAÇÃO DO PARANÁ, 2019 s.p).

Em foco, a abstração, uma das habilidades fundamentais (pilar) do pensamento computacional, possui estreita relação com a Matemática. Para Ribeiro, Foss e Cavalheiro (2017, p.8), “a abstração compreende dados (informações para solução do problema), processos (definição de instruções básicas para construção de algoritmos) e técnicas para criação de algoritmos (para chegar a solução de problemas)”.

Ainda segundo estes autores, em outra obra (2019),

**Em Matemática, se usam abstrações** para representar quantidades (que é um tipo de informação), que são os números. Para representar outros tipos de informação (muitas vezes bastante complexos) são necessárias abstrações adequadas. **Para construir algoritmos, também é necessário que sejam conhecidas as abstrações** que representam as operações fundamentais que nos permitem descrever processos (RIBEIRO; FOSS; CAVALHEIRO, 2019, p. 9, grifo nosso).

Barcelos (2012) considera a abstração matemática fundamental para que os alunos não decorem fórmulas ou resolvam exercícios que não instiguem o aluno a pensar. Para o autor, o aluno tem de ser capacitado pelo professor a realizar a abstração, por meio das simbologias ou outros elementos alternativos.

Valente et al. (2017) defendem que um modelo computacional é equivalente a uma abstração lógico-matemática, de forma que, para desenvolver um modelo computacional, é necessário fazer uso do pensamento matemático.

Outra relação existente entre pensamento computacional e a Matemática são os algoritmos. Silva e Silveira (2020, p.9) destacam que é possível “[...] observar a utilização dos algoritmos em atividades de PC ao trabalhar como divisões mais longas ou fatoração por exemplo”.

Barcelos (2012, p. 32) exemplifica o uso de algoritmo para resolver um problema do PC, dizendo: “um conjunto de procedimentos que deve ser utilizado, tendo como objetivo a solução do problema”. Na Matemática, também se adota um conjunto de procedimentos para resolver problemas da área, ou seja, também são construídos algoritmos.

O reconhecimento de padrões também é uma habilidade que configura tanto na Matemática quanto no PC. Barcelos e Silveira (2012) observam isso, quando relatam:

A formação de sequências numéricas é um conteúdo matemático frequentemente explorado em experiências didáticas envolvendo a **identificação de padrões** associada a recursos computacionais. Mor e Noss (2008) relatam três episódios de atividades pedagógicas, todas envolvendo a identificação das regras de formação de sequências por alunos a partir da elaboração de algoritmos em um *software* educativo. **A identificação da regularidade** na formação das sequências surge na narrativa dos alunos, e sua transformação, inicialmente em algoritmo e depois na representação matemática, parece contribuir para essa identificação (BARCELOS; SILVEIRA, 2012, p. 8, grifo nosso).

Aqui, Barcelos e Silveira (2012) observam que o reconhecimento de padrão transita entre a Matemática (sequências numéricas) e o pensamento computacional, e ainda se relaciona com o desenvolvimento de algoritmos e a abstração.

Moretti (2019) destaca como o reconhecimento de padrões é importante para a solução de problemas, seja de ordem matemática seja computacional.

Algoritmos pensados e desenvolvidos anteriormente podem ser reutilizados ou ajustados para a uma nova classe de cenários: no momento que se identifica que um dado problema pertence a essa classe pode-se aplicar a solução genérica (MORETTI, 2019, p. 26).

Como observado nas citações dos diversos autores trazidos neste texto e/ou considerando Barcelos et al. (2016), existem relações entre a Matemática e pensamento computacional. Além disso, parafraseando Barcelos (2015), a Matemática pode ser muito beneficiada pelo desenvolvimento do pensamento

computacional. Assim, estratégias e recursos que aliam o ensino da Matemática e o desenvolvimento do pensamento computacional podem ser potencializadoras das habilidades de ambas as áreas.

### 3. O USO DO SCRATCH PARA O ENSINO DA MATEMÁTICA E O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL

Neste capítulo, apresentaremos e discutiremos o uso do o *software* Scratch como um recurso pedagógico potencializador dos processos de ensino e aprendizagem da Matemática e do desenvolvimento do pensamento computacional. Ao final deste capítulo, apresentaremos também a plataforma Dr. Scratch, como uma ferramenta para avaliar habilidades do pensamento computacional relacionadas à programação.

#### 3.1 BREVE APRESENTAÇÃO DO SCRATCH

O Scratch é uma plataforma de linguagem de programação gráfica, idealizada por Mitchel Resnick<sup>9</sup> e desenvolvida e disponibilizada pelo Media Lab do MIT (Instituto de Tecnologia de Massachusetts), no ano de 2007, embora seu desenvolvimento tenha iniciado no ano de 2003 (CASTRO, 2017).

Outra curiosidade é sobre o surgimento do nome Scratch, Farias e Motta (2017) afirmam que,

O nome 'Scratch' vem da técnica utilizada pelo hip-hop e *disc jockeys* de mixar músicas com discos de vinil, movimentando-os com as mãos para frente e para trás. A Plataforma Scratch foi desenvolvida considerando essa mesma ideia de mixar, mas neste caso, ao invés de músicas, são misturadas diferentes mídias, como imagens, animações, fotos e músicas (FARIAS E MOTTA, 2017, p. 4).

A plataforma Scratch tem como precursor a linguagem de programação LOGO, criada por Papert na década de 1960, mas, diferentemente dela, o Scratch funciona com blocos de programação prontos, eximindo o usuário de possuir conhecimentos prévios de linguagens de programação para as montagens desses blocos. O que pode facilitar que um maior número de pessoas tenha contato com esse tipo de programação.

---

<sup>9</sup> Mitchel Resnick é professor de Pesquisas Educacionais do Laboratório de Mídia do MIT, ligado à Fundação LEGO, há muito tempo mantida pelo MIT. Lá, desenvolve novas tecnologias e atividades para envolver principalmente crianças nas suas experiências criativas. Criou um grupo de pesquisa que desenvolveu o Scratch, um *software*, mas, sobretudo, uma comunidade online, usada por milhões de pessoas pelo mundo afora. Esse grupo também colabora com a LEGO no desenvolvimento de novas ideias em educação e novos produtos, incluindo o LEGO Mindstorms e o WeDo, kits de robótica.

Outra vantagem do Scratch é ser totalmente gratuito. As pessoas podem criar objetos digitais de aprendizagem e colocar na plataforma online (scratch.mit.edu). Nessa plataforma, após criar uma conta, os Scratchers<sup>10</sup> podem se utilizar de outras criações já construídas e dar segmentos aos projetos, deixando claro que, ao tomar posse do objeto construído, é necessário citar o primeiro autor e também permitir as mesmas condições para outros após os seus projetos serem postados no site.

Nesse mesmo site, professores podem criar uma turma com os seus alunos e observar todas as suas criações, podendo realizar comentários e dar dicas para melhorar os projetos.

Também há a plataforma Scratch Junior, que é uma versão para smartphones e tablets, mas possui menos recursos que a versão online e off-line para computadores, essa versão é para crianças de 5 a 7 anos iniciar na programação, podendo criar suas histórias e jogos (CASTRO, 2017).

O Scratch também pode ser utilizado no modo off-line, para tal, deve ser baixado e instalado uma versão do Scratch no computador ou smartphone. Nesse caso, os usuários podem executar seus projetos sem o uso da internet, o que é viável quando há limitação/restrição desse serviço. Mesmo assim, caso o usuário deseje compartilhar seu projeto na rede, basta conectar o computador na internet e fazer o upload do seu projeto para sua conta Scratch na rede.

Todas as criações com o Scratch que são disponibilizadas em seu site possuem licença Creative Commons (CC), com licença CC do Scratch a pessoa está livre para remixar, baixar e usar comercialmente, um adendo ao uso comercial é que,

O uso comercial do Scratch, conteúdo gerado pelo usuário e materiais de suporte é permitido sob a licença Creative Commons Attribution-ShareAlike 2.0. No entanto, o Scratch Team reserva-se o direito de bloquear qualquer uso comercial do Scratch que, a critério exclusivo do Scratch Team, seja prejudicial à comunidade. O uso comercial prejudicial inclui spam ou propaganda repetida por meio de projetos, comentários ou postagens em fóruns ([https://scratch.mit.edu/terms\\_of\\_use](https://scratch.mit.edu/terms_of_use)).

Nesse sentido, a equipe do Scratch pode analisar o uso comercial e, se entender que prejudica a comunidade, Scratch pode bloquear o conteúdo.

Então, para reusar, remixar e usar comercialmente, são necessários que sejam atendidos dois critérios: a) dar crédito sem endossar a si mesmo ou ao criador original

---

<sup>10</sup> Pessoas que criam projetos no Scratch.

e b) distribuir a obra sob a mesma licença, observando que nem todas licenças CC são dessa forma (SCRATCH 2 - en. scratch-wiki.info).

Todo o conteúdo gerado pelo usuário que você envia ao Scratch é licenciado para e através do Scratch sob a licença Creative Commons Atribuição-Compartilhamento pela mesma Licença. Isso permite que outras pessoas visualizem e remixem seu conteúdo. Esta licença também permite que a Equipe Scratch exiba, distribua e reproduza seu conteúdo no site Scratch, através de canais de mídia social e em outros lugares. Se você não deseja licenciar seu conteúdo sob esta licença, então não o compartilhe no Scratch ([https://scratch.mit.edu/terms\\_of\\_use](https://scratch.mit.edu/terms_of_use)).

Ou seja, podem ser copiadas, modificadas e remixadas por qualquer outro usuário que desejar criar um projeto a partir do que está publicado. Essa característica apresenta uma vantagem importante para o aprendizado dos estudantes, já que, para um iniciante, observar um projeto pronto o ajuda a entender toda a mecânica da plataforma, e, para um usuário que já tem experiência, mas que não consegue realizar alguns comandos, pode-se utilizar outro projeto para observar o que lhe falta e alcançar o intento.

Assim, os projetos que estão no site apoiam os iniciantes e experientes e ainda podem ser baixados para estudos e modificações.

A disponibilização do Scratch em vários idiomas também é outro ponto positivo dessa plataforma. No momento da escrita, ele está disponível em quarenta idiomas, incluindo o português, o que facilita na utilização por crianças dos primeiros anos escolares. Por esse e outros motivos, a plataforma Scratch é utilizada em mais de 150 países (<http://www.scratchbrasil.net.br/>).

O Scratch pode ser encontrado em várias versões, conforme Quadro3. A versão mais moderna até essa escrita é a 3.18.1, mas esse projeto foi trabalhado na versão 3.6.0.

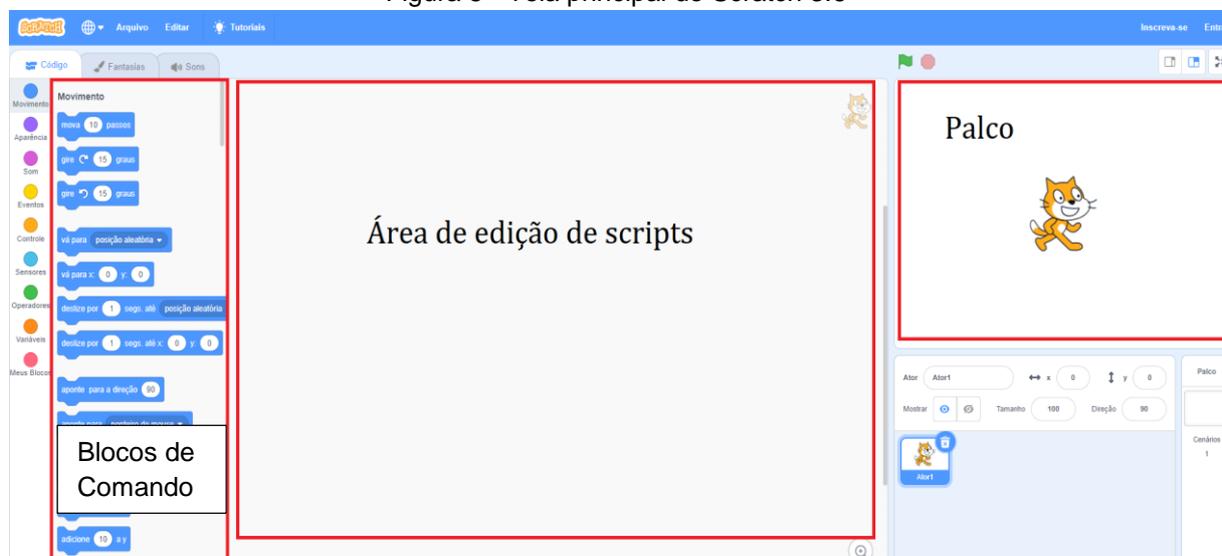
Quadro 3 - Versões do Scratch

<b>Versão</b>	<b>Ano</b>		<b>Versão</b>	<b>Ano</b>
1.4	4/8/2010		3.12.0	08/07/2020
3.6.0	10/10/2019		3.15.0	17/09/2020
3.10.0	08/05/2020		3.17.1	29/10/2020
3.11.1	24/06/2020		3.18.1	14/01/2021

Fonte: <https://scratch.br.uptodown.com/windows/versions>.

O layout do Scratch é intuitivo e permite misturar diversos tipos de mídias, possibilitando a criação de histórias, animações, simulações e jogos. Na Figura 5, é apresentada a imagem da tela principal do Scratch 3.6

Figura 5 - Tela principal do Scratch 3.6



Fonte: Horbach (2020, p. 20).

Os três retângulos em vermelho (na Figura 5) destacam as três principais áreas de trabalho. A área dos **Blocos de comando** é onde são escolhidos os blocos de comandos para criar os algoritmos. A **Área de edição de scripts** é onde os blocos escolhidos são largados para serem encaixados de acordo com o objetivo do usuário. Em outras palavras, é a área edição que possibilita a criação do projeto ou a programação de eventos (ou “scripts”). O **Palco** é onde os objetos (ou “sprites”) e cenários (ou “palcos”), escolhidos pelo usuário, vão interagir de acordo com a programação implementada.

Para se construir um objeto no Scratch, requer a escolha de comandos da linguagem de programação.

A edição de um projeto de programação no Scratch envolve elementos gráficos, alguns já estão na plataforma, como palco, atores, rotinas de ações, e podem ser alinhados com os parâmetros desejados, objetos, trajes e sons.

O Scratch é uma ferramenta que pode ser utilizada nas quatro grandes áreas do conhecimento no que tange à educação escolar, Santana e Oliveira (2019) nos ensina que “programar um computador requer habilidades como analisar problemas,

além de desenvolver e testar soluções computacionais, tais como: aplicativos e jogos digitais”.

Aqui é observado que os autores citam as habilidades que se desenvolvem ao programar, mais adiante, nessa mesma obra, eles relatam que, para desenvolver essas habilidades, utilizaram o Scratch em todas as áreas e de forma interdisciplinar no Ensino Fundamental II na Educação Básica de uma escola de Alagoas, e descrevem que

Os principais resultados obtidos mostram que a utilização da ferramenta de programação contribuiu positivamente para o desenvolvimento do trabalho colaborativo, raciocínio lógico, interatividade com o usuário, representação de dados e sincronização, bem como, enfatiza a necessidade de melhorar outras características essenciais relacionadas ao PC (paralelismo, controle de fluxo e abstração), habilidades fundamentais do XXI (SANTANA; OLIVEIRA, 2019, p. 159)

Aqui é referenciado pelos autores o uso do *software* nas diversas áreas, e o desenvolvimento das habilidades de programação e do Pensamento computacional. Na seção seguinte, discorreremos sobre a utilização do Scratch especificamente na área da Matemática e desenvolvimento do PC.

### 3.2 SCRATCH NO ENSINO DA MATEMÁTICA E DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL

Nos dias atuais, tem ficado evidente que a Matemática é essencial na vida de todos os indivíduos, desenvolvendo o pensamento lógico, o olhar crítico sobre os conceitos construídos para realizar uma boa argumentação e tirar conclusões com evidências, ser um consumidor prudente (RAMOS, 2017).

As Orientações Curriculares - Volume 2, Ciências da Natureza, Matemática e suas tecnologias para o Ensino Médio (2006), traz que, com a Matemática, é possível que os alunos resolvam

Problemas práticos do cotidiano; para modelar fenômenos em outras áreas do conhecimento; compreendam que a Matemática é uma ciência com características próprias, que se organiza via teoremas e demonstrações; percebam a Matemática como um conhecimento social e historicamente construído; saibam apreciar a importância da Matemática no desenvolvimento científico e tecnológico (BRASIL, 2006, p. 69).

D’Ambrósio (2001) afirma que:

O cotidiano está impregnado dos saberes e fazeres próprios da cultura. A todo instante, os indivíduos estão comparando, classificando, quantificando, medindo, explicando, generalizando, inferindo e, de algum modo, avaliando,

usando os instrumentos materiais e intelectuais que são próprios à sua cultura (D'AMBROSIO, 2001, p.22).

Em concordância com os autores, constatamos que a Matemática é uma ciência necessária para desenvolver as capacidades intelectuais que ajudam na estruturação dos raciocínios lógico e dedutivo, indispensáveis na resolução de problemas, tanto na vida social como no contexto escolar.

Já o pensamento computacional, conforme Wing (2006) e outros autores, é uma das competências para o século XXI (CONFORTO et al. 2018), e é potencializador de práticas que estimulam o raciocínio, além de torna-las mais significativas (CONFORTO et al. 2018).

O Scratch, por ser uma plataforma de programação, pode ser o elo entre Matemática e PC. Bussmann e Savioli (2020), parafraseando pesquisadores como Roy Pea (2010) e Ursula Wolz (2010), afirmam que:

para se aprender essa linguagem, é necessário entender de programação. Além disso, esse é um dos elementos principais para uma compreensão do Pensamento Computacional (BUSSMANN; SAVIOLI, 2020, p. 94).

Segundo os autores, para que se aprenda o PC, é necessário que se entenda de programação. Já para Omine (2014), entender a computação e a programação faz com que o indivíduo tenha uma interação melhor com os sistemas computacionais e com o entendimento desses.

Ensinar programação para quem não tem habilidades com algoritmos e com a lógica de programação não é tarefa simples, e muito menos de fácil compreensão (RESNIK et al. 2009). Por isso, quando o Scratch foi criado, o desejo do grupo de pesquisadores responsáveis era desenvolver uma plataforma de programação para pessoas que não tivessem contato com esse tipo de abordagem (RESNICK et al., 2009).

Santos e Bezerra (2017) destacam o Scratch

Como um importante ambiente de auxílio ao ensino da programação, por utilizar o conceito de desenvolvimento orientado ao design, permitindo com que mesmo usuários iniciantes (ou até mesmo, crianças) consigam criar um primeiro projeto em alguns minutos (SANTOS; BEZERRA, 2017, p. 70).

Esses autores também consideram o Scratch como um recurso importante para desenvolvimento do conhecimento matemático e do pensamento computacional, visto que, ao criarem seus projetos nesta plataforma, os usuários “vão absorvendo os conceitos básicos da computação, como iteração e condicionais, além de outros

conceitos matemáticos como coordenadas, variáveis e números aleatórios, por exemplo” (SANTOS; BEZERRA, 2017, p. 71).

Ainda nesse sentido, Wangenheim, Nunes e Santos (2014, p. 2) afirmam que: “criando programas de *software* com o Scratch, crianças aprendem a pensar criativamente, a trabalhar de forma colaborativa e a pensar de forma sistemática na solução de problemas”.

O desenvolvimento da criatividade para a solução de problemas está atrelado ao desenvolvimento do pensamento computacional.

O Scratch permite desenvolver projetos integrando diferentes elementos multimídias como sons, imagens, textos e outros, dando origem a objetos digitais como jogos, histórias, simulações e animações.

Para Horbach (2020),

A criação de jogos digitais utiliza muitos elementos, como som, efeitos visuais, regras de como o jogo funciona, a história do jogo, a programação, entre outros. Todos esses elementos juntos tem o potencial de poder integrar diferentes áreas do conhecimento, normalmente desintegradas na organização do currículo tradicional (HORBACH, 2020, p. 15).

O uso do Scratch nas simulações “oferece uma melhor visualização de diversos fenômenos, permitindo o desenvolvimento de atividades ou a criação de um mundo-do-faz-de-conta, onde certas atividades não são passíveis de serem desenvolvidas no mundo real” (HORBACH, 2020, p. 15-16).

Nesse diapasão, o usuário ainda pode aprender criando histórias no Scratch. Um professor de Matemática, por exemplo, pode explorar a história da Matemática nessa linguagem, explorando conceitos matemáticos utilizados para compor a história.

Para a produção de jogos matemáticos, o uso do Scratch mostra-se motivador, como afirmam os autores Cruz et al. (2020, p. 606) ao analisarem os resultados de sua pesquisa. Segundo eles, “o desafio da criação de jogos no Scratch entusiasmou, motivou e envolveu todos os alunos no trabalho de conceitos matemáticos, com diferente grau de envolvimento”.

Cano (2018, p. 57), destaca que: “Devido a facilidade do uso, muitos alunos conseguiram intuitivamente detectar os principais blocos de Scripts e juntá-los através da observação de como os elementos se encaixam uns nos outros”.

Dada as facilidades de manipulação da plataforma e de programação, estudantes podem iniciar a criação de códigos ao mesmo tempo que aperfeiçoam o desenvolvimento do seu pensamento computacional e também da lógica Matemática.

De acordo com Marji (2014), ao utilizar o Scratch, os usuários desenvolvem habilidades relacionadas à resolução de problemas, as quais tem sua relevância destacada não só na programação, mas em todos os âmbitos da vida, já que o *software* não está ligado somente à Matemática.

Santos e Bezerra (2017) esclarecem que

O Scratch pode auxiliar no ensino da lógica e colaborar para o desenvolvimento do pensamento computacional, bastando para isso que o aluno seja estimulado através de desafios e exercícios que utilizam os operadores lógicos, aritméticos e relacionais (SANTOS; BEZERRA, 2017, p. 97).

De acordo com os autores, os alunos, para trabalhar com a lógica utilizando o Scratch, necessitam ser estimulados por atividades desafiadoras, e, para essas atividades, deve-se buscar explorar, o máximo, os blocos de comandos que satisfaçam essas características.

Na próxima seção, faremos uma revisão sistemática da literatura, analisando abordagens que vêm sendo trabalhadas para o desenvolvimento do pensamento computacional, aliado ao ensino da Matemática, utilizando a plataforma Scratch.

### 3.3 ABORDAGENS DO DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL, ALIADO AO ENSINO DA MATEMÁTICA, UTILIZANDO O SCRATCH: UMA REVISÃO SISTEMÁTICA DA LITERATURA

Neste item, apresentaremos as abordagens do desenvolvimento do pensamento computacional. Essa abordagem estará aliada ao ensino da Matemática, utilizando o Scratch, a partir da revisão sistemática da Literatura.

Kitchenham (2004) esclarece que

Uma revisão sistemática da literatura é um meio de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para uma questão de pesquisa específica, ou área de tópico ou fenômeno de interesse. Os estudos individuais que contribuem para uma revisão sistemática são chamados de estudos primários; uma revisão sistemática é um estudo secundário (KITCHENHAM, 2004, p. 1).

Considerando esse entendimento, nesta revisão de literatura, procuramos analisar os trabalhos relevantes para evidenciar, como afirmamos, abordagens que

vêm sendo tomadas para o desenvolvimento do pensamento computacional, aliado ao ensino da Matemática, utilizando a plataforma Scratch.

O processo de levantamento bibliográfico foi guiado por um protocolo de pesquisa definido, que contém a questão problema e os critérios de inclusão e exclusão. De modo geral, esta revisão foi realizada em três fases: planejamento, identificação dos estudos primários e análise de dados.

### **Planejamento**

Na fase de planejamento foi elaborado o protocolo de pesquisa. Foi considerado para este somente a revisão de trabalhos completos do tipo artigo científico, que estivessem na Língua Portuguesa e publicados na base de dados do Google Acadêmico, no período de 2017 a 2020.

O recorte temporal levou em consideração o período dos últimos quatro anos para incluir somente estudos mais recentes sobre a temática, considerando como marco temporal, o ano de implementação da BNCC, em 2017, a qual passou a considerar o pensamento computacional como habilidade necessária para o desenvolvimento intelectual dos alunos do ensino básico.

Para o protocolo da pesquisa, consideramos trabalhos envolvendo os temas: Matemática, PC e Scratch - já que estes são os principais objetos da nossa pesquisa de mestrado.

As *strings* de busca utilizadas na pesquisa foram: “Pensamento computacional”, “Scratch”, “Matemática” e “Educação”. Utilizamos o conectivo lógico “AND” entre estas *strings*, iniciando com as duas primeiras e, depois, adicionando as demais, uma a uma, para refinar, de forma gradativa, a busca na base de dados.

Como critérios de inclusão, consideramos: artigos em Língua Portuguesa; trabalhos voltados para a Educação Básica (fundamental e médio); produções em que o PC estivesse atrelado ao ensino da Matemática e utilização do Scratch e; trabalhos nacionais (ou internacionais que tenham brasileiros como autores).

Como critérios de exclusão, adotamos: trabalhos que não sejam em Língua Portuguesa; voltados para o Ensino Superior; que não versavam, concomitantemente, sobre PC, Matemática e Scratch, e; trabalhos do tipo resumo ou revisão de literatura.

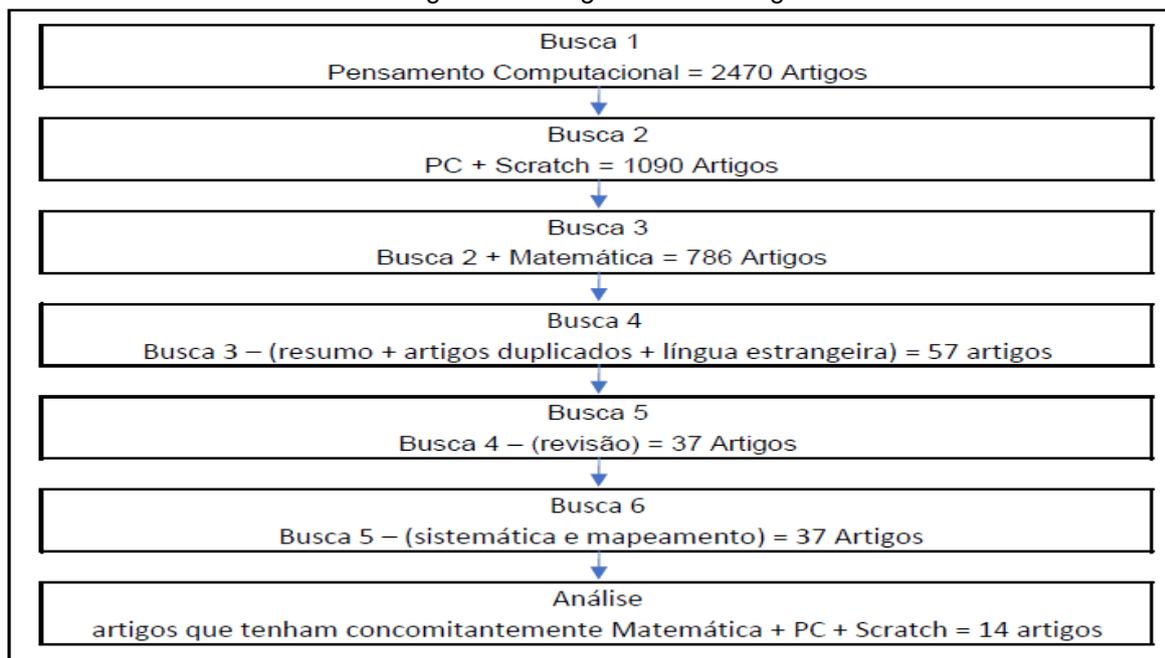
Na primeira busca na base de dados, utilizando somente a *string* “pensamento computacional”, foram encontrados 2470 trabalhos. Em seguida, com as *strings* “PC”

e "Scratch", resultaram em 1090 trabalhos e, por último, adicionando a *string* "Matemática", resultou em 786 artigos.

### Identificação dos estudos primários

Nesta fase, é feita a identificação/seleção dos estudos primários na base de dados pesquisada. Em outras palavras, é a etapa da revisão em que são selecionados os trabalhos que atendem ao protocolo de pesquisa inicialmente estabelecido.

Figura 6 - *Strings* de busca artigos



Fonte: Dos autores

Com as *strings* "resumo", "revisão", "sistemática" e "mapeamento", os resultados foram os seguintes:

"Resumo" e análise dos trabalhos duplicados ou em outra língua ficaram 57 artigos, mas, quando fizemos a exclusão da *string* "revisão" ficaram 39 artigos; com a exclusão da *string* "sistemática", sobraram 35 artigos; e, quando excluída a *string* "mapeamento", restaram 34 artigos; e, quando realizada a análise para observar se todos os trabalhos tinham concomitantemente Scratch, pensamento computacional e matemática, restaram 14 trabalhos.

Quando os critérios de inclusão e exclusão foram aplicados sobre esse conjunto de trabalhos selecionados, o quantitativo foi reduzido a 14 artigos. No fluxograma da Figura 6, são apresentadas as etapas de busca e refinamento dos trabalhos que compõem o rol de estudos primários dessa revisão.

No Quadro 4, são apresentados os estudos primários dessa revisão sistemática de literatura.

Quadro 4- Resultados da seleção de artigos do Google Acadêmico, no período de 2017-2020.

Trabalho(T)	Título	Objetivo	Autor	Ano
(T1)	Uma experiência com o Scratch no ensino das ciências e da Matemática.	Promover integração de saberes de diferentes áreas curriculares, literacia digital e o pensamento computacional	Marisa Correia e Raquel Santos	2017
(T2)	O Scratch promotor do pensamento computacional na geometria do ensino básico	Através do Programa Matemática do Ensino Básico, proporcionar o pensamento computacional, através do processo de resolução de problemas, criatividade e imaginação, privilegiando a habilidade fundamental e não mecânica.	Rui Ramalho e Ana Ventura	2018
(T3)	Recursos computacionais no ensino da matemática: aliando o pensamento computacional e as tecnologias da informação e comunicação	Apresentar desdobramento de eventos desenvolvidos, a partir de atividades exploratórias, em ambientes de ensino, com base no desenvolvimento de habilidades relacionadas com o pensamento computacional.	Gilson Pedroso dos Santos; José Ricardo e Souza Mafra	2019
(T4)	Desenvolvendo o pensamento computacional utilizando Scratch e lógica matemática.	Discutir a importância do pensamento computacional e do Scratch, ferramenta esta que possibilita a criação de um ambiente onde é possível, dentre outras coisas, aprender e ensinar sobre lógica matemática, utilizando operadores. Lógicos, aritméticos e relacionais.	Gilson Pedroso dos Santos; Ronilson dos Santos Bezerra	2017
(T5)	Reflexões sobre o ensino da Matemática e as tecnologias	Trazer uma reflexão sobre o ensino da Matemática e as tecnologias, tanto em relação ao ensino e à aprendizagem de alunos quanto em relação à formação continuada de professores, evidenciando sua importância no ensino da Matemática.	Anne Desconsi Hasselmann Bettin e Valdir Pretto	2020
(T6)	Praticando a Extensão para Promover Inclusão Digital com Computação Desplugada e Pensamento computacional.	Estimular o pensamento computacional como forma de sanar as dificuldades dos alunos de escolas públicas, principalmente em relação a compreensão de texto ou em aspectos de compreensão da matemática.	Rafaela da Silva Bobsin; Vithória Silveira Batista; Vitória de Souza Fabrício; Natália Bernardo Nunes; e Anelise Lemke Kologeski	2020

(T7)	O pensamento computacional na perspectiva da teoria dos registros de representação semiótica no ensino de geometria.	Apresentar a ferramenta e demonstrar como desenvolver atividades que trabalhem com a construção de registros figurais através das apreensões dos olhares que indiquem ângulos e lados através de comandos computacionais.	Jessica Rohden Schlickmann; Méricles Thadeu Moretti	2020
(T8)	A construção de conhecimento no ensino de lógica de programação usando o Scratch: um relato de experiência.	Relatar a experiência das oficinas do SCRATCH em uma escola pública de Santarém-PA.	Angel Pena Galvão; Juarez Benedito da Silva e Paulo Marcelo Pedroso Pereira	2019
(T9)	Clube de programação nas escolas: novas perspectivas para o ensino da computação.	Desenvolver um projeto para o ensino de programação com os alunos da educação básica, para isso a ferramenta selecionada foi o Scratch.	João Aloísio Winck; Marcelo Augusto Riske; Mariel Avozani; Adão Caron Cambraia e Marcia Fink	2018
(T10)	O ensino de programação e de robótica nos anos iniciais do ensino fundamental: uma sequência didática.	Apresentar uma pesquisa na área de tecnologia educacional que está em andamento no centro de ciências tecnológicas da universidade do estado de Santa Catarina, em Joinville/SC, vinculada ao projeto OBLABI – observatório e laboratório de práticas inovadoras em educação.	Stephanie Johansen Longo Basso e Luciane Mulazani dos Santos	2019
(T11)	Uma perspectiva de inclusão de adolescentes por meio da robótica educacional	Proporcionar experiências na área da tecnologia e da robótica, antes inexistentes, para os alunos da educação básica do município e, juntamente com isso, desenvolver o interesse dos mesmos pela área da informática como um todo.	Fabricia Araujo de Oliveira; Isys Danielly Tavares dos Santos; Janielly da Silva Santos e Marçal José de Oliveira Moraes	2018
(T12)	Robótica educacional: do conceito de robótica aplicada à concepção dos kits.	Entender a história dos kits de robótica educacional, ao passo que discutimos algumas das características físicas e lógicas desses kits e como essas características podem impactar nas salas de aula em que são utilizados.	José Pacheco de Almeida Prado e Gustavo Morceli	2018

(T13)	Tecnologias digitais para uma aprendizagem ativa e inovadora.	Utilizar as tecnologias para ajudar os estudantes a aprender e como utilizá-las de forma ativa, em modelos híbridos, para que o aluno seja protagonista.	José Moran	2018
(T14)	Gamificação na Matemática.	Realizar o levantamento e análise das atuais ferramentas para o ensino de fundamentos relacionados a tecnologias e a gamificação como prática pedagógica voltada a crianças e adolescentes, além de observarmos como são tratadas as questões de Matemática neste contexto.	Camille Braga Nader da Silva; Jan Martins Teixeira e Priscila Cardoso Petito	2019

Fonte: Elaborado pelo autor.

### **Análise dos trabalhos selecionados**

Nesta última fase, os dados retirados dos trabalhos foram sintetizados e analisados para a elaboração de metatexto. A análise foi guiada para evidenciar as principais abordagens que vêm sendo usadas para o desenvolvimento do pensamento computacional, aliado ao ensino da Matemática, utilizando a plataforma Scratch.

Analisando os estudos primários apresentados no Quadro 3, verificamos que o único trabalho realizado exclusivamente para professores (em formação) foi o de Correia e Santos (2017) (T1). As autoras desse trabalho realizaram uma experiência com licenciandos do 1º ciclo em Portugal, com o objetivo de analisar a possibilidade de se utilizar o *software* Scratch nas aulas de Matemática e de outras disciplinas dos anos iniciais da Educação Básica. Segundo elas, embora 75% dos participantes do estudo não tivessem contato prévio com o Scratch, consideraram que a realização de atividades interdisciplinares com essa plataforma de programação é motivadora para aprendizagem da Matemática e desenvolvimento do PC.

Ramalho e Ventura (2017) (T2) utilizaram o Scratch a partir do Projeto Matemática do Ensino Básico para desenvolver o PC em processos de ensino e aprendizagem da geometria. Constataram que a utilização do Scratch teve impacto positivo nesses processos.

Nesse mesmo diapasão, Santos e Mafra (2019) (T3) trabalharam o pensamento computacional no contexto escolar de Santarém-PA, com o auxílio das

TICs aplicadas ao ensino da Matemática na Educação Básica, apresentando o potencial de vários programas, dentre eles o Scratch.

Santos e Bezerra (2017) (T4) discutiram a importância do pensamento computacional e do Scratch para o ensino da Matemática. Os autores consideraram que o Scratch é um recurso que possibilita a criação de um ambiente favorável para, entre outras coisas, ensinar e aprender lógica Matemática, por meio da utilização dos operadores lógicos, aritméticos e relacionais.

Betin e Pretto (2020) (T5) procuraram demonstrar as formas de se utilizar diferentes tecnologias na educação básica, sendo uma delas o Scratch. Para os autores, o Scratch é considerado uma ferramenta de autoria, com grande potencial para a interdisciplinaridade da Matemática com outras áreas do conhecimento e pode ser usada em todos os níveis de ensino, privilegiando e auxiliando o desenvolvimento do raciocínio e do pensamento computacional na aprendizagem de programação.

Bobsin et al. (2020) (T6) procuraram estimular os estudantes de escolas públicas para o desenvolvimento do pensamento computacional como forma de sanar dificuldades, principalmente em relação à compreensão de texto e à Matemática, com atividades simples e lúdicas, utilizando o Scratch e o Code.org. Nesse trabalho, eles relataram que, mesmo Scratch necessitando de um tempo maior do que o Code.org para realizar as atividades, os alunos apresentaram uma evolução de 11,9% na aprendizagem Matemática usando o Scratch, frente aos 2,9% usando o Code.org.

No quesito gamificação, Moran (2018) (T13), numa perspectiva teórica, anunciou que os jogos e as aulas roteirizadas com a linguagem de jogos estão cada vez mais presentes nas escolas e são estratégias importantes de encantamento e motivação para uma aprendizagem mais rápida e próxima da vida real (MORAN, 2018). Segundo esse autor, atualmente, há um grande estímulo ao ensino de programação (desenvolvendo o pensamento computacional) e a aprendizagem a partir de jogos e/ou de materiais que utilizam técnicas dos jogos.

Para esse autor, os jogos de construção aberta, como o *Minecraft* construído com o Scratch, são excelentes para despertar a criatividade, a fantasia e a curiosidade, já que coloca em prática uma série de teorias que são ensinadas em Física, Matemática e Química e lhes permite pôr em prática suas ideias, transformá-las em produtos (MORAN, 2018).

Nessa linha da gamificação, Silva, Teixeira e Petito (2019) (T14) realizaram uma pesquisa com o objetivo de analisar ferramentas para a introdução das

tecnologias e a gamificação no ambiente escolar. Eles trabalharam práticas pedagógicas com crianças e adolescentes com foco no uso do lúdico no processo de ensino e aprendizagem. Eles analisaram os *softwares* Merge Cube, Google Expeditions, Code.org, quais as suas funcionalidades e usabilidade. O Scratch foi apresentado de forma mais detalhada, sendo o principal entre os quatro.

Relataram que a gamificação se mostra uma forma interessante e promissora de introduzir e rever conceitos matemáticos. Os autores destacam a importância da plataforma Scratch para introdução à lógica da criação de algoritmos e sua relação com a Matemática escolar. Para eles, o Scratch permite que os alunos sejam estimulados a desenvolver o pensamento computacional, por meio da interpretação e da elaboração de algoritmos.

O trabalho de Galvão, Silva e Pereira (2019) (T8) relata uma experiência com oficinas para uso do Scratch em uma escola pública de Santarém-PA. A proposta visava trabalhar a lógica de programação usando a plataforma Scratch. As atividades foram realizadas com 20 alunos de uma escola pública, do 9º ano do Ensino Fundamental. De acordo os autores, os alunos puderam experimentar o programa e aprender Matemática exercitando práticas computacionais na realização de desafios que lhes eram apresentados.

Já Winck et al. (2018) (T9) realizaram um trabalho com o intuito de disseminar o ensino da computação nas aulas de Matemática e desmistificar a aprendizagem de linguagens de programação por meio do Scratch. Para isso, foi escolhido um grupo de alunos que apresentava bom desempenho na disciplina de Matemática para participar das aulas, em turno inverso, com a intenção de que tivessem um tempo dedicado apenas para as atividades, para isso, utilizaram o *software* Scratch.

Os autores relatam que o trabalho com o Scratch promoveu a interdisciplinaridade da Matemática com outras disciplinas e PC, melhorando, conseqüentemente o desempenho dos alunos.

Ainda ressaltam a importância das iniciativas de tornar o ensino da computação na Matemática por meio do Scratch algo acessível e constituir essa ferramenta como um dos fatores determinantes que contribuem para a melhoria na qualidade da educação. (WINCK et al., 2018).

A robótica também configurou entre os trabalhos analisados. Johansen, Basso e Santos (2019) (T10), ao trabalharem com Matemática, PC e Scratch, visando evidenciar uma forma de trabalho com tecnologias digitais na Educação Básica,

afirmaram que perceberam que há espaço, na vida escolar, para estimular, especialmente nas crianças, os processos criativos e de lógica de programação. Segundo os autores, isso possibilitaria desenvolver a capacidade de organização de ideias e pensamento estratégico na solução de problemas.

Oliveira et al. (2018) (T11) ressaltam a importância da robótica educacional aliada à disciplina de Matemática, PC utilizando o Scratch, já que, proporciona aos alunos experiências que vão além do espaço escolar, permitindo que eles agucem o seu poder investigativo e de exploração.

Essa conexão estimula o raciocínio lógico matemático ao aplicar os conhecimentos adquiridos em sala de aula na montagem e desenvolvimento da programação de robôs utilizando o Scratch.

Prado e Morceli (2018) (T12) apresentam o Scratch como um exemplo popular de linguagem de programação visual e relatam que muitos kits de robótica adotaram o Scratch como ambiente de programação padrão (inclusive o utilizado por eles), sendo produtivo dentro da Matemática e desenvolvimento do PC.

Por fim, Schlickmann e Moretti (2020) (T7) desenvolveram uma pesquisa voltada para o desenvolvimento do pensamento computacional criando registros com representações de quadriláteros por meio da plataforma Scratch. Os registros são as diversas formas de representação de uma figura, como, por exemplo, um quadrado desenhado na lousa com giz pelo professor e outro construído no Scratch pelos alunos.

Os resultados do estudo evidenciam a semiótica entre a Matemática e o Scratch, além disso, os autores afirmam que, por meio da construção de figuras geométricas usando o Scratch, é possível propiciar aos alunos a percepção da figura com diferentes olhares no que tange ao PC.

### **Síntese da revisão sistemática**

Como é possível observar, várias abordagens vêm sendo adotadas para o desenvolvimento do pensamento computacional, aliado ao ensino da Matemática, com o uso da plataforma Scratch.

Foram encontrados trabalhos envolvendo a gamificação, a robótica, a programação, formação de professor, semiótica e os demais envolvendo a tecnologia de forma geral, todos envolvendo a ligação entre Matemática, PC e Scratch.

A robótica foi realizada com alunos em sala de aula e em laboratórios de informática, os trabalhos foram desenvolvidos por meio dos kits de robótica que as escolas possuíam.

Na robótica, o Scratch aparece como ambiente de programação padrão, aliado à Matemática e ao PC, ajuda a desenvolver a capacidade de organização de ideias e pensamento estratégico na solução de problemas, aguçando os processos criativos e de lógica de programação, desenvolvendo nos alunos o seu poder investigativo e de exploração.

A programação aconteceu em laboratórios de informática, os alunos puderam experimentar o Scratch e aprender Matemática exercitando práticas computacionais, na realização de desafios que lhes eram apresentados nas aulas de Matemática, desmistificando a aprendizagem de linguagens de programação através do Scratch.

Na gamificação, os autores salientam que os jogos, construídos com o Scratch, são excelentes para despertar a criatividade, a fantasia e a curiosidade, já que coloca em prática uma série de teorias que são ensinadas em Física, Matemática e Química; além disso, mostra uma forma interessante e promissora de introduzir e rever conceitos matemáticos.

Sobre os trabalhos de gamificação, um está no campo teórico, enquanto outro foi realizado com alunos.

A formação de professor também aparece; o trabalho acontece com futuros professores que consideraram que a realização de atividades interdisciplinares com o Scratch é motivadora para aprendizagem da Matemática e desenvolvimento do PC.

A semiótica também aparece e os autores afirmam que a construção de figuras geométricas, usando o Scratch, é possível propiciar aos alunos a percepção da figura com diferentes olhares buscando o desenvolvimento do PC. Esse trabalho ficou no campo teórico. Os demais trabalhos envolvem a tecnologia em trabalhos com alunos, os quais, em âmbito geral, aliam a educação básica ao ensino da Matemática para desenvolver o PC em processos de ensino e aprendizagem: da geometria, lógica matemática, raciocínio lógico. Alinhados ao Scratch, é considerado uma ferramenta que faz as pessoas serem autoras e protagonistas e pode ser usado em todos os níveis de ensino.

### 3.4 AVALIANDO CONCEITOS DO PENSAMENTO COMPUTACIONAL NO SCRATCH

Para avaliar habilidades de programação e, sob certos aspectos, do pensamento computacional, as quais são mobilizadas em projetos produzidos com o Scratch, foi criada uma ferramenta *on-line* e gratuita chamada Dr. Scratch.

