



Mateus Dauã de Morais

**Pensamento Computacional e matemática: como emergem em projetos com o *Scratch* no Ensino Remoto?**

Dissertação de Mestrado apresentada ao Instituto de Matemática e Estatística da Universidade Federal do Rio Grande do Sul como requisito parcial para a obtenção do título de Mestre em Ensino de Matemática.

Orientador: Prof. Dr. Rodrigo Dalla Vecchia

Porto Alegre

2022

## **Dados Internacionais de Catalogação na Publicação (CIP)**

Mateus Dauã de Morais

**PENSAMENTO COMPUTACIONAL E MATEMÁTICA: COMO EMERGEM EM  
PROJETOS COM O *SCRATCH* NO ENSINO REMOTO?**

Dissertação de Mestrado apresentada ao Instituto de Matemática e Estatística da Universidade Federal do Rio Grande do Sul como requisito parcial para a obtenção do título de Mestre em Ensino de Matemática.

Aprovado em: 18 de abril de 2022

**BANCA EXAMINADORA**

---

Prof. Dr. Clodis Boscaroli – UNIOESTE

---

Profa. Dra. Débora da Silva Soares – UFRGS

---

Prof. Dr. Maurício Rosa – UFRGS

---

Prof. Dr. Rodrigo Dalla Vecchia – UFRGS (orientador)

Dedico este trabalho aos meus pais: Paulo de Moraes e Rose Maria Funke. Aprendi muitas coisas durante estes meus anos de estudos, entretanto, quando coloco um olhar mais atento à minha trajetória, fica claro que as coisas mais essenciais que aprendi não foram com meus professores ou com meus estudantes, mas com meus pais que, mesmo com grandes dificuldades, fizeram de tudo para oportunizar que eu conseguisse alcançar os meus sonhos.

Obrigado por me mostrarem meios para alçar voo.

Amo vocês.

Talvez, não haja melhor demonstração das tolices e vaidades humanas que essa imagem distante do nosso pequeno mundo. Ela enfatiza nossa responsabilidade de tratarmos melhor uns aos outros, e de preservar e estimar o único lar que nós conhecemos.

O pálido ponto azul.

Carl Sagan

## AGRADECIMENTOS

Esta pesquisa só foi possível graças ao apoio que tive de diversas pessoas em toda a trajetória que já percorri até o momento que escrevo esses agradecimentos.

Agradeço ao meu orientador, Rodrigo Dalla Vecchia, por ter me instigado a ir além do que eu achava que era capaz de ir e ter me ensinado muito sobre o que é ser um pesquisador em Educação Matemática.

Aos membros da banca examinadora, Clodis Boscarioli, Débora da Silva Soares e Maurício Rosa, pelas conversas que geraram sugestões, críticas e elogios ao meu trabalho, tanto na banca de qualificação quanto na defesa da dissertação, e que fizeram com que eu conseguisse seguir por um caminho excepcional para a produção e organização desta pesquisa

Aos meus pais, Paulo e Rose, que sempre estiveram ao meu lado me incentivando aos estudos e, mesmo com algumas dificuldades, foram o principal suporte para que eu pudesse chegar nesse momento.

À Larissa, por estar comigo desde antes do início da minha caminhada na Pós-Graduação, por ser compreensiva e ter me ajudado em diversos momentos que precisei de forças para continuar.

Aos meus amigos e familiares que também me apoiaram nas escolhas que fiz referente aos meus estudos, que foram receptivos nos momentos que precisei do apoio deles e compreensíveis quando não consegui participar de alguns eventos.

Aos professores que foram uma inspiração para mim durante a minha trajetória até aqui, da Educação Infantil à Pós-Graduação. Em especial ao professor Fabiano de História do Colégio Dr. Wolfram Metzler em Novo Hamburgo que, no ano de 2010, foi a minha principal referência para que eu decidisse me tornar professor.