Idealizada por Moreno-León e Robles (2015), a ferramenta oferece *feedback* aos usuários da plataforma Scratch, na forma de pontuações a um conjunto de conceitos relacionados à construção de algoritmos, os quais a plataforma identifica, automaticamente, ao analisar a estrutura da codificação/programação implementada em projetos no Scratch.

Considerando a dificuldade existente de se avaliar minuciosamente a estrutura do código implementada pelos alunos, dado o demasiado tempo necessário para essa tarefa, ou ainda pela falta de experiência com programação computacional do professor avaliador, o Dr. Scratch é uma alternativa viável para identificar e avaliar certos conceitos ligados à programação e, por conseguinte, ao pensamento computacional, quando a plataforma Scratch é utilizada para o desenvolvimento para esse fim.

O Dr. Scratch detecta maus hábitos de programação ou erros potenciais, como nomes de *sprites*<sup>11</sup> não significativos, repetição de código, código que nunca é executado e a inicialização incorreta dos atributos do objeto. Na Figura 7, é apresentada uma imagem do resultado da análise do Dr. Scratch de um artefato.

No resultado da avaliação mostrado na Figura 8, é possível observar que o Dr. Scratch avalia um total de sete habilidades/conceitos, com pontuação variando de 1 a 3. No Quadro 4, são apresentadas algumas informações que são consideradas pela ferramenta para atribuir a pontuação de cada habilidade avaliada.

Barcelos et al. (2017) ressaltam que o Dr. Scratch:

associa o nível de complexidade e frequência das estruturas de programação utilizadas a sete categorias conceituais relacionadas ao Pensamento Computacional: abstração, paralelismo, lógica, sincronização, controle de fluxo, interação com o usuário e representação de dados. A cada categoria é atribuída uma pontuação 1, 2 ou 3 em função da complexidade das estruturas e estratégias de programação utilizadas (BARCELOS et al., 2017, p. 934).

---

<sup>11</sup> Uma área de definição dos objetos

Figura 7 - Tela de avaliação do Dr. Scratch

Dr. Scratch  
Analyze your Scratch projects here!

HELP DR. SCRATCH(VERSÃO BETA)

Pontuação: 17/21 [Tweet](#)

The level of your project is...  
**MASTER!**  
Você é o mestre do universo!!!  
[Volte para o seu projecto Scratch.](#)

**Maus hábitos**

- 1 scripts duplicados.
- 0 nomeação das personagens.
- 0 nomeação de cenário.
- 1 código morto.

Level up	Nível
Lógica	2/3
Paralelismo	3/3
Interatividade com o usuário	2/3
Representação de dados	2/3
Controle de fluxo	2/3
Sincronização	3/3
Abstração	3/3

Fonte: Site do Dr. Scratch

Na página do Dr. Scratch, é possível encontrar algumas orientações sobre cada conceito para que os usuários melhorem seus projetos. O Quadro 5 sistematiza essas informações.

Quadro 5 - Nível de desenvolvimento para cada conceito/habilidade do pensamento computacional.

Conceito do Pensamento Computacional	Nível		
	Básico (1 ponto)	Em desenvolvimento (2 pontos)	Proficiente (3 pontos)
Abstração e Decomposição de problemas	Mais de um comando e mais de um ator	Definição de blocos	Uso de clones
Paralelismo	Dois comandos na bandeira verde	Dois comandos na tecla pressionada ou dois comandos a partir de um click em um mesmo ator	Dois comandos "quando recebo a mensagem", criar clone ou dois comandos "quando o cenário muda para", em paralelo
Lógica	Comando "se"	Comando "se então"	Operadores lógicos
Sincronização	Comando "espere"	Comandos: "transmissão", "quando recebe mensagem", "para tudo", "para o comando"	Comandos: "espere até", "quando o fundo mudar para", "transmita e aguarde"
Controle de fluxo	Sequência de blocos	Comando repita sempre	Comando repita até
Interatividade com o usuário	Evento disparador bandeira verde	Blocos: "tecla pressionada", "perguntar e esperar", "blocos de mouse"	Interação a partir da entrada de áudio ou vídeo
Representação de dados	Modificadores de propriedades do ator	Uso de variáveis	Uso de listas

Fonte: Moreno-León e Robles, 2015, tradução AMORIM 2020

No Quadro 6, buscamos explicitar a relação entre os pilares do pensamento computacional e as habilidades de codificação do PC, para ajudar nas análises das habilidades mobilizadas que estão nos resultados da pesquisa no Capítulo 5.

Quadro 6 - Orientações e definições dos conceitos do PC

Habilidades de Codificação	Orientações (habilidades de codificação)
<b>Lógica</b>	Podem ajudar seus projetos a se tornar dinâmicos, de modo a se comportarem de forma diferente dependendo da situação.
<b>Representação de dados</b>	Cada projeto Scratch precisa de um conjunto de informações sobre os personagens para funcionar corretamente. Por exemplo, precisamos saber a posição de cada personagem, a direção que está apontando, tamanho, etc. Além disso, programadores podem criar novos repositórios de informação para armazenar dados como o nível em que nos encontramos, o tempo decorrido, a classificação, as vidas, as recompensas coletadas.
<b>Paralelismo</b>	É a possibilidade de que várias coisas ocorram simultaneamente, ou seja, ao mesmo tempo.
<b>Interatividade com o usuário</b>	Permite que quem rode seus projetos possa criar situações novas no projeto. Por exemplo, eu poderia usar o teclado ou mouse para mover um personagem, você poderia responder perguntas ou usar a webcam, entre outras possibilidades.
<b>Sincronização</b>	A forma mais fácil de sincronizar o comportamento de seus personagens é usando um bloco 'espere', que faz o personagem

	esperar uma quantidade de segundos que você define como parâmetro do bloco
<b>Abstração</b>	A habilidade de abstrair e decompor problemas lhe ajuda a quebrar um problema em partes menores, mais fáceis de entender, a programar e depurar (corrigir os erros)
<b>Controle de Fluxo</b>	Ajuda a controlar o comportamento de seus personagens, fazendo, por exemplo, alguns blocos que são repetidos um número de vezes ou até que uma situação ocorra.

Fonte: Site Dr. SCRATCH, 2020.

## 4. METODOLOGIA E PROCEDIMENTOS DA PESQUISA

### 4.1 CARACTERIZAÇÃO DA PESQUISA

A abordagem metodológica adotada para esta pesquisa foi a qualitativa. A escolha dessa metodologia se deu pelo fato de ela possibilitar compreender os aspectos subjetivos do fenômeno estudado, ou seja, possibilitar responder como os professores de Matemática, de Tangará da Serra-MT, avaliam (em termos de potencialidades e dificuldades) a utilização do *software* Scratch para o ensino da Matemática e para o desenvolvimento do pensamento computacional?

A pesquisa é de cunho interpretativo, valendo-se da observação e de alguns instrumentos de coleta para alcançar os intentos pretendidos. Esse fato também corroborou a escolha da abordagem qualitativa, visto que se pretendeu buscar não só a aparência do fenômeno, mas sua essência.

Segundo Triviños (1987, p. 131), “na pesquisa qualitativa de forma muito geral, segue-se a mesma rota ao realizar uma investigação. Isto é, existe uma escolha de um assunto ou problema, uma coleta e análise das informações”.

Na visão de Denzin e Lincoln (2006), quando evocamos a terminologia qualitativa, a palavra qualitativa implica uma ênfase sobre as qualidades das entidades e sobre os processos, os quais não têm como serem aferidos, não tem como medi-los experimentalmente no que concerne às quantidades, tais como, volume, intensidade ou frequência, para eles a pesquisa qualitativa enfatiza o processo.

Nesse entendimento, a validade da pesquisa não se dá pelo tamanho da amostra, como na pesquisa quantitativa, mas, sim, pela profundidade com que o estudo é realizado. Assim, a escolha pela pesquisa qualitativa se sustenta pelo fato de que proporciona uma avaliação próxima de um resultado ao tema da pesquisa em tela, em relação à análise e à coleta de dados.

Diferentemente dos estudos quantitativos que enfatizam o ato de medir e analisar as relações causais entre variáveis, a pesquisa qualitativa não faz essas análises. Marconi e Lakatos (2011, p. 274) esclarecem que a observação qualitativa “implica em conhecer e aprofundar situações sociais mantendo uma reflexão contínua e observando detalhes dos sucessos, dos eventos e das interações”.

Esse tipo de pesquisa também consiste num processo de interiorização e aprofundamento das interações homem-tecnologia digital (Scratch).

Para contemplar os objetivos da nossa pesquisa, foram: 1) identificadas as principais características do Scratch que os professores consideraram pedagogicamente viáveis para os processos de ensino e aprendizagem da Matemática e do pensamento computacional; 2) evidenciadas as principais vantagens e desvantagens que os professores apontaram ao utilizarem o Scratch, na perspectiva do ensino da Matemática e desenvolvimento do pensamento computacional, e, por fim; 3) reveladas as habilidades do pensamento computacional e da Matemática, que foram mobilizadas pelos professores com o auxílio do Scratch.

#### 4.2 PERFIL DOS SUJEITOS

Os sujeitos da pesquisa foram 14 professores de Matemática, que trabalhavam com alunos do sexto ano ao ensino médio, da Educação Básica, no município de Tangará da Serra – MT..

Para fins didáticos, os cursistas serão identificados pelo codinome P. Como foram 14 cursistas, o codinome ocorrerá do P1 ao P14.

Em 2020, esses professores estavam alocados da seguinte forma: 2 trabalhavam exclusivamente no ensino médio (P2 e P5); 3 deles no ensino médio, EJA e fundamental (P7, P8 e P14); e 9 deles, exclusivamente, no ensino fundamental, com turmas do sexto ao nono ano (P1, P3, P4, P6, P9, P10, P11, P12 e P13).

Ressaltamos que todos já trabalharam, em algum momento da carreira, em ambas as etapas (fundamental e médio) da educação básica. Além disso, 10 professores trabalham em escolas situadas na zona urbana e 4 em escolas do campo (P7, P9, P11 e P13).

Sobre a formação dos 14 professores sujeitos da pesquisa, apenas um deles possui uma segunda graduação, todos os demais só possuem a Licenciatura em Matemática, sendo 1 graduado, 11 especialistas, 1 mestrando e 1 mestre.

Quanto ao tempo de carreira, 4 professores declararam ter entre 0 e 5 anos; metade (7) declarou ter de 5 a 10 anos de docência; e 3 professores, entre 10 e 15 anos, no trabalho com docência.

No que tange ao vínculo funcional, 8 (57%) são professores efetivos e 6 (43%) contratados. Esses números foram arredondados para a análise nessa seção.

Dessa forma, pelos anos trabalhados, pela formação acadêmica, observamos que a maioria dos professores tem experiência na docência e esse fato nos ajudou em vários aspectos de condução do Curso, pois, com essa bagagem, os professores pretendiam melhorar sua prática pedagógica e também puderam colaborar no andamento do Curso.

Quanto ao nível de letramento digital, 10 professores (71%) se declararam possuir nível básico, enquanto somente 4 deles (29%) declararam ser proficientes. Ou seja, os sujeitos desta pesquisa possuíam um nível básico de letramento digital.

Na avaliação da formação acadêmica para o uso de tecnologias digitais na educação, 43% (6) entenderam que é boa; 43% (6) regular, e 14% (2), péssima.

Nessa análise, observamos que os professores possuem uma formação acadêmica entre boa e regular e isso ajudou a desenvolver o Projeto, e, a partir dessa análise, buscamos, nos atendimentos com o tutor, dar ênfase aos cursistas que disseram que a formação era péssima.

Quando perguntados sobre o número de cursos de formação continuada voltados para o uso de tecnologias digitais na educação, 9 deles (64%) disseram que são insuficientes e 5 (36%) disseram que os cursos são suficientes.

Como a maioria entende que esse número é insuficiente, entendemos que os professores necessitam de mais cursos para o uso das tecnologias digitais..

No quesito de aparelhos ou tecnologias digitais: 8 professores disseram sempre usar celular e 3 nunca o utilizaram; 3 já o utilizaram alguma vez. Sobre *smartphones*: 7 sempre, 2 já utilizaram; e 5 nunca. Quanto aos *tablets*: 10 nunca o utilizaram; 3 já alguma vez e 1 sempre o utilizou. Internet todos já a utilizaram e *notebook* também.

O aparelho mais utilizado é o *notebook*, por apresentar maior facilidade de trabalhar com ele, inclusive no Curso todos trabalharam com notebook, já que o celular ou *smartphone* tende a ter uma tela menor, o que pode prejudicar no momento do trabalho, mas, mesmo assim, a pesquisa revelou que, dos professores observados, um grande número usa celular ou *smartphone*.

Nas respostas sobre pensamento computacional, 1 disse ter tido contato no curso de graduação, 6 tiveram contato nos cursos de formação complementar, e 7 disseram que já fizeram leituras sobre esse tema.

Quando perguntados se sabiam que a BNCC traz dentro da competência Cultura digital a expressão “pensamento computacional”, 85% disseram que sim, e

15%, não; e também 85% responderam que o PC tem de ser trabalhado por profissionais de todas as áreas; 15% disseram somente por Matemática; todos disseram que o PC pode contribuir com o processo de ensino e aprendizagem.

No quesito PC, a maioria disse que já teve contato, o que nos ajudou na parte teórica e no entendimento por parte deles, pois iniciar o curso com um tema tão complexo como o pensamento computacional poderia ser um fator complicador, já que o Curso estava pautado somente em dois momentos exclusivamente teóricos.

Eles não haviam se aprofundado sobre o tema em outras formações, mas entendiam o que era, e, nas pós-leituras, os professores realizaram substancial discussão, e isso favoreceu o Curso, pois de Matemática e ensino eles tinham experiências; PC eles entenderam o movimento, portanto, só faltava trabalhar com o *software* para analisar se esse poderia ser um potencializador dos outros dois.

#### 4.3 A PRODUÇÃO DE DADOS

A produção dos dados ocorreu por meio de um curso de formação continuada constituído de 10 módulos, ofertado de forma remota com carga horária total de 80 horas, sendo 30 horas de atividades síncronas e 50 de atividades assíncronas. No Quadro 7, é apresentada uma descrição dos módulos do curso e a distribuição da carga horária.

Quadro 7- Organização do curso de formação ofertado, com as datas de cada módulo, os temas e a distribuição da carga horária das atividades síncronas (S) e assíncronas (AS).

Formação	Dia	Estudos	Carga Horária
1º Encontro	29/05/2020	Estudos teóricos sobre pensamento computacional – Leituras de artigos, vídeos e livros sobre o tema	10 horas assíncronas + 3 horas síncronas:  <b>Total de 13 horas</b>
2º Encontro	03/06/2020	Uso do Scratch no Ensino da Matemática e no Desenvolvimento do Pensamento computacional – Estudos Teóricos	7 horas assíncronas + 3 horas síncronas :  <b>Total de 10 horas</b>
3º Encontro	08/06/2020	Iniciação ao <i>software</i> Scratch: Construindo Simuladores com os professores de Matemática de Tangará da Serra.	3 horas assíncronas + 3 horas Síncronas :  <b>Total de 6 horas</b>
4º Encontro	15/06/2020	Discussão sobre a construção do simulador e início da construção de Histórias no Scratch.	7 horas assíncronas + 3 Síncronas :  <b>Total de 10 horas</b>

5º Encontro	22/06/2020	Apresentações das histórias e simuladores, após assistirem vídeos sobre as construções.	7 horas assíncronas + 3 Síncronas: <b>Total 10 horas</b>
6º Encontro	29/06/2020	Estudo dos vídeos, análise de projetos no site do Scratch para construir Jogos no Scratch	7 horas assíncronas + 3 Síncronas: <b>Total de 10 horas</b>
7º Encontro	01/07/2020	Debate Sobre as construções dos artefatos construídos até o momento e sorteio dos artefatos para apresentação final;	<b>2 horas Síncronas</b>
8º Encontro	02/07/2020	Preparação do plano de aula para execução dos artefatos dos grupos (2h); Execução da proposta (4h).	10 Assíncronas + 3 horas Síncronas: <b>Total de 13 horas</b>
9º Encontro	17/07/2020	Apresentação dos planos: Nesse momento, os cursistas apresentaram os resultados do seu aprendizado ao longo do curso.	<b>3 horas Síncronas</b>
10º Encontro	23/07/2020	Fechamento do Curso com as discussões gerais: Quais as percepções que os cursistas tiveram? Como avaliaram o curso e o aprendizado para aplicar os conhecimentos em sala? Aqui foram discutidas a Matemática envolvida no curso e as possibilidades de desenvolvimento do pensamento computacional nesse período.	<b>3 horas Síncronas</b>

Fonte: Próprio autor (2021).

O Curso foi estruturado e executado na plataforma gratuita Google Sala de Aula. Nessa plataforma, foram disponibilizados 10 módulos (conf. Quadro 7), com estrutura (conf. Figura 8) inicial do Curso.

As atividades síncronas foram viabilizadas pelo Google Meet, plataforma *on-line* de comunicação por vídeo, que, no formato *free*, não permite a gravação das aulas. Por esse motivo, para realizar todas as atividades do Curso, inclusive as gravações das aulas, utilizamos o *e-mail* da Unemat.

Assim, todas as atividades realizadas entre cursistas e tutor foram gravadas para análise posterior, foi gerado um grande número de documentos de áudio e vídeo, os quais foram utilizados nesta pesquisa.

A metodologia empregada durante o Curso seguiu o viés da Aprendizagem Baseada em Problemas (ABP), cuja premissa está baseada em situações da vida real e cotidiana das pessoas para estimular o desenvolvimento conceitual, atitudinal e procedimental dos estudantes (BOROCHOVICIUS; TORTELLA, 2014). O detalhamento dessa metodologia é feito no Apêndice C na seção 8.3.

Os dois primeiros módulos do curso foram dedicados ao aprofundamento teórico sobre pensamento computacional e suas relações com a Matemática, bem como sobre o uso da plataforma Scratch. Os demais módulos tiveram um caráter mais prático, ou seja, neles foram realizadas atividades de exploração e produção com o Scratch, com foco no desenvolvimento de artefatos que envolvessem conceitos matemáticos. Especificamente, no módulo 9, solicitamos que os cursistas apresentassem um plano de aula que envolvesse o uso do Scratch em processos de aprendizagem da Matemática e do desenvolvimento de habilidades do pensamento computacional. Uma parte da estrutura do curso é apresentada na Figura 8.

Figura 8 - Estrutura do Curso de Formação – Google Sala de Aula

The screenshot shows a Google Classroom interface for a course titled 'a e Pensamento Computacional com Scratch'. The page is divided into three main sections: Módulo 1, Módulo 2, and Módulo 3. Each module contains a list of activities and materials. The left sidebar shows a list of all topics from Módulo 1 to Módulo 10. The main content area shows the details for each module, including the title, a list of items with icons, and their respective dates or posting times.

Module	Item	Date / Posting Time
Módulo 1	Termo de consentimento	Item postado em 11 de jun. d...
	Nos estudos realizados sobre pensame... 9	Data de entrega: 2 de jun. de ...
	Material do Módulo 1 3	Última edição: 22 de mai. de ...
Módulo 2	Materiais BNCC e Roteiro	Item postado em 17 de jun. d...
	Relatório do Encontro	Última edição: 3 de jun. de 2...
	As figuras geométricas construídas co... 2	Data de entrega: 7 de jun. de ...
	Como você utilizaria o pensamento comput...	Item postado em 3 de jun. de...
Módulo 3	Tutorial gol da menina	Item postado em 16 de jun. d...
	Tutorial Isolamento Social	Item postado em 16 de jun. d...

Fonte: Dos autores (2021).

Como instrumentos de produção de dados foram utilizados relatórios elaborados pelos professores cursistas sobre as atividades desenvolvidas, questionário (entrada) e entrevistas semiestruturadas, além dos artefatos produzidos no Scratch pelos cursistas.

Os relatórios eram aplicados após os módulos do Curso, e respondidos diretamente na plataforma Google Sala de Aula. Ao todo, foram solicitados 7 relatórios.

#### 4.3.1 Questionário

O questionário foi aplicado no início do curso para analisar o perfil dos participantes, e a entrevista foi aplicada após o término do Curso.

Em fase final da elaboração, de acordo com Nogueira (2002), os questionários podem ser:

- questionários abertos, que têm como vantagem a característica de explorar todas as possíveis respostas a respeito de um item, servindo de base para a futura elaboração de um questionário fechado;
- questionários fechados, que apesar de se apresentarem de forma mais rígida do que os abertos, permite a aplicação direta de tratamentos estatísticos com auxílio de computadores e elimina a necessidade de se classificar respostas à posteriori, possivelmente induzindo tendências indesejáveis;
- questionários diretos, que apresentam a vantagem de se coletar diretamente a resposta desejada;
- questionários indiretos, alternativas utilizadas para os casos em que não é possível obter uma resposta precisa às questões por impossibilidade ou por se tratar de um assunto delicado;
- questionários assistidos, que permitem ao pesquisador acompanhar e coordenar diretamente as perguntas aos entrevistados, porém que podem induzir os respondentes a expressarem-se de acordo com ênfases do pesquisador, ao invés de suas próprias;
- questionários não assistidos, que se por um lado eliminam a possibilidade de contaminação por parte de um aplicador, podem não ser respondidos ou respondidos por pessoas não adequadas (NOGUEIRA, 2002, p. 2).

Para o referido trabalho, utilizamos o questionário fechado. Segundo Marconi e Lakatos (1999, p.100), os questionários “são instrumentos desenvolvidos cientificamente, compostos de um conjunto de perguntas ordenadas de acordo com um critério predeterminado”.

Nesta pesquisa, o questionário foi composto de perguntas fechadas e duas perguntas abertas, conforme Apêndice D, da seção 8.4. Foi aplicado para visualizar o entendimento dos professores sobre pensamento computacional, se eles, na formação inicial, estudaram o tema, letramento digital e o que sabem do *software* Scratch, o que permitiu identificar o perfil dos participantes.

### 4.3.2 Entrevistas

A entrevista, segundo Marconi e Lakatos (1999, p. 94), é o “encontro entre duas pessoas, a fim de que uma delas obtenha informações a respeito de um determinado assunto”. Complementarmente, Gil (1999) e May (2004) asseguram que a entrevista pode ser estruturada, semiestruturada e não estruturada.

Uma entrevista é estruturada quando as perguntas são bem definidas. Segundo Gil (1999, p. 121), esse tipo de entrevista “desenvolve-se a partir de uma relação fixa de perguntas, cuja ordem e redação permanece invariável para todos os entrevistados, que geralmente são em grande número”.

Nas entrevistas semiestruturadas, o entrevistado responde às perguntas dentro de sua concepção, mas não fala livremente. Caso o entrevistado desvie do tema original o pesquisador busca pela sua retomada (GIL, 1999).

Já a entrevista não-estruturada é totalmente aberta, é flexível e busca compreender a concepção do entrevistado. May (2004, p. 149) ensina que a entrevista não-estruturada “permite ao entrevistado responder perguntas dentro da sua própria estrutura de referências”.

Neste estudo, foram utilizadas entrevistas semiestruturadas, as quais foram realizadas de forma remota, via Google Meet, nos horários definidos com cada cursista. Fizemos perguntas para os participantes, observando e redigindo suas respostas para ver se realmente o *software* Scratch tem viabilidade em processos de ensino da Matemática e desenvolvimento do PC.

Para não caracterizar os entrevistados, adicionamos apenas os excertos das entrevistas com cada Professor cursista, que está no Apêndice B, na seção 8.2.

### 4.3.3 Artefatos

Os artefatos produzidos no Scratch, durante o curso, também foram considerados na análise, com intuito de identificar quais habilidades do pensamento computacional e da Matemática foram mobilizadas pelos professores cursistas em cada produção.

Os cursistas produziram jogos, simulações e histórias. No início, foi proposto o desenvolvimento de produções básicas, com histórias que mudavam de cenário, no máximo uma vez, para que os professores cursistas fossem adquirindo familiaridade

com a plataforma Scratch. Ao todo, durante o curso, foram produzidos 10 jogos, 11 histórias e 10 simulações. No entanto, somente os últimos artefatos que foram produzidos por eles, ligados a um plano de aula, é que constituíram o corpus de análise da pesquisa.

Com esses instrumentos de coleta de dados, buscamos obter um conjunto de dados que, após a análise, pudessem apontar as potencialidades e dificuldades do *software* Scratch para o ensino da Matemática e desenvolvimento do pensamento computacional no ambiente escolar.

A metodologia empregada durante o Curso seguiu o viés da Aprendizagem Baseada em Problemas (ABP) cuja premissa se baseia em situações da vida real e cotidiana das pessoas para estimular o desenvolvimento conceitual, atitudinal e procedimental dos estudantes (BOROCHOVICIUS; TORTELLA, 2014). O detalhamento dessa metodologia é feito no Apêndice C na seção 8.3.

#### 4.4 ANÁLISE DOS DADOS

A análise dos dados produzidos seguiu a abordagem da Análise Textual Discursiva (ATD), de Moraes e Galliazzi (2016, p. 13), que a definem como “uma metodologia de análise de informações de natureza qualitativa com a finalidade de produzir novas compreensões sobre fenômenos e discursos”. Esse método busca um aprofundando criterioso e rigoroso durante a análise do corpus da pesquisa a fim de obter um arcabouço de informações existentes sobre o tema estudado.

A Análise Textual Discursiva se concretiza através de um conjunto de documentos chamado de corpus. Para Moraes (2003),

Esse conjunto representa as informações da pesquisa e para a obtenção de resultados válidos e confiáveis, requer uma seleção e delimitação rigorosa. Seguidamente não trabalhamos com todo o corpus, mas é necessário definir uma amostra a partir de um conjunto maior de textos (MORAES, 2003, p. 194).

Para esse autor, a matéria prima do *corpus* são as produções textuais. Porém, não se deve seguir a interpretação literal da palavra “textuais”, pois o autor também afirma que “o termo deve ser entendido num sentido mais amplo, incluindo imagens e outras expressões linguísticas” (MORAES, 2003, p. 194).

Neste trabalho, o corpus da pesquisa foi constituído por: um questionário que foi aplicado aos cursistas antes do início do curso, para analisar os perfis dos sujeitos da pesquisa; relatórios elaborados pelos professores cursistas após os encontros

síncronos; os objetos digitais que eles produziram no Scratch; e, por fim, as entrevistas individualizadas realizadas após o término do curso.

Uma vez constituído o corpus da pesquisa, os procedimentos da análise, na ATD, focaram-se em quatro ciclos: a desmontagem de textos; o estabelecimento de relações; captação do novo emergente; e, por fim, o processo auto-organizado. “Os três primeiros compõem um ciclo, no qual se constituem como elementos principais desta abordagem de análise” (MORAES, 2003, 191). Na Figura 10, apresentamos uma representação da ATD na forma de um fluxograma.

Na **desmontagem de textos**, também chamada de unitarização, os textos são examinados em seus detalhes e fragmentados de forma a produzir unidades, “enunciados frente aos fenômenos estudados” (MORAES, 2003, 191).

A fase de **estabelecimento de relações** é um processo denominado categorização. Nela, os textos da base (as unidades) são relacionados e sofrem comparações definidas no início do processo, esses são classificados e combinados para formar “conjuntos mais complexos”, que correspondem às categorias (MORAES, 2003, 191).

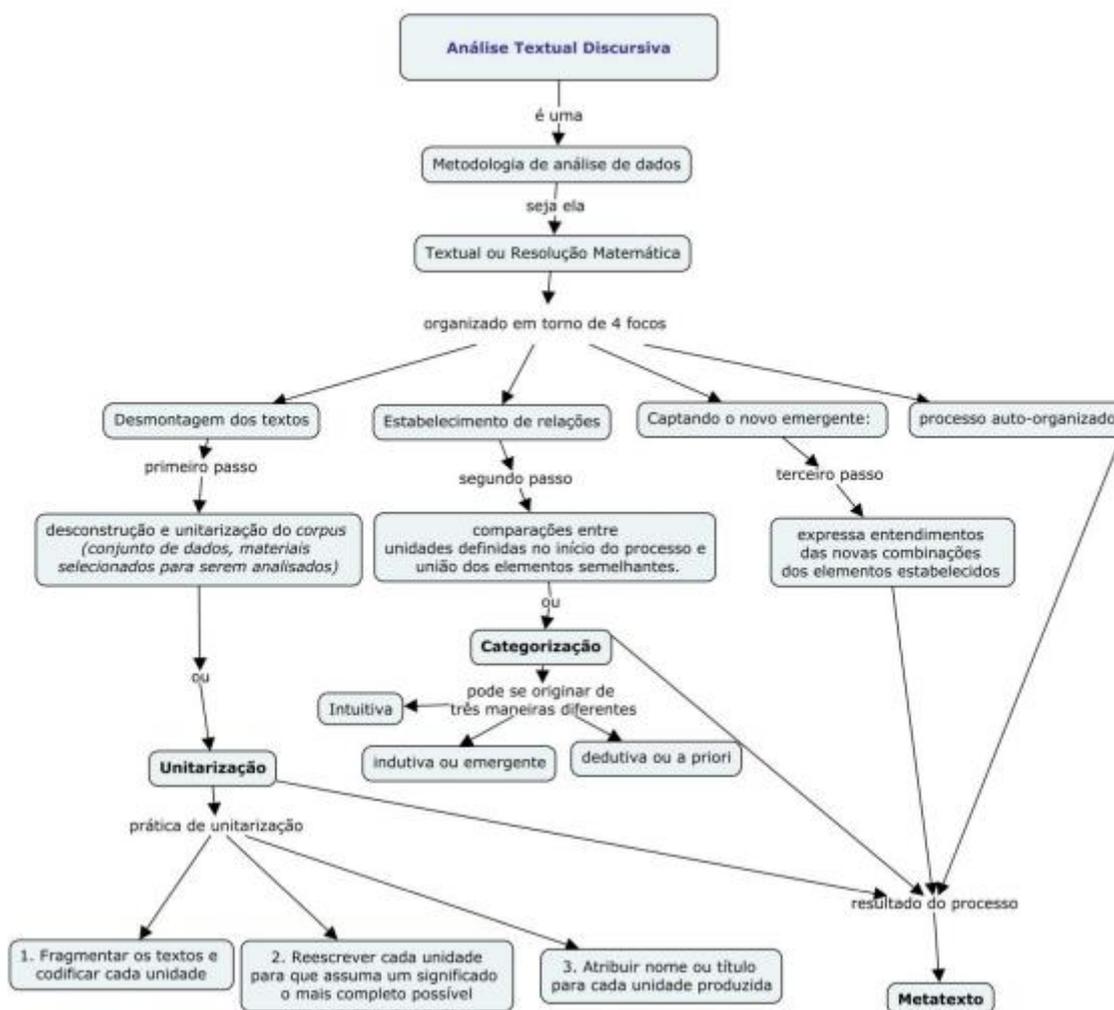
A terceira fase do ciclo, a **captação do novo emergente**, é caracterizada pelo surgimento do novo, ou seja, de uma compreensão renovada sobre o fenômeno analisado. O novo emergente é um metatexto, obtido após uma intensa impregnação dos materiais de análise, produzidos nas duas primeiras fases do ciclo, o qual procura explicar a compreensão que se revela ao pesquisador (MORAES, 2003).

Por fim, no processo da ATD, vem o **processo auto-organizado**, que, segundo Moraes e Galliazzi,

[...] corresponde a um processo auto-organizado composto de dois movimentos principais. O primeiro deles é de desconstrução, de desmontagem dos sentidos e conhecimentos existentes, de aproximação ao caos. O segundo é de reconstrução, de organização das unidades de sentido produzidas pela desconstrução, com emergência de categorias e textos expressando os novos entendimentos construídos no processo (MORAES E GALLIAZZI, 2006, p. 126).

Partindo da premissa que envolve todo o movimento da ATD observado na Figura 9, os resultados finais, originais e criativos, não podem ser previstos, já que é uma pesquisa e essa pode levar a resultados não previstos inicialmente, mas “Mesmo assim é essencial o esforço de preparação e impregnação para que a emergência do novo possa concretizar-se” (MORAES, 2003, p. 192).

Figura 9 - Mapa Conceitual da ATD



Fonte: CONCENTINO et al. (2017)

Conforme supracitado, as categorias preestabelecidas foram: características do software Scratch (vantagens e desvantagens) e habilidades mobilizadas da matemática e do pensamento computacional, caracterizadas no segundo passo, conforme Figura 9, estabelecimento de relações.

Nesta pesquisa, primeiramente, foram definidas as duas categorias supracitadas, com base nos objetivos almejados com a pesquisa. Após isso, foi realizada a fase de unitarização do corpus constituído pelo questionário, relatórios, entrevistas e 7 artefatos produzidos no Scratch.

A unitarização foi trabalhada a partir do corpus da pesquisa. Primeiramente, foram escolhidas as categorias, pois era necessário de partir delas para realizar a unitarização.

O questionário e relatórios se apresentavam em formato de texto, enquanto as entrevistas estavam em arquivo audiovisual e, os artefatos no Scratch, em arquivos de mídia específicos da plataforma contendo dados textuais, de áudio e imagens. Para uniformização do formato do corpus, as entrevistas foram transcritas na íntegra e os artefatos do Scratch foram analisados e parcialmente descritos, quando se observava alguma unidade de análise. Ao longo dessa etapa de organização do corpus da pesquisa, cada professor foi rotulado com um código de P1 a P14, conforme esclarecimento anterior.

Para uma melhor organização dos dados, as unidades de análise e suas respectivas codificações foram ordenadas numa planilha eletrônica, para que relações pudessem ser mais facilmente estabelecidas. A identificação das relações entre unidades e categorias favoreceu a elaboração dos metatextos, os quais são apresentados no capítulo 5 deste trabalho.

## 5. APRESENTAÇÃO E ANÁLISE DOS DADOS

Neste capítulo, serão apresentados os principais dados produzidos no curso e os seus resultados à luz do método da Análise Textual Discursiva.

Primeiramente, apresentaremos o Curso de formação e todos os seus 10 módulos. Em seguida, iniciaremos a análise do corpus da pesquisa, considerando duas categorias pré-estabelecidas a partir dos objetivos da pesquisa, sendo elas: características favoráveis e desfavoráveis do Scratch para o ensino e aprendizagem da Matemática e desenvolvimento do pensamento computacional; habilidades mobilizadas (da Matemática e do pensamento computacional) com o uso do Scratch; que favorece os processos de ensino e aprendizagem da Matemática e o desenvolvimento do PC partindo da análise dos professores cursistas.

### 5.1. O CURSO DE FORMAÇÃO

O curso foi realizado 100% *on-line*, utilizando a plataforma Google Sala de Aula, com momentos de atividades síncronas e assíncronas. O modelo adotado para o curso seguiu o viés do modelo híbrido, com os encontros presenciais substituídos por momentos síncronos *on-line*.

Já o modelo teórico adotado no curso se aproximou do *À La Carte*, no qual, segundo Bacich (2016),

O estudante é responsável pela organização de seus estudos, de acordo com os objetivos gerais a serem atingidos, organizados em parceria com o educador; a aprendizagem, que pode ocorrer no momento e local mais adequado, é personalizada. Nessa abordagem, pelo menos uma disciplina é feita inteiramente *on-line*, apesar do suporte e organização compartilhada com o professor. A parte *on-line* pode ocorrer na escola, em casa ou em outros locais (BACICH, 2016, p. 683).

Nesse modelo, o estudante é responsável pelos seus estudos, a aprendizagem ocorre no momento e no local mais apropriado e é personalizada.

Todas as atividades e materiais do curso foram disponibilizadas na plataforma Google Sala de Aula e os módulos foram pensados e organizados de forma a evidenciar a metodologia da Aprendizagem Baseada em Problemas. Para tirar as dúvidas ou marcar reuniões extraordinárias tinha o grupo de *WhatsApp*, exclusivo do curso.

### **5.1.1 Módulo 0 - Momentos que antecederam o curso**

Primeiramente, o curso foi todo planejado e implementado na plataforma Google Sala de Aula para que pudéssemos apresentar aos professores, em um momento anterior ao início das atividades, a estrutura geral e o cronograma de execução do curso.

Também, nesse momento, foi criado um grupo de *WhatsApp* para que os cursistas expressassem suas dúvidas iniciais sobre o curso e para servir de canal de comunicação durante toda a formação.

Após o término dessa organização, marcamos uma reunião, via *Google Meet*, para apresentar a plataforma do curso (na plataforma Google Sala de Aula), com seus módulos e atividades. Aproveitamos esse momento para indicar as leituras e vídeos iniciais (atividades assíncronas), cujos temas seriam discutidos no primeiro módulo do curso. Também pedimos para responderem a um questionário sobre atuação profissional, letramento digital e pensamento computacional.

### **5.1.2 Módulo 1 – Início do curso - estudos teóricos sobre pensamento computacional e suas possíveis relações com a Matemática**

De certa forma, o primeiro módulo do curso iniciou com a apresentação da proposta aos professores, quando foi detalhada toda a estrutura do curso a ser desenvolvido. Contudo, optamos por contar como módulo 1, o primeiro momento síncrono, voltado para as discussões sobre pensamento computacional e suas relações com a Matemática e de como esse pensamento poderia ser desenvolvido durante as aulas desse componente curricular.

Ao término desse módulo, os professores cursistas elaboraram um relatório dos principais pontos discutidos e compreendidos sobre pensamento computacional. Alguns recortes do que foi observado pelos professores são apresentados nos excertos abaixo.

Nas aulas de Matemática, o PC pode servir como mediação em situações que envolvam problemas matemáticos, principalmente em sua estruturação por fluxogramas e algoritmos dos variados caminhos possíveis de resolução (P2, Relatório 1).

Nos variados conceitos e conteúdos matemáticos podemos vislumbrar possibilidades na contextualização, estruturação organizacional de algoritmos, repartição de ideias almejando resultados variados e novas

alternativas e reformulação ou criação de novos problemas derivados da depuração (P2, Relatório 1).

O pensamento computacional pode facilitar na resolução de situações problemas, pois possui etapas que se forem seguidas, torna mais prático alcançar o objetivo (P3, Relatório 1).

Através de situações problemas apresentaria para os alunos que com a organização de ideias por meios de princípios do pensamento computacional é possível encontrar a solução de forma prática sem mesmo necessitar do uso de um computador ou qualquer tecnologia digital (P4, Relatório 1).

Quando falamos em Matemática, falamos sobre problemas reais a serem resolvidos, mas não de qualquer forma, precisa-se de um passo a passo para que esse problema seja realizado de forma eficiente e organizado (P5, Relatório 1).

O pensamento computacional compartilha com o pensamento matemático. Alguns métodos que normalmente usamos para resolver problemas, é um pensamento complexo que se baseia na lógica dedutiva (P12, Relatório 1).

O pensamento computacional está ligado à lógica, da mesma forma que a lógica está ligada a Matemática (P14, Relatório 1).

Esses excertos deixam claro que os professores perceberam a existência de relações entre PC e Matemática, demonstrando que o objetivo do módulo foi atingido.

### **5.1.3 Módulo 2 - Estudos teóricos sobre Scratch e suas relações com pensamento computacional e a Matemática**

Neste módulo, foram discutidos os aspectos teóricos sobre o uso do Scratch como recurso pedagógico e a simbiose do *software* com o PC e a Matemática. Para tal, foram disponibilizados no ambiente *on-line* do curso artigos e vídeos sobre o tema.

O estudo do material pelos cursistas ocorreu de forma assíncrona, enquanto as discussões sobre o assunto aconteceram de forma síncrona, via Google Meet, com todos os cursistas.

Após o estudo e discussão do material (ocorridos no módulo), foi orientado que os cursistas assistissem uma série de vídeos que instruíam sobre o uso da plataforma Scratch. Entre os vídeos, alguns deles ensinavam a construir figuras geométricas planas usando o Scratch. Essa atividade foi realizada de forma assíncrona, no intervalo entre os módulos 2 e 3, com o propósito de instruí-los ao uso da plataforma.

Complementarmente, foram marcados momentos extraclasse para auxiliá-los com a compreensão das atividades, visto que apenas um dos professores, P2, já possuía alguma experiência com o uso do Scratch. Quase todos os cursistas requereram auxílio, com exceção de 4 deles.

O que se observou, nesses dois primeiros módulos, foi que eles satisfazem quatro etapas da Aprendizagem Baseada em Problemas (ABP). Quando inserimos os materiais para os estudos sobre o ensino da Matemática, desenvolvimento do Pensamento computacional (PC) e uso do Scratch, trazemos a **realidade da vida dos cursistas** (Seção 4.4.2 – item I), pois esse fato está ligado diretamente aos seus trabalhos cotidianos, podendo vir a potencializar esses trabalhos.

A segunda etapa da ABP é caracterizada pelo **recebimento do contexto problemático** (Seção 4.4.2– item II), que ocorre quando os cursistas começam a conhecer os aspectos teóricos das situações-problemas que eles ajudarão a delinear.

A terceira etapa da ABP observada foi a da **investigação por meio dos diversos recursos disponibilizados pelo Tutor** (Seção 4.4.2– item III), já que eles, naquele momento, iriam iniciar as construções de figuras geométricas planas no Scratch, que seriam suas primeiras produções, e os vídeos e materiais deixados na plataforma Google Sala de Aula serviriam de base para ajudá-los a realizar essa atividade.

Para finalizar esse módulo, solicitamos que respondessem (na forma de texto) à seguinte pergunta: como você utilizaria o pensamento computacional com o *software* Scratch no ensino de Geometria? O objetivo da pergunta era verificar se os professores conseguiam perceber as possibilidades de uso do Scratch para desenvolver habilidades da Matemática e do pensamento computacional.

Foram selecionados alguns excertos representativos das respostas apresentadas pelos cursistas:

Usaria o *software* como aprofundamento dos conceitos mais abstratos da geometria básica em geral e como aplicação nas construções de figuras geométricas, principalmente às relacionadas as etapas que envolvam o uso de algoritmos e visualização de propriedades matemáticas (P2, Relatório 2).

Esse é meu primeiro contato com Scratch. O pouco do que aprendi até agora me fez entender que usamos alguns pilares do pensamento computacional nas construções de figuras geométricas sem perceber, tais como dividir em pequenos blocos, reconhecimento de padrões, montar um roteiro de ações para construir o passo a passo e por fim chegar na solução da construção. (P4, Relatório 2).

Com o pouco que já vi, percebo que a utilização do pensamento computacional é fundamental para que o projeto tenha êxito. No campo abstração, temos que identificar o que queremos realizar, escolher as ações no programa e desconsiderar ações que não irão auxiliar o projeto. No campo decomposição, podemos dividir cada ação dos personagens em pequenos

blocos para corrigir eventuais falhas e ir aperfeiçoando. No campo reconhecimento de padrões é possível perceber que algumas ações se repetem, com isso não é preciso construir novos blocos com a mesma ação. Basta duplicar ou usar a função REPETIR. No campo algoritmo, podemos enumerar os passos para que todos os alunos consigam construir, facilitando a compreensão do projeto (P3, Relatório 2).

Usaria o *software* como aprofundamento dos conceitos mais abstratos da geometria básica em geral e como aplicação nas construções de figuras geométricas, principalmente às relacionadas as etapas que envolvam o uso de algoritmos e visualização de propriedades matemáticas (P2, Relatório 2).

Primeiro ensinaria o básico do programa para os alunos, e após isso, solicitaria algumas ideias de resoluções para a construção de determinadas figuras e sólidos geométricos. Após as tentativas dos alunos e a apresentação de suas ideias, passaríamos para a construção todos juntos (P12, Relatório 2).

Esses excertos já dão indicativo de que os professores, mesmo na fase inicial do curso e de contato com o Scratch, já conseguem vislumbrar potencialidades do uso desse *software* para o desenvolvimento de habilidades do pensamento computacional e da Matemática.

#### **5.1.4 Módulo 3 e Módulo 5 – A construção de um simulador no Scratch**

Dando sequência à exploração do Scratch com os cursistas, combinamos com eles que iríamos realizar a construção de três diferentes artefatos: simulação, histórias animadas e jogos.

No módulo 3, foi iniciada a produção de um simulador de contágio da Covid-19 com base na interação entre pessoas, conforme Figura 10.

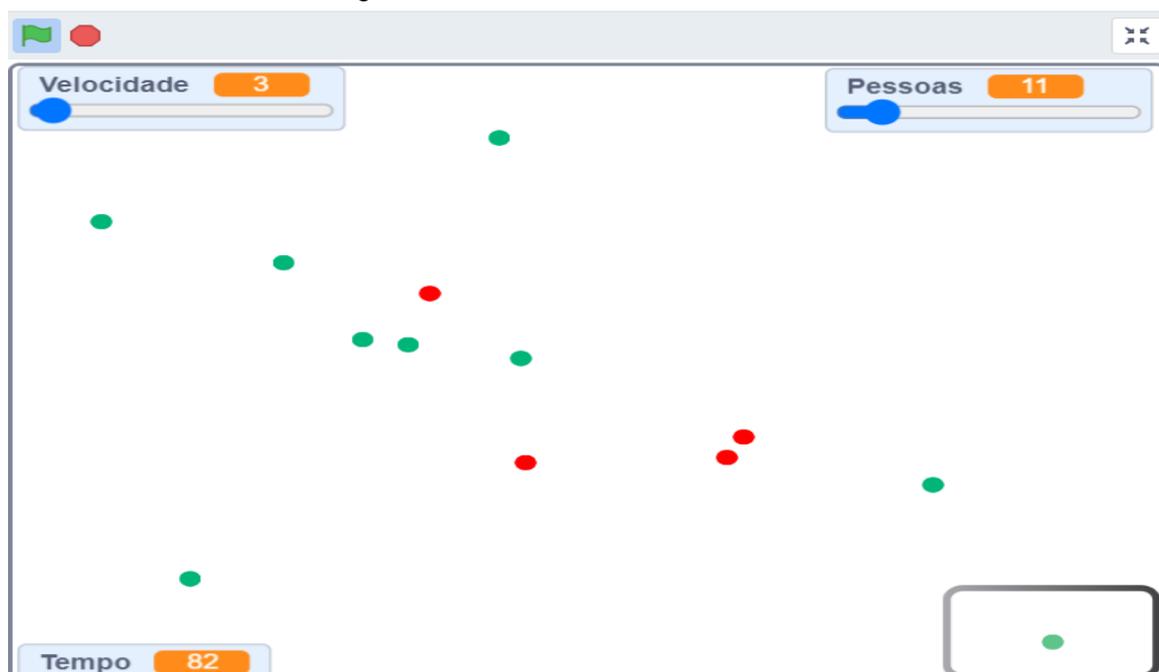
Antes de iniciar a atividade, foi orientado que os professores cursistas criassem uma conta, no Scratch *for Education*<sup>12</sup>, para postarem seus trabalhos de modo que pudéssemos acompanhar o desenvolvimento de cada projeto da turma, por dentro do ambiente proporcionado por essa conta.

Esse encontro objetivou desenvolver o artefato simulador, mas foi observado que, nesse momento, eles tiveram dificuldade para captar o que foi trabalhado pelo tutor. Consideraram o artefato complexo de produzir.

---

<sup>12</sup> Site disponibilizado para professores de Scratch criarem suas turmas e colocarem seus trabalhos, bem como explorar outros trabalhos já construídos por outros Scratchers.

Figura 10 - Simulador de Isolamento Social



Fonte: <https://scratch.mit.edu/projects/402495490/>

E como o tempo de curso era curto, com encontros seguidos, não poderíamos pular a simulação e somente realizar histórias e jogos, por esse motivo, logo em seguida ao encontro e percebida a dificuldade do grupo, resolvemos retomar a simulação em encontro futuro, após a história.

Assim, ao invés de trabalhar o simulador, foram estudadas as histórias, para que fossem apresentadas no módulo 4.

O material com vídeos e tutoriais para a criação de histórias já estava todo no Google Sala de Aula, assim, o simulador só foi discutido novamente no módulo 5, com um exemplo menos complexo, quando foi colocado o projeto no *Scratch for Educators* para análise.

No módulo 5, os professores apresentaram as 3 histórias que faltavam e iniciaram as apresentações dos simuladores.

Os professores conseguiram trabalhar bem com a questão do simulador. E, mesmo com algumas dificuldades, os simuladores ficaram muito bons, haja vista que o intento de todo o trabalho era que eles chegassem no módulo 9 com a capacidade de realizar criações mais elaboradas, e construíssem um bom plano de aula.

Nesse contexto, observamos o trabalho dos cursistas como o centro da aprendizagem, pois eles estavam crescendo em suas produções e potencializando os trabalhos iniciais.

Observamos também que o trabalho em grupo ganhava corpo, já que os cursistas conversavam entre si e se ajudavam durante as apresentações, como também nos seus estudos realizados.

Eles citavam os companheiros que ajudavam nas suas criações, e, quando essa ajuda não era possível, eles buscavam no professor tutor o apoio para o trabalho.

Nesse módulo, além das histórias, foram apresentadas mais 3 simulações e também foram realizados, nos 30 minutos finais, as tarefas que não tinham sido realizadas em relação aos simuladores que estavam no módulo 3 do Google Sala de Aula.

A pergunta para os simuladores estava ligada ao PC: Quais as habilidades do pensamento computacional foram desenvolvidas com a atividade de criação de simuladores? O objetivo foi observar se eles entendiam que estavam desenvolvendo o PC nas suas construções.

Eles a responderam da seguinte forma:

Todo o processo de desenvolvimento e realização do simulador envolve Matemática e pensamento computacional, por mais que para se explicar para os alunos se use metodologias pedagógicas diferentes, acredito que todos os pilares são trabalhados (P11, Relatório 5).

Apareceram: Interatividade com o usuário, Representação de dados, Controle de fluxo e Abstração. Faltaram: Lógica, Paralelismo e sincronização - Talvez por falta de prática e pouco conhecimento do *software*. (P11, Relatório 5).

A viabilidade do processo em si da construção do simulador é a parte mais pertinente pedagogicamente, pois demanda do uso e entendimento de conceitos computacionais e matemáticos (P2, Relatório 5).

Por esses excertos, observamos que os cursistas responderam à pergunta com vários conceitos. O primeiro entendeu habilidades baseadas em pilares. Em nosso entendimento, essa resposta deriva das leituras que fizemos e que agora eles conseguiam fazer a relação do artefato produzido com os pilares do PC. Nesse caso, tivemos uma evolução dos primeiros encontros, que entendemos ser natural.

O segundo avalia de acordo com o Dr. Scratch, inclusive foi pedido para todos os cursistas avaliarem seus projetos no programa, e, assim, esse projeto foi avaliado pelo Dr. Scratch. Segundo o cursista, as habilidades que apareceram foram interatividade com o usuário, representação de dados, controle de fluxo e abstração. O interessante é que, quando o cursista respondeu o que faltou, ele justifica que talvez tenha sido por falta de prática e pouco conhecimento do *software*.

Verificamos que essa fala realmente condizia com a realidade, pois, quanto mais se trabalhava e se produzia com o *software*, ocorriam substanciais melhoras na avaliação do projeto pelo Dr. Scratch. Esse fato ficou evidente quando os cursistas trabalharam com o último projeto do plano de aula no módulo 9.

#### **5.1.5 Módulo 4 – Histórias**

Os cursistas, conforme citado na seção anterior, estudaram as histórias nesta semana e apresentaram o que fizeram de histórias.

Durante a criação dos trabalhos, os grupos comunicavam entre si para saber o que os outros estavam fazendo; tiravam dúvidas, e, apenas quando essas dúvidas ultrapassavam seus conhecimentos, pediam ajuda ao professor Tutor.

No dia estipulado para a apresentação dos trabalhos, os sete grupos estavam com suas histórias prontas para apresentar. Nesse encontro, os professores apresentaram sua primeira construção com o *software* Scratch. O intuito era trabalhar o desenvolvimento do pensamento computacional e o ensino da Matemática.

Foram apresentadas 4 histórias, eles tiveram toda liberdade para apresentar suas dificuldades e facilidades para construir as histórias, ficando apenas três grupos para apresentar no módulo 5.

Ainda, nesse módulo, responderam ao relatório disponibilizado no Google sala de aula e, com as histórias prontas, eles puderam observar o módulo 5 (simulador), pois, no módulo 3, esse conhecimento não ficou muito claro.

As perguntas disponibilizadas para esse módulo foram: 1) na construção da história em quadrinhos, o que você achou mais interessante para aplicar no seu trabalho em sala de aula? 2) você encontrou dificuldades na construção da história em quadrinhos (relate a experiência abordando aspectos positivos e negativos)? 3)

que habilidades do pensamento computacional foram mobilizadas? Algumas respostas foram bem ilustrativas:

Achei muito interessante para se trabalhar coordenadas e a parte lúdica com os alunos. As dificuldades encontradas foram na sincronização dos personagens, mas conforme íamos voltando o projeto conseguíamos perceber onde tínhamos que melhorar. As habilidades do pensamento computacional utilizadas foram reconhecimento de padrões, decomposição e o pensamento algorítmico (P4, Relatório 6).

Pontos positivos: explora muito o raciocínio e habilidades de tecnologia digital. Porque precisa pensar na imagem, nos atores, nos fundos, cenários, áudios, no sincronismo das falas, baixar a imagem, pensar num roteiro, enfim, são muitas habilidades a ser exploradas (P13, Relatório 6).

A aplicação da Matemática em uma situação real, do cotidiano dos estudantes, e uma forma de interagir por meio de um vídeo ou a produção de material com um *software*, desta forma estaremos ensinando os estudantes sobre as tecnologias digitais. Sim, principalmente nas codificações do programa e na sincronização das mensagens dos autores [se referindo a segunda pergunta]. A sequência lógica do roteiro com o tempo das falas e mensagem (P8, Relatório 6)

A possibilidade de transformar a história num vídeo mp4 para mostrar aos alunos. A história proposta mostra uma situação real, onde os alunos podem ver a matemática aplicada a situações do dia a dia. A própria construção da história envolvendo os alunos num outro momento, para que entendam todo o processo de construção, onde os mesmos percebam a necessidade dos comandos para a realização de qualquer ação. Sendo possível também o trabalho sobre os conteúdos de potenciação e sequências, citados na história. O tempo maior gasto foi na construção dos atores no *paint* e na busca dos cenários na internet. Após iniciar a construção da história, a sincronização das falas dos personagens deu bastante trabalho. A história foi construída com os comandos básicos do Scratch, então não tivemos dificuldade de inserir os comandos. No entanto foi preciso fazer uso de muitos comandos, por vezes repetidos para a efetivação das ações dos personagens, o que demandou muito tempo para o ajuste dos comandos com a sincronização. O nosso projeto teve 1/3 na abstração, 1/3 no paralelismo, 1/3 na interatividade com o usuário, 1/3 na representação de dados, 2/3 no controle de fluxo e 3/3 na sincronização (P12, Relatório 6).

As habilidades do PC que tiveram mais destaque na historinha foram: a sincronização de dados e controle de fluxo. Já paralelismo, interatividade, representação de dados e abstração foram pontuadas no Dr Scratch, porém foram de menor índice. Não tivemos pontuação de lógica, na historinha (P1, Relatório 6).

Como podemos perceber, os cursistas estavam começando a entender como funcionava os comandos do Scratch e como era a inserção de cenários, fundos, atores, realização da sincronização, colocação de figuras, construção de figuras no *paint* e como buscá-las na *internet*, construção de roteiros etc.

Podemos observar que eles estavam construindo matemática e PC com o Scratch, trabalhando com coordenadas, potenciação, sequências, as bases numéricas, a parte lúdica, buscando figuras, desenhando no *paint*, a sincronização,

comandos, elaborando roteiros, com a possibilidade de trabalharem a realidade de sala de aula.

Eles apresentavam, naquele momento, a dificuldade de qualquer iniciante em um ambiente de programação. Nos excertos, podemos entender que essas dificuldades estão ligadas, principalmente, à sincronização de eventos, pois eles ainda não tinham trabalhado com falas, atores, sons e vários cenários. Por isso, acreditamos, que, naquele momento, algumas das dificuldades relatadas se justificavam pela falta de habilidade com a lógica de programação e com o pouco conhecimento da plataforma Scratch.

No que concerne às habilidades do pensamento computacional, pedimos para todos os cursistas colocarem suas criações no Dr. Scratch e observarem quais as habilidades eram desenvolvidas.

Percebemos que o conceito que o Dr. Scratch menos observou foi a lógica, já que estavam iniciando e o processo da lógica requer que os produtores tenham muita abstração para a sua inserção, por isso, acreditamos que não seria possível aparecer a lógica na maioria dos trabalhos naquele momento.

#### **5.1.6 Módulo 6 - Jogos**

Neste módulo, foram apresentados mais 4 simuladores e 3 jogos via plataforma Google Meet.

Observamos nos trabalhos a evolução dos professores cursistas em relação às suas criações e ao manuseio do *software*, os cursistas já estavam buscando projetos do site da MIT e ampliavam seus conhecimentos com cursos oferecidos pela internet. Naquele momento, eles já estavam realizando criações muito boas, iniciavam o domínio básico do *software*.

Ao final, realizamos uma discussão para levantar a opinião dos cursistas em relação ao que foi produzido até aquele momento. Eles disseram que:

Os jogos são uma marca ideal para chamar a atenção dos estudantes, porém, temos também a criação de animações e cenários para uma aprendizagem satisfatória (P14).

Gostei muito de todos os projetos nos quais participei e acho que todos podem ser usados no trabalho em sala de aula. No entanto, o que achei mais completo, foi a construção de jogos, pois nele podemos testar todas as finalidades dos comandos usados. (P12, Entrevista).

Para o ensino da Matemática vi que é possível desenvolver qualquer conteúdo em uma história, simulador ou jogo (P13).

Avalio que os jogos sejam mais atrativos para os estudantes por já conhecerem, no entanto precisa conhecer as ferramentas do Scratch (P7).

Nesse módulo, não realizamos relatórios por entender que já havia evolução necessária para que os cursistas pudessem apresentar um plano de aula condizente com o aprendizado, já que a evolução das duplas aconteceu de forma satisfatória, deixando o último para o módulo 10, para realizar um apanhado geral do curso.

### **5.1.7 Módulo 7 - Jogos, Discussão sobre os Artefatos Construídos**

Neste módulo, foram apresentados os três últimos jogos desenvolvidos pelos cursistas. Como já mencionado, os professores já conseguiam executar os comandos e estavam aptos para criar jogos, histórias e simuladores. Esse era o intuito do início do curso, pois eles necessitavam chegar nesse módulo com a teoria aprofundada, e o domínio do *software*, para que, nos módulos 8 e 9, pudessem usar sua criatividade para realizar as produções sendo os protagonistas, executando as suas ideias.

Realizamos discussões sobre os artefatos produzidos e dividimos os 7 grupos por artefatos: 3 ficaram com jogos, dois com simuladores e 2 com histórias. Os cursistas que ficaram com simuladores puderam trabalhar com jogos ou histórias, aqui eles iniciaram o plano de aula que se encontra no anexo I.

Durante as discussões, os cursistas avaliaram que, de forma geral, com qualquer artefato produzido, pode-se trabalhar Matemática e PC, mas deram mais ênfase aos jogos, pois consideraram serem eles mais atrativos aos alunos.

Mesmo com essa constatação dos jogos serem mais atrativos aos alunos, os professores consideraram todos os artefatos produzidos como importantes para o trabalho da Matemática e do PC, e não consideraram existir uma hierarquia entre jogos, história e simulação, pois o contexto que irá definir o momento de utilizar cada um deles.

### **5.1.8 Módulo 8 - Desenvolvimento do plano de aula**

No módulo 8, no encontro via Google Meet, o objetivo era observar as dificuldades que os professores estavam encontrando nos seus planos de aula,

presentes no Apêndice A, na seção 8.1, e as possíveis contribuições do tutor e dos outros grupos.

Cada grupo colocava sua ideia de apresentação para o módulo 9, quando um grupo apresentava o que estava construindo, os outros esperavam e, no final, ajudavam apontando possíveis melhorias para o plano de aula, conforme Apêndice A, na seção 8.1. Nesse encontro, os professores cursistas, por meio do Scratch, apresentaram bons planos, relacionando PC e Matemática. Ficou evidente um ganho de maturidade dos cursistas em relação ao Scratch, pois se mostraram bem ambientalizados com a plataforma.

Depois desse encontro via Meet, os professores tiveram 15 dias para montar seus planos e apresentá-los. Continuaram se comunicando pelo grupo de WhatsApp entre eles e também com o professor tutor, os contatos não pararam até o dia da apresentação.

A atividade para pós-encontro foi a finalização do plano de aula e sua execução.

### 5.1.9 Módulo 9 - Apresentação dos objetos produzidos no Scratch

Neste módulo, os professores cursistas apresentaram os objetos desenvolvidos de acordo com o que haviam planejado e colocado no plano de ensino elaborado.

Os artefatos apresentados foram: história, simulador e jogos. No Quadro 8, são listados todos os artefatos produzidos, seguidos dos *links* para sua visualização na plataforma *on-line* do Scratch, onde estão hospedados.

Quadro 8 - Quadro resumo dos artefatos produzidos

Nome do artefato	Tipo de artefato	Link
Função Exponencial	História	<a href="https://scratch.mit.edu/projects/410097593/">https://scratch.mit.edu/projects/410097593/</a>
Brincando com a simetria	Simulação	<a href="https://scratch.mit.edu/projects/412094139/">https://scratch.mit.edu/projects/412094139/</a>
Critérios de divisibilidade	Jogo	<a href="https://scratch.mit.edu/projects/412099569/">https://scratch.mit.edu/projects/412099569/</a>
Problema dos 35 camelos	História	<a href="https://scratch.mit.edu/projects/412149079/">https://scratch.mit.edu/projects/412149079/</a>
Merilu Bica	Jogo	<a href="https://scratch.mit.edu/projects/412146306/">https://scratch.mit.edu/projects/412146306/</a>

Jogo do Miko 3.1	Jogo	<a href="https://scratch.mit.edu/projects/408749283/">https://scratch.mit.edu/projects/408749283/</a>
Base dez e base sexagesimal	História	<a href="https://scratch.mit.edu/projects/412004988/">https://scratch.mit.edu/projects/412004988/</a>

Fonte: Dos autores

### 5.1.9.1. Projeto Função Exponencial

P5 e P14 produziram uma história para abordar os conceitos matemáticos de potenciação e função exponencial e suas aplicações em problemas de tomada de decisão envolvendo valorização e desvalorização de um bem (caminhão).

Figura 11 - Função exponencial – empresário Murilo realizando simulação de um caminhão novo, para observar a desvalorização desse caminhão.



Fonte: <https://scratch.mit.edu/projects/410097593/>

A história retratava uma situação de tomada de decisão referente à compra de um caminhão, considerando duas opções de compra. O personagem principal, o empresário Murilo, em destaque na Figura 11, ligava para duas empresas para consultar o preço de um mesmo caminhão e elas lhe fornecem, além do valor, a taxa de desvalorização anual, conforme pode ser visto na Figura 12 (a).

Após esses diálogos com os atendentes das lojas, o comentarista da história, apresentava uma série de cálculos para mostrar como calcular a desvalorização dos caminhões após um determinado período de tempo, usando potenciação e funções exponenciais, conforme pode ser visualizado na Figura 12 (a). A partir dessa discussão, o comentarista apresentava a opção mais vantajosa de compra, conforme Figura 12(b).

Nesse objeto, todos os cálculos foram apresentados como texto nos balões de conversa e/ou nos *frames* que compuseram a história.

Figura 12 – Frames do projeto chamado função exponencial: Figura 12 (a) apresenta números de desvalorização do caminhão durante 5 anos, enquanto a Figura 12 (b) apresenta a figura do caminhão e seu valor com taxa de desvalorização mensal e, logo em seguida, uma fórmula de função exponencial.

**Caminhão A:**  
 Dados: Tempo e Taxa na mesma unidade de tempo (ano)

1) O valor do caminhão durante 1 ano:  
 $100\% - 15\% = 85\% = 85 / 100 = 0,85$

1.1)  $150\ 000 \cdot 0,85 = \text{R\$ } 127\ 500,00$

2) O valor do caminhão durante 2 anos:  
 1.2)  $127\ 500 \cdot 0,85 \cdot 0,85$

Ou

$127\ 500 \cdot 0,85^2 = \text{R\$ } 108\ 375,00 = \text{Função Exponencial.}$

3) Durante os 5 anos, o caminhão valerá:  
 1.3)  $150\ 000 \cdot 0,85^5 = \text{R\$ } 66\ 555,79$

4) Houve uma desvalorização de **R\$ 83 444,21.**

**b**

Ele é um ótimo caminhão, com desvalorização de 1,5% a.m. segundo a Tabela Fipe, e o seu valor é de R\$ 130 000,00.

Vamos trabalhar com a função exponencial, cuja fórmula é  $f(t) = a^t$ , com  $a > 0$  e  $a \neq 1$

Fonte: <https://scratch.mit.edu/projects/410097593/>

#### 5.1.9.2. Brincando com a simetria

No projeto P1 e P11, os cursistas apresentaram uma simulação: a primeira tela (Figura 13a) é o início da simulação; para iniciar teria de clicar primeiro na borboleta, que passava algumas informações sobre a simulação.

Após as explicações, se o usuário clicasse em simulação (Figura 13a), ele iria direto para a produção de simulações, conforme Figura 14 (a e b).

Figura 13 – Na Figura 13 (a) é apresentado a tela inicial do jogo, a Figura 13 (b) são apresentados hiperlinks e explicações transformações matemáticas.

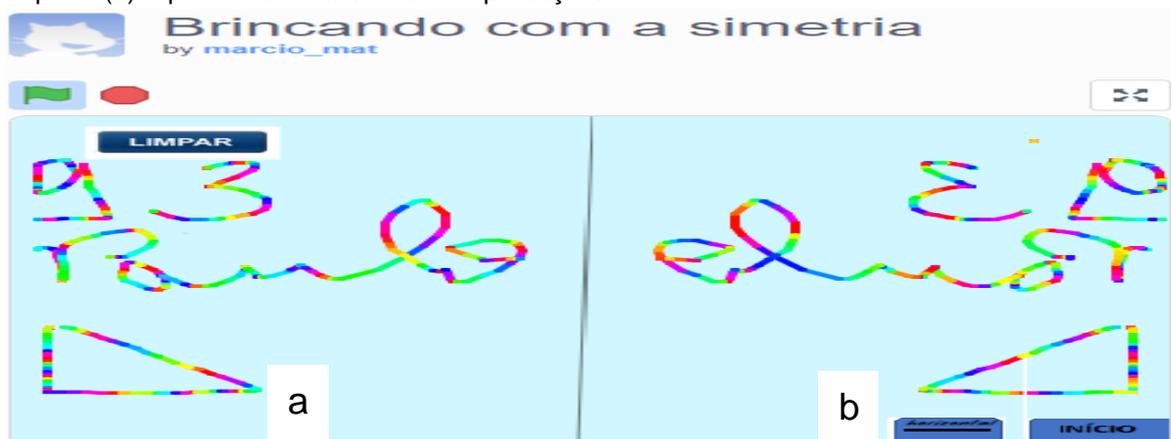


Fonte: <https://scratch.mit.edu/projects/412094139>

Se clicar em conteúdo, Figura 13 (a), o usuário seria levado a hiperlinks em dois textos, ambos matemáticos, em que um aborda o que é simetria e o outro relata quais são os movimentos de rotação, translação e reflexão. Podemos observar essa explicação na Figura 13 (b).

A Figura 14 (a) apresenta a parte em que o usuário realiza a sua produção, enquanto a Figura 14 (b) representa a simetria de reflexão dessa produção.

Figura 14 – Brincando com a simetria, parte (a) da figura representa o que foi produzido pelo usuário, e a parte (b) representa a reflexão dessa produção.



Fonte: <https://scratch.mit.edu/projects/412094139>

### 5.1.9.3. Acerto o Pato – Critérios de divisibilidade

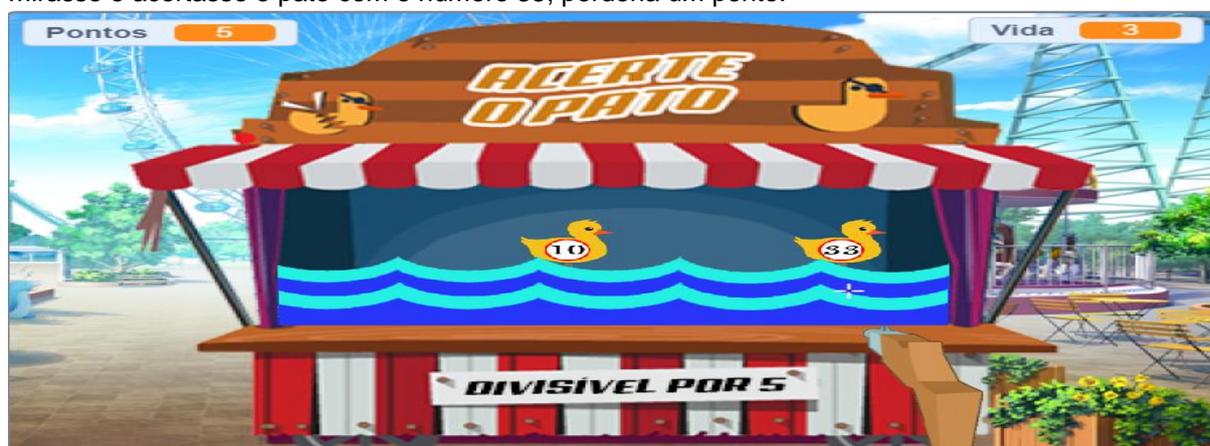
Na criação de P3 e P4, eles trabalharam a matemática abordando critérios de divisibilidade, em um jogo chamado Acerte o Pato.

A produção consistia em quem acertasse o pato correto ganharia 1 ponto, e se acertasse o pato errado, perderia uma vida. O usuário iniciava com 3 vidas (ao perder as 3 vidas, o jogo encerrava) e, a cada 5 pontos, o jogo mudava de nível. Nos cantos superiores esquerdo e direito da Figura 15, é possível observar pontos e vidas.

O jogo foi dividido em fases, iniciava com a apresentação da atriz Ana e logo após aparecia o primeiro critério de divisibilidade, que era por 2; se acertasse 5 patos corretos, marcaria 5 pontos e passaria para o critério de divisibilidade 5; com mais 5 pontos, passaria para o critério 3 e, por fim, com mais 5 pontos, iria para os números divisíveis por 6. Na Figura 15, há um exemplo da tela inicial do jogo..

Para atirar, eles criaram uma espingarda para cada fase e também uma mira por fase: com o clique do mouse, o usuário atirava e, se acertasse o pato, que compreende o critério pedido, ganharia 1 ponto, mas se acertasse o pato, que não compreende o critério correto, perderia uma vida.

Figura 15 – Acerte o Pato – frame do jogo no momento em que o usuário estava nos critérios de divisibilidade por 5, se ele mirasse e acertasse o pato com o número 10, ganharia 1 ponto, mas, se mirasse e acertasse o pato com o número 33, perderia um ponto.

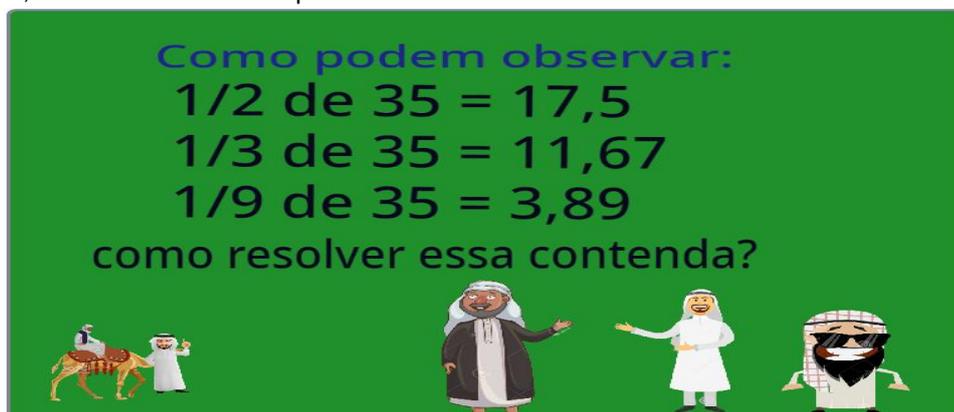


Fonte: <https://scratch.mit.edu/projects/412099569/>

### 5.1.9.4. História dos 35 camelos

P7 e P8 apresentaram sua história sobre o livro o Homem que Calculava de Malba Tahan (ver Figura 16). Nesse projeto, eles abordaram os conceitos matemáticos de fração, divisão, multiplicação e soma.

Figura 16 – História do livro Homem que calculava – trabalho com frações – Nesse *frame*, o matemático fez a divisão dos 35 camelos, que os três irmãos tinham de herança, dividindo os 35 camelos deixados conforme testamento. Arredondando 17,5 para 18; 11,67 para 12, e 3,89 para 4, em que a soma resultaria em 34, sobrando um camelo para o matemático.



Fonte: <https://scratch.mit.edu/projects/412149079/>

Nessa história em especial, os cursistas trabalharam com a matemática a partir da história narrada. Nesse caso, os blocos lógicos eram para simular as ações dos atores, a matemática viria com os frames.

Nesse caso, são abertas possibilidades para se criar histórias no Scratch em qualquer componente curricular.

#### 5.1.9.5. Merilu Bica

P9 e P13 produziram um jogo para abordar conceitos matemáticos de adição e subtração de números naturais.

Após breve introdução com imagens e sons em ambiente de fazenda, com grilos, vacas, campo, insetos, galinha, galo, pintinhos, celeiro, palhas, foi apresentado o objetivo do jogo por um personagem. A Figura 17 reflete parte das imagens do cenário.

O objetivo era bicar o máximo de grilos verdes no tempo de 1 minuto, para tanto, eram utilizadas as setas para movimentar os animais e a tecla “a” para reiniciar o jogo.

A intenção do jogo era justamente trabalhar a soma e a subtração, fazendo com que o aluno entendesse que, quanto mais grilos verdes bicasse, maior seria a pontuação.

Os comandos foram ajustados para que, no momento em que a galinha tocasse o bico no grilo verde, somasse um ponto, mas, quando bicasse o grilo vermelho, perderia dois pontos, conforme Figura 18.

Figura 17 – Merilu Bica – cenário inicial do jogo e apresentação da personagem principal, a galinha Merilu



Fonte: <https://scratch.mit.edu/projects/412146306/>

Figura 18 - Merilu Bica – galinha capturando os grilos (grilos verdes somam um ponto e grilos vermelhos perdem 2) – na parte superior esquerda, estão os pontos e o tempo para a captura do grilo.



Fonte: <https://scratch.mit.edu/projects/412146306/>

Os criadores relatam que o jogo foi criado para ensinar a somar e a subtrair e também auxilia na coordenação motora fina, pois, para movimentar a galinha, é necessário trabalhar as setas do teclado e encostar o bico da galinha no grilo.

#### 5.1.9.6. Jogo do Miko 3.1

O trabalho de P2 e P12 se referia a um conteúdo matemático introdutório sobre figuras geométricas para os anos iniciais, intitulado Jogo do Miko 3.1.

O usuário teria de levar o Miko (macaco) até as figuras pedidas; para isso, eles teriam de usar as setas do teclado (direita e esquerda, para cima e para baixo), o Miko pediria para qual figura desejava ir (Figura 19).

Para cada figura geométrica que o Miko encostasse, o usuário ganharia 10 pontos; se encostasse em figura errada, não perderia pontos. Como exemplo, tomamos a Figura 19, onde o Miko pediu para o usuário levá-lo até os retângulos; se o usuário fosse até o retângulo, ele ganharia 10 pontos, mas se encostasse em outra figura, não perderia pontos, mas também não os ganharia.

Esse jogo é muito interessante, pois, além de trabalhar a geometria, o usuário também trabalha a coordenação motora fina e noções de lateralidade.

Nesse contexto, entendemos que a geometria pode ser trabalhada tranquilamente com o Scratch desde os anos iniciais até o ensino médio, entendendo que o professor necessita realizar as adaptações necessárias de acordo com os objetos de conhecimento (conteúdo) para o ano em que irá ministrar sua aula.

Figura 19 - Jogo do Miko 3.1 – Miko se deslocando até as figuras geométricas que são pedidas pelos balões (comandos criados). Na parte superior esquerda, estão os pontos; e, na superior direita, o tempo.



Fonte: <https://scratch.mit.edu/projects/408749283>

#### 5.1.9.7. Base dez e sexagesimal

P6 e P10 produziram essa história para abordar conceitos matemáticos envolvendo bases numéricas, e como essas bases são aplicadas no mundo real.

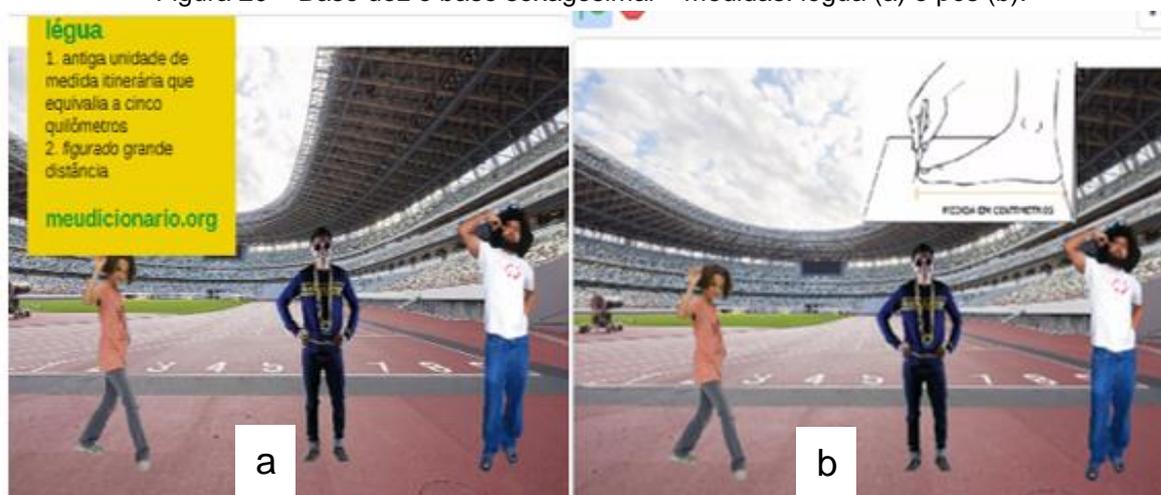
A intenção dessa história era apresentar as diferentes bases numéricas existentes em nosso sistema numérico. Nessa história, foi trabalhada a base 10 e a base 60.

Para isso, criaram uma história entre um Professor e dois alunos; primeiramente, em sala de aula, e, logo após, vão para uma pista de atletismo.

Os cursistas apresentaram algumas formas de medir, por exemplo, léguas (Figura 20 (a)) e pé (Figura 20 (b)). Buscaram na história alguns atletas que fizeram história nos 100 metros, e deram ênfase ao brasileiro Paulo André, que correu essa distância em 9,9 segundos, apresentando uma velocidade média de aproximadamente 10,1 metros por segundo.

Esse trabalho apresentava aos alunos uma narrativa de possibilidades de trabalhar bases diferentes ao mesmo tempo. No caso concreto, a base 10 e a base 60, mas também indicava que, nem sempre, trabalhamos com a base 10, pois existem outras bases, que, mesmo sendo pouco usadas, fazem parte do nosso sistema.

Figura 20 – Base dez e base sexagesimal – medidas: léguas (a) e pés (b).



Fonte: <https://scratch.mit.edu/projects/412004988/>

### 5.1.10 Módulo 10 - Avaliação e encerramento do curso

No último módulo do curso, foi realizada uma avaliação sobre uso do Scratch, buscando evidenciar as dificuldades e potencialidades que os professores perceberam ao longo das atividades de produção de objetos durante o curso.

Foi dada a liberdade para que eles falassem livremente. As falas dos professores, nessa roda de conversa, constituem a base de dados analisadas nesta pesquisa. Complementarmente, eles responderam a um questionário com perguntas

gerais sobre o curso e sobre as atividades desenvolvidas. Essas respostas também foram adicionadas à base de dados analisada.

Finalizada a etapa de produção de dados, iniciamos o processo de sistematização e análise dos dados à luz da ATD. Os metatextos produzidos serão apresentados nas seções que seguem.

## 5.2 CATEGORIA 1: CARACTERÍSTICAS FAVORÁVEIS E DESFAVORÁVEIS DO SCRATCH PARA O ENSINO E APRENDIZAGEM DA MATEMÁTICA E DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL

Para melhor apresentar e discutir os dados sobre essa categoria, dividimo-la em duas subcategorias: características favoráveis e desfavoráveis; as quais são abordadas separadamente abaixo.

### **5.2.1 Subcategoria 1: características favoráveis do Scratch**

As características favoráveis da utilização do Scratch mais evidenciadas pelos professores cursistas foram: interdisciplinaridade, trabalho colaborativo, criatividade, ludicidade, criação de diferentes objetos, interatividade e protagonismo.

#### 5.2.1.1 Interdisciplinaridade

A interdisciplinaridade não foi criada para dividir ou separar disciplinas e colocar cada uma em sua caixa, mas, sim, para manter as individualidades de cada uma, pois estas estão envolvidas em um processo de construção e evolução histórica da vida humana. Isso é destacado pelos Parâmetros Curriculares Nacionais do Ensino Médio (1999).

A interdisciplinaridade não dilui as disciplinas, ao contrário, mantém sua individualidade. Mas integra as disciplinas a partir da compreensão das múltiplas causas ou fatores que intervêm sobre a realidade e trabalha todas as linguagens necessárias para a constituição de conhecimentos, comunicação e negociação de significados e registro sistemático dos resultados (BRASIL, 1999, p.22).

Embora a pesquisa realizada não tivesse o intuito de evidenciar esse movimento durante a formação, e, sim, fazer com que os cursistas percebessem as possibilidades de trabalho favorecidas pelo Scratch, as análises dos dados produzidos evidenciaram o aspecto do trabalho interdisciplinar que a plataforma favorece.

Nesse sentido, os professores cursistas relataram:

O Scratch traz possibilidades muito amplificadas na interdisciplinaridade em sala de aula (P4, Relatório 7).

Sua utilização pode propiciar a união de duas ou mais disciplinas, [promover] a interdisciplinaridade (P12, Relatório 7).

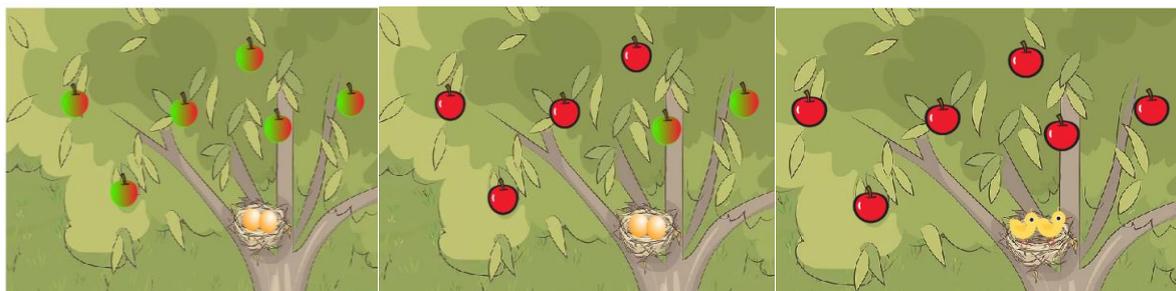
O desenvolvimento de habilidades na Matemática pode ser atrelado a outras disciplinas com o uso do software Scratch, desde que o professor perceba sua aplicabilidade atuando como orientador (P13, Relatório 7).

Posso trabalhar juntamente com o Professor de Geografia, mapas, gráficos e coordenadas (P8, Relatório 7).

Para P4, o Scratch amplia as possibilidades do trabalho interdisciplinar em sala de aula. Segundo ele, o *software* amplia os horizontes nessa perspectiva.

Ao analisar esse excerto de P12, lembramos que, durante o curso, ele sempre pensava em uma atividade interdisciplinar, mas o Scratch fez com que ele vislumbrasse novas possibilidades desse trabalho. Em uma de suas criações com o Scratch, P12 produziu uma simulação sobre o tempo de amadurecimento de maçãs e o nascimento de pássaros no período, conforme ilustrado na Figura 21, que contém alguns frames dessa animação.

Figura 21 – Simulação do amadurecimento de frutos e nascimento de pássaros.



Fonte: <https://scratch.mit.edu/projects/403192271/>

Nessa simulação, as maçãs iniciavam verdes e, enquanto essas maduravam, os ovos, que estavam em um ninho, na mesma árvore, eclodiram fazendo surgir novos pássaros. Nesse trabalho de plano, foram mobilizadas habilidades das ciências naturais e da Matemática.

Observamos que, durante as produções, os cursistas argumentavam que o Scratch poderia ir além da Matemática, como mencionou P13. Contudo, evidenciaram que, para um trabalho mais articulado com outras disciplinas, seria necessário ensinar também aos professores dessas disciplinas trabalharem com o Scratch.

Na experiência de Correa e Santos (2017) com futuros professores, as autoras concluíram que os sujeitos pesquisados conseguiram observar diversas perspectivas

interdisciplinares com o uso do Scratch, assim como ocorreu nessa turma. Já Castro (2017, p. 49) salienta que: “Ao utilizar o Scratch é possível promover a interdisciplinaridade entre a programação e os conteúdos do currículo”. A autora vislumbra a interdisciplinaridade assim como Correa e Santos (2017), na relação programação e conteúdos do currículo.

Ao analisarmos as citações dos professores cursistas, percebemos que eles destacaram outras disciplinas para desenvolver um trabalho de interdisciplinaridade com a Matemática. Segundo os cursistas P6 e P10, as produções com o Scratch, voltadas para a Matemática, também poderiam contemplar aspectos da disciplina de Educação Física e Ciências (P6 e P10). Um dos projetos deles envolveu uma pista de atletismo para trabalhar a base dez e a base sexagesimal. A história envolvia várias atletas de vários países. Nesse caso, buscaram elementos da área da Educação Física para contextualizar os temas matemáticos que pretendiam explorar. Na Figura 22, serão apresentados alguns frames do objeto criado no Scratch.

Figura 22 – Projeto Base dez e base sexagesimal - frame



Fonte: <https://scratch.mit.edu/projects/412004988/>

Em acordo com os professores cursistas, também consideramos que o Scratch é um *software* interdisciplinar, por essência. Como destacado na Seção 3.1 deste trabalho, a plataforma Scratch foi criada com a ideia de misturar, mixar criações, mídias, imagens, animações e fotos, mas como foi evidenciado, neste trabalho, também pode mixar conhecimentos de várias áreas.

#### 5.2.1.2 Trabalho colaborativo

Um dos pilares da metodologia ABP utilizada no curso, o trabalho colaborativo, foi favorecido pelo Scratch, principalmente pela plataforma *on-line*, como fica evidente nas falas de alguns professores:

Eu vou fazer igual aquele do P3 e P4, o projeto dos patinhos que trabalhava os critérios de divisibilidade, aquele projeto da P12 e do P2 achei muito legal.

**Eu fui na página do Scratch ver como eles fizeram** (P13, Entrevista, destaque nosso).

[o Scratch] Cria um ambiente de discussão com foco no planejamento de um programa em que crianças e jovens se ajudam buscando desenvolver e conhecer soluções cada vez melhor (P7, Relatório 7).

No excerto, é possível notar que P13 foi na plataforma *on-line* do Scratch para visualizar as construções que seus colegas tinham produzido, bem como para procurar por outras que poderiam auxiliá-lo com a produção do seu objeto.

Para favorecer esse processo de compartilhamento das produções, foi criada uma sala virtual na plataforma Scratch para Educadores, na qual todos os professores cursistas tinham acesso às produções que os outros colegas estavam produzindo.

P14 visualizou que, devido à facilidade de se observar projetos já construídos e disponibilizados na plataforma *on-line* Scratch, como jogos, histórias e simulações, os alunos poderão descobrir processos envolvendo o uso de conceitos matemáticos, evidenciando o trabalho colaborativo da plataforma Scratch e sua importância na construção de outros objetos.

Também, nesse sentido, P7 evidenciou o ambiente colaborativo proporcionado pela plataforma Scratch e destacou a importância dessa característica para o desenvolvimento de projetos mais complexos, o que pode ser interpretado como uma contribuição para o desenvolvimento de habilidades do pensamento computacional.

Essa característica de trabalho colaborativo, favorecido pelo Scratch, conforme apontado pelos professores participantes da pesquisa, também foi destacado por Silva (2015, p.19), que destaca que o “Scratch [...] proporciona ao estudante uma inserção e uma interação com a sociedade por estimular o trabalho colaborativo e facilitar o desenvolvimento e a comunicação de ideias”.

De modo geral, percebemos que os professores participantes se valeram muito dessa característica da plataforma para melhorar seus projetos, principalmente com relação à programação, desenvolvendo habilidades do pensamento computacional. Por esse motivo, e projetando o uso pedagógico do Scratch com seus alunos, os professores apontaram que tal característica é favorável para os alunos desenvolverem, principalmente, as habilidades do pensamento computacional.

#### 5.2.1.3 Produções de diferentes objetos

Durante o curso, foi pedido aos cursistas a produção de três artefatos: história, simulações e jogos, mas os objetos criados, dentre os artefatos, foram os mais variados no campo da Matemática.

Foram feitas produções de objetos para o trabalho com geometria (ângulos, figuras geométricas), bases numéricas, as quatro operações, raciocínio lógico, critérios de divisibilidade, história da matemática, matemática financeira, valorização e desvalorização, quizes matemáticos, potências, plano cartesiano etc.

Seguem algumas das falas dos cursistas a esse respeito:

Nos artefatos produzidos conseguimos trabalhar os seguintes conteúdos: Na produção da história trabalhamos o conteúdo de teorema de Tales. Na simulação dos carros dá para trabalhar velocidade média, tempo e distância. No jogo trabalhamos alguns critérios de divisibilidade (P3, Relatório 7).