À diretora da E.E.E.F. Ayrton Senna do Brasil, Sheila Kieling, e à supervisora Caroline Cunha, que confiaram no meu trabalho durante a produção dos dados desta pesquisa mesmo com as dificuldades que apareceram com a pandemia.

À diretora Tereza (Teka) da Escola Conquistadora, que sempre foi compreensiva referentes aos horários que precisei fazer para conseguir estudar na UFRGS.

Aos colegas orientados pelo professor Rodrigo Dalla Vecchia, pelas diversas contribuições recebidas durante estes anos.

## Resumo

Este trabalho tem como objetivo apresentar uma pesquisa de caráter qualitativo que buscou responder à seguinte questão: **como o Pensamento Computacional e a matemática emergem de projetos desenvolvidos com o ambiente de programação Scratch em atividades remotas?** Para isso, procuramos sustentação teórica em referências relacionadas ao uso de Tecnologias Digitais na Educação Matemática, nas ideias construcionistas de Papert (1986, 1988, 2008), em conceitos relacionados ao Pensamento Computacional (PC), principalmente pautados por Wing (2006, 2008, 2011), e na proposta de Aprendizagem Criativa de Resnick (2020). Para este estudo, além do entrelaçamento teórico realizado, produzimos dados a partir de encontros que, devido às restrições para conter a pandemia do Coronavírus SARS-CoV-2 (Covid-19), aconteceram de forma remota via *Google Meet* com estudantes dos anos finais do ensino fundamental de uma escola pública estadual localizada em Novo Hamburgo/RS. O principal recurso tecnológico utilizado pelos participantes e analisado nesta pesquisa foi o *Scratch*. Também utilizamos o *Machine Learning For Kids*, site que permite o treinamento de dados por algoritmos de Aprendizado de Máquina e sua utilização junto à linguagem do *Scratch*. Observamos que o PC e a matemática atuaram em movimentos complementares na determinação e na condução da resolução dos problemas, pois enquanto o Pensamento Computacional se mostra como uma possibilidade de potencializar a resolução, a matemática, algumas vezes, emerge como parte de um problema a ser resolvido. Também destacamos que a Decomposição de problemas, um dos pilares do PC, não é algo fixo, direto ou linear, podendo emergir de diferentes formas na busca pela solução do problema. Assim, notamos que as habilidades relacionadas ao PC são processos que acontecem ao longo de muitas transformações e que a matemática, em alguns casos, contribui no processo para determinar o encaminhamento da solução do problema.

**Palavras-chave:** Educação Matemática; Pensamento Computacional; Ensino Remoto; Scratch.

## Abstract

This work aims to present qualitative study that sought to answer the following question: how does Computational Thinking and mathematics emerge from projects developed with the Scratch programming environment in remote activities? For this, we seek theoretical support in references related to the use of Digital Technologies in Mathematics Education, in the constructionist ideas of Papert (1986, 1988, 2008), in concepts related to Computational Thinking (CP), mainly guided by Wing (2006, 2008, 2011), and in the proposal of Creative Learning of Resnick (2020). For this research, in addition to the theoretical intertwining carried out, we produced data from meetings that, due to restrictions to contain the Coronavirus Sar-Cov-2 (Covid-19) Pandemic, took place remotely via Google Meet with students from the Final Grades Elementary School from a state public school located in Novo Hamburgo/RS. The main technological resource used by the participants and analyzed in this study was Scratch. We also use Machine Learning For Kids, a site that allows data training by Machine Learning algorithms and its use with the Scratch language. We observed that the PC and mathematics acted in complementary movements in the determination and conduction of the resolution of the problems. While Computational Thinking shows itself as a possibility to enhance the resolution, mathematics sometimes emerges as part of a problem to be solved. sorted out. We also emphasize that the Decomposition of problems, one of the pillars of the PC, is not something fixed, direct, or linear, and can emerge in different ways in the search for the solution to the problem. Thus, we noticed that the skills related to the PC are processes that take place over many transformations and that mathematics, in some cases, contributes to the process to determine the direction of the problem solution.