Na história, trabalhamos os conceitos de potenciação, podendo ser ampliado para sequências numéricas e equações exponenciais. Na simulação os conceitos principais foram os volumes e áreas de cubos. No jogo o conceito explorado foi de figuras geométricas planas – triângulo, quadrado, retângulo e círculo (P12, Relatório 7).

Os artefatos que produzimos, utilizamos conceitos matemáticos, que podem ser trabalhados em sala de aula (P14, Relatório 7).

Nas explicações dos cursistas, podemos verificar que eles citaram vários objetos que podem ser produzidos para o trabalho da Matemática com a programação do Scratch.

P3 relata os conteúdos que podem ser trabalhados com cada objeto produzido durante o curso; P12 segue a mesma linha dos demais, apresentando os artefatos que geraram objetos digitais de aprendizagem e, conseqüentemente, os conteúdos matemáticos.

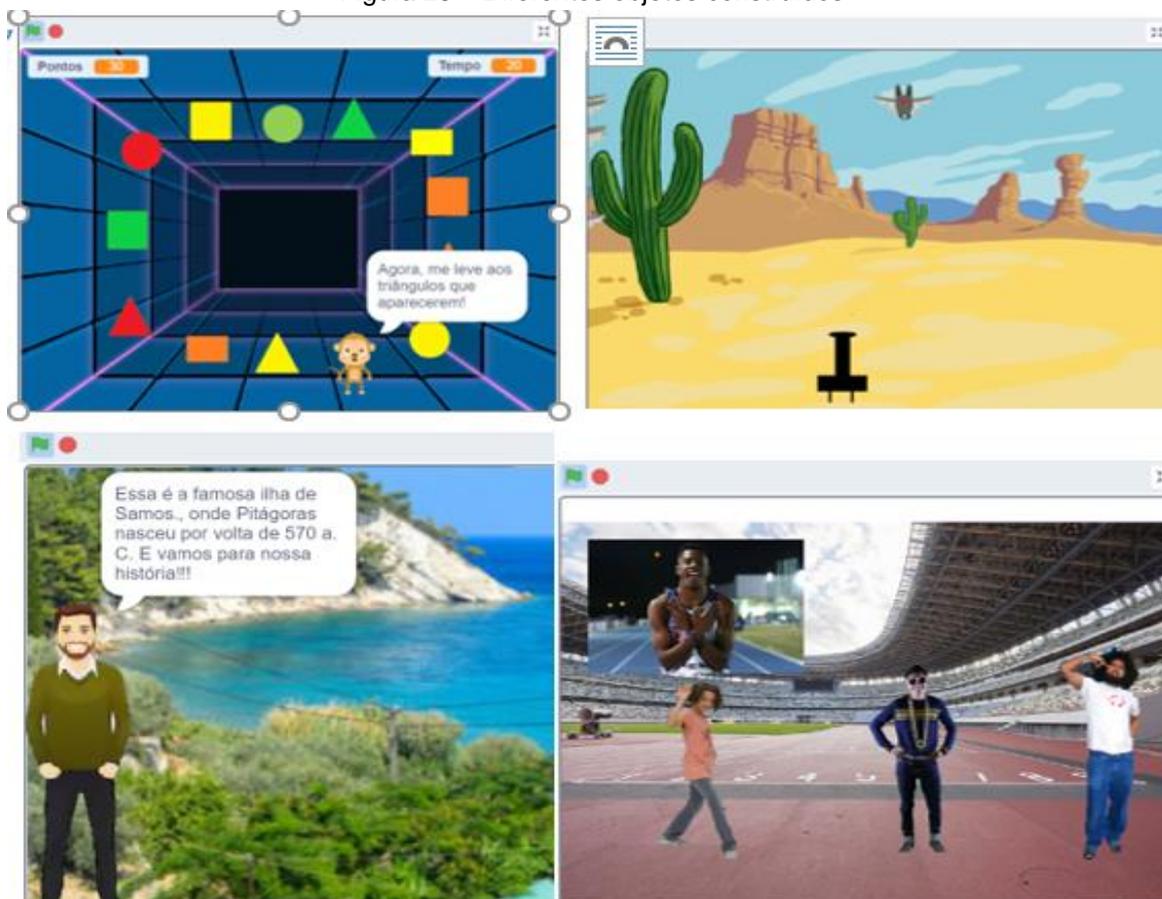
Por fim, P14 relacionou os objetos produzidos com o seu trabalho em sala, relatando que esses objetos podem ser trabalhados no ambiente, desde que planejado com antecedência, preparando os alunos para o trabalho criativo; a figura 23 ilustra alguns objetos produzidos durante o curso.

Os autores Souza e Costa (2018), ao descreverem o Scratch, afirmam que ele, “É uma ferramenta voltada para a aprendizagem a partir de projetos e da construção de objetos virtuais significativos pelos alunos e, portanto, contribui para uma aprendizagem também mais significativa” (SOUZA; COSTA, 2018).

Realmente, entendemos que as produções variadas contribuem para uma aprendizagem significativa, pois cada objeto tem sua significação, e os professores podem trabalhar conteúdos diferentes, mas, com os alunos realizando as produções, nesse momento, entra a aprendizagem significativa. São apresentadas, na Figura 23, algumas produções que revelam essa variedade significativa de aprendizado que podem ser realizadas na matemática com o Scratch.

O Scratch é uma ferramenta poderosa para a produção de objetos digitais de aprendizagem. Observamos, durante o curso, a riqueza de produções criadas pelos professores para o trabalho matemático em sala de aula.

Figura 23 – Diferentes objetos construídos



Fonte: Site do Scratch

#### 5.2.1.4 Ludicidade

O Scratch é um *software* lúdico, que possibilita aos produtores aprenderem brincando. Nesse sentido, os professores salientam:

Achei interessante a parte lúdica do programa, onde é possível chamar a atenção do aluno com personagens, sendo possível inserir áudios, mudar cenários (P3, Relatório 7).

Achei muito interessante para se trabalhar coordenadas a parte lúdica com os alunos (P4, Relatório 6).

Como é um *software* interativo, o Scratch chama a atenção dos alunos. Quanto mais lúdico for, mais atrativo a aula será (P4, Relatório 7).

As vantagens é que no Scratch pode-se trabalhar de uma forma mais lúdica e divertida, quebrando um pouco o método tradicional, onde o programa faz com que o aluno aprenda a interpretar uma situação problema e após saber resolver (P5, Relatório 7).

P3 entende que o lúdico pode chamar a atenção do aluno, pois, com a parte criativa e desafiadora, que é inerente às produções no Scratch, o aluno pode buscar áudios e cenários de acordo com sua necessidade de aprendizagem.

P4 vê que as coordenadas podem ser trabalhadas de forma lúdica, e o programa já traz um cenário específico com os fundos das coordenadas e malhas quadriculadas, basta somente fazer com que o aluno navegue, conforme Figura 24.

Para P5, o trabalho com o Scratch pode ocorrer de forma lúdica e divertida, quebrando um pouco o método tradicional, que faz referência à lousa, ao canetão ou giz, no qual não há uma ação criativa por parte do aluno, que apenas recebe as informações e as armazena. Para ele, o Scratch ajuda a ter essa relação com a ludicidade, apoiada na criação, interpretação e resolução.

Eles entenderam que essa característica irá atrair os alunos para o aprendizado da Matemática e do PC, pois os alunos querem e tem o interesse de criar, e o Scratch é uma fonte para que essa habilidade seja aperfeiçoada por meio do lúdico.

De forma geral, a ludicidade foi apontada pelos professores como uma das vantagens do Scratch, vários cursistas relataram, nos encontros síncronos, como foi bom o trabalho com o Scratch: “você aprende brincando sem ver a hora passar”.

Os cursistas criaram trabalhos, que exigiram horas em frente ao computador, e afirmaram que não se sentiram cansados, já que a forma lúdica e os desafios propostos de aprender a manusear e a criar projetos os instigavam a permanecerem conectados mais tempo para alcançar o objetivo.

P3 salienta que “estava trabalhando no projeto já fazia umas 3 horas e nem percebi o tempo passar”.

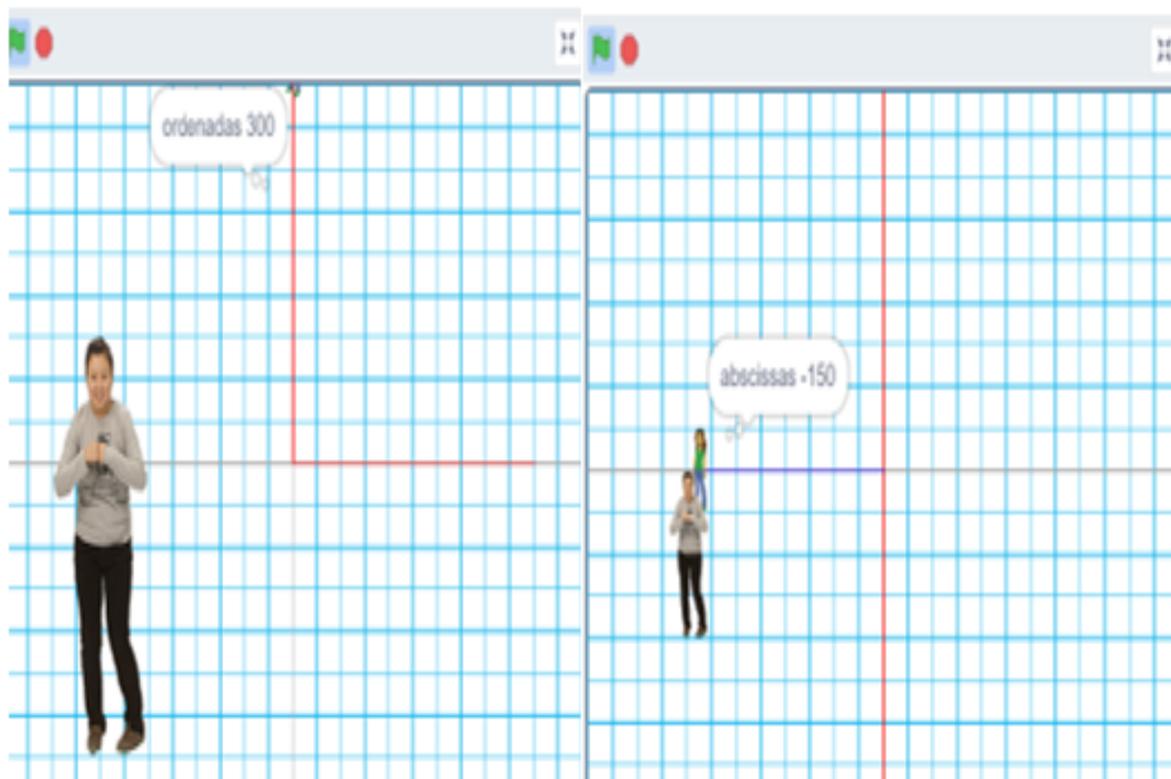
Gordinho (2009) afirma que,

O Scratch como interface de comunicação e ludicidade, aponta para a possibilidade de desenvolvimento de novas competências cognitivas, e novos laços de entajuda e novas relações adulto/criança. Sendo assim, o Scratch é um elemento catalisador, capaz de contribuir para criatividade, desinibição, espontaneidade, socialização, comunicação, ludicidade, entre outros (GORDINHO, 2009, p. 96).

Podemos perceber que a fala dos cursistas coaduna com a teoria sobre o Sscrth.

Por esses motivos, apontamos a ludicidade como uma das características vantajosas do Scratch para o trabalho dos professores em sala de aula.

Figura 24 – Trabalhando coordenadas com o Scratch



Fonte: Site do Scratch

#### 5.2.1.5 Criatividade

Quando os professores do curso se depararam com o Scratch não sabiam o que fazer, e até argumentavam “o que vamos fazer com esse *software*” (vários cursistas).

Na aula introdutória ao *software*, alguns cursistas pensaram em desistir por pensar que não seria possível aprender e realizar os projetos que seriam pedidos no curso, “Professor, sou muito perfeccionista, e ao me deparar com o Scratch, pensei que não iria conseguir produzir, então pensei em desistir no início” (P13). Todavia, logo que fizeram as primeiras produções, que foram com figuras geométricas, observaram que poderiam criar, e isso foi observado durante o curso: a evolução das criações cada vez mais aguçadas, com maior número de códigos, até mesmo as notas do Dr. Scratch foram aumentando com o tempo. Nesse ponto, P13 já estava com outro pensamento “Professor amei fazer as figuras geométricas, fiz de 2 formas” (P13). E os outros disseram:

Propicia a utilização de metodologias ativas, a autonomia e criatividade dos alunos, tanto na construção como na utilização de artefatos produzidos por meio do mesmo [...] O usuário pode desenvolver sua criatividade, raciocínio lógico tudo isso atrelado ao pensamento computacional (P12, Relatório 7).

Podemos realizar a união desse *software* com outros, para a criação de vídeos, por exemplo, instigando a criatividade de cada um (P7, Relatório 7).

Estudar de forma agradável, reconhecer comandos, desenvolver raciocínio lógico matemático, autonomia ao aluno de criar, desenvolver habilidades computacionais (P13, Relatório 7).

Criação de situações Matemáticas que colaboram com a aprendizagem dos estudantes, como jogos, histórias e simuladores (P14, Relatório 7).

No que tange à criatividade, P12 fez uma simbiose entre criatividade, metodologias ativas, autonomia, raciocínio lógico e PC. Ele entendeu que, para criar com o Scratch, não existe hierarquia entre esses, pois essas características são inseparáveis.

P7 vislumbrou que as criações no Scratch podem ser colocadas em outros *softwares* para a produção de vídeos, instigando os seus usuários a uma maior capacidade de criação, pois possibilita unir os trabalhos com o Scratch a outros softwares para produção de vídeos.

Nesse prisma, tivemos produções, apresentadas no módulo 9, com mais de 20 atores, vários cenários diferentes (inclusive alguns cenários foram produzidos pelos próprios cursistas), músicas, sincronização, falas, desenhos feitos pelos próprios cursistas e implementados na produção do Scratch.

Silva (2019) relata que “dessa forma, o *software* aliado a ações previamente elaboradas é uma forma de os discentes irem além das produções feitas no papel, pois atua na subjetividade e criatividade” (SILVA, 2019, p. 145).

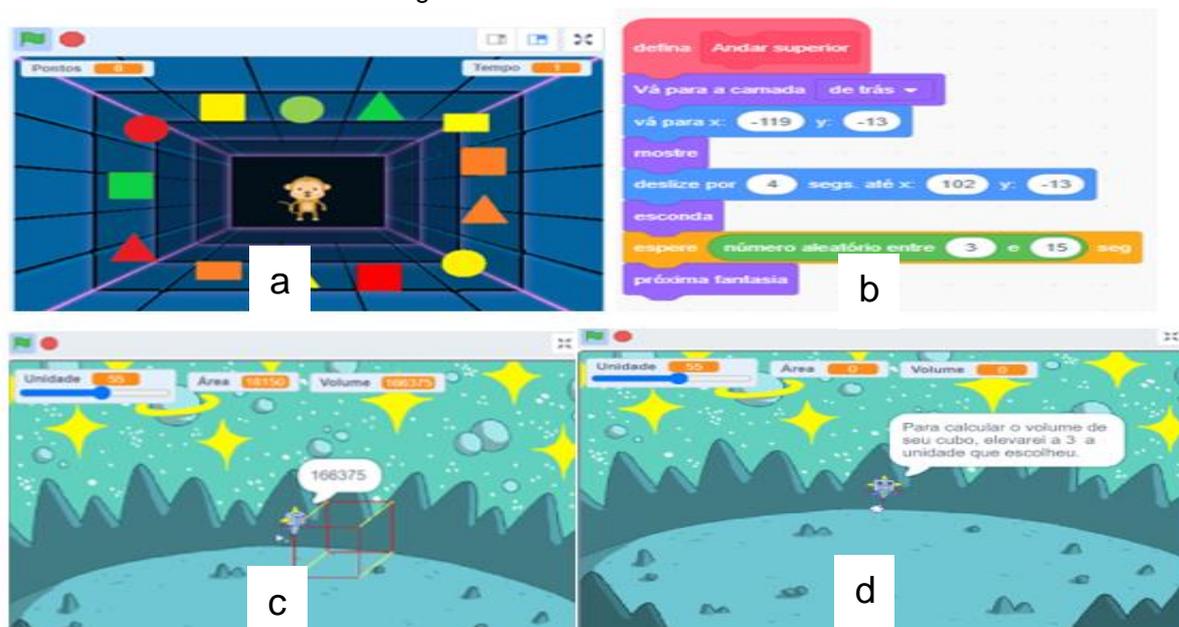
Na Figura 26, são apresentadas as formas criativas que os cursistas realizaram em seus trabalhos: uma dupla pensou em um *quiz* com o recruta zero, enquanto a outra partiu para os cálculos de área, volume de um cubo a partir da unidade (comprimento do lado) que pode ser escolhida através do controle deslizante.

Podemos observar o quão criativo é o trabalho com o *software*., a primeira imagem apresenta o Miko que foi levado para as figuras geométricas pelo usuário (Figura 25 (a)); a segunda demonstra a criatividade do programador ao realizar o bloco de comandos “andar superior”, incorporando outros comandos (Figura 25 (b)); a terceira (Figura 25 (c)) e a quarta (Figura 25 (d)) apresenta o Robô, construindo um cubo e demonstrando o número de arestas e o volume a partir da unidade de medida dada pelo usuário.

Em outras palavras, o Scratch aguça a criatividade das pessoas que vão produzir nele, pois o criador parte da imaginação e necessita colocá-la em prática

através da criação de comandos, sons, atores, cenários, figuras, conteúdo, todos eles sincronizados para a apresentação.

Figura 25 – Criatividade no Scratch



Fonte: Site do Scratch

A partir de um ator e de um fundo, utilizando a criatividade, os cursistas construíram produções de conteúdos matemáticos por meio de artefatos jogos, história e simulação explorados pelos *Scratchers*.

#### 5.2.1.6 Interatividade

Para Lemos (2002, p. 119), a interatividade “representa uma nova qualidade de interação a que define como ação dialógica entre o homem e os objetos tecnológicos”. Assim, de acordo com o autor, o curso de formação foi relevante em aspectos de interatividade, já que os professores cursistas que, no início do curso não conheciam a ferramenta Scratch, durante o curso, iniciaram suas produções e, ao final, conseguiram trabalhar um plano de aula de Matemática para a apresentação.

Por esse aspecto, podemos afirmar que o curso foi interativo, já que houve a dialógica entre homem e os objetos digitais, conforme afirma Lemos (2002). Entretanto, há toda uma rede de interatividade, mesmo que a que ocorre com o usuário seja a mais visível, há também interatividade de personagem com o cenário, sons, comandos, diálogos, movimentos, todos coordenados e acontecendo simultaneamente, o que Silva (2019) chama de paralelismo.

Os cursistas entenderam da seguinte forma a interatividade,

Achei muito simples, fácil, o *layout* dele que é bem interativo você pode pegar um desenho animado e colocar ele como ator (P8, Entrevista).

Por ser um *software* interativo, o Scratch chama a atenção dos alunos (P4, Relatório 7)

Apareceram: Interação com o usuário, Representação de dados, Controle de fluxo e Abstração (P11, Relatório 5).

P8 acredita que, pelo *layout* ser interativo, pode-se pegar um desenho animado, colocar nome no ator, ou seja, ele entende que o Scratch produz por interação, o que possibilita criar os personagens que se desejar e fazê-los interagir com o usuário, pois se torna mais significativa o estudo.

P4 também afirmou que o Scratch é interativo e, por ser um *software* que trabalha com programação, oferece uma linguagem básica e, quanto mais o criador for habilidoso e experiente em criações, mais interativo será o trabalho.

P11 apresenta os conceitos que apareceram no Dr. Scratch no seu projeto, segundo ele, a interatividade com o usuário apareceu no seu projeto na avaliação do Dr. Scratch.

O que os cursistas relataram está de acordo com Rocha (2018), pois ele “revelou que todos os projetos apresentavam características de interatividade, o que sugere que o Scratch é uma ferramenta propícia para programação de objetos de aprendizagens interativos” (ROCHA, 2018, p. 106).

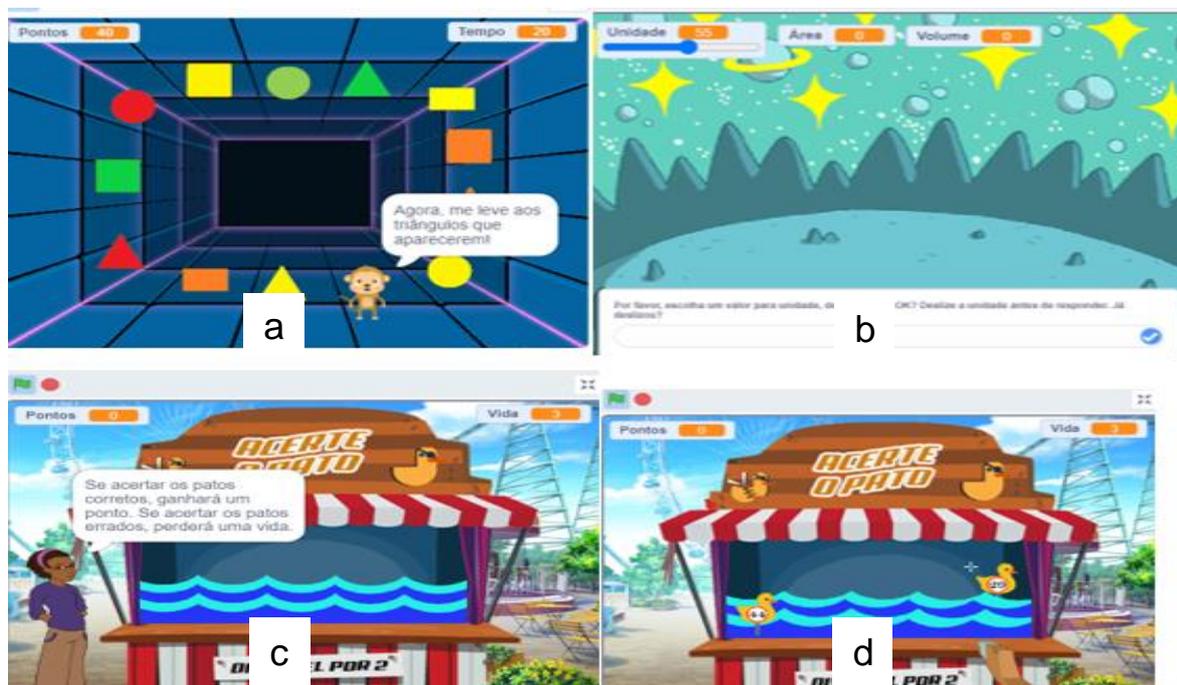
Entendemos da mesma forma que Rocha (2018) que o Scratch é uma ferramenta que proporciona interatividade, seja com os elementos internos, com os criadores seja com os usuários.

Dos 7 trabalhos apresentados no curso de formação que realizamos, todos apresentaram interatividade com o usuário, alguns menos e outros mais, conforme Dr. Scratch avaliou.

Na Figura 26, são apresentados alguns exemplos de interatividade produzidas pelos cursistas em seus objetos digitais durante o curso. Podemos observar que, no Jogo do Miko 3.1, o ator pede para o usuário levá-lo aos triângulos (Figura 26 (a)); na simulação Volume e área do cubo, o ator pede para o usuário escolher o tamanho da unidade (Figura 26 (b)), no jogo Acerte o pato, primeiro a atriz apresenta informações sobre o jogo (Figura 26 (c)) e, depois, inicia o jogo com o usuário tendo de acertar os patos corretos (Figura 26 (d)).

Os cursistas entenderam que a interatividade foi uma característica bem explorada durante a formação, já que eles trabalharam nos projetos essa característica, como pode ser observado nos excertos e na Figura 28, que ilustra essa afirmação.

Figura 26 – Interatividade do usuário com o Scratch. A Figura 28 (a) apresenta o Miko pedindo para o usuário levá-lo aos triângulos; a Figura 28 (b) pede para o usuário colocar a unidade para calcular a área e o volume; já a Figura 28 (c) informa os usuários as regras do jogo Acerte o Pato, enquanto o *frame* (d) apresenta o usuário jogando e tentando acertar o pato correto.



Fonte: Site do Scratch

#### 5.2.1.7 Protagonismo

Outra característica favorável do Scratch, que os cursistas relataram, e utilizada durante o curso, foi o protagonismo. Eles se sentiram protagonistas e já projetaram o trabalho em sala conforme excertos abaixo.

Destacamos o protagonismo do estudante na construção dos artefatos propostos, fomentamos o desafio do acesso da ferramenta ao estudante (P10, Relatório 7).

Acredito que o Scratch pode ser utilizado nas aulas, pois é uma ferramenta divertida e até simples para criar uma história ou simulador, mas tenho alunos com bastante habilidade com tecnologia e faria jogos excepcionais, com o protagonismo dos alunos, tenho certeza disso (P13, Relatório 7).

A utilização do Scratch para o ensino é uma excelente ferramenta para potencializar o aprendizado, por ser desafiador e de acesso aos estudantes, pois os mesmos já estão familiarizados como o meio digital, eles serão os protagonistas (P14, Relatório 7).

A intenção do trabalho é o aprendizado, vamos fazendo a mediação, mas os alunos terão a autonomia para produzir, e não o professor (P7, Entrevista).

De forma geral, os professores, ao se depararem com as construções que produziram durante o curso no trabalho com o Scratch, projetaram em seus alunos a criatividade e o protagonismo para o trabalho em sala.

Pelos excertos, percebemos que dialogaram com essa percepção, já que P10, P13, P14 e P7 citaram que os alunos serão protagonistas, revelando que o Scratch proporciona essa característica, pois, mesmo que os professores peçam um conteúdo, o executor de todo o projeto será o aluno.

Para Sobreira, Takinami e Santos (2013), o processo de construção do Scratch tem uma característica cíclica, que parte do imaginar, criar, corrigir, partilhar, refletir e voltar ao ciclo para criá-lo novamente. Segundo as autoras, esse processo acaba “colocando o aluno num papel singular no processo de aprendizagem e de construção do conhecimento, mediado por computador. Este papel promove essencialmente a autoria e o protagonismo” (SOBREIRA; TAKINAMI; SANTOS, 2013, p. 135).

Esse processo citado por elas (2013) sintoniza com o relato de P4 quando ele afirma que: “para terminar o processo tinha que dar uma volta, refletir e depois conseguia trabalhar novamente nele”.

Diante do exposto, verificamos que os professores consideraram que o Scratch favorece o protagonismo, pois partiram do projeto baseando-se em ideias autônomas de criação, e, assim, exerceram seu protagonismo.

## **5.2.2 – Subcategoria 2: características desfavoráveis/desvantagens do Scratch**

Os cursistas apontaram algumas desvantagens que o Scratch possui, dentre as mais importantes, citaram o tempo necessário para as construções, pouco espaço para visualizar os comandos e a falta de recurso para pausar a execução de um código. Outros aspectos também foram apontados durante o curso, principalmente na fase inicial, na qual os professores ainda não estavam muito familiarizados com o *software*, como, por exemplo: otimizar códigos, curso no início, sincronização, trocas de cenário e a música que não mudava e também trabalhar com os operadores.

### **5.2.2.1 – Tempo necessário para as construções**

O tempo foi apontado pelos cursistas como fator necessário para as produções. P7 relatou que o tempo para preparar a atividade pode ser um dificultador, pois

Um vídeo de 10 minutos, a gente pode gastar aí cinco seis dias para fazer essa produção. Mas a intenção não é essa, a intenção do trabalho é o aprendizado, vamos fazendo a mediação, mas os alunos terão a autonomia para produzir, e não o professor (P7, Entrevista).

A fala de P7 revela que o tempo pode ser uma desvantagem ao produzir um vídeo, por demorar vários dias para se construir uma produção de 10 minutos, mas também ressalta o protagonismo dos alunos, na perspectiva de uso do Scratch, com o professor como mediador do processo.

P5 faz o seguinte relato,

Praticamente para ele fazer o que a gente fez nesses dois meses de curso, ele vai levar 4 a 6 meses, para ele pegar fotos, ele ver essa questão do palco de cenário, de colocar voz no ator entendeu sim, o que vejo na escola é isso, os alunos são muito rápido se tiver tempo e computador eles vão pegar fácil (P5, Entrevista).

Esse é um fato que temos de observar, pois, no início do relato, ele acreditava que o tempo do aluno seria maior do que o do professor, mas, no final, ele ressaltou que, com as condições necessárias, os alunos irão aprender rápido.

P5 se junta a P7, no que tange ao tempo, pois, eles atribuíram ao tempo uma desvantagem, entretanto, eles relataram que, se a escola tiver computador e o próprio tempo para o trabalho, os alunos conseguirão fazer a atividade, pois eles aprendem rápido.

Acreditamos que a questão do tempo seja, sim, um empecilho, principalmente, no início de qualquer aprendizado. Contudo, devido às produções irem se tornando cada vez mais complexas, trabalhar a sincronização, mesmo para o *Scratcher* experiente, vai levar mais tempo em suas produções, mas, concordando com P7, o importante é o aprendizado e o Scratch permite a formação de muitas habilidades.

#### 5.2.2.2 – Pouco espaço para visualizar os comandos

Às vezes, alguns atores necessitam de mais comandos na área de produção de *Scripts*, pois são eles que dão andamento às produções quando o jogo tem níveis, fases e outras sequências, como revela o Jogo do Miko 3.1 na Figura 27.

Na Figura 29, estão presentes alguns comandos do Miko que é o ator principal do jogo, a tela de produção de *Scripts* fica muito extensa, haja vista que esses comandos são somente para esse ator.

Como os comandos ficam extensos, o iniciante sente dificuldade para os visualizar e, por vezes, um toque equivocado pode mudar o projeto. Isso fica evidenciado nas palavras dos cursistas abaixo.

A Montagem dos blocos também eu penso que tem que melhorar, pouco espaço na visualização, faz a gente sofrer, o quanto mais prático for o programa, melhor para aplicá-lo em um determinado momento com um aluno em um projeto de uma aula mais interativa, então eu acho que pode sim ser melhorar esses aspectos, deixar de forma mais organizada (P3, Entrevista).

Nos jogos lá tinha um bloco que o P3 mexeu sem querer no mouse ele foi para cima de outro bloco, aí nós não sabíamos onde estava, depois estava faltando bloco, muito apertado para trabalhar lá, não sabia onde que estava faltando (P3, Entrevista).

P3 salientou que pode ser melhorada a questão da visualização dos blocos, para tornar o programa mais prático e atrativo para trabalhar nas aulas, pois o seu intuito é trabalhar o Scratch em forma de projeto com os alunos, e, para isso, é necessário que os criadores e desenvolvedores do Scratch pensem uma forma de expandir essa visualização.

P4 relata que, durante a construção dos jogos, ao rodar os projetos, observaram que não acontecia conforme eles pensavam. Quando voltavam na programação, percebiam que o espaço era estreito para a tela de produção de scripts, o que dificultava a visualização dos comandos. Como exemplo, juntamos alguns comandos do Jogo do Miko 3.1, na Figura 27, para explicitar os relatos dos cursistas.

De forma geral, os cursistas relataram que a tela de produção de *scripts* era de difícil visualização e, por ter comandos muito próximos, dificultava as produções, principalmente, no momento de rever e organizar as produções.

Figura 27 – Alguns comandos do Jogo do Miko 3.1



Fonte: <https://scratch.mit.edu/projects/408749283>

### 5.2.2.3 – Pause para visualização

Nos trabalhos desenvolvidos, os cursistas apontaram que não conseguiam realizar a visualização do projeto enquanto ele rodava, pois tinham de assistir o projeto todo. Nos projetos com introdução longa ou com jogos de níveis, ou com histórias com longas conversas, eles tinham dificuldades de assisti-los.

Eles afirmaram que sabiam que os blocos de comandos pertenciam a uma parte, e essa parte poderia ser melhorada, mas, quando iam rodar o projeto e assisti-lo, o programa não dava a opção de pausar e o projeto voltava ao início, é o que observamos nos relatos de P12, P9 e P10:

Tive muita dificuldade no momento de assistir meu projeto, pois sempre voltava no início para ver os erros (P12, Entrevista).

Eu me perdia um pouco quando ia analisar o projeto, pois não tinha parada (P9, Entrevista).

Para ser sincero, acabei o curso e não consegui entender uma forma de pausar o projeto, tinha que assistir ele todo, pois só parava no botão vermelho (P10, Entrevista).

P12 revelou que sua dificuldade era voltar ao início para ver os erros, pois, mesmo sabendo que, se consertasse o erro, ele tinha de parar, e o Scratch não tem *pause* para rodar a partir dali.

Já P9 disse se perder dentro do projeto, pois, quando ia arrumá-lo sempre mudava alguns comandos. Ele acrescentou que, se houvesse o *pause*, seria mais fácil realizar essas retomadas.

P10 reconheceu que a única forma de parar os projetos seria apertar o botão vermelho, e, como os demais cursistas, ele disse que, quando encontrava um erro, assistia ao vídeo todo ou clicava no botão vermelho.

Realmente, é complexo o trabalho com o Scratch no sentido de visualizar o projeto, porque, quando necessitava pausar, não havia essa possibilidade. Não encontramos uma forma para ajudar os cursistas. Nesse sentido, eles teriam mesmo de assistir o projeto até o final. Apresentamos que, nos blocos de comando, pode-se ver o erro, isso ajudou, mas, de forma geral, eles teriam de assistir o trabalho para visualizar onde poderiam melhorá-lo.

### 5.2.2.4 – Outras desvantagens

Os cursistas apontaram algumas outras desvantagens do Scratch, e, embora entendemos que, para cada nível, existe uma dificuldade, é importante apresentá-las

para que outros trabalhos possam vislumbrar o que os iniciantes terão de dificuldades, porque o trabalho em sala de aula com estudantes pode apresentar essas características.

A sincronização foi apontada por P5 que salientou “ah eu acho que isso aqui é uma desvantagem dele, cara, eu acho que os desenvolvedores deveriam facilitar mais essa sincronização, que se perde muito tempo para sincronizar, uma história eu levei 3 a 4 horas para montar tudo é muito tempo” (P5, Entrevista).

Nessa fala de P5, observamos que projetos mais simples demoram pouco tempo, mas, quando os *Scratchers* vão realizando criações mais complexas, demanda um tempo maior. Nesse projeto, ele utilizou voz, imagens, sons, e, realmente, isso requer um tempo maior para a sincronização.

Outro ponto, agora apontado por P13, foi o trabalho com operadores, “o trabalho com operadores foi complicado para mim. Consegui utilizar alguns, mas foi complexo” (P13, Entrevista).

Realmente, até na análise do Dr. Scratch (lógica), o trabalho com os operadores “e, ou”, foi empecilho para as notas dos projetos, mas acreditamos que, com a construção de mais projetos, essa desvantagem possa ser sanada, pois, dentro do curso, já obtivemos êxito nesse critério com outros cursistas.

Outra desvantagem apontada foi a mudança de cenário e a continuidade da música: “Eu estava trabalhando com músicas e outros áudios, eu não conseguia fazer a música parar no momento da troca de cenário” (P11, Entrevista).

Esse relato de P11 ocorreu na criação de sua primeira história, quando houve a dificuldade para trocar cenário e a música parar, nessa troca; foi um empecilho, no momento, mas, depois, ele conseguiu realizar essa tarefa com certa facilidade.

Assim, no início do curso, é imprescindível a atenção aos cursistas, pois, como já anunciamos, alguns pensaram em desistir, mesmo antes de conhecer o Scratch e suas características, por pensarem na dificuldade, conforme podemos observar nestes relatos:

Professor, sou muito perfeccionista, e ao me deparar com o Scratch, pensei que não iria conseguir produzir, então pensei em desistir no início (P13).

Professor de Deus, o início foi muito complexo, fui construir meu primeiro objeto e não conseguia (P9).

O início foi desgastante, queria entender e não conseguia (P1).

O início de qualquer aprendizado é complexo, e o Scratch, como viemos apresentando em relação ao PC e à Matemática, foi bem desafiador, mas, conforme

o curso foi se desenrolando, os cursistas apresentaram um desenvolvimento satisfatório e esses percalços, observados no início, desapareceram.

### 5.3 – CATEGORIA 2: HABILIDADES MOBILIZADAS DA MATEMÁTICA E DO PC COM O USO DO SCRATCH

Nesta seção, elencaremos as habilidades mobilizadas, tanto da Matemática quanto do pensamento computacional, as quais os professores evidenciaram durante as produções de artefatos com o Scratch.

Com relação às habilidades da Matemática, analisamos aquelas elencadas pelos professores em seus planos de aula, elaborados como uma atividade dentro do curso, os quais nortearam a produção dos artefatos no Scratch. Ao todo foram sete planos de aula.

Complementarmente, destacamos outras habilidades que, a nosso ver, também foram mobilizadas, mesmo que os criadores do objeto não as tenham elencadas inicialmente em seu plano.

#### **5.3.1 – Subcategoria 2.1: habilidades mobilizadas da Matemática com o uso do Scratch**

Nesta seção, evidenciaremos as habilidades da Matemática, de acordo com a BNCC, as quais os professores mobilizaram em cada um dos projetos/artefatos criados com o Scratch. Optamos por apresentar as habilidades de cada projeto, individualmente.

##### 5.3.1.1. Projeto Função Exponencial

No projeto de P5 e P14, que consistia numa história sobre a negociação de um caminhão (ver Seção 5.1.9 – item 1), o objeto do conhecimento abordado foi a função exponencial, usada para avaliar a desvalorização de um caminhão ao longo do tempo e orientar o comprador (personagem da história) a tomar a decisão mais acertada.

De acordo com o plano de aula, elaborado pelos professores autores desse projeto, as habilidades mobilizadas da Matemática são as que seguem no Quadro 9.

Quadro 9 – Função exponencial – habilidades mobilizadas

Habilidade	Descrição da habilidade
(EM13MAT301)	Resolver e elaborar problemas do cotidiano, da Matemática e de outras áreas do conhecimento, que envolvem equações lineares simultâneas, usando técnicas algébricas e gráficas, incluindo ou não tecnologias digitais.
(EM13MAT303)	Resolver e elaborar problemas envolvendo porcentagens em diversos contextos e sobre juros compostos, destacando o crescimento exponencial.
(EM13MAT304)	Resolver e elaborar problemas com funções exponenciais nos quais é necessário compreender e interpretar a variação das grandezas envolvidas, em contextos como o da Matemática Financeira e o do crescimento de seres vivos microscópicos, entre outros.
(EM13MAT406)	Utilizar os conceitos básicos de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.

Fonte: Plano de aula de P5 e P14

Podemos observar que as habilidades apontadas pelos professores são relativas ao ensino médio, com ênfase no objeto do conhecimento Funções de Uma Variável Real, do tipo Exponencial.

É possível visualizar a primeira habilidade 301, na imagem retratada pela Figura 28 (a), no que diz respeito ao caminhão A.

Já a habilidade 303, é possível observá-la na mesma Figura 28 (b), quando o vendedor relata a porcentagem de desvalorização ao mês.

A habilidade 304 é representada também na Figura 28 (c), quando os professores trabalharam a questão da função exponencial.

E, na Figura 28 (d), está representada a habilidade 406, apresentando os comandos para que a história Matemática fosse concretizada.

Como supracitado, essa separação é apenas didática, já que, durante o curso, vislumbramos que existe o elo entre pensamento computacional e a Matemática, e, conforme o curso foi se aprofundando, os cursistas praticamente assimilaram essa ligação.

Figura 28 – Projeto função exponencial – matemática, PC com programação envolvidos no trabalho – Figura 28 (a) representa a habilidade 301, a Figura 28 (b) representa a habilidade 303, a Figura 28 (c) a habilidade 304 e a Figura 28 (d) a habilidade 406 de matemática prevista na BNCC.

**Caminhão A:**  
 Dados: Tempo e Taxa na mesma unidade de tempo (ano)

1) O valor do caminhão durante 1 ano:  
 $100\% - 15\% = 85\% = 85 / 100 = 0,85$

1.1)  $150\ 000 * 0,85 = \text{R\$ } 127\ 500,00$

2) O valor do caminhão durante 2 anos:  
 1.2)  $127\ 500 * 0,85 * 0,85$

**Função Exponencial.**

2) Durante os 5 anos, o caminhão valerá:  
 $130\ 000 * 0,985^60 = \text{R\$ } 52\ 494,90$

3) Houve uma desvalorização de **R\$ 77 505,10**

quando o cenário mudar para loja scania

sempre

mostre

diga Boa Tarde senhor Murilo! Temos em no

toque o som vermelha até o fim

transmita Caminhão vermelho e espere

espere 1 seg

Fonte: <https://scratch.mit.edu/projects/410097593/>

No que concerne às habilidades mobilizadas da Matemática também entendemos que são mobilizadas nesse projeto as habilidades informadas no Quadro 12.

Quadro 10 – Projeto função exponencial - habilidades matemáticas da BNCC mobilizadas com o uso Scratch

Habilidade	Descrição da habilidade
(EF07MA02)	Resolver e elaborar problemas que envolvam porcentagens, como os que lidam com acréscimos e decréscimos simples, utilizando estratégias pessoais, cálculo mental e calculadora, no contexto de educação financeira, entre outros.
(EF07MA09)	Resolver e elaborar problemas que envolvam as operações com números racionais.

Fonte: Brasil, 2017.

Apresentamos na Figura 28 algumas habilidades matemáticas presentes na BNCC, as quais também contemplam o projeto apresentado pelos professores. Não pretendemos esgotar todas as habilidades, mas apresentar as que entendemos ser concernentes ao projeto.

### 5.3.1.2. Problema dos 35 camelos

Para este projeto, os cursistas entenderam que trabalharam as seguintes habilidades da BNCC, conforme Quadro 11.

Quadro 11–Problema dos camelos - habilidades matemáticas da BNCC mobilizadas com o uso Scratch

Habilidade	Descrição da habilidade
EF07MA06	Reconhecer que as resoluções de um grupo de problemas, que têm a mesma estrutura, podem ser obtidas utilizando os mesmos procedimentos.
EF07MA08	Comparar e ordenar frações associadas às ideias de partes de inteiros, resultado da divisão, razão e operador.
EF07MA09	Utilizar, na resolução de problemas, a associação entre razão e fração, como a fração $2/3$ para expressar a razão de duas partes de uma grandeza para três partes da mesma ou três partes de outra grandeza.

Fonte: Plano de Aula - P7 e P8

As habilidades selecionadas pelos cursistas foram as 06, 08 e 09 do 7º ano, por entenderem que o projeto se encaixa nelas.

Nesse caso, podemos elencar também a habilidade presente no Quadro 12, porque entendemos que foi trabalhada no projeto.

Quadro 12–Problema dos camelos - habilidades matemáticas da BNCC mobilizadas com o uso Scratch

Habilidade	Descrição da habilidade
(EF07MA05)	Comparar e ordenar frações associadas às ideias de partes de inteiros, resultado da divisão, razão e operador

Fonte: Brasil, 2017

O projeto contempla as habilidades da BNCC elencadas nos quadros 11 e 12; de acordo com essas habilidades e com o plano de aula dos professores, o projeto pode ser trabalhado com alunos do 7º ano.

Podemos observar, na Figura 29, como essas habilidades foram trabalhadas.

Figura 29 –História dos 35 camelos – representação da habilidade matemática do Quadro 12 trabalhada no projeto – Imagens extraídas do projeto digital.



Fonte: <https://scratch.mit.edu/projects/412149079/>

### 5.3.1.3. Jogo do Miko 3.1

Neste trabalho, P2 e P12 trabalharam um conteúdo voltado para os anos iniciais, com o objetivo de reconhecimento de figuras planas.

No jogo, o usuário tem de movimentar o macaco, cujo nome é Miko, com as setas do teclado (para cima, para baixo, para direita e para esquerda) para encontrar as figuras geométricas que o ator pede para marcar pontos, conforme Figura 30.

Figura 30–Jogo do Miko 3.1 – Reconhecendo figuras geométricas, imagem extraída do projeto digital.



Fonte: <https://scratch.mit.edu/projects/408749283/>

Os professores entenderam que foi mobilizada a habilidade descrita no Quadro 13.

Quadro 13 – Jogo do Miko 3.1 - habilidades matemáticas da BNCC mobilizadas com o uso Scratch

Habilidade	Descrição da habilidade
(EF02MA15)	Reconhecer, comparar e nomear figuras planas (círculo, quadrado, retângulo e triângulo), por meio de características comuns, em desenhos apresentados em diferentes disposições ou em sólidos geométricos.

Fonte: Plano de aula P2 e P12

Eles entenderam que, de acordo com o projeto que realizaram, foi mobilizada a habilidade descrita no Quadro 13, para ser aplicado no segundo ano do ensino fundamental.

Entendemos que, no projeto construído, também estão presentes as habilidades no Quadro 14.

Quadro 14 – Jogo do Miko 3.1 - habilidades matemáticas da BNCC mobilizadas com o uso Scratch

Habilidade	Descrição da habilidade
(EF01MA11)	Descrever a localização de pessoas e de objetos no espaço em relação à sua própria posição, utilizando termos como à direita, à esquerda, em frente, atrás.
(EF01MA14)	Identificar e nomear figuras planas (círculo, quadrado, retângulo e triângulo) em desenhos apresentados em diferentes disposições ou em contornos de faces de sólidos geométricos.

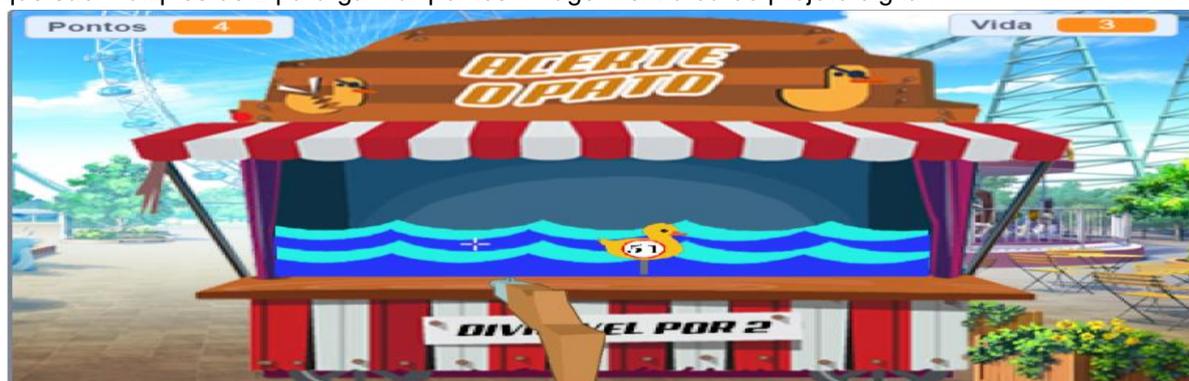
Fonte: BRASIL, 2017.

Esse projeto pode ser trabalhado com os alunos dos anos iniciais do ensino fundamental, mais precisamente primeiro e segundo ano.

#### 5.3.1.4. Acerte o Pato

Neste projeto, P3 e P4 trabalharam com critérios de divisibilidade, o jogo é trabalhado por fases, inicia no critério de divisibilidade por 2, 3, 4, 5 e 6.

Figura 31 – Jogo: Acerte o pato – frame: divisibilidade por 2; nele, o usuário teria de acertar os patos que são múltiplos de 2 para ganhar pontos: imagem extraída do projeto digital.



Fonte: <https://scratch.mit.edu/projects/412099569/>

O intuito é acertar o pato de acordo com a divisibilidade exigida na fase, por exemplo, se o critério de divisibilidade for 2, o usuário teria de acertar os patos terminados em 0, 2, 4, 6 e 8; acertando nos patos corretos ganharia pontos; errando, perderia vidas, conforme observado nos cantos superiores da Figura 31.

Nesse projeto, os professores entenderam que foi mobilizada a habilidade presente no Quadro 15.

Quadro 15 – Acerte o Pato - habilidades matemáticas da BNCC mobilizadas com o uso Scratch.

Habilidade	Descrição da habilidade
(EF06MA05)	Classificar números naturais em primos e compostos, estabelecer relações entre números, expressas pelos termos “é múltiplo de”, “é divisor de”, “é fator de”, e estabelecer, por meio de investigações, critérios de divisibilidade por 2, 3, 4, 5, 6, 8, 9, 10, 100 e 1000.

Fonte: Plano de aula P3 e P4

Os cursistas entenderam que essa habilidade, descrita no Quadro 15, foi a mobilizada pelo projeto; seria uma habilidade para o 6º ano do ensino fundamental.

Entendemos que pode também ser trabalhada nesse projeto uma habilidade descrita na BNCC para o 7º ano, conforme Quadro 16.

Quadro 16 – Acerte o Pato - habilidade matemática da BNCC mobilizada com o uso Scratch

Habilidade	Descrição da habilidade
(EF07MA01)	Resolver e elaborar problemas com números naturais, envolvendo as ideias de múltiplos, divisores e divisibilidade.

Fonte: BRASIL, 2017.

Critério de divisibilidade especificamente é um tema não muito explorado na BNCC, acreditamos que o documento pode ser ampliado nesse quesito.

#### 5.3.1.5. Brincando com a simetria

Neste projeto, P1 e P11 trabalharam uma simulação com hiperlinks, a ideia era estimular a percepção visual na construção de figuras simétricas.

No início da simulação, foi apresentado para o usuário se ele desejaria realizar a simulação ou ver o conteúdo que seria trabalhado, conforme Figura 32.

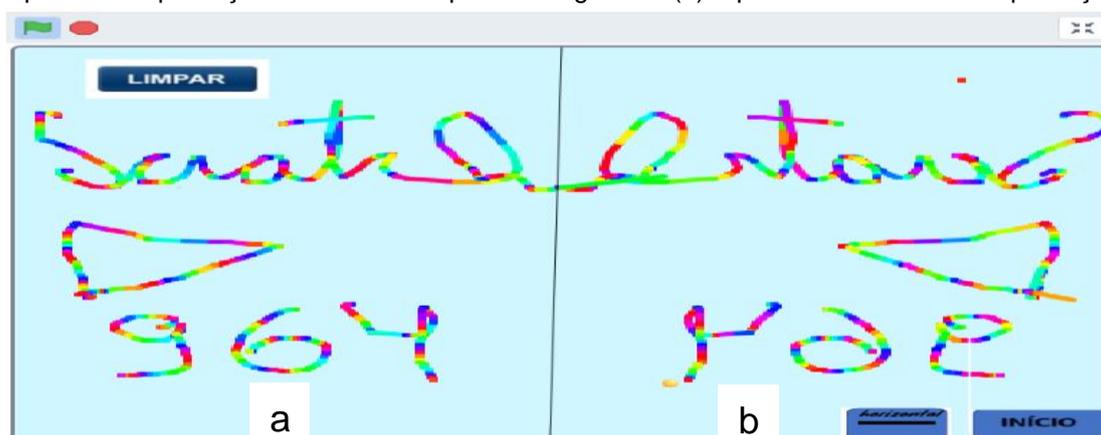
Figura 32 – Brincando com a simetria – imagem inicial da simulação extraída do projeto digital



Fonte: <https://scratch.mit.edu/projects/412094139/>

Se escolhesse simetria, o usuário poderia brincar com a reflexão, colocando nomes, números, figuras geométricas, conforme Figura 33.

Figura 33 – Representação da simetria de reflexão extraída do projeto digital em que a Figura 33 (a) representa a produção do usuário enquanto a Figura 33 (b) representa a reflexão da produção.



Fonte: <https://scratch.mit.edu/projects/412094139/>

As habilidades que os professores entenderam que foram mobilizadas nesse projeto estão descritas no Quadro 17.

Os cursistas entenderam que esse projeto pode ser trabalhado no quarto ano do ensino fundamental, de acordo com as habilidades que foram mobilizadas.

Quadro 17 – Brincando com a simetria - habilidades matemáticas da BNCC mobilizadas com o uso Scratch.

Habilidade	Descrição da habilidade
(EF04MA18)	Reconhecer ângulos retos e não retos em figuras poligonais com o uso de dobraduras, esquadros ou <i>softwares</i> de geometria.
(EF04MA19)	Reconhecer simetria de reflexão em figuras e em pares de figuras geométricas planas e utilizá-la na construção de figuras congruentes com o uso de malhas quadriculadas e <i>softwares</i> de geometria.

Fonte: Plano de Aula P1 e P11.

Entendemos também que podem ser mobilizadas algumas habilidades da BNCC além das traçadas pelos cursistas, as quais serão apresentadas no Quadro 18.

Quadro 18 – Brincando com a simetria - habilidades matemáticas da BNCC mobilizadas com o uso Scratch.

Habilidade	Descrição da habilidade
(EF08MA15)	Reconhecer e construir figuras obtidas por composições de transformações geométricas (translação, reflexão e rotação), com o uso de instrumentos de desenho ou de <i>softwares</i> de geometria dinâmica.
(EF07MA17)	Reconhecer e construir figuras obtidas por simetrias de translação, rotação e reflexão, usando instrumentos de desenho ou <i>softwares</i> de geometria dinâmica e vincular esse estudo a representações planas de obras de arte, elementos arquitetônicos, entre outros.
(EM13MAT105)	Utilizar as noções de transformações isométricas (translação, reflexão, rotação e composições destas) e transformações homotéticas para analisar diferentes produções humanas como construções civis, obras de arte, entre outras

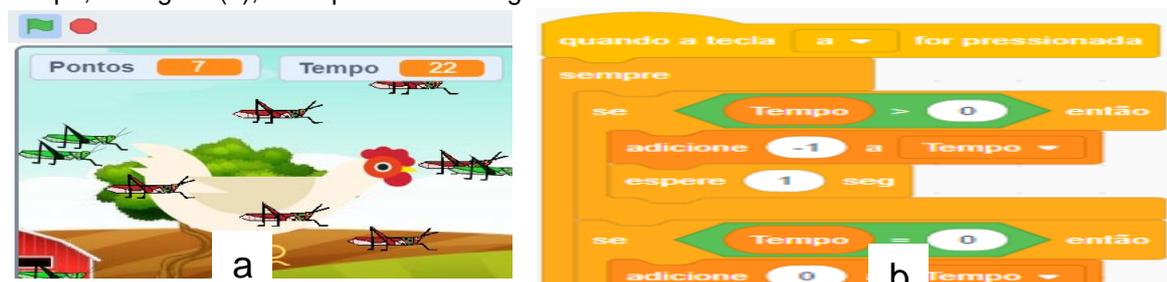
Fonte: BRASIL, 2017.

Observamos que essa construção pode ser trabalhada no ensino fundamental, anos iniciais, finais e ensino médio de acordo com as habilidades previstas na BNCC.

#### 5.3.1.6. Merilu Bica

Neste projeto, P9 e P13 trabalharam com o jogo Merilu bica, o objetivo do jogo era fazer a galinha (Merilu) capturar os grilos verdes e ganhar maior número de pontos em 1 minuto, mas, quando tocasse os grilos vermelhos, perderia pontos; o jogo foi estruturado por fases, movimentando a galinha pelas setas do teclado. Na Figura 34 (a), será apresentado o cenário do jogo e a captura dos grilos, enquanto a Figura 34 (b) representa uma parte da programação.

Figura 34 – Merilu Bica – Figura 34 (a) cenário do jogo, captura dos grilos e a marcação de pontos e tempo, na Figura (b), são apresentados alguns comandos



Fonte: <https://scratch.mit.edu/projects/412146300>

As Professoras entenderam que, para esse jogo, foram mobilizadas as habilidades disponíveis no Quadro 19.

Quadro 19 – Merilu bica - habilidades matemáticas da BNCC mobilizadas com o uso Scratch

<b>Habilidade</b>	<b>Descrição da habilidade</b>
(EF07MA03)	Comparar e ordenar números inteiros em diferentes contextos, incluindo o histórico, associá-los a pontos da reta numérica e utilizá-los em situações que envolvam adição e subtração.
(EF07MA04)	Resolver e elaborar problemas que envolvam operações com números inteiros.

Fonte: Plano de aula dos Professores

P9 e P13 entenderam que o projeto pode ser trabalhado no sétimo ano de acordo com as habilidades citadas. Entretanto, entendemos que também podem ser consideradas outras habilidades, nesse projeto, descritas no Quadro 20.

Quadro 20 – Merilu bica - habilidades matemáticas da BNCC mobilizadas com o uso Scratch

<b>Habilidade</b>	<b>Descrição da habilidade</b>
(EF04MA03)	Resolver e elaborar problemas com números naturais envolvendo adição e subtração, utilizando estratégias diversas, como cálculo por estimativa, cálculo mental e algoritmos.
(EF03MA06)	Resolver e elaborar problemas de adição e subtração com os significados de juntar, acrescentar, separar, retirar, comparar e completar quantidades, utilizando diferentes estratégias de cálculo, incluindo cálculo mental e estimativa.

Fonte: BRASIL, 2017.

De acordo com as habilidades mobilizadas, esse projeto também pode ser trabalhado no terceiro e quarto anos e, realizando algumas adaptações, seria um bom projeto para iniciar a adição e a subtração nos anos iniciais.

### 5.3.1.7 História da base decimal e sexagesimal

Neste trabalho, P6 e P10 trabalharam as bases numéricas 10 e 60. Para isso, utilizaram a pista de atletismo em uma corrida de 100 metros (Figura 35).

Além dos metros e segundos, os professores apresentaram outras formas de medir como légua e pés.

A intenção dos professores foi mostrar que pode utilizar bases diferentes para essa corrida, em que a distância é medida por metros (utiliza a base 10) e o tempo em segundos (utiliza a base 60).

Figura 35 – História da base 10 e sexagesimal – imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412004988/>

Para esse trabalho, foi utilizada a habilidade do ensino fundamental, que pode ser utilizada no quinto ano (Quadro 21).

Quadro 21 – História da base 10 e sexagesimal - habilidades matemáticas da BNCC mobilizadas com o uso Scratch

Habilidade	Descrição da habilidade
(EF05MA19)	Resolver e elaborar problemas envolvendo medidas das grandezas comprimento, área, massa, tempo, temperatura e capacidade, recorrendo a transformações entre as unidades mais usuais em contextos socioculturais.

Fonte: Plano de aula dos professores

Entendemos que, para esse trabalho, pode-se utilizar também as habilidades descritas no Quadro 22.

Quadro 22 – História da base 10 e sexagesimal - habilidades matemáticas da BNCC mobilizadas com o uso Scratch

Habilidade	Descrição da habilidade
(EF04MA22)	Ler e registrar medidas e intervalos de tempo em horas, minutos e segundos em situações relacionadas ao seu cotidiano, como informar os horários de início e término de realização de uma tarefa e sua duração

Fonte: BRASIL, 2017.

As habilidades mobilizadas com esse projeto podem ser trabalhadas no quarto, quinto ano e, como reforço, no sexto ano.

### 5.3.2 Metatexto relativo à subcategoria 2.1

Nas percepções dos cursistas, o trabalho com o Scratch pode abarcar os mais diversos conteúdos matemáticos e habilidades correlatas de todos os anos da educação básica. Segundo P13, “para o ensino da Matemática vi que é possível desenvolver qualquer conteúdo em uma história, simulador ou jogo.” (P13, Relatório 7).

Essa potencialidade do Scratch para trabalhar qualquer conteúdo da Matemática, observada por P13 e também por outros professores, alinha-se às observações de Kleinubing (2016):

Características possibilitam a aplicação da ferramenta Scratch em contextos escolares, possibilitando ser uma ferramenta de grande auxílio do ensino-aprendizagem nas escolas. Projetado especialmente para auxiliar no aprendizado de crianças de 8 a 16 anos, o Scratch se tornou uma ferramenta útil para qualquer idade. Milhões de usuários criam projetos todo dia para utilizar em escolas, museus, bibliotecas, centros comunitários entre outras instituições (KLEINUBING, 2016, p. 28).

A variedade de objetos do conhecimento matemático e de habilidades abordadas nos artefatos desenvolvidos no curso corroboram o fortalecimento da elasticidade do Scratch para o uso como recurso para auxiliar os processos de ensino e aprendizagem da Matemática na educação básica.

Nessa seara, P5 afirmou: “[...] utilizei a matemática financeira, explorando os conteúdos de função exponencial, potenciação, adição, subtração, valorização e desvalorização.” (P5, Relatório 7).

A viabilidade do uso do Scratch para o desenvolvimento de trabalhos sobre educação financeira, no ensino básico, perpassa a matemática financeira, e também foi destacada no trabalho de Farias et al. (2019) envolvendo um jogo no Scratch:

Observamos que 71,5% dos participantes concordam que a prática educativa ‘Finance Math Game’ permitiu planejar e executar ações envolvendo a criação de jogos digitais no controle de orçamento familiar e a educação financeira, enquanto que 71,4% indicam que a prática promoveu a reflexão crítica, acerca de conceitos relacionados a educação financeira (FARIAS et al. 2019, p. 1362).

O professor P11 destacou que é: “[...] possível realizar [no Scratch] as 3 formas de movimento da matemática, translação, rotação e reflexão [...]”. Temas estes que são importantes no desenvolvimento matemático dos alunos, conforme destacam Salles et al. (2012):

O estudo da simetria facilita a aprendizagem dos alunos em diversos conteúdos de matemática, por exemplo, quando é introduzido o conceito de números simétricos, que são trabalhados no decorrer das séries finais do Ensino Fundamental e também quando é trabalhado Função do Segundo Grau, no Ensino Médio, no momento em que é mencionado o conceito de parábola. (SALLES et al. 2012, p. 2).

Os critérios de divisibilidade também foram contemplados em um dos objetos criados. “Fizemos o jogo para que os alunos aprendam ludicamente os critérios de

divisibilidade” (P11, Notas de Aula). Todavia, por meio desse jogo, podem ser trabalhadas outras habilidades, como destacam os PCNs e a BNCC.

Conceitos como os de ‘múltiplo’ e ‘divisor’ de um número natural ou o conceito de ‘número primo’ podem ser abordados neste ciclo como uma ampliação do campo multiplicativo, que já vinha sendo construído nos ciclos anteriores, então como assunto novo, desvinculado dos demais. Além disso, é importante que tal trabalho não se resuma à apresentação de diferentes técnicas ou de dispositivos práticos que permitem ao aluno encontrar, mecanicamente, o mínimo múltiplo comum e máximo divisor comum sem compreender as situações-problema que esses conceitos permitem resolver (BRASIL, 1998, p. 66).

Classificar números naturais em primos e compostos, estabelecer relações entre números, expressas pelos termos ‘é múltiplo de’, ‘é divisor de’, ‘é fator de’, e estabelecer, por meio de investigações, critérios de divisibilidade por 2, 3, 4, 5, 6, 8, 9, 10, 100 e 1000 (BRASIL, 2017, p. 301).

Para demonstrar a elasticidade do Scratch com alunos de diferentes anos escolares, P14 destacou que o projeto dele “[...] foi pensado para o ensino médio”. Já P10 menciona que “utilizei uma linguagem para alunos do sexto ano”, enquanto P2 afirmou que seu “[...] projeto é para introdução dos alunos das séries iniciais em geometria”.

Nos sete artefatos analisados neste estudo, foram mobilizadas, ao todo, 26 habilidades da Matemática, sendo 11 relacionadas aos anos finais do ensino fundamental, 4 ao ensino médio e 11 habilidades mobilizadas para os anos iniciais do ensino fundamental.

Observamos que, pelas criações serem voluntárias e sem pedido específico para se trabalhar com etapas da educação básica, os cursistas, com a liberdade de criação, desenvolveram projetos tanto para o ensino fundamental quanto para o médio.

Diante do exposto, o Scratch se apresenta como um recurso que pode ser usado em qualquer fase do ensino fundamental ou médio, embora, ao longo deste curso, não tenha sido criado nenhum objeto voltado aos primeiros anos do ensino fundamental. Contudo, acreditamos que isso não aconteceu devido ao público do curso ser constituído de professores das outras duas etapas de ensino.

Por fim, nas 26 habilidades da BNCC, desenvolvidas pelos cursistas em seus projetos no módulo 9 do curso, serão apresentadas as potencialidades do Scratch para desenvolver habilidades matemáticas, demonstrando várias possibilidades do trabalho com essa disciplina em sala de aula

### **5.3.3 Subcategoria 2.2: habilidades do pensamento computacional mobilizadas com o uso do Scratch**

Nesta seção, analisaremos os sete artefatos produzidos (em duplas) no Scratch pelos 14 professores de Tangará da Serra, com vistas ao desvelamento das habilidades do pensamento que foram mobilizadas.

Para isso, analisaremos cada artefatos/objetos com relação às quatro habilidades fundamentais do pensamento computacional (pilares), conforme apresentadas e discutidas na Seção 2.4 deste estudo. Outra fonte de dados considerada, nesta análise, foram os planos de ensino elaborados pelos professores cursistas, os quais serviram para nortear a construção dos objetos.

#### **5.3.3.1 Decomposição**

Este pilar, conforme já definido na Seção 2.4.1, está relacionado ao processo pelo qual os problemas são divididos em partes menores para facilitar a sua resolução. Compreende também a prática de analisar problemas a fim de identificar quais partes podem ser separadas e também de que forma podem ser reconstituídas para solução de um problema global.

Para a análise dos projetos no que tange à habilidade da decomposição, foi realizado o desmembramento do objeto, de acordo com nossa observação e guiado pelas falas dos cursistas, desvelando como eles chegaram ao objeto final considerando as partes.

##### **5.3.3.1.1 Função Exponencial**

Neste projeto, a decomposição foi evidenciada a partir da definição do problema a ser resolvido. Diante da proposta de produzir um objeto no Scratch que mobilizasse habilidades da Matemática, a dupla de professores decidiu abordar aplicações da função exponencial na tomada de decisão frente a duas opções de compra. Assim, o problema que se apresentou foi: Que tipo de artefato produzir e como ele será constituído?

Diante desse problema, a decomposição se tornou evidente quando optaram por, primeiramente, decidir o tipo de artefato e depois pensar em como iriam produzi-lo. Na fala de P5, fica evidenciada essa decomposição do problema: “Entendemos que era melhor projetar em forma de história, e as contas irem aparecendo em forma de slides” (P5, Notas de Aula).

Depois disso, perceberam que a produção do artefato também poderia ser decomposta em partes, entre elas: produzir o enredo da história, buscar imagens na internet para compor o cenário; construir os frames e a codificação da história no Scratch. “Buscamos na internet imagens de caminhões, de vendedores e de concessionárias que vendem caminhões” (P14, Notas de aula). (Figura 36, a e b)

Sobre o enredo da história, P5 afirmou: “definido os primeiros passos foi pensado em fazer orçamentos e desvalorização de acordo com a tabela Fipe”.

A própria história também foi decomposta em partes, como podemos notar na fala de P5: “primeiro realizamos o cálculo do caminhão A, logo em seguida realizamos o cálculo do caminhão B. Depois dos cálculos é pedido para a pessoa que está manuseando o Scratch ajudar a escolher a melhor opção.” (P5, NA).

Figura 36 – Exemplos de decomposição do Projeto Função Exponencial na imagem (a) apresenta o ator simulando a compra de um caminhão, e a imagem (b) apresenta a análise da Tabela Fipe e os juros ao mês – imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/410097593/>

Durante o trabalho da programação do artefato no Scratch, observamos que a decomposição também ocorreu, pois o funcionamento do programa exigiu a programação em separado dos atores e cenários que compuseram a história. Isso também ficou evidente para os professores cursistas, como podemos ver na fala de um deles:

Para o ator narrador foi criado um personagem, no primeiro momento foi criado um código de apresentação que ficou somente nesse ator [...] Em seguida, foi criado um personagem para o comprador, o Sr. Murilo, que ligava para as empresas para saber qual caminhão seria mais viável (P14) (Figura 37, imagens a e b).

Figura 37 – Decomposição projeto Função Exponencial – Na Figura 38 (a), Ana pergunta qual o nome do usuário; na Figura 38 (b), após os cálculos, chega-se à conclusão de que o melhor caminhão é o B, já que o caminhão A teve maior desvalorização - extraídas do projeto digital.



Fonte: <https://scratch.mit.edu/projects/410097593/>

Na Figura 38, será apresentado como os cursistas realizaram parte da decomposição, através dos cálculos da função criada.

Figura 38 – Decomposição - Função Exponencial – Figura 38 (a) cálculo da taxa de desvalorização; Figura 38 (b) apresenta a porcentagem de desvalorização mensal juntamente com a Tabela Fipe; a Figura 38 (c) cálculo da desvalorização do caminhão em 5 anos, e a Figura 38 (d) representa a porcentagem de desvalorização mensal juntamente com a tabela Fipe

**Caminhão A:**

Dados: Tempo e Taxa na mesma unidade de tempo (ano)

1) O valor do caminhão durante 1 ano:  
 $100\% - 15\% = 85\% = 85 / 100 = 0,85$

1.1)  $150\ 000 * 0,85 = \text{R\$ } 127\ 500,00$

2) O valor do caminhão durante 2 anos:

1.2)  $127\ 500 * 0,85 * 0,85$

Ele é um ótimo caminhão, com desvalorização de 1,5% a.m. segundo a Tabela Fipe, e o seu valor é de R\$ 130 000,00.

**Função Exponencial.**

2) Durante os 5 anos, o caminhão valerá

1.1)  $130\ 000 * 0,985^{60} = \text{R\$ } 52\ 494,90$

Houve uma desvalorização de R\$ 77 505,10

Ele é um ótimo caminhão, com desvalorização de 1,5% a.a segundo a Tabela Fipe, e o seu valor é de R\$ 150 000,00.

Fonte: <https://scratch.mit.edu/projects/410097593/>

O exercício da programação, em separado atores e cenários, forçou o exercício da decomposição, já que o Scratch proporciona esse trabalho, pois, ao realizar o planejamento para a construção do artefato, o criador necessita realizar esse trabalho intenso de decomposição.

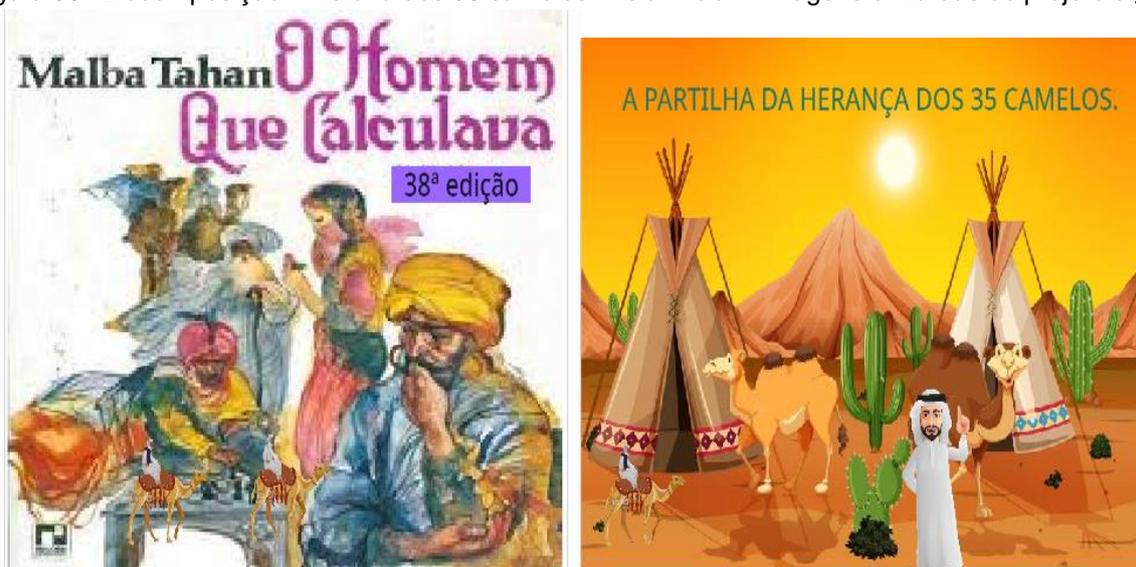
#### 5.3.3.1.2 Problema dos 35 Camelos

Este trabalho também foi baseado em uma história, que foi projetada em um plano de aula.

Esse artefato também teve foco na criação de uma história, mais especificamente, a “História dos 35 camelos”, do livro “O homem que calculava”, de Malba Tahan (1994) (Figura 39).

Embora os autores tenham optado por reproduzir uma história existente, compreenderam que a implementação dela, no Scratch, exigia um planejamento/estratégias, que, por sua vez, exigia decompor o problema em partes menores.

Figura 39 - Decomposição - História dos 35 camelos – tela inicial – imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412149079/>

Assim, os professores produtores desta história foram definindo as partes e suas relações. A fala de um deles evidencia parte desse movimento: “Escolhemos figuras na internet para representar os camelos, Sheiks, homens árabes e cenários que contemplassem a região” (P8).

No que tange aos comandos, “criamos a história através da sincronização entre as imagens escolhidas e escritas que iriam dentro dos balões” (P8, Notas de Aula).

Na Figura 40, observamos como a decomposição foi realizada. Os criadores a trabalharam a partir das criações dos códigos, ou seja, por função e foi por essa realização que conseguiram fazer a sincronização.

Como o projeto anterior, essa é uma história, os comandos ficaram baseados na sincronização, conforme Figura 40.

Figura 40 – Decomposição - História dos 35 camelos – sincronização de comandos – imagens extraídas do projeto original



Fonte: <https://scratch.mit.edu/projects/412149079/>

### 5.3.3.1.3 Jogo do Miko 3.1

O jogo do Miko 3.1, no que tange ao pilar da decomposição, “primeiramente partiu do ator Miko, que tem no próprio Scratch, o primeiro cenário que é o campo de futebol também é um cenário do Scratch” (P12, Entrevista). Nesse primeiro passo, escolheram ator e cenário do Scratch.

Partindo daí e para dar uma introdução ao conteúdo do jogo, “trabalhamos algumas perguntas, primeiro do nome do participante, logo em seguida uma pergunta pessoal, envolvendo o campo de futebol e algumas figuras geométricas que estão contidas nele”, (Figura 41).

Após esse primeiro momento, “separamos triângulos, retângulos, quadrados e círculos, também sons para que o jogo se tornasse agradável” (P12, Entrevista).

“Optamos por escolher as setas do teclado para movimentar o Miko, e após isso colocar as figuras ao redor do Miko” (P2, Entrevista).

Os criadores utilizaram outros jogos, que utilizam as setas do teclado. Nesse aspecto, para chegar às figuras pedidas se fez necessário utilizar as setas, o que facilitou a adaptação do usuário em relação ao jogo.

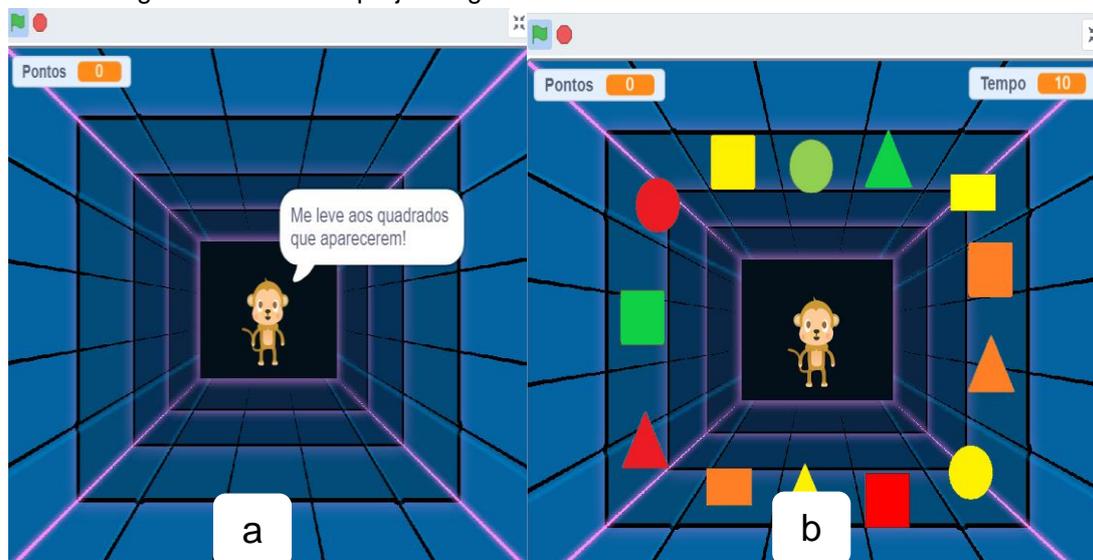
Figura 41 – Decomposição – Jogo do Miko 3.1 – imagens extraídas do projeto original.



Fonte: <https://scratch.mit.edu/projects/408749283/>

Em seguida, foi “dado o comando para a figura, para que o jogador movimentasse o Miko” (P2, Entrevista), conforme Figura 42 (a); ao encontrar a figura pedida pelo comando, essa desaparecia, conforme Figura 42 (b).

Figura 42 – Decomposição–2 - Jogo do Miko 3.1 – Imagem (a) o Miko pede para o usuário leva-lo a figura geométrica pedida, na imagem (b) é apresentado o cenário do jogo e as figuras criadas pelos cursistas – imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/408749283/>

Entretanto, algo mais precisaria ser feito. Então, “criamos a pontuação e para cada figura correta que o Miko encostasse, o jogador levaria 10 pontos” (P12, Entrevista), conforme se perceberá na Figura 43.

No que tange aos comandos diferentes dos dois projetos anteriores analisados, o jogo do Miko possui uma gama maior de comandos que envolvem a Matemática.

“Partimos dos comandos para o Miko, dar o *start*, para o jogador fizessem os demais comandos” (P12, Entrevista), conforme ilustrado na Figura 44.

Nesse “monólogo do Miko”, temos praticamente comandos semelhantes aos da história, já que é como se eles estivessem contando uma história, mas há uma interação com o usuário, perguntando se está bem, nome do usuário, quantos retângulos têm no campo, e quantos círculos têm no meio do campo.

Figura 43 – Decomposição-3 - Jogo do Miko 3.1 – sincronização de comandos conforme programação – imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/408749283/>

Em seguida, ao começar o jogo, “criamos comandos para a lateralidade” (P2, Entrevista). Observamos aqui que eles utilizaram o plano cartesiano nos eixos x e y, para definirem esses movimentos (Figura 44).

Após isso, eles definiram os comandos para os pontos, conforme Figura 45 (a), e para direção desejada, conforme Figura 45 (b), todos no mesmo comando.

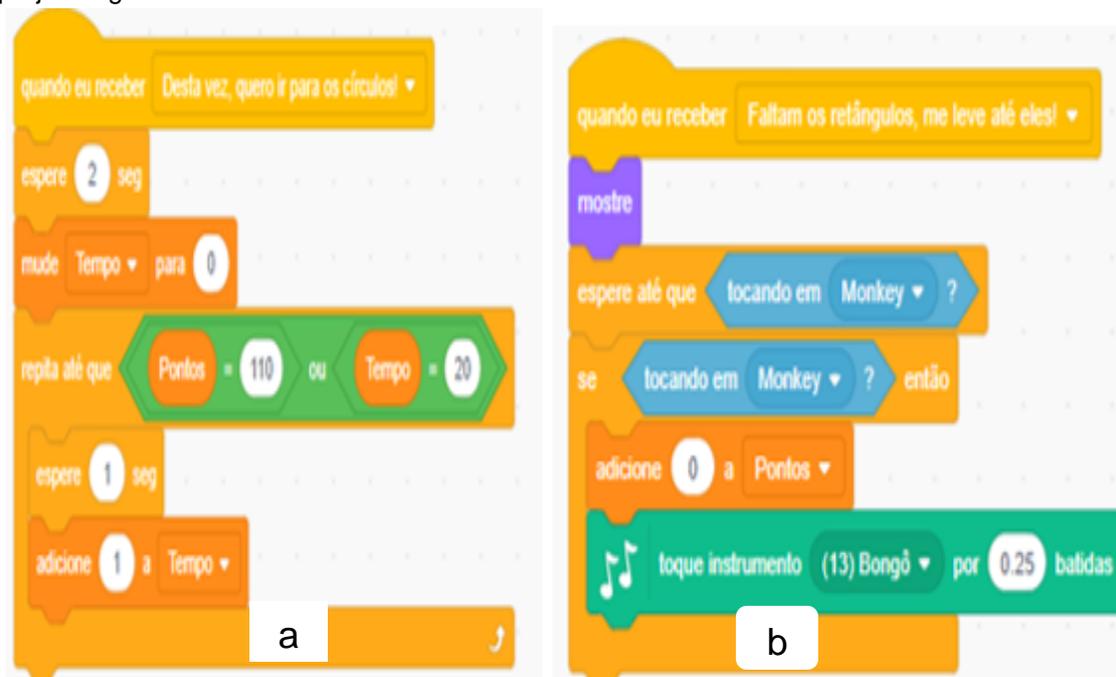
Figura 44 – Decomposição - Jogo do Miko 3.1 – comandos para realizar a lateralidade – imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/408749283/>

A Figura 45 (a) também representa um comando que eles dão para o Miko e a Figura 45 (b) revela um comando dado para a imagem do quadrado que, ao tocar no Miko, não aparecem pontos, pois agora a pontuação ocorre somente quando se toca nas imagens dos retângulos.

Figura 45 – Decomposição - Jogo do Miko 3.1 – comandos para pontos e direção – imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/408749283/>

O que observamos nos comandos desse jogo é que ele já possui a matemática envolvida, pontos, tempo, conectivo “ou”, adicione, lógica “se, então”.

#### 5.3.3.1.4. Acerte o pato

Acerte o pato é um jogo que teve um grande número de atores e cenários. Para produzi-lo, a decomposição do problema foi essencial.

Primeiramente, eles tiveram de “trabalhar o cenário, com a personagem principal do Scratch chamada Ana que dava os comandos gerais do jogo” (P3, Entrevista). Depois, tiveram de separar os patos, pois o jogo foi dividido em fases. “Dividimos os patos em andares, um na parte superior e outro na parte inferior” (P3). Os patos certos só apareceriam uma vez, “foram criados 100 patos no *fotoshop*” (P4).

Eles separaram o jogo em quatro fases ou níveis. Separaram os patos em E1 para errado (caso atirasse no pato que não pertencia àquele critério de divisibilidade, perdia vidas), e C1 para correto (caso atirasse no pato que pertencia àquele critério de divisibilidade, ganhava pontos).

Para os patos não aparecerem no momento errado: “utilizamos o bloco *script* do ator, para que ao invés do jogo parar, somente o pato pararia” (P3), pois os critérios de divisibilidade são diferentes, e, no momento de trocar do critério de divisibilidade de 2 para 3, por exemplo, os patos dos critérios de divisibilidade por 2 da fase anterior recebiam comandos para não se movimentarem mais.

Os patos foram programados para aparecerem aleatoriamente de acordo com as fases do jogo: “utilizamos a lógica para que acontecesse durante o jogo, senão o jogo não teria uma pontuação fixa, mas graças ao entendimento da lógica conseguimos” (P3, NA).

“Utilizamos comando para que as ondas realizassem simulação do movimento” (P4, NA).

Eles tiveram dificuldade com as armas e “para resolver o problema foi criado uma espingarda para cada nível” (P3, NA).

Ao invés de tirar pontos, eles escolheram perder vidas, os pontos só somam, e, se errar, não perde pontos e, sim, vidas. A cada cinco pontos mudava de fase.

Os *Scratchers* alegaram que “quando perdia pontos a programação não rodava corretamente, por isso tiraram a perca dos pontos” (P3, NA).

O usuário poderia errar três vezes o pato; assim, ao errar mais de três vezes, o jogo acabava, pois eles colocaram apenas 3 vidas.

Na Figura 46, seguem as apresentações das fases, e os cenários são bem parecidos, mas mudavam a cada fase, já que os critérios de divisibilidade eram diferentes a cada fase, e os patos certos e errados também mudavam.

Quanto aos comandos, primeiro eles realizaram o comando para a apresentadora Ana, que, conforme Figura 46, aparece no início e, a cada fase que se inicia, ela explica quais são os pontos-chaves para cada critério de divisibilidade.

Ana é que aciona todas as fases, dando os comandos, só após a fala dela que se iniciava cada fase. A instrução está na Figura 47, e os comandos para a fala na Figura 46.

Figura 46 – Decomposição – Acerte o pato – imagem (a) apresentação do jogo por Ana, personagem principal da narração; a imagem (b) critérios de divisibilidade por 2 – imagens extraídas do projeto digital.

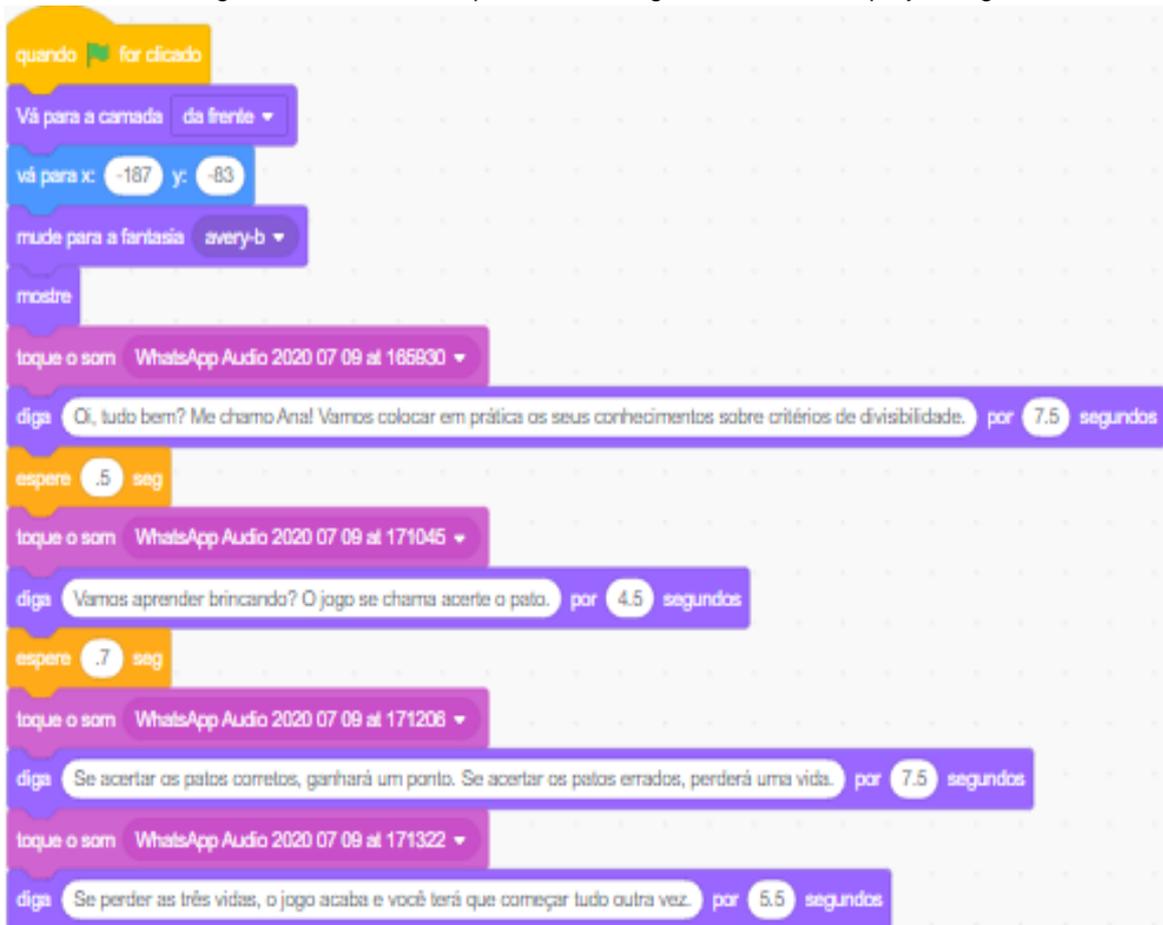


Fonte: <https://scratch.mit.edu/projects/412099569/>

Também realizaram a decomposição do comando de cada arma, para cada fase uma arma diferente; podemos observar o comando da arma do nível 1 na Figura 48 (a) e, do nível 2, na Figura 48 (b).

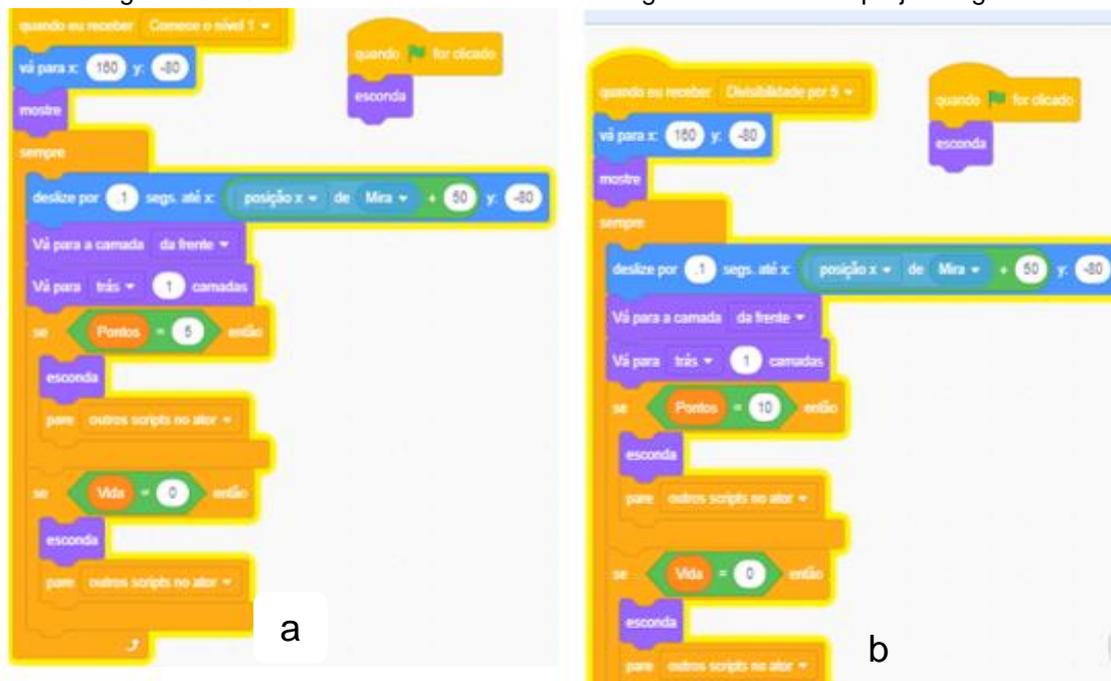
Há uma variedade de detalhes, a Matemática presente nos comandos é muito rica, temos a lógica nos comandos (se então; sempre; operadores, plano cartesiano com as coordenadas  $x$  e  $y$ ) e, para montar esses comandos, tem de ser trabalhado o PC, pois são vários atores trabalhando em sincronia. E, quando os pontos atingem 5, 10, 15, as armas também mudam, basta observarmos os comandos.

Figura 47 – Comandos para Ana - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412099569/>

Figura 48 – Comandos arma 1 e arma 2 - imagens extraídas do projeto digital



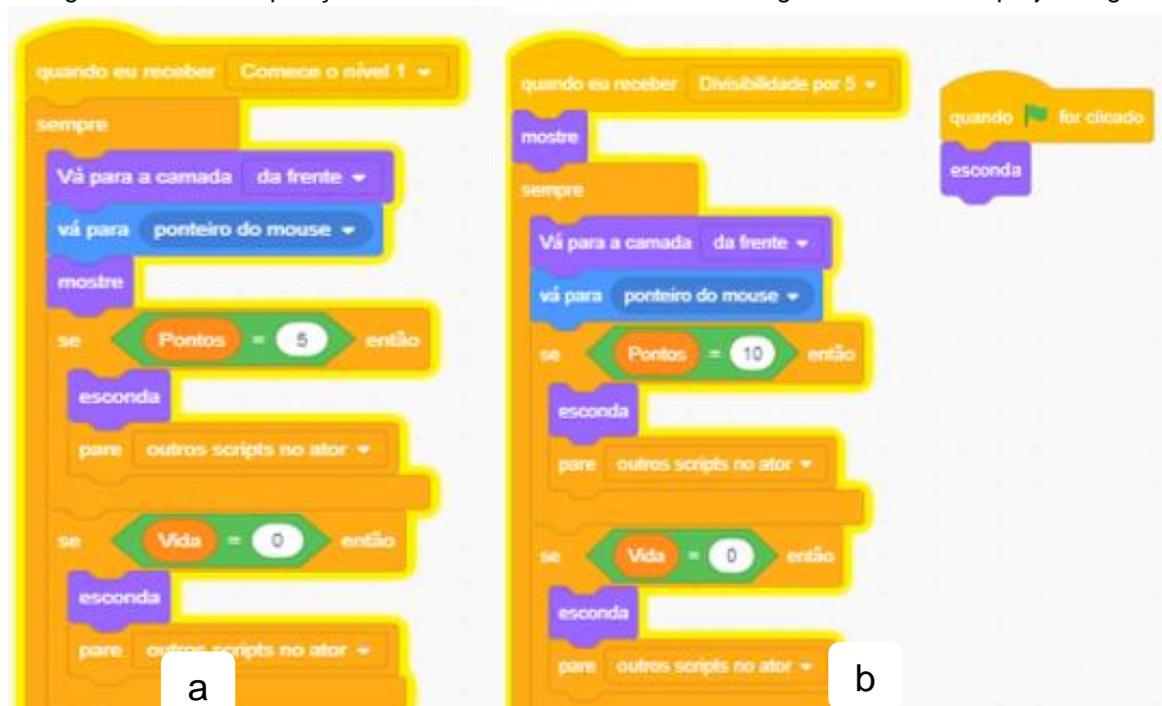
Fonte: <https://scratch.mit.edu/projects/412099569/>

As miras também tinham sua programação, “o padrão eram os mesmos das armas, inclusive da pontuação, já que cada mira pertencia a uma arma” (P3). Observamos a mira 1 Figura 49 (a) e mira 2 na Figura 49 (b).

Os tiros também possuíam comandos específicos, mas, nesse caso, foi criado apenas um formato de tiro para todas as armas, com o comando, conforme Figura 50.

Esse projeto foi complexo, podemos observar o processo de elaboração: foram

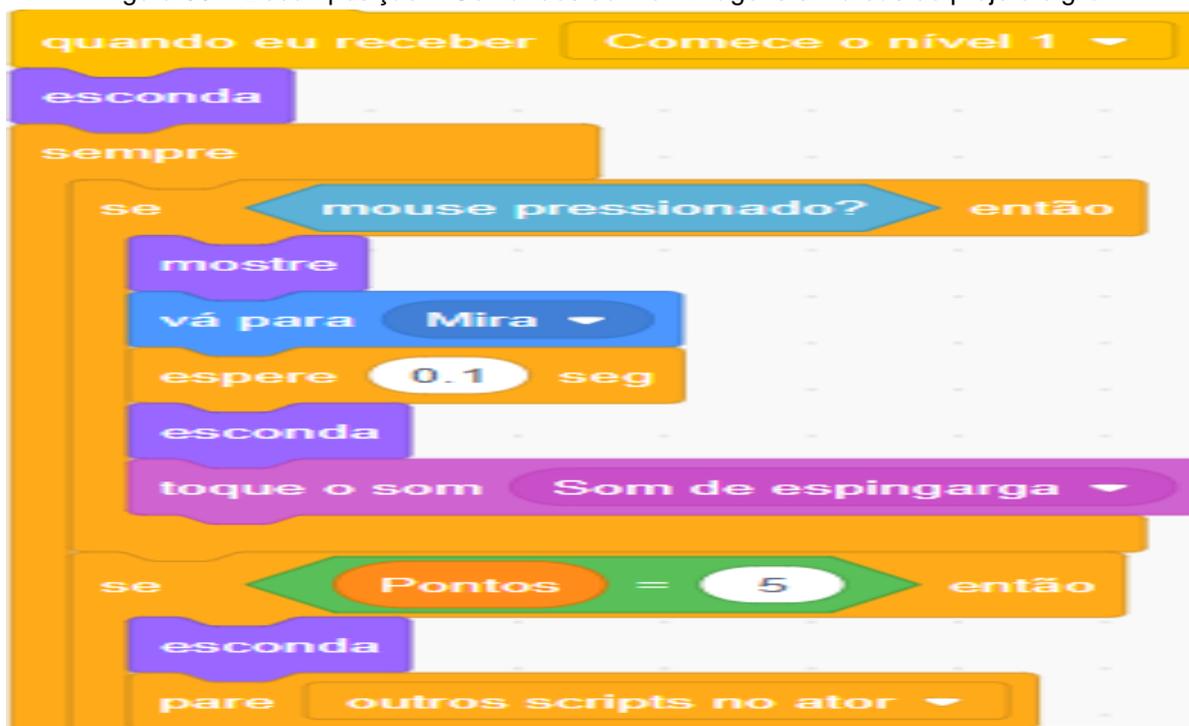
Figura 49 – Decomposição - Comandos mira 1 e mira 2 - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412099569/>

escolhidas figuras na internet, criados atores (patos) no photoshop e, em camadas, as ondas se sobrepondo em duas camadas. Os comandos, que foram inicialmente para a apresentadora Ana, depois foram para arma, mira, pontos, vidas. Nesse jogo, a decomposição teve de ser muito bem programada para que o produto final saísse com exatidão.

Figura 50 – Decomposição - Comandos do tiro - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412099569/>

#### 5.3.3.1.5. Brincando com a simetria

Eles separaram os conteúdos de simetria, duas páginas: a primeira com conceito de simetria; e a segunda com explicação das transformações geométricas, a translação, rotação e reflexão, explicando cada uma delas. Após isso, passa para a simulação. “Conseguimos fazer para apresentar só a simulação da reflexão, a intenção era fazer das três, mas dá muito trabalho, e o tempo hábil deu para fazer só essa, gastamos duas semanas” (P11, NA).

“Pensamos em uma borboleta que já tinha no Scratch e construímos um cenário de fundo introduzimos um som de fundo” (P1, NA). “Para iniciar tem que clicar na borboleta” (P1) (Figura 51).

“Na simulação tem um texto explicando como fazer para obter as transformações e como voltar ao início. Quando você clica no conteúdo por exemplo da reflexão o texto aponta o que é uma reflexão e assim para as outras duas” (Figura 52).

P1 disse: “colocamos na parte debaixo para o usuário escolher entre iniciar a simulação ou ler os conteúdos” (P1) (Figura 53).

Figura 51 – Decomposição Brincando com a Simetria – tela inicial - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412094139/>

Figura 52 – Transformações matemáticas - imagens extraídas do projeto digital

**SIMETRIA**  
SIMETRIA

Existe simetria quando as duas partes de um elemento dividido ao meio são iguais. Esse conceito é usado na Matemática, na Arte, na natureza e na Arquitetura.

O termo simetria vem do grego *syn* (junto) + *metron* (medida ou qualidade do que tem a mesma medida).

Na geometria, um objeto apresenta simetria quando se parece o mesmo depois de uma transformação como reflexão, rotação ou translação.

O eixo de simetria é uma linha real ou imaginária, que atravessa o centro da figura.

SEMI-HORIZONTAL

TRANSLAÇÃO - É o termo usado para "mover" formas, usando a direção e a magnitude.

ROTACÃO - É o "giro" de uma forma ao redor de um ponto chamado centro de rotação. A distância ao centro de rotação se mantém constante.

REFLEXÃO - Ocorre através de uma reta chamada de eixo. O ponto original e seu correspondente na reflexão tem a mesma distância ao eixo.

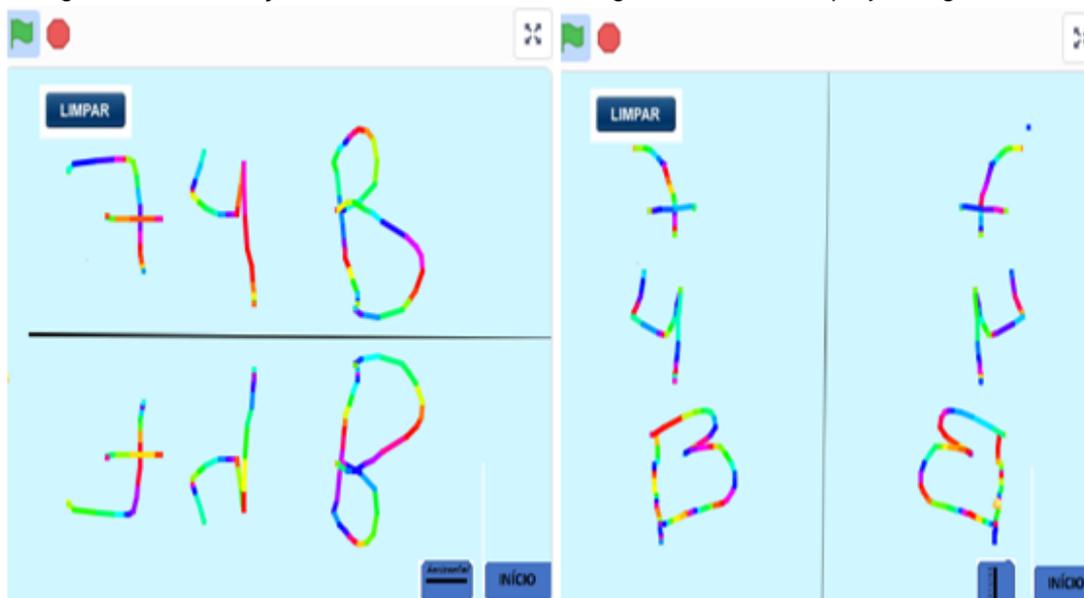
TRANSLAÇÃO      REFLEXÃO      ROTACÃO

PRÓXIMO   INÍCIO   FIM      SIMULAÇÃO   INÍCIO

Fonte: <https://scratch.mit.edu/projects/412094139/>

Logo, em seguida, eles pensaram em adicionar como seria apresentado “colocamos a opção da apresentação, se o usuário queria na vertical ou horizontal”. (P11) (Figura 53).

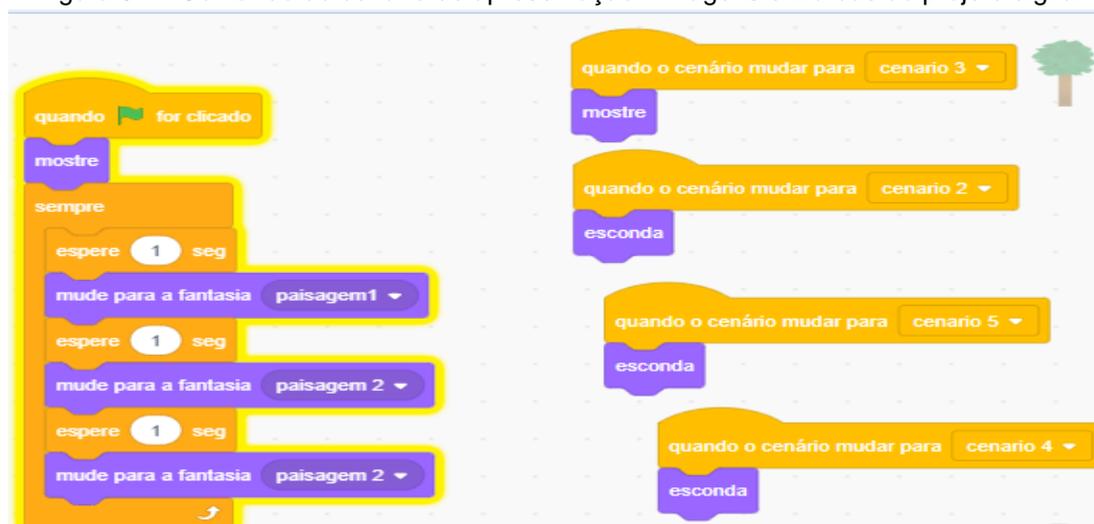
Figura 53 – Simulação horizontal e vertical - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412094139/>

No que concerne aos comandos, iniciaram com os comandos para a tela inicial: “utilizamos comandos de 2 cenários, só para trazer a sensação de movimentos, e todos os demais cenários da simulação estavam ligados a ele” (P11), (Figura 54).

Figura 54 – Comando do cenário de apresentação - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412094139/>

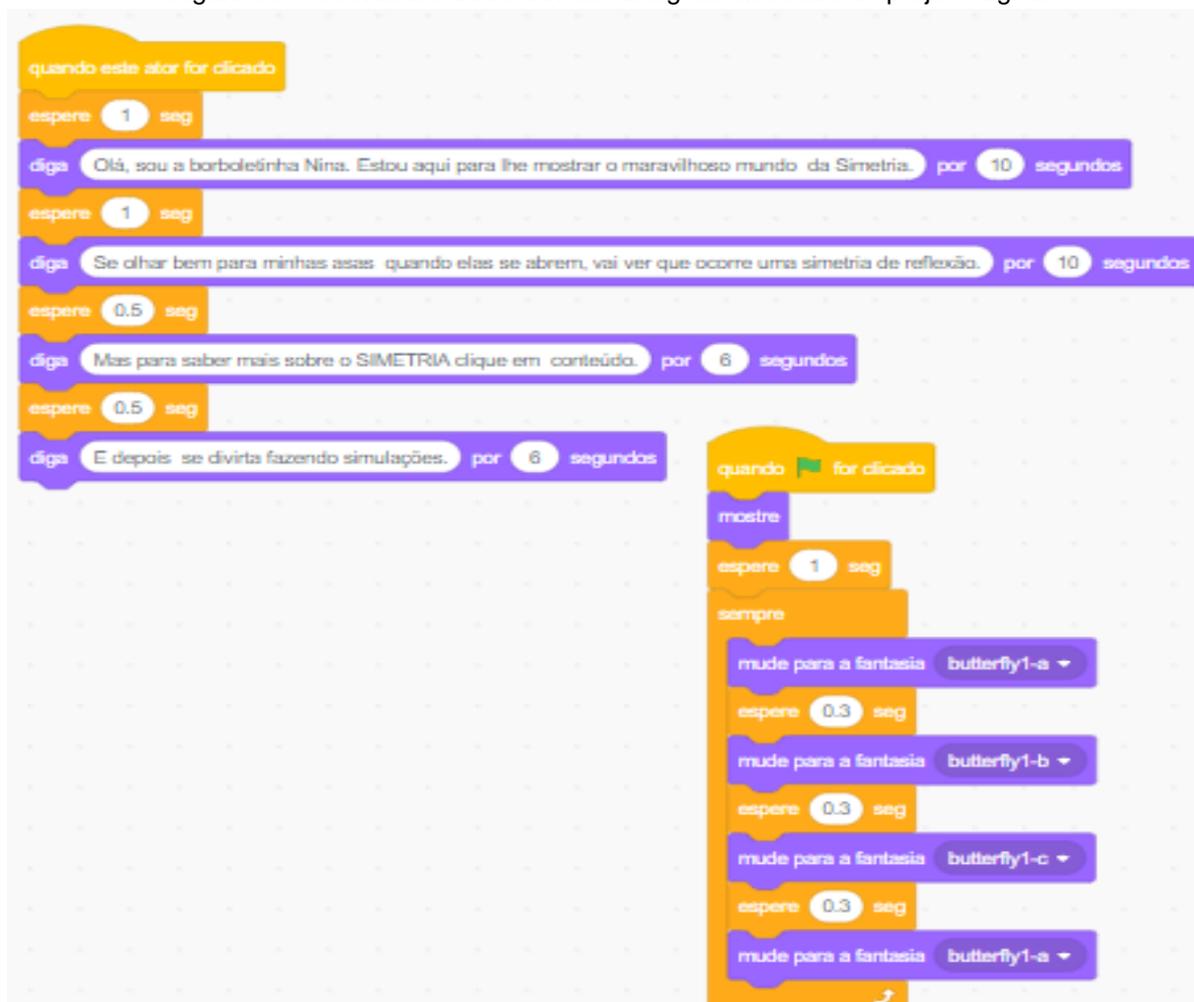
Em seguida, pensaram nos comandos para a borboleta, pois é ela que dá início à apresentação da simulação. P11 disse: “escolhemos a borboleta, pois pensamos que as criações são livres, sem amarras” (Figura 55).

Depois escolheram os comandos do Sol, “esse comando é só para o cenário inicial do jogo, depois eles são escondidos” (P11).

Os comandos citados foram para dar início à produção, já que, pensando em decomposição, os criadores necessitavam realizar as criações por partes e ir fazendo a sincronização com as demais ideias, seguindo o roteiro.

Depois disso, “criamos os comandos para simulação e conteúdo” (P11). Assim, quando clicava na simulação, aparecia uma nova tela, e, para iniciar a simulação, era só colocar o mouse na escrita “clique aqui”.

Figura 55 – Comandos da borboleta - imagens extraídas do projeto digital



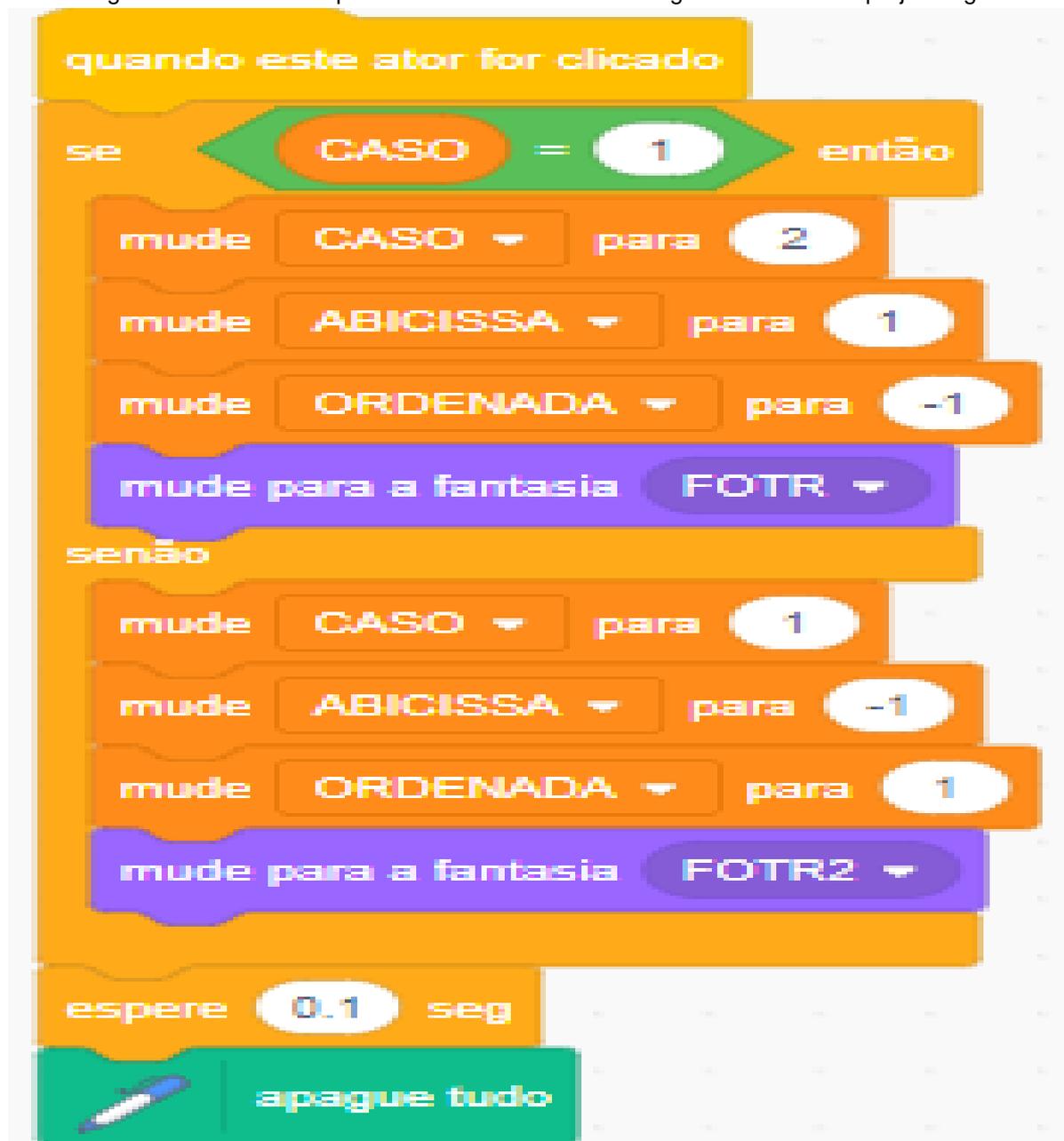
Fonte: <https://scratch.mit.edu/projects/412094139/>

Após isso, o usuário poderia escolher se queria desenvolver na horizontal ou vertical, para isso eles criaram os comandos, conforme Figura 56.

Depois de “criado as horizontais e verticais criamos o comando para as escritas, linhas para gerar o que as pessoas escreviam, criamos o ator pincel” (P11) (Figura 57).

Esse comando é de extrema complexidade, por isso, recorreremos a ele quando formos descrever o pilar abstração. Os cursistas relataram que “Esse comando foi o mais importante, foi nele que conseguimos realizar a simetria”. (P1).

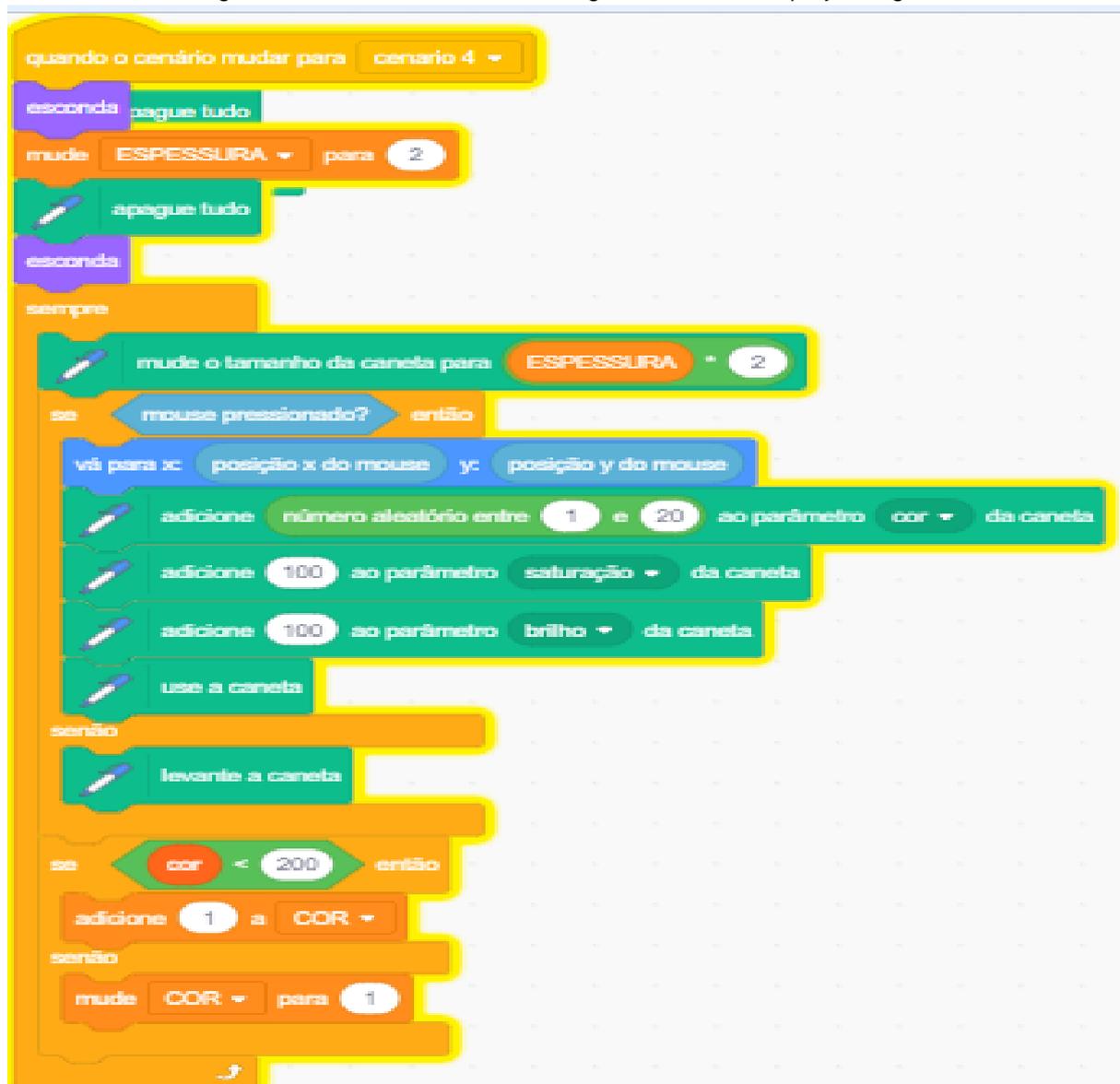
Figura 56 – Comandos para horizontal e vertical - - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412094139/>

Após isso, “pensamos em criar os comandos para limpar a tela” (P1), ou seja, eles criaram esse comando para apagar tudo que foi criado e deixar a tela virgem para novas criações (Figura 58).

Figura 57 – Comando Pincel - - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412094139/>

Figura 58 – Limpa tela - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412094139/>

A simulação com a história “Brincando com a simetria”, no campo do pilar decomposição, partiu da criação do cenário inicial com a borboleta, sol, árvores, menu do jogo e se consolidou nos comandos horizontal e vertical, pincel e limpa a tela, os

quais, como observado, foram complexos para gerar a simetria que o pincel fazia no campo da reflexão.

#### 5.3.3.1.6 Merilu Bica

**Merilu Bica** foi um projeto que partiu do cenário rural, os criadores P9 e P13, escolheram vozes de pessoas para explicar o início do jogo, os personagens foram animais como galinhas e pintinhos e o cenário tinha celeiro, grama, palhas, grilos, sons de grilo, cata-vento. Como disse P13: “queríamos construir um cenário para o jogo que contemplasse a nossa realidade” (Figura 59).

Figura 59 – Cenário Inicial Merilu bica - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412146306/>

Depois, os construtores pensaram como adicionar a matemática: “fizemos o jogo pensando nas quatro operações com números naturais” (P9).

A princípio, eles pensaram em colocar a galinha se movimentando aleatoriamente e, para pontuar, seria necessário tocar no grilo. Mas, com o passar do projeto, eles conseguiram fazer a galinha se movimentar por meio das setas do teclado, “perguntei para P3, e ele me respondeu que era possível fazer ela andar com as setas, definimos os eixos e a quantidade de passos, velocidade e tempo” (P13) (Figura 60).

Após isso, a ideia era fazer com que a galinha, ao bicar no grilo verde, ganhasse pontos, mas, se encostasse no grilo verde, perderia dois pontos. Para isso, tiveram de realizar alguns comandos bem específicos, “a nossa ideia era fazer com que a galinha bicasse o grilo” (P9).

“Perguntei ao professor tutor, como fazer para que o bico da galinha, ao se encostar no grilo verde, contasse um ponto; e, se encostasse no grilo verde, perdesse

dois pontos, ele disse que era necessário ver a cor correta do bico da galinha, e deu certo” (P13), conforme Figura 61, que mostra os comandos dos grilos para contagem de pontos.

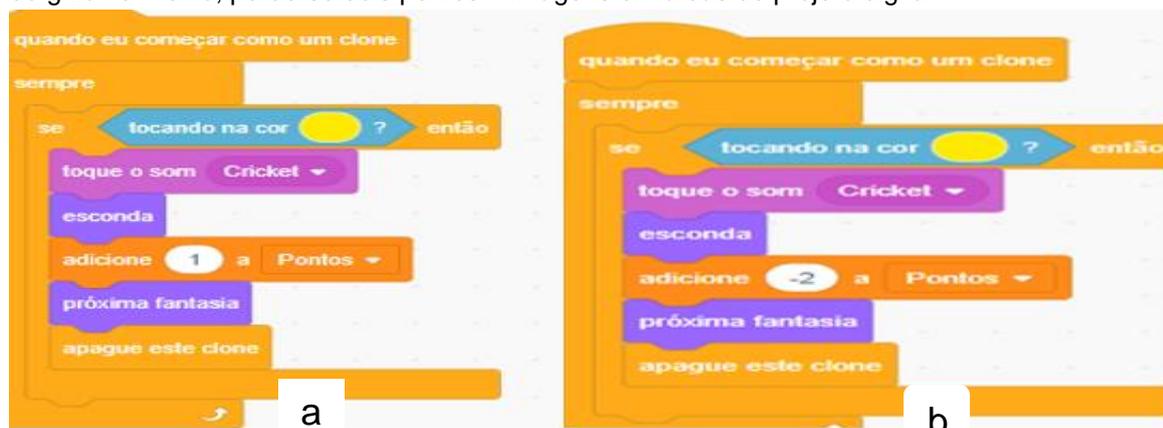
Figura 60 – Setas de comandos - - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412146306/>

Observamos aqui os comandos dos grilos: ao tocar na cor amarela ou adiciona 1 ponto, conforme Figura 61 (a); ou subtrai -2 pontos, conforme Figura 61 (b). A cor amarela é o bico da galinha, e essa cor foi definida por meio do código sensores.

Figura 61 – Comando da contagem de pontos – na imagem (a), se o clone do grilo verde tocar no bico amarelo da galinha, é somado um ponto; na imagem (b), se o bico amarelo da galinha tocar no clone do grilo vermelho, perde-se dois pontos – imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412146306/>

Os cursistas tiveram de clicar no código sensores, “tocando na cor”, conforme Figura 62 (a), e pegar o conta-gotas e ir até o bico da galinha (Figura 62 (b)).

Figura 62 – Contagem de pontos – imagem (a) sensor para encontrar a cor correta do bico da galinha na imagem (b) - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412146306/>

O jogo Merilu Bica, no que tange à decomposição, partiu, como descrevemos anteriormente, da criação do cenário rural com galinhas, pintinhos, celeiro, grama, palhas, grilos, sons de grilo, cata-vento, e se desenvolveu com os comandos setas de comandos, comandos para contagem de pontos. A respeito da Matemática, foi trabalhada a questão dos números naturais (adição e subtração) e criaram como atores principais a galinha e os grilos (verde e vermelho).

#### 5.3.3.1.7 Base dez e base sexagesimal

Nesta história, os cursistas pensaram em trabalhar a história da base dez e da base sexagesimal, com uma linguagem simples para alunos do sexto ano.

Separaram os atores: dois alunos e um professor, escolheram para o trabalho as bases numéricas 10 e 60. Optaram, inicialmente, pela sala de aula e depois pela pista de atletismo, ambas figuras retiradas da internet, assim como os atores (Figura 63).

Figura 63 – Sala de aula e pista de atletismo – imagens extraídas do projeto digital



-Fonte: <https://scratch.mit.edu/projects/412004988/>

“A base 10 representa os metros e a base 60 os segundos” (P6). A ideia do projeto dos cursistas era apresentar para aos alunos que as duas bases podem ser trabalhadas ao mesmo tempo. “Fizemos esse trabalho pensando no trabalho em sala de aula com sexto ano, com linguagem com cuidado maior”.

Apresentaram as medidas como légua e pé, conforme figura 24. “A nossa ideia é para que os alunos pudessem pesquisar essas medidas pela internet” (P10).

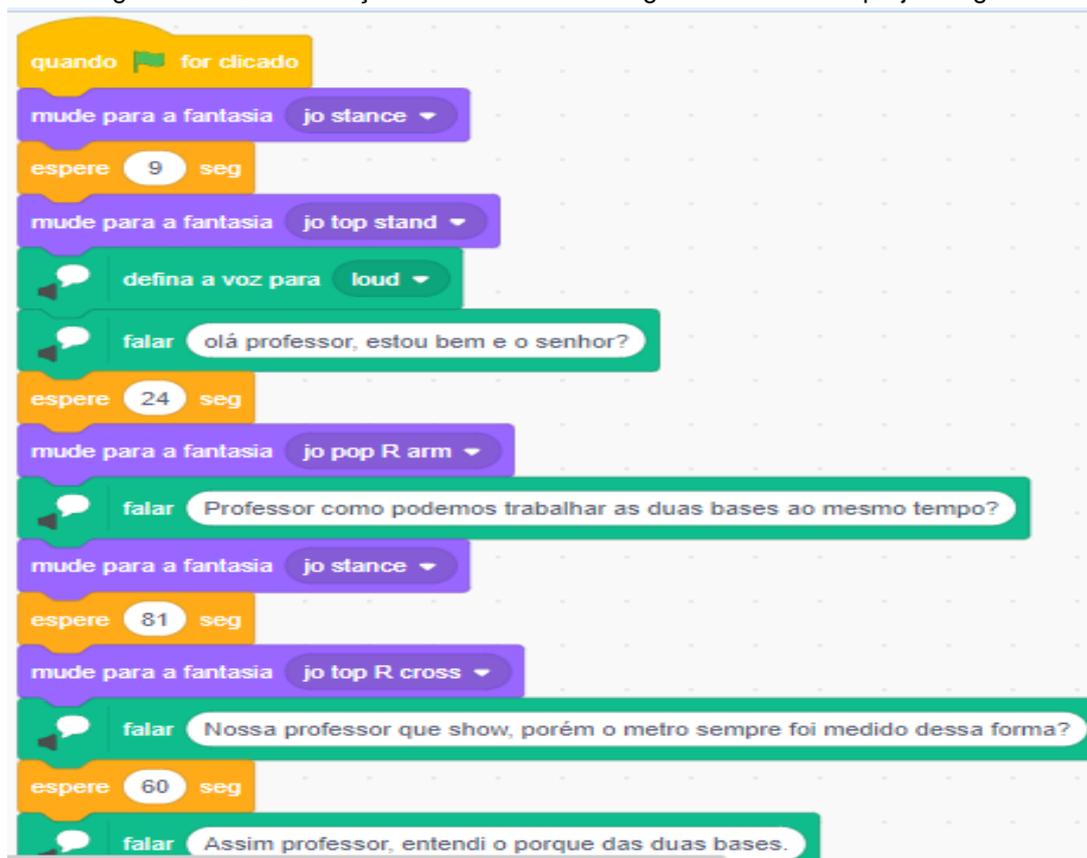
Eles pensaram em trabalhar de forma interdisciplinar e escreveram seu plano de trabalho: “Trabalhamos a Matemática na disciplina de Educação Física” (P10).

No que tange aos comandos, eles trabalharam a história, e essa fica situada na sincronização apresentada na Figura 64.

No que concerne à decomposição, os cursistas escolheram a história para trabalhar o sistema de numeração de bases 10 e 60, escolheram as imagens de atores, pista e sala de aula da internet.

O público-alvo eram os alunos do sexto ano; escolheram uma linguagem no projeto para conversar com esse público; os comandos se basearam na sincronização da história.

Figura 64 – Sincronização de comandos – imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412004988/>

### 5.3.3.2 Reconhecimento de Padrões

A definição desse pilar foi caracterizada no item 2.4.2 e está relacionado à identificação de características comuns entre os problemas como também em suas soluções. Nessa seção, faremos uma análise sobre alguns dos artefatos apresentados pelos cursistas com vistas nas evidências do pilar reconhecimento de padrão.

#### 5.3.3.2.1 – Jogo do Miko 3.1

Neste jogo, partimos do reconhecimento de padrões presente na personagem principal Miko (na programação recebe nome de monkey), conforme Figura 65 (a). Assim que o personagem tocasse nas figuras geométricas, que estavam ao seu redor, a configuração do jogo somava 10 pontos para o usuário.

As figuras geométricas supracitadas estão na Figura 65 (e), como é um jogo criado para a iniciação à geometria, para os primeiros anos do ensino fundamental, os criadores trabalharam apenas com quadrados, círculos, quadrados e os retângulos.

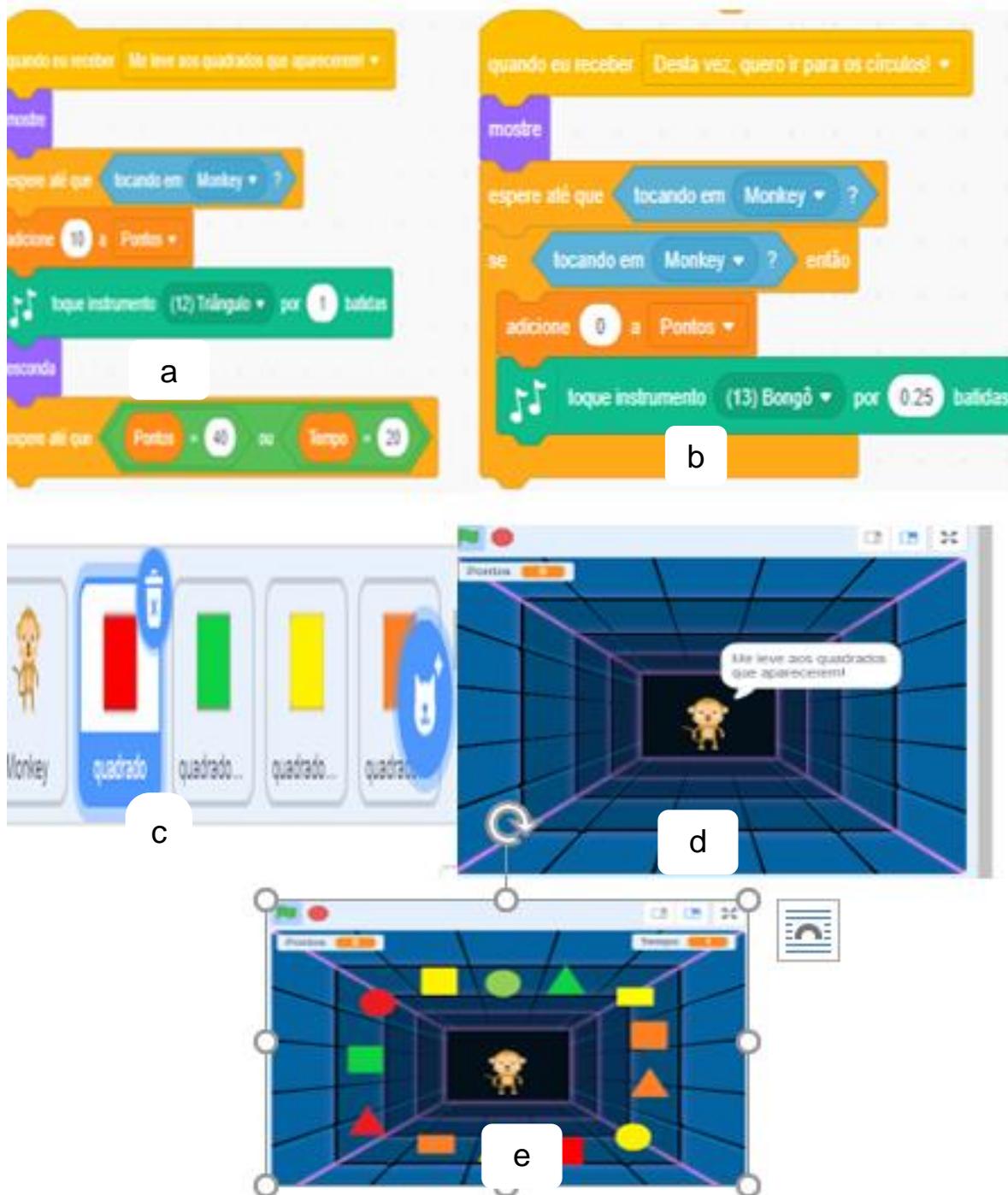
Na Figura 65 (c), apresentamos os comandos do quadrado, o ator aparece na figura em vermelho. Nos comandos, podemos observar que, quando o ator Miko dá o comando para levar aos quadrados, e se realmente levá-los (na Figura 65 (d)), o usuário ganharia 10 pontos ao encostar na cor vermelha, mas, se levar a outra figura, conforme Figura 65 (b), a programação criada previa que o usuário não ganharia nenhum ponto.

Isso acontece, pois, devido à programação, se o usuário levasse o Miko até a figura incorreta, ele não perderia pontos, mas também não ganharia.

De acordo com o reconhecimento de padrões no Jogo do Miko 3.1, analisamos que ele ficou por conta da percepção das cores que foram criadas pelos professores cursistas.

Assim, quando o ator principal Miko encostasse nas figuras geométricas corretas, reconhecendo as cores, sendo essas figuras atores também, conforme observamos na Figura 65 (e), o usuário somaria 10 pontos, do contrário, como já citado, o usuário não contabilizaria pontos.

Figura 65 – Encostar nas figuras geométricas – na imagem (a), ao tocar o quadrado no Miko, ganharia 10 pontos; na imagem (b), ao tocar nos círculos, ganharia 10 pontos; na imagem (c), comando do ator (quadrado) ao tocar no Miko; na imagem (d), ao encostar o Miko na cor vermelha, o usuário ganharia 10 pontos; na imagem (e), ao encostar nas figuras geométricas (cores), o usuário ganharia 10 pontos - imagens extraídas do projeto digital.



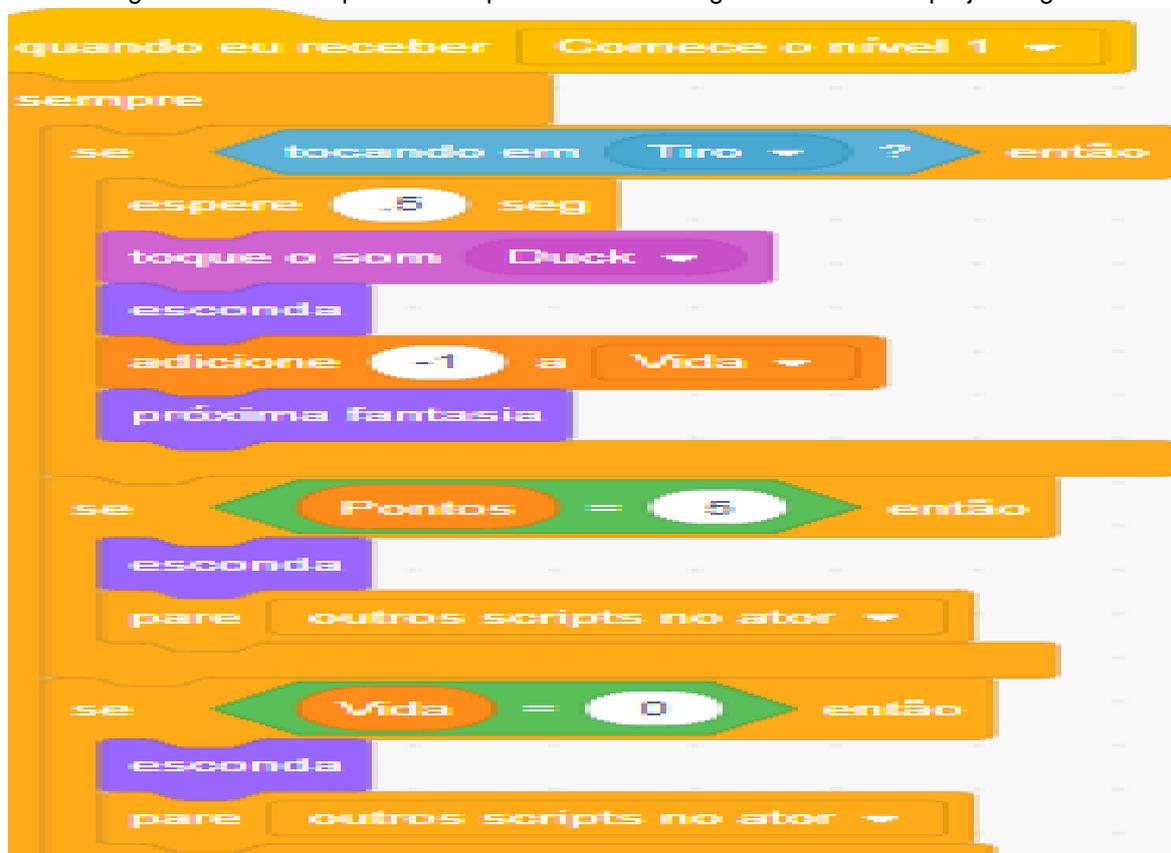
Fonte: <https://scratch.mit.edu/projects/408749283/>

#### 5.3.3.2.2 – Acerte o pato

O jogo acerte o pato, quanto à habilidade reconhecimento de padrão, baseia-se em acertar o pato correto e ganhar pontos, ou atirar no pato errado, perder uma

vida. Segundo P3, “criamos 100 patos no photoshop, e colocamos os patos errados em duas pastas e os patos certos, que apareceriam uma vez só, na tela, para o aluno não acertar no chute. Como o critério de divisibilidade, no primeiro nível, seria 2, separamos os patos pares em uma pasta e os patos errados em duas” (P3, NA).

Figura 66 – Acertar pato errado perderia vida - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412099569/>

Assim, no jogo Acerte o pato, o reconhecimento de padrão se baseia em reconhecer qual pato é o correto e qual é o errado. Podemos observar que o tiro, ao acertar o pato correto, é adicionado 1 ponto; mas, se o tiro acertasse o pato errado, que não faz parte do critério de divisibilidade, o usuário perderia 1 ponto; e, se vida chegasse a 0, o jogo seria parado, conforme Figura 66.

#### 5.3.3.2.3 Merilu Bica

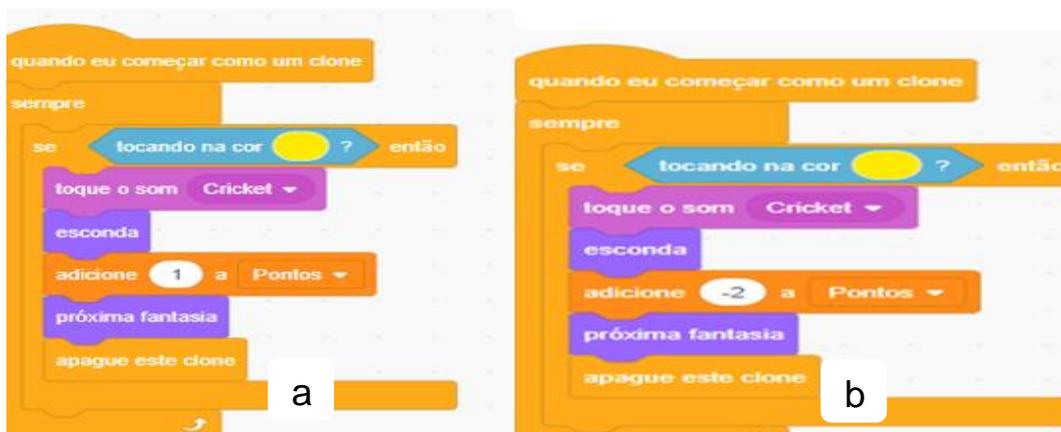
No jogo Merilu Bica, o reconhecimento de padrões ocorre com base no reconhecimento de padrão de cores.

O usuário ao fazer o bico da galinha encostar no grilo verde, soma um ponto e, se encostar no grilo vermelho, perde dois pontos. P13 salienta: “pensamos que,

quando a galinha bicar o grilo correto, soma um ponto, e, quando errar, perde 2, para o aluno caprichar e somar o maior número de pontos possíveis” (P13).

Essa afirmação pode ser observada nas Figuras 67 e 68, quando é utilizado o conta-gotas para reconhecer a cor do bico da galinha.

Figura 67 – Comando da contagem de pontos – imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412146306/>

Figura 68 – Contagem de pontos – imagem (a) conta-gotas para marcar a cor correta do bico da galinha Merilu na imagem (b). Imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412146306/>

Assim, no projeto Merilu Bica, o reconhecimento de padrões se dá pelas cores do bico da galinha e pelas cores dos grilos (verde e vermelho), já que, ao tocar a cor amarela do bico da galinha nos grilos verdes, o usuário ganharia um ponto, e o mesmo bico tocando nos grilos vermelhos, perderia dois pontos.

### 5.3.3.3 – Abstração

A abstração, conforme definição apresentada na seção 2.4.3, é entendida como o conceito que envolve a filtragem dos dados e sua classificação, ignorando elementos que não são necessários, e visando aos que são relevantes.

Como o trabalho foi realizado com professores de Matemática, apresentaremos a abstração da seguinte forma: Os conhecimentos matemáticos que os professores já possuem serão transpostos para a linguagem computacional. Analisaremos os projetos, considerando as particularidades do curso, e observando, neles, como foi realizada a transformação dos conhecimentos matemáticos em linguagem computacional, conforme balizado no item 2.4.3.

Nesta seção, separaremos os projetos por artefatos, sendo três histórias, 3 jogos e 1 simulação.

#### 5.3.3.3.1 – Jogo do Miko 3.1 – Merilu bica – Acerto o Pato

No Jogo do Miko 3.1, a abstração da Matemática para a linguagem computacional aconteceu por meio de seus comandos (Figura 69 e 70).

Corroboramos que, para que todos os projetos se concretizem, eles necessitam da sincronização, mas alguns, por pertencerem a alguns artefatos específicos, contêm a Matemática mais presente em seus comandos e não só na apresentação.

Nesta seção, à transposição da Matemática para a linguagem computacional chamaremos de abstração.

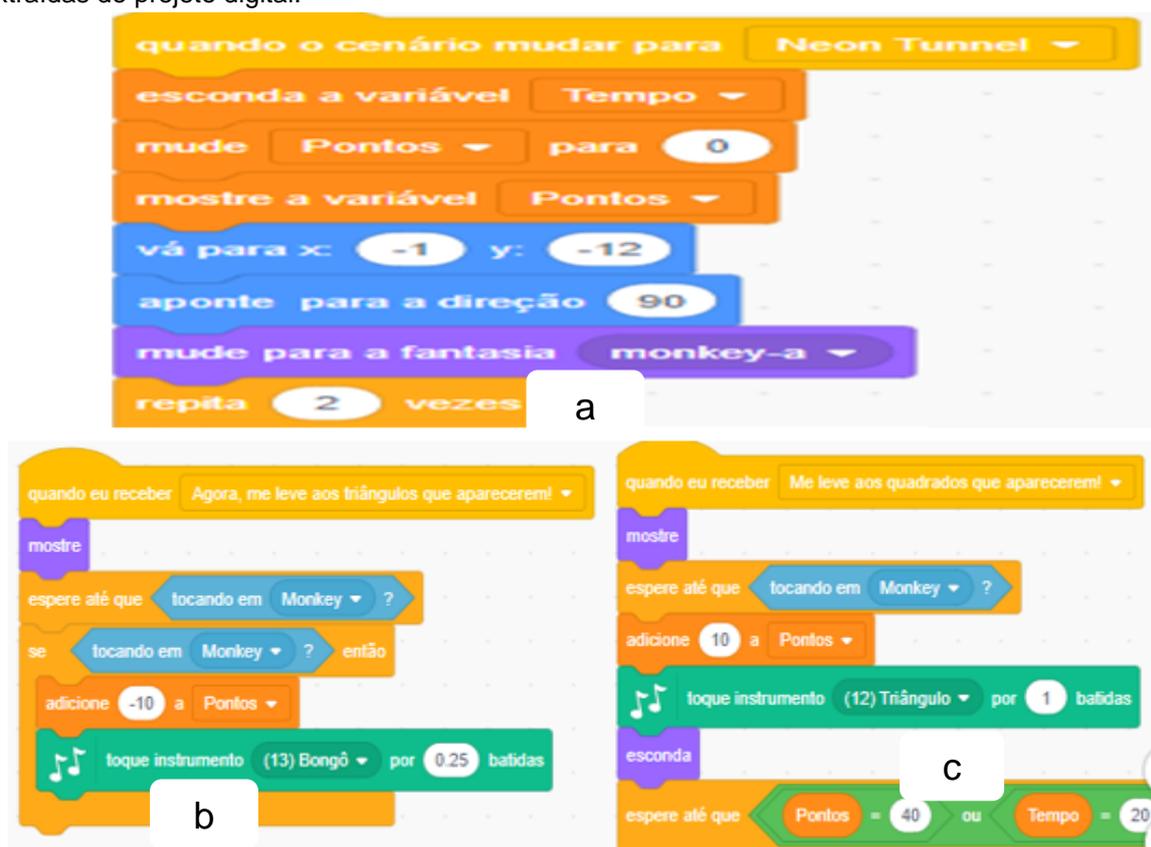
Essa abstração pode ser observada na Figura 69, em que, além da sincronização, existem os comandos de variável, a transposição matemática para a linguagem computacional está presente na localização (eixos x e y) (Figura 69 (a)).

Outro comando que reluz a transposição da matemática para a linguagem computacional está presente no comando de lógica (se então) na Figura 69 (b). Podemos incluir nesse quesito também a contagem de pontos presente nos 3 *frames*, e ainda o tempo no comando “espere até que tocando repita” (Figura 69 (c)).

Nessa transposição, verificamos a questão dos ângulos quando é utilizado o comando “aponte para a direção 90” (Figura 69 (a)).

Nesse projeto, a transposição dos conhecimentos matemáticos para a linguagem computacional não está apenas na sincronização de atores, cenários, sons etc., mas também em atribuir comandos matemáticos na construção do projeto.

Figura 69 – Jogo do Miko 3.1 – imagem (a) localização e variável eixos x e y; na imagem (b), a lógica matemática em “se; então”; e, na imagem (c), o comando “espere até que tocando repita” - imagens extraídas do projeto digital.



Fonte: <https://scratch.mit.edu/projects/408749283/>

Nessa mesma linha, está o projeto Merilu Bica, pois esse projeto (Figura 70) está marcado por comandos que caracterizam a abstração que vimos ressaltando, praticamente são comandos matemáticos semelhantes entre esse projeto e o Jogo do Miko 3.1.

No projeto Acerte o pato, a abstração ocorre nos mesmos moldes dos outros dois projetos, ou seja, apresentam-se alguns comandos novos, transpondo alguns comandos matemáticos para a linguagem computacional.

As ondas estão apresentadas em camadas. Dessa forma, a mira do jogo também trabalha seguindo essas camadas (Conf. Figura 71 (a)).

Ainda, nessa mesma figura, temos os comandos deslizar e também a aleatoriedade (probabilidade), que podem ser visualizados na parte (a) da figura, nos comandos: “espere número aleatórios entre 3 e 15 segundos para mudar de fantasia”.

A abstração aqui ilustrada no jogo acerte o Pato tem uma diversidade de comandos que faz com que a Matemática esteja presente no interior do projeto.

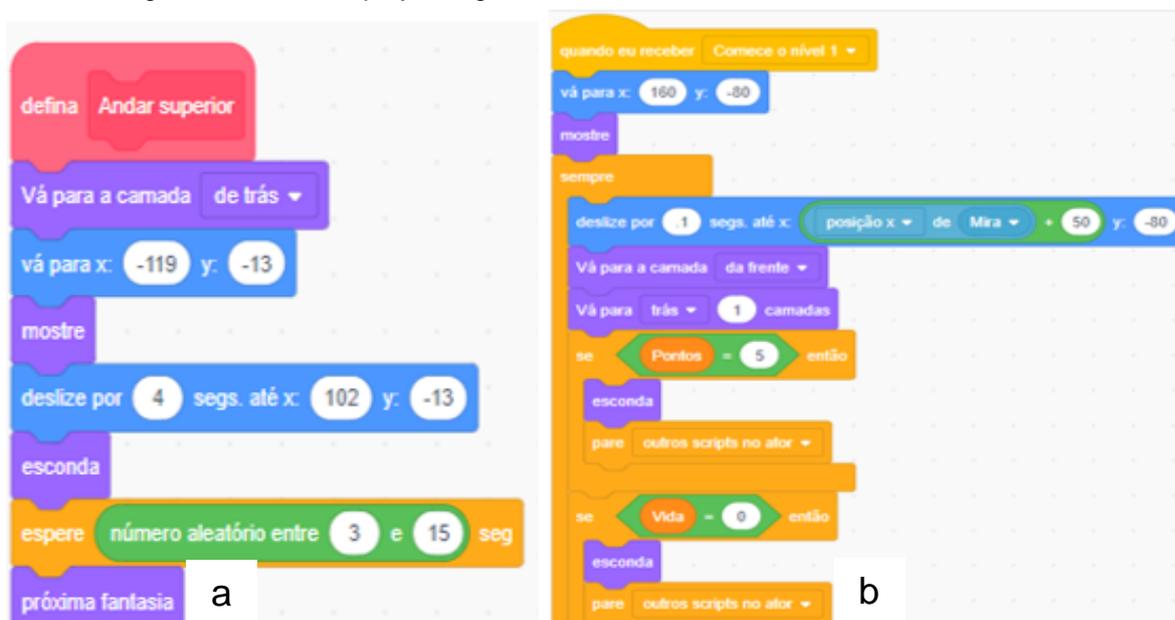
Figura 70 – Merilu Bica – Abstração – na imagem (a), a transposição está na lógica, “se, então” “sempre”, no tempo, na contagem de pontos; já na imagem (b), temos as variáveis, blocos lógicos-imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412146306/>

Os jogos apresentados, por ter algumas características específicas no que tange à abstração, apresentam vários comandos que demonstram a Matemática presente, conforme Figuras 69, 70 e 71.

Figura 71 – Acerte o pato – blocos de comando na imagem (a) são para criar as ondas, que servem de base para os patos saírem e para a arma atirar; já na imagem (b), temos a probabilidade no comando deslize - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412146306/>

### 5.3.3.3.2 – Brincando com simetria

O projeto Brincando com a simetria representou uma simulação no que tange à abstração aqui entendida como a transposição dos conhecimentos matemáticos para a linguagem computacional. Ele oferece comandos interessantes, conforme Figura 72.

Nessa imagem, constatamos a presença da Matemática. Temos o eixo das abscissas e ordenadas, o qual tem valor de -1 para as abscissas e +1 para as ordenadas. É possível observar a linguagem Matemática em simbiose com a linguagem computacional, pois, com seu conhecimento matemático, vislumbramos como se realiza essa abstração.

Além desse conceito, também na Figura 72, é visto a questão da espessura, da lógica (se; então; senão), e também das probabilidades, quando existe um número aleatório entre um e 20.

Figura 72 – Produção matemática - imagens extraídas do projeto digital



Fonte: <https://scratch.mit.edu/projects/412094139/>

Trabalhamos os conceitos citados pelo autor em relação à simetria com o Scratch; a tecnologia proporcionou a visualização automática do que estava se fazendo instantaneamente, desenhava de um lado e aparecia do outro; assim, com a utilização do *software*, o aluno entende os conceitos de simetria de uma forma lúdica.

#### 5.3.3.4 – Algoritmo

Como definido na seção 2.4.4, algoritmo é um plano, uma estratégia, com um conjunto de instruções claras e necessárias para solução de um problema.

Para avaliar a mobilização da habilidade de algoritmo pelos professores cursistas ao produzirem artefatos no Scratch, utilizamos um recurso digital desenvolvido especificamente para identificar um conjunto de conceitos de algoritmo (Conf. Quadro 5, na seção 3.4) na programação de artefatos produzidos nessa plataforma. Esse recurso é chamado de Dr. Scratch.

Inegavelmente, não podemos reduzir a avaliação do desenvolvimento da habilidade de algoritmo a uma mera análise baseada na identificação automática de trechos de códigos, realizada por um sistema, e associada a um certo grau de domínio de alguns conceitos da programação, como, por exemplo, a identificação de blocos de operadores relacionais e lógicos estar associada a um certo nível de domínio de lógica computacional. Contudo, essa varredura automatizada no código de programação de um artefato no Scratch pode, sim, dar-nos indicativo de certo nível de complexidade do algoritmo implementado.

Assim, foi usado esse recurso (Dr. Scratch) e analisamos projeto por projeto, iniciando pelas três histórias, seguidas pelos três jogos e, por último, a simulação, buscando estabelecer relações entre a pontuação dada pelo programa avaliador e o desenvolvimento da habilidade de algoritmo.

##### 5.3.2.4.1 – Função exponencial

Neste projeto, conforme podemos observar na Figura 73, não foi obtida a nota de lógica, pois, conforme observamos nos outros módulos, esse projeto não foi caracterizado pela apresentação dos blocos lógicos “se, então”; por ser uma história, em alguns projetos não foi necessária a sua utilização para que a história ocorresse.

Em paralelismo e sincronização, obtiveram a nota máxima três (3), justamente por terem utilizado os comandos “quando recebo a mensagem” e “transmita e aguarde” de forma sistemática.

A interatividade com o usuário obteve nota dois (2) no Dr. Scratch, essa avaliação se deu, porque, além de disparar o projeto na bandeira verde, ele pergunta o nome do usuário e foi esperado dele essa ação de se nomear.

Na representação de dados, a nota foi um (1), pois foram usados apenas os modificadores do ator, não foram utilizadas variáveis e nem listas de comandos.

Figura 73 – Nota da história Função exponencial



Fonte: Relatório do Dr. Scratch

No controle de fluxo, a nota foi dois (2), pois, no projeto, existe o comando “sempre”; se tivesse o comando “repita até”, a nota seria três (3).

Na abstração, a nota foi um (1), pois, nesse projeto, o criador não utilizou blocos nem clones; utilizou apenas mais de um comando e mais de um ator.

Nesse projeto, a nota obtida foi doze (12), por se tratar do artefato história em que alguns comandos de lógica, de abstração, representação de dados não são necessários para sua concretização, e, portanto, não apareceram de forma destacada nos projetos, já que, para que a história fosse construída, alguns comando não foram necessários.

#### 5.3.3.4.2 – Problema dos 35 camelos

Esse projeto também é uma história e podemos observar que sua nota de lógica também não aparece (Ver Figura 74), pelo mesmo motivo citado no projeto da Função exponencial. Nos conceitos de paralelismo e sincronização, também as notas foram as máximas. Citamos os mesmos motivos do item 5.3.2.4.1 para representação de dados e abstração.

Nesse projeto, conforme ilustra a Figura 74, a interatividade com o usuário obteve a nota um (1). Essa nota se justifica pelo projeto ter como disparador a bandeira verde, e nem mais uma interação.

Esse projeto ainda apresenta o controle de fluxo com nota um (1), pois só apresentou a sequência de blocos, não apresentando o comando repita.

Figura 74 – Nota do Problema dos 35 camelos



Fonte: Relatório do Dr. Scratch

A nota obtida foi dez (10), pois se tratou de uma história e, conforme projeto anterior, alguns conceitos não foram utilizados, porque, na história, esses conceitos não interferem no produto final.

#### 5.3.3.4.3 – Base dez e base sexagesimal

Conforme Figura 75, podemos observar que a nota de lógica não apareceu pelo motivo já citado nos dois primeiros projetos.

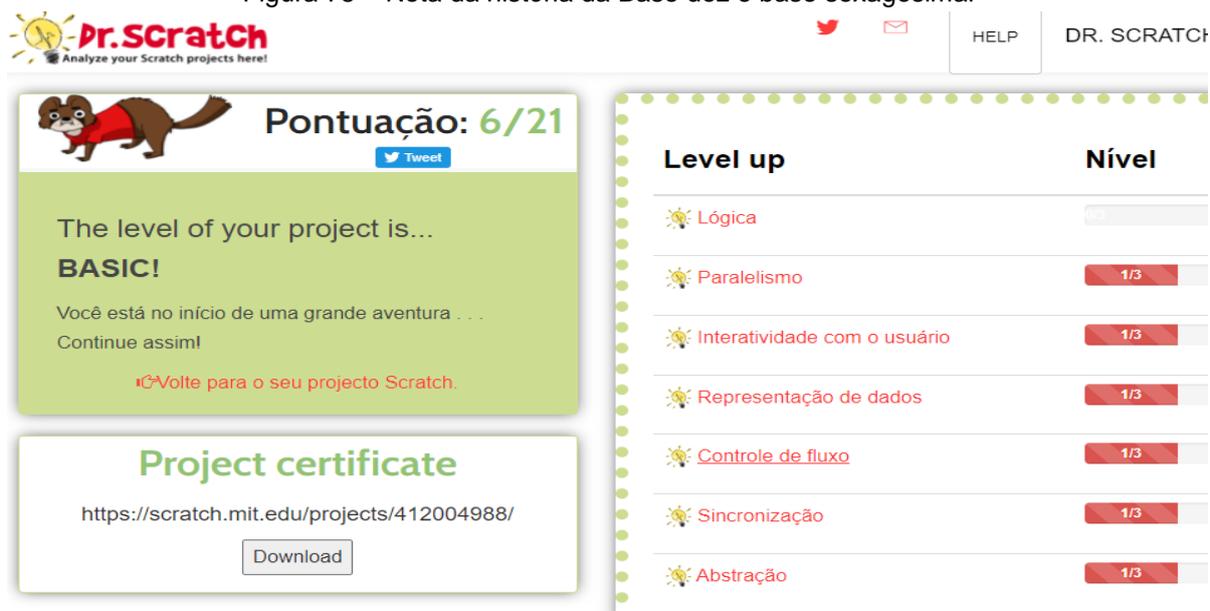
Diferentemente das duas primeiras histórias, esse projeto obteve a nota um (1) no paralelismo, porque o projeto utilizou apenas dois comandos na bandeira verde. Não foram utilizados dois comandos na tecla pressionada, ou dois comandos para um mesmo ator (nota dois), e nem quando “recebo comando” ou “quando cenário muda” (nota 3), tudo isso ao mesmo tempo.

Quanto à sincronização, também ficou com a nota um (1), pois só foi utilizado o comando “espere”, ou seja, não foram utilizados comandos como “transmita, receba, pare tudo, espere até”.

A interatividade com usuário ocorreu apenas ao acionar a bandeira verde para iniciar o jogo, por isso a nota foi um (1). A mesma nota obtida para representação de dados, haja vista que não foram utilizadas nem as variáveis nem as listas, ficando apenas nos modificadores de propriedades do ator.

O controle de fluxo também obteve nota um (1), já que os comandos se detiveram na sequência de blocos. Nesse projeto, a nota obtida foi seis (6). Já no que concerne à abstração, a nota também foi um (1), pois o projeto não utilizou blocos ou clones.

Figura 75 – Nota da história da Base dez e base sexagesimal



Fonte: Relatório do Dr. Scratch

No geral, observamos que os projetos de história apresentaram algumas peculiaridades: tiveram nota máxima em sincronização e paralelismo, não apareceu a lógica, e, assim, a interatividade com o usuário, a representação de dados, o controle de fluxo e a abstração ficaram comprometidas.

Reafirmamos que essas notas não interferem tanto na qualidade do produto final no que diz respeito ao artefato história, pois elas, em geral, baseiam-se no conteúdo trabalhado para um público definido, pautando-se apenas na sincronização e no paralelismo.

#### 5.3.3.4.4 – Acerte o Pato

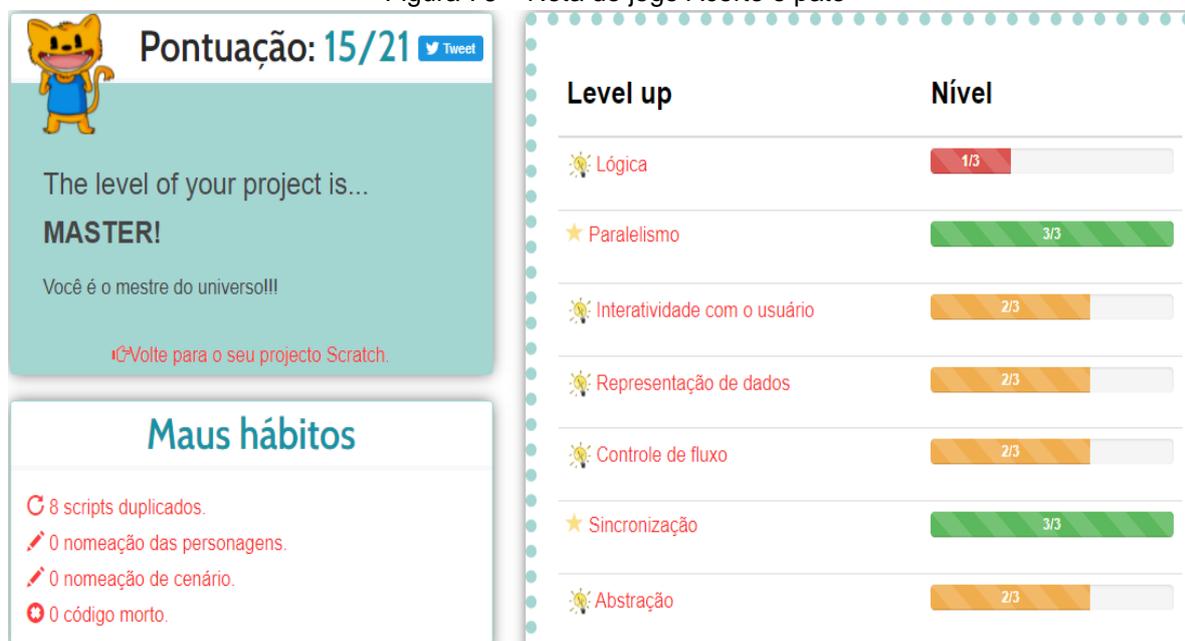
No jogo Acerte o pato, já observamos o aparecimento da lógica, aqui o projeto tirou nota um (1) nesse conceito (Conf. Figura 76). Essa nota ocorreu pelo fato de eles não terem utilizado o comando “se então; senão”, que daria nota dois (2), ou operadores lógicos “e, ou”, conforme demonstrado na Figura 76.

O paralelismo e a sincronização, nesse projeto, obtiveram notas máximas (nota 3), pois foi utilizado o comando “quando receber mensagem”, “espere 5 s”. Em interatividade com o usuário, a nota foi dois (2), embora o projeto tenha sido bem interativo, faltou interação com áudio ou vídeo; se houvesse essa interação, a nota seria três (3).

A representação de dados também obteve a nota dois (2), P3 e P4 utilizaram variáveis, mas não utilizaram listas, por isso, a nota foi dois (2). O controle de fluxo

também teve nota dois (2), pois faltou o comando “repita sempre”. A abstração também ficou com nota dois (2), porque faltou a utilização de clones. A nota recebida no projeto foi quinze (15).

Figura 76 – Nota do jogo Acerte o pato



Fonte: Relatório do Dr. Scratch

O Jogo do Miko 3.1 foi o único projeto que obteve nota máxima (nota 3) em lógica, pois eles utilizaram um dos comandos de operadores lógicos (“ou”), conforme Figura 77.

#### 5.3.3.4.5 Jogo do Miko 3.1

Figura 77 – Nota do Jogo do Miko 3.1



Fonte: Relatório do Dr. Scratch

Em Paralelismo e sincronização, pelos motivos já citados, o projeto tirou a nota três (3) em ambos.

Figura 78 – Comando interno com o bloco lógico "ou".



Fonte: <https://scratch.mit.edu/projects/408749283/>

Na interatividade com o usuário, devido não ter interação com áudio ou vídeo, obteve a nota dois (2). O controle de fluxo também foi o único projeto que obteve a nota três (3), pois tem o comando 'repita até', conforme Figura 79.

Figura 79 – Comando repita até



Fonte: <https://scratch.mit.edu/projects/408749283/>

Na abstração, o projeto obteve a nota um (1), pois os criadores não utilizaram blocos nem clones. O Jogo do Miko 3.1, na avaliação do Dr. Scratch, alcançou a nota dezessete (17) (Figura 77), utilizando critérios do Quadro 5 da seção 3.4.

#### 5.3.3.4.6 – Merilu Bica

O Jogo Merilu bica, no que tange à lógica, na avaliação do Dr. Scratch, obteve a nota um (1), pois não utilizou o comando "se então; senão" para tirar a nota dois (2) (Figura 80). No paralelismo, o projeto tirou a nota máxima, pois havia mais que dois comandos "quando eu receber a mensagem".

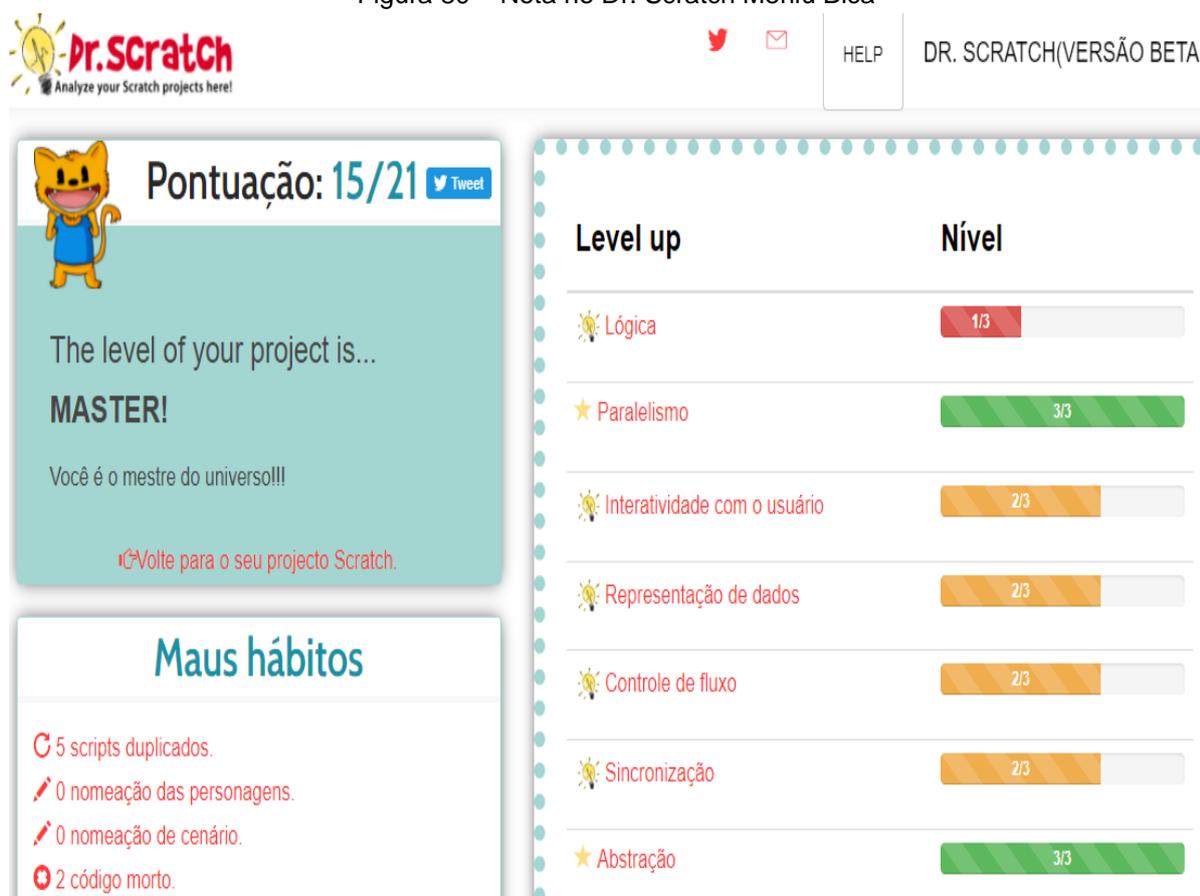
No conceito interatividade com o usuário, obteve a nota dois (2), por não ter a interatividade por áudio e vídeo com o usuário. Mesma nota para representação de dados, pois não foi criada lista de variáveis. Para o controle de fluxo, a nota foi dois (2), já que não utilizaram o comando "repita até", se fosse feito isso, teria tirado três (3).

Na sincronização, a nota foi dois (2), pois a criação não utilizou os comandos “espere até; quando o fundo mudar”, “transmita e aguarde”. Em abstração, o projeto obteve a nota máxima três (3) pois os criadores utilizaram clones para os grilos.

O Jogo Merilu bica, na avaliação do Dr. Scratch, alcançou a nota quinze (15) (Figura 80), utilizando critérios estabelecidos.

Observamos que a avaliação do Dr. Scratch para os jogos, em relação às notas, estas são maiores do que nas histórias, isso se dá, conforme afirmamos anteriormente, pelo fato de que os jogos necessitam de mais comandos para rodar, ou seja, os jogos preenchem mais critérios, apresentados no Quadro 5 da seção 3.4, e, por esse motivo, as notas são maiores.

Figura 80 – Nota no Dr. Scratch Merilu Bica



Fonte: Relatório Dr. Scratch

#### 5.3.3.4.7 – Brincando com a simetria

O projeto Brincando com a simetria foi a única simulação. Esse projeto, no que se refere à utilização dos comandos, assemelha-se aos jogos citados.

Ele obteve a nota dois (2) no conceito lógica (Conf. Figura 81); ele preenche os dois primeiros critérios “se então” para nota um (1); e “se então; senão” para a nota dois (2), mas não possui operadores lógicos “e” ou “ou”, os quais, na Matemática, chamamos de conectivos.

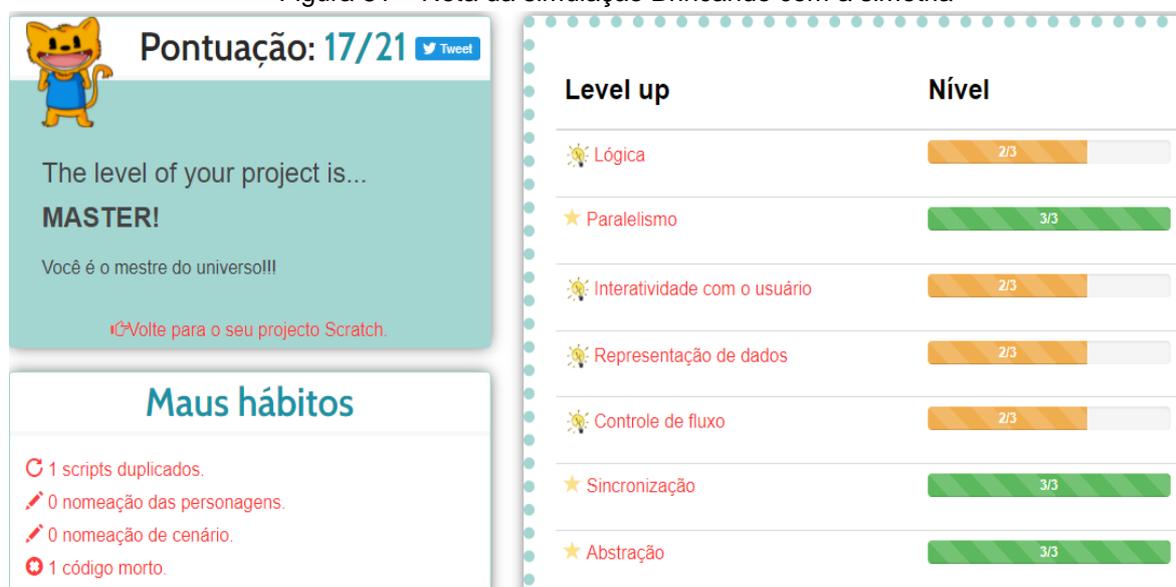
No conceito paralelismo, ele preencheu todos os requisitos, por isso tirou a nota três (3). Já na interatividade com o usuário, por não utilizar áudios e vídeos para essa interação, obteve a nota dois (2). Essa mesma nota foi para representação de dados, por não utilizar a lista de variáveis (são coleções de variáveis do mesmo tipo agrupadas por um único identificador). Nenhum projeto conseguiu tirar a nota máxima, nesse quesito, pela complexidade de se realizarem as listas de variáveis.

Para o controle de fluxo, a nota também foi dois (2), pois utilizaram o “comando de blocos” e “repita sempre”, mas não utilizaram o comando “repita até”.

No conceito sincronização, o projeto obteve a nota três (3), por utilizar todos os critérios previstos para esse conceito, conforme Quadro 5, na seção 3.4. Na abstração, também obteve a nota máxima três (3), pois, no projeto, está o comando clone (para o ator borboleta), conforme Figura 82.

No que tange à utilização da habilidade algoritmo, entendemos ser melhor a análise do Dr. Scratch, que foi utilizada pelos cursistas durante suas construções no curso.

Figura 81 – Nota da simulação Brincando com a simetria



Fonte: Relatório do Dr. Scratch

O algoritmo, utilizado pelo Dr. Scratch para apresentação do relatório, não estigmatiza os projetos com notas inferiores, mas demonstra quais comandos foram utilizados para determinados artefatos.

Figura 82 – Comando clone projeto Brincando com a simetria



Fonte: <https://scratch.mit.edu/projects/412094139/>

Observamos que as histórias obtiveram notas inferiores a jogos e simulação, isso se deve ao fato de, como afirmado, a história não necessitar de tantos comandos ou de comandos mais complexos para a sua construção.

Nesse contexto, o Dr. Scratch é uma importante ferramenta para avaliar construções, pois ele aponta em que o *Scratcher* pode melhorar seu projeto na questão dos comandos, aprendendo e usufruindo do máximo que o *software* possui.

Como exemplo do supracitado, podemos demonstrar a questão da lista de variáveis, pois nenhum projeto conseguiu apresentá-la, esse pode ser um começo para outro estudo e outras produções se debruçarem sobre o tema e utilizar esse comando para obter a nota máxima nesse conceito.

### 5.3.4 Metatexto relativo à subcategoria 2.2

Para essa análise, atemo-nos aos quatro pilares do PC, assim chamadas as quatro habilidades fundamentais do pensamento computacional.

No campo da decomposição, como ficou evidenciado na seção anterior, o Scratch favoreceu o exercício da decomposição, principalmente para o planejamento da programação. Nesse sentido, P2, em sua entrevista, afirmou: “você separa os passos, você separa tudo que você fez, esse caminho é o mais importante que o produto final, acredito que seria a decomposição, isso fazemos na matemática para resolução de problemas” (P2, Entrevista).

O caminho apontado por P2 é de extrema relevância já que é nele que passamos e voltamos diversas vezes, separamos o problema em partes menores para buscar a resolução. Esse também foi o caminho dos demais projetos.

Sobre isso, Moretti (2019) enfatiza:

Quebrar as tarefas em fragmentos menores torna o processo de solução mais claro e erros podem ser identificados com maior facilidade (processo de depuração). Programadores profissionais pensam em termos de decomposição o tempo todo (MORETTI, 2019, p. 25).

Se profissionais da programação pensam em decomposição em todos os momentos de sua criação, imaginamos que os iniciantes também devem pensar, desenvolvendo assim essa habilidade. Ao menos isso foi evidenciado por P13 quando afirmou: “primeiro fui na introdução, esse momento foi fácil, depois eu fui pensando parte por parte para não complicar” (P13, Entrevista).

Essa afirmação de P13 também estava na argumentação dos demais cursistas. P5, por sua vez, disse: “eu gosto de resolver o problema passo-a-passo” (P5, Entrevista). Sobre isso, Brackmann (2017) destaca:

Quando um problema não está decomposto, sua resolução é muito mais difícil. Ao lidar com muitos estágios diferentes ao mesmo tempo, torna-se mais difícil sua gestão. Uma forma de facilitar a solução é dividir em partes menores e resolvê-las, individualmente (BRACKMANN, 2017, p. 34).

Uma característica do Scratch, observada e evidenciada pelos professores, a qual faz mobilizar a habilidade da decomposição, foi a estruturação da plataforma para que a programação de diferentes atores e cenários, que irão compor um artefato, seja realizada de forma separada um do outro. Isso faz com que o planejamento para a implementação do todo seja pensado em partes, desenvolvendo assim, a habilidade da decomposição.

Portanto, diante das diversas evidências, ficou claro para nós que os professores consideram que o uso do Scratch contribui para o desenvolvimento desse pilar (decomposição) do pensamento computacional.

Quanto ao pilar reconhecimento de padrões, esse foi mais destacado pelos professores cursistas diretamente relacionado ao trabalho de programação no Scratch. Conforme os cursistas foram se ambientando ao *software*, iam reconhecendo padrões da programação, como fica claro na fala de P6: “quando eu fiz a primeira história foi complexa, mas na segunda foi mais simples, eu já conhecia os comandos para fazer uma história simples” (P6, Entrevista).

Realmente, o início de qualquer aprendizado é mais desafiador, por não se ter padrões estabelecidos, como destacou P9: “tem que ter paciência, no começo é difícil, mas depois que aprende vai embora” (P9, Entrevista).

Observamos essa facilidade e a melhora dos projetos conforme os cursistas foram criando, testando, errando, fazendo novamente; em todos esses momentos, os cursistas estavam evoluindo, para tanto, tomamos como exemplo a narração de P3: “você cria um bloco para dizer que aquele bloco é o bloco andar, depois de criado o bloco é só colocar que ele sempre vai fazer a mesma coisa, não são necessários criar todos comandos” (P3, Entrevista).

Como observamos, em alguns projetos, foram criados blocos para não ficar repetindo os comandos várias vezes, ou seja, algoritmos que foram criados para resolver um problema, poderiam resolver vários.

Brackmann (2017) reforça o que expusemos quando afirma que:

Através do reconhecimento de padrões, é possível simplificar a solução de problemas e replicar esta solução em cada um dos subproblemas, caso haja semelhança. Quanto mais padrões se consegue encontrar, mais dinâmico e rápido a macro solução é encontrada (BRACKMANN, 2017, p. 37).

Sobre a abstração, com foco na transposição da linguagem e saberes matemáticos para a linguagem de programação, os professores evidenciaram que a programação no Scratch favorece o desenvolvimento dessa habilidade, pois, além de simples, consideram a linguagem do programa atrativa para os estudantes, o que os motivaria para produzirem artefatos relacionados aos objetos do conhecimento matemático abordados em sala de aula, esses artefatos, por sua vez, exigiriam a transposição de linguagem.

Nesse sentido, P3 argumenta que: “como professor de matemática pode ser o primeiro a introduzir meus alunos no ambiente de programação, fazendo com que eles construam jogos matemáticos”. Aqui, o professor demonstra compreender a importância da transposição da linguagem matemática para a computacional como forma para desenvolver a abstração e também habilidades relativas à Matemática.

O prisma pelo qual analisamos o desenvolvimento da abstração nos artefatos que os professores cursistas desenvolveram refrata o que afirmam Marques et al. (2017) sobre este pilar do pensamento computacional:

A abstração dos objetos envolvidos não representa apenas quantidades ou figuras e precisam ser representados por estruturas adequadas que incluam as informações relevantes. A identificação e representação destas abstrações para diferentes tipos de processos são o foco do PC (MARQUES et al., 2017, p. 316).

Já no pilar do algoritmo, buscamos o Dr. Scratch para entender o processo das notas obtidas através dos relatórios fornecidos pelo programa em todos os projetos.

P5, ao afirmar: “eu tive 12 de 21 [...]”, entende que a nota do Dr. Scratch se deu, porque “[...] a lógica não consegui ainda, talvez seja pelo motivo de ser uma história e é a primeira história que faço para alunos, achei uma nota boa 12, já que estou iniciando a linguagem de programação para o ensino”.

Realmente, por ser uma história e estar iniciando com a programação, a nota de P5 foi considerada uma boa nota, justamente, por todos os critérios visualizados na seção anterior, e pela dificuldade de se colocar os comandos de lógica em histórias.

No cômputo geral, no que se refere a histórias, a lógica praticamente não apareceu. Nesse sentido, Amorim (2020) explica que

O critério lógica foi comumente avaliado com notas baixas, acredita-se que esse resultado pode ser explicado, pois este critério é avaliado pelas estruturas de controle e operadores lógicos constantes no sistema, e do uso de operadores lógicos, e estes conceitos possuem uma complexidade maior (AMORIM, 2020, p. 19)

Amorim argumenta que a lógica, analisada no projeto dos seus cursistas, obteve notas baixas; em nosso curso, as notas baixas apareceram nas histórias, mas, nos jogos e na simulação, isso não aconteceu.

Acreditamos que, pelo artefato produzido ser a história, esse conceito ficou comprometido. P10 entendeu essa situação e afirmou: “nossa nota ficou comprometida por ser uma história e utilizamos uma linguagem simples para aluno de 6º ano, por isso acredito que não teve lógica”.

Temos o mesmo entendimento do cursista P10, pois, dependendo de como proceder a história e a linguagem utilizada, a nota no Dr. Scratch pode ser baixa, só analisarmos que os jogos e simulação obtiveram notas mais altas, justamente pelos comandos que os artefatos propõem.

P11 fez um relato interessante: “eu tinha tirado 15, minha abstração estava com a nota 1, fui lá no Dr. Scratch para entender como ele avaliava a abstração, e percebi que se eu utilizasse um clone, tirava nota máxima, fiz o clone da borboleta e tirei 3”.

Mesmo sem saber dos critérios, previstos no Quadro 5 desse trabalho, P11 foi procurar no Dr. Scratch e observou que o algoritmo usado pelo programa tinha alguns caminhos que melhorariam sua nota e foi assim que ele fez e conseguiu, pois buscou entender o algoritmo que atribuía a nota.

Não podemos afirmar que o Dr. Scratch é a melhor forma de avaliar um projeto no Scratch, mas ele nos abre possibilidades de ampliar os conhecimentos através da busca de novos comandos para evoluir na nota e nas habilidades. Além disso, ele nos dá um norte para a evolução na construção de projetos mais audaciosos.

Dentre as habilidades desenvolvidas com os quatro pilares, não observamos hierarquia entre elas, pois, ao mesmo tempo que dissemos que, sem a decomposição, fica difícil a construção de um projeto, porque ela é importante por realizar a quebra do projeto em partes menores, também afirmamos que não existe um projeto sem um plano, uma estratégia (algoritmo). E, assim, para melhorar o projeto, há necessidade de se reconhecer padrões até para ganhar tempo e fazer projetos melhores. Por fim, a abstração foi significativa para o curso, pois realizou a transposição dos conhecimentos matemáticos em linguagem computacional.

Após a contextualização apresentada nas seções 5.3.2 e seguintes, ficou evidenciado que os professores mobilizaram e perceberam tais habilidades do PC quando se utiliza o *software* Scratch para o desenvolvimento de artefatos.

## 6. CONSIDERAÇÕES FINAIS

Neste estudo, foi proposto um curso de formação de 80 horas com 14 professores de Matemática de Tangará da Serra para analisar as vantagens e desvantagens do *software* Scratch para o ensino da Matemática e desenvolvimento do pensamento computacional.

A análise nos permitiu observar que os professores não foram unânimes quanto aos artefatos que podem ter um melhor efeito pedagógico, pois quatro cursistas (P6, P7, P12 e P13) acreditam que o contexto pode apresentar o que é melhor para o momento, esses acreditam que os três artefatos (história, simulação e jogos) podem ser utilizadas; cinco deles (P1, P2, P4, P9 e P14) acreditam que o jogo pode produzir esse efeito; e quatro (P3, P5, P8, P10) acreditam que a história tem um melhor efeito pedagógico, mas, para P11, o melhor efeito é história com simulação.

A partir desses dados, consideramos que todas as produções foram relevantes, porque, mesmo sendo menos citadas individualmente, quatro professores acreditam que, em vários contextos, podem ser usadas as simulações, e um professor (P11) acredita que a soma de história e simulação é a melhor pedagogicamente.

No que concerne as habilidades matemáticas, foi possível observar nos planos de aulas dos professores que trabalharam 14 habilidades ligadas a BNCC, contemplando desde os anos iniciais do ensino fundamental ao ensino médio, salientando que para a educação infantil não foi trabalhada habilidade, pois dentre os professores não havia nenhum dessa etapa.

Quanto às habilidades mobilizadas do pensamento computacional, os cursistas de u forma geral acreditam que podem ser mobilizadas todas elas, pois, dependendo do artefato a ser criado, podem mobilizar mais ou menos habilidades. Entretanto, durante o curso, eles entenderam que foram mobilizadas todas as habilidades nas produções realizadas. As habilidades estão presentes no Quadro 3, na junção de pilares com os conceitos (habilidades de programação desenvolvidas), e descritas no item 5.3.3 e seus subitens desse trabalho.

No que tange às características do Scratch, quanto as vantagens eles entenderam que a interdisciplinaridade é uma delas, também é possível realizar o trabalho colaborativo com produções diferentes, a ludicidade também apareceu nessa característica e que o *software* contribui no desenvolvimento do protagonismo.

Nessa linha, analisaram que o Scratch é um potencializador nas aulas de Matemática para desenvolver o pensamento computacional, já que ele impulsiona as habilidades de criação, implantando junto novas metodologias, por exemplo, a ABP.

Quanto às desvantagens, alguns cursistas apontaram que o tempo das produções foi extenso, como foi o caso de P5, que trabalhou quatro horas no projeto do caminhão, e que, embora recompensado pela bela produção, acredita que o pouco espaço para colocar os comandos foi o motivo dessa demora.

Os professores disseram que a sincronização é um pouco complicada de se trabalhar, pois, quando as produções começam a exigir mais comandos, a sincronização fica um pouco prejudicada até pelo pouco espaço que tem para colocar os códigos.

Nesse sentido, outra desvantagem citada foi a questão de que, quando colocavam comandos, eles travavam tudo, principalmente os operadores. Entretanto, eles acreditam que o *software* pode melhorar no sentido de otimizar esses códigos.

Por fim, os professores disseram que o *software* Scratch é um potencializador para o ensino da Matemática e desenvolvimento do pensamento computacional em sala de aula, vislumbraram várias formas de se utilizar o *software* para esse trabalho e acreditam que os alunos aprenderão rapidamente a programar com o Scratch.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Maria Elizabeth Bianconcini de; VALENTE, José Armando. Pensamento computacional nas políticas e nas práticas em alguns países. **Revista Observatório**, [s. l.], v. 5, n. 1, p. 202–242, 2019.

AMORIM, Louis dos Santos; **O Scratch e o pensamento computacional: uma experiência vivenciada pelos graduandos de pedagogia do Uniceplac**. Brasília, 2020, 22 p. Monografia (Curso de Sistemas de Informação), Uniceplac.

ARAUJO, Ana Lis Souto de Oliveira; ANDRADE, Wilkerson; GUERRERO, Dalton Serey. Um Mapeamento Sistemático sobre a Avaliação do Pensamento Computacional no Brasil. In: **Anais dos workshops do Congresso Brasileiro de Informática na Educação 2016**, Anais... [s.l: s.n.]

AUGUSTO, Cleiciele Albuquerque. José Paulo de Souza; Eloise Helena Livramento Dellagnelo; Silvio Antonio Ferraz Cario . Pesquisa Qualitativa: rigor metodológico no tratamento da teoria dos custos de transação em artigos apresentados nos congressos da Sober (2007-2011). **Rev. Econ. Sociol. Rural**, Brasília , v. 51, n. 4, p. 745-764, Dec. 2013.

Bacich, Lilian. (2016) Ensino Híbrido: Proposta de formação de professores para uso integrado das tecnologias digitais nas ações de ensino e aprendizagem. In: V Congresso Brasileiro de Informática na Educação, 2016, Uberlândia, **Anais do XXII Workshop de Informática na Escola**. 2016. p. 679-687.

BALANSKAT, Anja; ENGELHARDT, Katja **Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe**, out. 2015. European Schoolnet. Disponível em: <[http://www.eun.org/c/document\\_library/get\\_file?uuid=3596b121-941c-4296-a760-0f4e4795d6fa&groupId=43887](http://www.eun.org/c/document_library/get_file?uuid=3596b121-941c-4296-a760-0f4e4795d6fa&groupId=43887)>. Acesso em: 1 dez. 2015.

BARCELOS, Ricardo José dos Santos. **O processo de construção do conhecimento de algoritmos com o uso de dispositivos móveis considerando estilos preferenciais de aprendizagem**. 2012. Tese (Doutorado em Informática e Educação), Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

BARCELOS, Thiago Schumacher; SILVEIRA, Ismar Frango **Pensamento Computacional e Educação Matemática: relações para o ensino de Computação na educação básica**. In: **Anais do XXXII CSBC**, Curitiba, 2012.

BARCELOS, Thiago; SOUZA, Alexandra; SILVA, Leandro; MUÑOZ, Roberto; VILARROEL, Rodolfo. Mensurando o desenvolvimento do Pensamento Computacional por meio de mapas auto-organizáveis: um estudo preliminar em uma Oficina de Jogos Digitais. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, [s.l.], p. 932-941, out. 2017. ISSN 2316-8889.

BARCELOS, Thiago Schumacher; MUÑOZ, Roberto; VILARROEL, Rodolfo; SILVEIRA, Ismar Frango. Relações entre o Pensamento computacional e a Matemática: uma Revisão Sistemática da Literatura. **Anais dos Workshops do**

**Congresso Brasileiro de Informática na Educação**, [s.l.], p. 1369-1378, out. 2015. ISSN 2316-8889.

BARCELOS, Thiago; BORTOLETTO, Rodrigo; ANDRIOLI, Mary. Formação online para o desenvolvimento do Pensamento computacional em professores de Matemática. In: **Anais dos workshops do Congresso Brasileiro de Informática na educação 2016**, Anais... [s.l: s.n.]

BARELL, John. **Problem-Based Learning**. An Inquiry Approach. Thousand Oaks: Corwin Press. 2007.

BARR, Valerie; STEPHENSON, Chris. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? **ACM Inroads**, v. 2, n. 1, p. 111-122, 2011.

BARRET, Terry. What is problem-based learning? In: O'NEILL, G.; McMULLIN, B.; MOORE, S. (Eds.). **Emerging issues in the practice of university learning and teaching**. Dublin: AISHE, 2005. p. 55-66.

BARRET, Terry; MOORE, Sarah. **New approaches to problem-based learning: revitalizing your practice in higher education**. New York: Routledge, 2011.

BERBEL, Neusi Aparecida Navas. A problematização e a aprendizagem baseada em problemas: diferentes termos ou diferentes caminhos?. **Interface** (Botucatu), Botucatu, v. 2, n. 2, p. 139-154, Feb. 1998.

BLIKSTEIN, Paulo. O Pensamento computacional e a Reinvenção do Computador na Educação, 2008. Disponível em: <<http://bit.ly/1IXIbNn>>. Acesso em: 28 abr. 2018.

BRACKMANN, Christian Puhlmann. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica**. 2017. Universidade Federal do Rio Grande do Sul, [s. l.], 2017.

BOROCHOVICIUS, Eli; TORTELLA, Jussara Cristina Barboza. Aprendizagem Baseada em Problemas: um método de ensino-aprendizagem e suas práticas educativas. **Ensaio: aval. pol. públ. Educ.**, Rio de Janeiro, v. 22, n. 83, p. 263-294, June, 2014 .

BRASIL. Base Nacional Comum Curricular (BNCC). **Educação é a Base**. Brasília, MEC/CONSED/UNDIME, 2017. Disponível em: <http://basenacionalcomum.mec.gov.br/a-base>. Acesso em: 10 maio 2019.

\_\_\_\_\_. Base Nacional Comum Curricular (BNCC). Ensino Médio. **Educação é a Base**. Brasília, MEC/CONSED/UNDIME, 2017.

\_\_\_\_\_. MINISTÉRIO DA EDUCAÇÃO. **Parâmetros Curriculares Nacionais: Ensino Médio**. Ciências da natureza, matemática e suas tecnologias. Brasília: Ministério da Educação, Secretaria de Educação Básica, 2006. 135 p. (Orientações curriculares para o ensino médio; volume 2).

\_\_\_\_\_. MINISTÉRIO DA EDUCAÇÃO. **Parâmetros Curriculares Nacionais: Ensino Médio**. Brasília: MEC/SEB, 1999.

BUSSMANN, Christian James de Castro; SAVIOLI, Angela Marta Pereira das Dores. Pensamento matemático-computacional. **Cadernos UniFOA**, Volta Redonda, n. 42, p. 91-100, abril, 2020.

CANO, Marcos Daniel. Programação de jogos digitais: Ferramenta Scratch unindo estudantes do ensino médio. **Nexus Revista de Extensão do IFAM**. Manaus, v.4, n.1, p. 53-59, jun. 2018.

CARVALHO, Joethe; NETTO, José Francisco; ALMEIDA, Thais. Revisão Sistemática de Literatura sobre Pensamento computacional por Meio de Objetos de Aprendizagem. In: **Brazilian Symposium on Computers in Education** (Simpósio Brasileiro de Informática na Educação - SBIE). 2017.

CASTRO, Adriane de. O Uso Da Programação Scratch para o Desenvolvimento de Habilidades em Crianças do Ensino Fundamental. 2017. Dissertação (Ensino de Ciência e Tecnologia). Programa de Pós-Graduação em Ensino de Ciência e Tecnologia. Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2017.

CIEB. Centro de Inovação para a Educação Brasileira. **Currículo de referência em tecnologia e computação da educação infantil ao ensino fundamental**. Disponível em: <http://curriculo.cieb.net.br/>. Acesso em: 07 set. 2019.

CITTOLIN, Simone Francescon. **A convivência na aprendizagem baseada em problemas no contexto da educação superior**. 2019. Tese (Doutorado em Educação), Universidade Tuiuti do Paraná. Curitiba, 2019. Disponível em <https://tede.utp.br/jspui/handle/tede/1666>. Acesso em: 06 nov. 2020.

CONCENTINO, Jéssica; COSTA, Juliana Aparecida Alves da; FERRUZZI, Elaine Cristina; WAIDEMAN, Adriele Carolini; CARGNIN, Claudete. Encaminhamentos da metodologia de análise de dados: análise textual discursiva. **Encontro Paranaense de Educação Matemática**. Unoeste Cascável, 21 a 23 de setembro, 2017.

CONFORTO, Debora; CAVEDINI, Patrícia; MIRANDA, Roxane; CAETANO, Saulo. Pensamento computacional na educação básica: interface tecnológica na construção de competências do século XXI. **Revista Brasileira de Ensino de Ciências e Matemática**. Passo Fundo, v. 1, n. 1, p. 99-112, jan./jun. 2018.

CRUZ, Sara.; BENTO, Marco.; LENCASTRE José Alberto. Construção de jogos em Scratch sobre conceitos da Matemática. **Atas do 5º Encontro sobre Jogos e Mobile Learning**. Universidade de Coimbra. Centro de Estudos Interdisciplinares do Século 20 (CEIS20). Coimbra 2020.

DANTAS, Sergio Carrazedo. **Design, implementação e estudo de uma rede sócio profissional online de professores de Matemática**. UNIVERSIDADE ESTADUAL PAULISTA (UNESP) Instituto de Geociências e Ciências Exatas. Rio Claro, 2016.

DEWEY, John. **Vida e educação**. Tradução Anísio Teixeira. 8. ed. São Paulo: Melhoramentos, 1973.

DR. SCRATCH. Disponível em: <http://www.drscratch.org/learn/Logic/>. Acesso em: 20 fev. 2021.

FARIA, Airan Priscila de.; MOTTA, Marcelo Souza. Aprendendo Matemática ao se programar no Scratch. **Revista Educação & Tecnologia**, n. 16. UTFPR/Curitiba (2017). ISSN Eletrônico 2179-6122.

FARIAS, Fernando Lucas de Oliveira; BRITO, Everton da Silva; MELO, Elvis; BANDEIRA, Lais Michele de Souza Araújo; COSTA, Felipe Jhonanta Ferreira da; DINIZ, Igo Joventino Dantas; TRINDADE, Sidney Soares; NUNES, Isabel. (2019). Práticas Pedagógicas Colaborativas utilizando Ferramentas Digitais: Um Relato de experiência na formação de educadores. **Anais do XXIV Workshop de Informática**. 2018.

FAZENDA, Ivani (org). **O que é interdisciplinaridade?** São Paulo: Cortez, 2006.

FERRAZ, Ana Paula do Carmo Marcheti; BELHOT, Renato Vairo. Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. **Gestão & Produção**, São Carlos, v. 17, n. 2, p. 421-431, 2010.

FERREIRA, Ruy. **Interatividade educativa: uma visão pedagógica**. 2008. 199 p. Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Educação, Campinas, SP.

FREIRE, Paulo. **Pedagogia do Oprimido**. São Paulo: Paz e Terra, 1996.

HU, Cheng Computational Thinking – What It Might Mean and What We Might Do About It. In: 16TH Annual joint Conference on Innovation and Technology in Computer Science Education 2011, **Anais...** [s.l: s.n.]

GAZALE, Raquel Alves. **Aprendizagem baseada em problemas: uma proposta para as séries finais do ensino fundamental**. 2018. Dissertação (Mestrado em Projetos Educacionais de Ciências). Escola de Engenharia de Lorena, Universidade de São Paulo, Lorena, 2018.

GOOGLE FOR EDUCATION. Future of the Classroom Emerging Trends in K-12 Education Global Edition. **Relatório Global**, 2018, p. 57.

GREGORUTTI, Juliana de Lima. **Construção dos critérios de divisibilidade com alunos de 5ª série do ensino fundamental por meio de situações de aprendizagem**. 2009. 147 f. Dissertação (Mestrado em Educação) - Pontifícia Universidade Católica de São Paulo, São Paulo, 2009.

HORBACH, Ivan Carlos. **Semelhança de triângulos: Um estudo propositivo através do Scratch**. Dissertação. Universidade Federal da Fronteira Sul Campus Chapecó

Mestrado Profissional em Matemática em Rede Nacional Profmat. Chapecó, p. 1-71. 2020.

ISTE. Explore os Padrões do Aluno; **Pensador computacional**, 2016, s/p.

KITCHENHAM, Budgen. (2004). **Procedures for performing systematic reviews**. Keele, UK, Keele University, 33 (TR/SE-0401), 28.

KLEINUBING, Jorge José. **Utilizando o scratch para o ensino da matemática**. Trabalho de conclusão de curso. Curso superior em licenciatura em informática. Universidade Tecnológica Federal do Paraná. Francisco Beltrao, 2016.

KOSCIANSKI, André; Glizt, Fabiana Rodrigues de Oliveira. O pensamento computacional nos anos iniciais do ensino fundamental. CINTED-UFRGS, **Novas Tecnologias na Educação**, v. 15, n. 2, dezembro, 2017.

LEE, Irene; MARTIN, Fred; DENNER, Jill; COULTER, Bob; ALLAN, Walter; ERICKSON, Jeri; SMITH, Joyce Malyn; WERNER, Linda. **Computational thinking for youth in practice**. ACM Inroads. 2011.

LEITE, Laurinda; AFONSO, Ana Sofia. A aprendizagem baseada na resolução de problemas: características, organização e supervisão. **Boletín das Ciencias**, Santiago de Compostela, v. 48, p. 253-260, 2001.

LEITE, Laurinda.; ESTEVES, Esmeralda. Ensino orientado para a aprendizagem baseada na resolução de problemas na licenciatura em ensino da física e química. In: Congresso Galaico-português de Psicopedagogia, 8, 2005, Braga. **Anais...** Braga: Universidade do Minho, 2005. p. 1751-1768.

LIMA, Árlon Chaves; SOUSA, Decíola Fernandes de; Desenvolvimento do Raciocínio Lógico e Algoritmo Através do Programa Institucional de Bolsas de Iniciação à Docência no Ensino Fundamental. **Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação**, p. 1379-1388. 2015.

MALTEMPI, Marcos Vinicius. Novas Tecnologias e Construção de Conhecimento: Reflexões e Perspectivas. In: **V Congresso Ibero-americano de Educação Matemática** (CIBEM). Porto, Portugal, 2005.

MAJED, Marji. **Title of English-language original: Learn to Program with Scratch**. published by No Scratch Press. Portuguese - language, by Novatec Editora Ltda. All rights reserved, 2014.

MAJED, Marji **Aprendendo a programar com Scratch**. São Paulo: Novatec Editora Ltda, 2014.

MANNILA, Linda. DAGIENE, Valentina; DEMO, Barbara; GRGURINA, Natasa; MIROLO, Claudio; ROLANDSSON, Lennart; SETTLE, Amber. (2014). Computational Thinking in K-9 Education. In: **Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference - ITICSE-WGR '14**.

MARQUES, Monica.; CAVALHEIRO, Simone; FOSS, Luciana; BORDINI, Adriana; ÁVILA, Cristiano; Uma Proposta para o Desenvolvimento do Pensamento Computacional Integrado ao Ensino de Matemática. Brazilian Symposium on Computers in Education (**Simpósio Brasileiro de Informática na Educação - SBIE**), [S.l.], p. 314, out. 2017.

MARQUES, Samanta Ghisleni,. **Implicação dos pilares do pensamento computacional na resolução de problemas na escola**. 2019. Tese (Doutorado em Educação), Unisc, Universidade de Santa Catarina.

MARQUES, Monica; CAVALHEIRO, Simone; FOSS, Luciana; BORDINI, Adriana Uma Proposta para o Desenvolvimento do Pensamento Computacional Integrado ao Ensino de Matemática. **VI Congresso Brasileiro de Informática na Educação (CBIE 2017)** 314. 10.5753/cbie.sbie.2017.314.

MIRANDA, Pauline Vielmo. A linguagem LOGO como alternativa lúdica de ensino. **II Forum Internacional de Educação**. Universidade de Santa Cruz do Sul, Santa Cruz do Sul, 2016.

MORAES, Roque. Uma tempestade de luz: a compreensão possibilitada pela análise textual discursiva. **Ciência & Educação**. Bauru, SP, v. 9, n. 2, p. 191-210, 2003.

MORAES, Roque; GALIAZZI, Maria do Carmo. **Análise Textual Discursiva**. UNIJUÍ. 3. ed. rev. e ampl. Ijuí : Editora Unijuí, 2016. – 264 P. (Coleção Educação em Ciências).

\_\_\_\_\_. **Análise Textual Discursiva**. Ijuí: Unijuí, 2007.

\_\_\_\_\_. Análise Textual Discursiva: Processo reconstrutivo de múltiplas faces. **Ciência & Educação**, v. 12, n. 1, p. 117-128, 2006.

MORENO-LEÓN, Jesús; ROBLES, Gregorio. **Analyze your Scratch projects with Dr. Scratch and assess your Computational Thinking skills**. Amsterdam, Netherlands, 2015.

MOVIMENTO PELA BASE. Dimensão e Desenvolvimento das Competências Gerais da BNCC.

NETO, João Coelho. ALTOÉ, Anair. Construcionismo e a Formação de Professores: Um Estudo com Alunos do Curso de Pedagogia aa Uenp Cp. **X Congresso Nacional de Educação**, Educere, Curitiba 2011. p. 2315 - 2325. Disponível em: [https://educere.bruc.com.br/CD2011/pdf/5807\\_2630.pdf](https://educere.bruc.com.br/CD2011/pdf/5807_2630.pdf) Acesso em: 20 nov. 2020.

NEVES, Fabiana Moraes das; JUNG, Hildegard Susana; ALTMANN, Idio Fridolino. O Pensamento matemático na perspectiva significativa:: uma revisão de estudos científicos. **Revista Internacional de Pesquisa em Didática das Ciências e Matemática**, [s. l.], v. 1, p. e020005, 2020.

OMINE, Eduardo Hiroshi. **Design gráfico computacional: computação aplicada no projeto e na produção de imagens dinâmicas e interativas** (Doctoral dissertation, Universidade de São Paulo, 2014).

PINHO, Gustavo; WEISSHAHN, Yuri; CAVALHEIRO, Simone; REISER, Renata; BRUM, Clause Fatima de.; FOSS, Luciana; AGUIAR, Marilton; DU BOIS, André: Pensamento Computacional no Ensino Fundamental: Relato de Atividade de Introdução a Algoritmos. **Anais do Workshop de Informática na Escola**, [s.l.], p. 261, nov. 2016. ISSN 2316-6541.

RAABE, André Luis Alice; BRACKMANN, Christian Puhlmann; CAMPOS, Flávio R. **Currículo de referência em tecnologia e computação: da educação infantil ao ensino fundamental**. São Paulo: CIEB, 2018. E-book em pdf.

RAMALHO, Rui; VENTURA, Ana. (2018). O Scratch promotor do pensamento computacional na geometria do ensino básico. In: **III Encontro Internacional de Formação na Docência (INCTE)**: livro de atas (pp. 232-242). Bragança: Instituto Politécnico. ISBN 978-972-745-241-5.

RAMOS, Henrique de Almeida. **Pensamento computacional na Educação Básica: uma proposta de aplicação pedagógica para alunos do quinto ano do Ensino Fundamental do Distrito Federal**. Monografia (Graduação). Brasília: UnB, 2014, p.119.

RESNICK, Mitchel; MALONEY John; HERNÁNDEZ, Andrés Monroy; RUSK, Natalie; BRENNAN, Karen; EASTMOND Evelyn; MILLNER, Amon; ROSENBAUM, Eric; SILVER, Jay (2009). **Scratch: programming for all**. Communications of the ACM 52.11: 60-67.

RIBEIRO, Leila.; FOSS, Luciana.; CAVALHEIRO, Simone André da Costa. Pensamento Computacional: Fundamentos e Integração na Educação Básica. **Jornada de Atualização em Informática na Educação**, [s.l.], p.1-25, nov. 2019.

\_\_\_\_\_. Entendendo o Pensamento Computacional. In: **arXiv:1707.00338**, 2017.

RIBEIRO, Rafael; BARCELOS, Thiago; SOUZA, Alexandra; SILVA, Leandro Augusto. Mensurando o desenvolvimento do Pensamento Computacional por meio de Mapas Auto-Organizáveis: Comparação de métricas de complexidade de *Software* com Dr. Scratch e CT-Test. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, [S.l.], p. 609, out. 2018. ISSN 2316-8889.

RICO, Rosi. Competência 5: Cultura Digital, Documento reforça a necessidade de usar bem a tecnologia. In: **Revista Nova Escola**. Disponível em: <https://novaescola.org.br/bncc/conteudo/9/competencia-5-cultura-digital#:~:text=Essa%20compet%C3%Aancia%20reconhece%20o%20papel,vida%20das%20pessoas%20e%20da>. Acesso em: 20 jan. 2020

RODRIGUES, William Costa. **Metodologia Científica**. FAETEC/IST, Paracambi, 2007.

SALLES, Eliciane Bruning de; ROOS, Liane Terezinha Wending; LUCION, Paula; ZÜGE, Vanessa. Arte e Matemática: O ensino de simetria é magia. **IV Jornada Nacional de Educação Matemática**. Universidade de Passo Fundo, RS, p. 12, 2012.

SANTANA, Sivaldo Joaquim de; OLIVEIRA, Wilk. Desenvolvendo o Pensamento computacional no Ensino Fundamental com o uso do Scratch. **Anais do Workshop de Informática na Escola**, [s.l.], p. 158-167, nov. 2019. ISSN 2316-6541.

SANTOS, Gilson Pedroso dos; BEZERRA, Ronilson dos Santos. Desenvolvendo o pensamento computacional utilizando Scratch e lógica Matemática. **Jornada de Atualização em Informática na Educação**, [s.l.], p. 66-99, out. 2017. ISSN 23167734.

SECRETARIA DE EDUCAÇÃO DO PARANÁ. **Educação dia a dia**. Disponível em: [http://www.educadores.diaadia.pr.gov.br/arquivos/File/inovacao\\_tecnologia/piramide\\_escola\\_ficha\\_o\\_que\\_e\\_na\\_escola.pdf](http://www.educadores.diaadia.pr.gov.br/arquivos/File/inovacao_tecnologia/piramide_escola_ficha_o_que_e_na_escola.pdf). Acesso em: 01 mar. 2021.

SCHFFER, Carmen Cristina Rodrigues. **Tecnologia computacional e desenvolvimento cognitivo**: estudo de caso na formação de psicólogos. São Paulo: Annablume; Belo Horizonte: FUMEC, 2004.

SCRATCH Brasil. **Você Conhece o Scratch?**. s/d. Disponível em: <http://www.scratchbrasil.net.br/index.php/sobre-o-scratch.html>. Acesso em: 28 out. 2019.

SCRATCH2. Disponível em: [https://en.scratch-wiki.info/wiki/Scratch\\_Project\\_License](https://en.scratch-wiki.info/wiki/Scratch_Project_License). Acesso em: 02 fev. 2021.

SILVA, Leonardo Cintra Lopes da.; **A relação do Pensamento Computacional com o ensino de Matemática na Educação Básica**, 2019, 131 p. Dissertação (PROFMAT – Mestrado Profissional em Matemática em Rede Nacional) Faculdade de Ciências e Tecnologia da Universidade Estadual Paulista Júlio de Mesquita Filho, Presidente Prudente.

SILVA, Paulo Vinicius Baptista da; MORO, Maria Lucia Faria. A interação de adolescentes marginalizados com a linguagem Logo. **Psicol. Reflexo. Crit.**, Porto Alegre, v. 11, n. 1, p. 35-58, 1998.

SILVA, Aleff Hermínio da. **Investigação Matemática**: contribuições dessa metodologia para o ensino-aprendizagem da Divisibilidade dos números naturais. Monografia Licenciatura em Matemática. UFP (Universidade Federal da Paraíba, Vinho Tinto, 2020, p. 75.

SILVA, Edeilson Ferreira da; SILVEIRA, Ismar. Intersecção entre pensamento computacional e a matemática: perspectivas de ensino na educação básica brasileira. **Anais do CIET:EnPED:2020** - (Congresso Internacional de Educação e Tecnologias | Encontro de Pesquisadores em Educação a Distância), São Carlos, ago. 2020.

SOUSA, Juciane de; DOS SANTOS, Andréia Nunes dos. A história da Matemática como instrumento de ensino e aprendizagem na educação básica. **Boletim Cearense de Educação e História da Matemática**, [s. l.], v. 7, n. 20, p. 451–458, 2020.

SOUZA, Samir Cristino de; DOURADO, Luis. Aprendizagem baseada em problemas (ABP): um método de aprendizagem inovador para o ensino educativo. **HOLOS**, v. 5, p. 182-200, 2015.

TAHAN, Malba. **O homem que calculava**. Rio de Janeiro: Record, 1994.

UNICAMP. **Tutorial Scratch** – Conceitos básicos - Unicamp (2011). Disponível em: eurydice.nied.unicamp.br › material-didatico ›. Acesso em: 28 out. 2020.

UNIVERSIDADE FEDERAL DE SÃO PAULO – UNIFESP. **Aprendizado Baseado em Problemas. São Paulo**, 2001. Disponível em: <http://www2.unifesp.br/centros/cedess/pbl/>. Acesso em: 11 ago. 2017.

URRUTH, Bolivar. Linguagem de Programação LOGO como Inclusão Digital no ensino de geometria plana com alunos surdos do PROEJA. **Revista Viver IFRS**, v. 3, n. 3, p. 48-50, 2015.

VALENTE, José Armando. Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. **Revista e-Curriculum**. e-Curriculum, [s. l.], v. 14, n. 03, p. 864–897, 2016.

\_\_\_\_\_. (Org.) **Computadores e conhecimento: repensando a educação**. 2. ed. Campinas, SP: UNICAMP/NIED, 1998.

VICK-VIEIRA, Marli Fatima; SANTANA, André Luis Maciel; RAABE, André Luis Alice. Do Logo ao Pensamento computacional: o que se pode aprender com os resultados do uso da linguagem Logo nas escolas brasileiras. **Tecnologias, sociedade e conhecimento**, Campinas, v. 4, n. 1, Dez. 2017.

VOELCKER, Marta Dieterich; Fagundes, Léa Cruz; Seidel, Susana. Fluência digital e ambientes de autoria multimídia. CINTED-UFRGS, **Novas tecnologias na educação**. Porto Alegre, v. 6, n. 1, jul-2008.

WANGENHEIM, Christiane Gresse Von; NUNES, Vinícius Rodrigues; SANTOS, Giovane Daniel dos. Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso. **Revista Brasileira de Informática na Educação**, v. 22, n. 3, 2014.

WENGER, Etienne. **Communities of practice**. Learning, Meaning and Identity. Cambridge: University Press, 2008.

WIELEWSKI, Gladys Denise. **Aspectos do pensamento matemático na resolução de problemas**: uma apresentação contextualizada da obra de Krutetskii. 2005. Tese (Doutorado em Educação Matemática). PUC, São Paulo.

WING, Jeannette PENSAMENTO COMPUTACIONAL - Um conjunto de atitudes e habilidades, que todos, não só cientistas da computação ficaram ansiosos para aprender e usar. **Revista Brasileira de Ensino de Ciências e Tecnologia**, Ponta Grossa, v. 9, n. 2, p. 1-10, mai./ago. 2016.

WILSON, Cameron, SUDOL, Leigh Ann; STEPHENSON, Chris; STEHLIK, Mark. (2010). **Running on empty**: the failure to teach k12 computer science in the digital age. *Commun. ACM.* 27, 30.

WING, Jeannette Computational Thinking. , 2007. Carnegie Mellon University.

WING, Jeannete. M. Computational Thinking: What and Why? , 17. out. 2010.

WING, Jeannette. Pensamento computacional – Um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar. *Revista Brasileira de Ensino de Ciência e Tecnologia*, v. 9, n. 2, 2016.

## 8. APÊNDICES

### 8.1 Apêndice A – Planos de Aula

Link de acesso:

<https://docs.google.com/document/d/1iA723SXXOmMNdasTRswu5hRD8bYvFy0/edit?usp=sharing&oid=113840944612938508421&rtpof=true&sd=true>

### 8.2 Apêndice B -Excertos

Link de acesso:

<https://docs.google.com/document/d/1RT-4qpcd3vk3lfbIR7BXQu7WZEWgiBP8/edit?usp=sharing&oid=113840944612938508421&rtpof=true&sd=true>

### 8.3 Apêndice C - Questionário

Link de acesso:

<https://docs.google.com/document/d/1oC9IWv4BeROSmwnSJF3WBcnJFZmZo9d7/edit?usp=sharing&oid=113840944612938508421&rtpof=true&sd=true>

### 8.4 Apêndice D - Aprendizagem baseada em problemas (ABP)

A Aprendizagem Baseada em Problemas tem como referência as teorias de Jerome Seymour Bruner e John Dewey, cuja teoria nos remete a uma educação para a vida, já que a aprendizagem ocorre o tempo todo e, mesmo entendendo a importância da acumulação de conhecimentos, o foco é o desenvolvimento das capacidades (DEWEY, 1973).

Gazale (2018) salienta que

A ABP oferece um caminho a ser percorrido até chegar ao resultado final que é a solução de uma situação-problema. O ponto de partida é sempre a análise do problema, passando pelos estudos auto gerenciados pelo aluno, pela troca de ideias entre os membros do grupo até chegar à solução do problema (GAZALE, 2018, p. 18).

Berbel (1998, p. 142) relata “que a primeira etapa da problematização é a **Observação da Realidade** social, concreta, pelos alunos, a partir de um tema ou unidade de estudo” (Grifos do autor).

Para o curso de formação realizado, o problema proposto foi o planejamento e execução de uma aula visando o desenvolvimento de habilidades relativas ao Componente Curricular Matemática e ao pensamento computacional, com o uso do Scratch.

Figura 83 - Figura 83 -Estrutura de organização das atividades



Fonte: Adaptado de Souza e Dourado (2015).

Na Figura 83, ilustramos a estrutura para a organização das atividades na ABP, partindo do contexto real, traçando um problema, quem o recebe busca a solução e, por fim, apresenta o resultado.

### **I. Aprendizagem Baseada em Problemas: características**

Analisamos, com o apoio de Souza e Dourado (2015), as características da Aprendizagem Baseada em Problemas (ABP). Segundo os autores, ela possui três características, a saber: 1. O aluno como centro da aprendizagem; 2. O trabalho em grupo e 3. O professor como tutor.

Relataremos cada uma dessas características, para embasar futuramente as etapas da formação que fizemos.

### **II. Aluno como centro da aprendizagem**

De acordo com Souza e Dourado (2015), a ABP contrapõe o modelo de aulas expositivas, no qual o professor assume um papel central no processo de aprendizagem e o aluno se posiciona como mero expectador e/ou receptor, o que Paulo Freire (1996) chama de educação bancária.

Na Aprendizagem Baseada em Problemas, como o próprio nome já diz, o foco é a problematização, dessa forma, as aulas tendem a ser interdisciplinares, fortalecendo a participação dos estudantes, fazendo com que eles tenham participação ativa no processo ensino e aprendizagem.

Assim, para os autores,

A opção por uma metodologia de aprendizagem centrada no aluno acentua a importância da ABP, vez que, por sua aplicabilidade, estaríamos possibilitando o desenvolvimento de atividades educativas que envolvem a participação individual e grupal em discussões críticas e reflexivas (SOUZA; DOURADO, 2015, p. 187).

Aqui temos duas vertentes, aluno como centro da aprendizagem e também como protagonista de sua aprendizagem. O currículo centrado na ABP vem com o intuito de mudança passando o foco do professor para o aluno, focando nos trabalhos individuais e coletivos (SOUZA; DOURADO, 2015).

### **III. O Trabalho em Grupo**

Essa forma de trabalho não é uma exclusividade da ABP, até mesmo os métodos de ensino mais tradicionais utilizam essa estratégia, principalmente nos cursos superiores, nos quais os alunos se organizam das mais variadas formas de modo a atenderem os objetivos das disciplinas.

Na ABP, no entanto, os alunos valorizam a convivência com os demais membros, valorizando o trabalho colaborativo, e, ao invés de serem forçados a entender um determinado conteúdo, conceito ou uma parte curricular do programa (muitas vezes para ganhar nota), ele se disponibiliza a realizar o trabalho, buscando assim o protagonismo individual e do todo, corroborando as ideias de Citollini (2019), quando ela afirma que

Práticas de aprendizagem cooperativa que buscam alcançar objetivos comuns incrementam a interdependência entre os membros do grupo, promovendo a confiança mútua desenvolvendo a responsabilidade, e suscitando a ajuda recíproca entre os indivíduos. (CITOLLINI, 2019, p. 13-14).

A autora vem fortalecendo a questão diferenciadora do trabalho em grupo e cooperativa trazido pela ABP para alcançar um objetivo comum, resolvendo um trabalho de relevância social.

#### IV. O Professor como tutor

Para Souza e Dourado (2015), a relação professor e aluno tem de seguir esses caminhos. Do ponto de vista da aprendizagem,

A ABP é um método que contempla como um dos pontos fundamentais de sua aplicação a relação entre o professor, o aluno e o conteúdo a ser estudado e aprendido. Nessa relação, o professor posiciona-se como um mediador, um guia que estimula os alunos a descobrir, a interpretar e a aprender (SOUZA E DOURADO, 2015, p.190).

O Professor mediador é aquele que entende que o aprendizado não se dá somente no ambiente escolar, aprende-se durante todo o tempo e não existe um lugar específico para o aprendizado, aprende-se nas relações interpessoais, em atividades do dia a dia (WENGER, 2008). Assim, o professor, no que compete ao seu trabalho de tutor, tem de buscar estratégias para que seu aluno seja curioso, criador, tenha uma boa interpretação (principalmente da realidade), que busque o conhecimento, utilizando-o em todos os momentos da sua vida. Em outras palavras, o professor deve estimular seus discentes a tomar decisões, habilidade necessária para o cidadão do século 21. A ABP objetiva preparar os estudantes para se tornarem eficientes e eficazes na solução de problemas e aprendizes autônomos.

Essa característica vem ao encontro da primeira que é o aluno como centro da aprendizagem, já que, para o aluno ser o centro da aprendizagem, o professor necessita buscar o seu papel de tutor, ficando responsável por acompanhar o processo junto do aluno e agindo no elo entre o aluno e o conhecimento.

Esse professor necessita ter o domínio dos conteúdos da sua disciplina e todos os conhecimentos adquiridos ao longo dos anos que a perpassam.

Conhecendo a profundidade da disciplina que atua e a realidade atual, o professor tutor necessita buscar meios para a inserção dos seus alunos no mundo das ciências e da tecnologia, valorizando os valores morais e éticos, como traz a BNCC (2017) quando se retrata da quinta competência que é a Cultura Digital

Utilizar tecnologias digitais de comunicação e informação de forma crítica, significativa, reflexiva e ética nas diversas práticas do cotidiano (incluindo as escolares) ao se comunicar, acessar e disseminar informações, produzir conhecimentos e resolver problemas. (BNCC, 2017, p. 18).

O professor como tutor tem de estar atento a essas informações, já que, nesse mundo, com tanto conteúdo que é disponibilizado pela internet via redes sociais, ele

tem de preparar seus alunos para a disseminação dos conteúdos verdadeiros, para não propagar notícias falsas. Este é, sem dúvida, um dos grandes desafios do Professor tutor: preparar seus alunos para este mundo digital “selvagem”.

### **Etapas da aprendizagem baseada em problemas**

De acordo com Leite e Afonso (2001), Leite e Esteves (2005), Barrett (2005) e Barrett e Moore (2011), a estrutura básica da ABP ocorre em quatro etapas, sendo elas: Contexto real da vida dos estudantes; recebimento do contexto problemático; investigação por meio de diversos recursos disponibilizados pelo tutor e apresentação dos resultados e autoavaliação

#### **I – Contexto real da vida dos estudantes (contexto problemático)**

Dentro da estrutura colocada pela ABP, a primeira fase é a escolha do contexto da vida real dos estudantes.

O objetivo dessa fase é identificar um problema que esteja presente na vida dos estudantes, que faça sentido para eles, para justamente buscarem soluções para a atividade investigativa. Do contrário, não haverá identificação deles com o problema.

Destarte, buscar o contexto real, que seja parte da vida dos envolvidos, é ponto fulcral nesse momento.

Cabe ao professor fazer essa dinâmica, para que os envolvidos se sintam motivados para desenvolver a atividade investigativa, essa etapa é de extrema relevância em todo percurso futuro que será percorrido (BARELL, 2007; BARRETT & MOORE, 2011).

Nesse trabalho, o professor prepara a sistematização, observando quais os materiais são necessários para o trabalho de investigação; a escolha dos materiais será de responsabilidade do professor.

Aqui o contexto problemático deve estar bem definido.

#### **II – Recebimento do contexto problemático**

Nessa etapa, os estudantes recebem do professor tutor o contexto problemático. Nesse momento, eles são divididos em grupos, iniciando o processo de argumentação.

Esse processo dá início à identificação das informações faltantes para elaborar as questões problema, ou a questão problema.

Também é responsável pelo desenvolvimento da investigação, e o que será necessário aprofundar.

Esse aprofundamento é com base nos conhecimentos prévios que os estudantes possuem sobre a investigação, com a identificação do que necessitam fazer para que o problema seja resolvido (LEITE & AFONSO, 2001).

### **III – Investigação por meio de diversos recursos disponibilizados pelo tutor**

Nesse momento, acontece todo o projeto de desenvolvimento a partir dos recursos que foram disponibilizados pelo professor tutor, definidos em momento anterior.

Apropriando-se dessas informações, os alunos iniciam a pesquisa, que pode ser em grupo ou individual, na ABP existe essa simbiose constante.

Após esse percurso, os alunos trazem as informações para o debate em grupo, levando em consideração a resolução da questão problema, definida na fase anterior.

Os estudantes, nessa fase, apropriam-se das informações por meio de leitura e análise crítica, pesquisam na internet, discutem em grupo o material coletado e levantam as hipóteses de solução (CITTOLIN,2019; LEITE & AFONSO, 2001).

### **IV – Apresentação dos resultados e auto avaliação**

Sendo essa a última etapa, os participantes elaboram um trabalho com a síntese das discussões e reflexões, relatam e sistematizam as soluções encontradas para o problema, preparam a apresentação para a turma e para o tutor, e promovem a autoavaliação e a avaliação da contribuição de cada membro no processo de aprendizagem realizada e que contenha as soluções para os problemas investigados de forma sistemática, bem como a elaboração e apresentação (CITTOLIN, 2019).

Na apresentação final o professor avaliará os processos, procedimentais, conceituais e atitudinais (LEITE & AFONSO, 2001).

### **V – Contextualização da metodologia: Na pesquisa**

As três **características** presentes da ABP foram utilizadas em nossa pesquisa:

a) Utilizamos o contexto da vida real dos participantes, trabalhamos com um *software* que eles utilizariam em sala de aula, bem como o PC para buscar a potencialização de suas aulas.

b) O trabalho em grupo também foi utilizado, a busca era a ajuda mútua em cada encontro, com isso, os participantes da pesquisa cresceram juntos.

c) O professor como tutor; eram marcados horários para tirar dúvidas dos cursistas, de acordo com as possibilidades e realidades de cada um.

As seguintes **etapas** da ABP foram trabalhadas dentro da pesquisa:

A) No contexto problemático, foi informado aos professores que eles teriam de preparar uma aula, com os requisitos mínimos e poderiam utilizar como exemplo o plano de aula deixado na plataforma Google Sala de Aula.

A aula seria uma apresentação de um artefato produzido com o Scratch, observando os pilares do pensamento computacional e o ensino da Matemática, apresentada no 9º encontro.

B) A investigação realizada por meio de recursos utilizados; deixamos vídeos, artigos, livros na plataforma Google Sala de Aula, para ajudá-los nas construções.

C) Os resultados foram mostrados por meio de apresentações, em que cada dupla realizou sua produção conforme plano de aula apresentado.

D) A avaliação do curso foi uma conversa final com todos os cursistas, e nela eles apresentaram as suas percepções de todo o processo, pois o objetivo da pesquisa é que eles avaliassem o que proposto tal entendimento ficou descrito no relatório da atividade do módulo 10.