**Keywords:** Mathematics Education; Computational Thinking. Remote Teaching; Scratch.



## LISTA DE FIGURAS

FIGURA 1 - OS QUATRO PILARES QUE SUSTENTAM O PENSAMENTO COMPUTACIONAL .....	42
FIGURA 2 - ESTRATÉGIA "DIVIDIR PARA CONQUISTAR" .....	44
FIGURA 3 - PARTES DA BICICLETA (DECOMPOSIÇÃO) .....	45
FIGURA 4 - SEQUÊNCIA DE FIBONACCI .....	46
FIGURA 5 - PANFLETO DE DIVULGAÇÃO DO PROJETO INTITULADO PROGRAMAÇÃO COM SCRATCH .....	55
FIGURA 6- MENSAGEM ENVIADA VIA WHATSAPP PARA OS GRUPOS DAS TURMAS DA ESCOLA.....	55
FIGURA 7 - COMPARTILHAMENTO DA TELA DO SCRATCH PELO ALUNO EDUARDO .....	58
FIGURA 8 - INTERFACE DO SCRATCH VERSÃO 3 .....	59
FIGURA 9 - CATEGORIAS DE CÓDIGOS DE PROGRAMAÇÃO DO SCRATCH.....	60
FIGURA 10 - BLOCO DE PROGRAMAÇÃO CONSTRUÍDO NA PARTE CENTRAL DO SCRATCH.....	60
FIGURA 11 - PALCO OU INTERFACE DO PROGRAMA COM SEUS RESPECTIVOS EIXOS COORDENADOS .....	61
FIGURA 12 - ÁREA PARA SELECIONAR OU ENVIAR ATORES E CENÁRIOS .....	61
FIGURA 13- PÁGINA INICIAL DO RECURSO ON-LINE MACHINE LEARNING FOR KIDS.....	63
FIGURA 14 - LISTAS DE PALAVRAS PARA SEREM TREINADAS COM APRENDIZADO DE MÁQUINA ....	63
FIGURA 15 - CAIXA DE TEXTO PARA TESTAR O MODELO .....	64
FIGURA 16 – BLOCO DE CÓDIGOS GERADO NO SCRATCH A PARTIR DO MODELO TREINADO NO SITE MACHINE LEARNING FOR KIDS .....	64
FIGURA 17- INTERFACE DO PIXILART .....	65
FIGURA 18 - A ESPIRAL DA APRENDIZAGEM CRIATIVA .....	67
FIGURA 19 - OS 4 PRINCÍPIOS DA APRENDIZAGEM CRIATIVA .....	68
FIGURA 20 - IMAGEM CONTENDO DOIS PROGRAMAS DISPONÍVEIS NO SITE DO PROJETO .....	74
FIGURA 21 - PROGRAMA SELECIONADO POR MARIA.....	80
FIGURA 22 - INTERFACE DO PROGRAMA QUE ESTAVA SENDO PROGRAMADO POR MIGUEL.....	83
FIGURA 23 - FLUXOGRAMA DO ENTRELAÇAMENTO DA MATEMÁTICA COM O PENSAMENTO COMPUTACIONAL NO EPISÓDIO 1 .....	92
FIGURA 24 - BLOCO DE PROGRAMAÇÃO PARA AS DIFERENTES TROCAS DE CENÁRIO .....	100
FIGURA 25 - FLUXOGRAMA COM ASPECTOS ENVOLVENDO A PROBLEMÁTICA DE EDUARDO .....	105
FIGURA 26 - BLOCO DE PROGRAMAÇÃO COM DESLOCAMENTO POR PONTOS DO PLANO CARTESIANO.....	106
FIGURA 27 - FLUXOGRAMA COM OS ASPECTOS ENVOLVIDOS NAS PROBLEMÁTICAS DE PAULO .....	115

## LISTA DE QUADROS

QUADRO 1- CRITÉRIOS DE SELEÇÃO DAS PRODUÇÕES DE OLIVEIRA (2021) .....	22
QUADRO 2- CONCEITOS/HABILIDADES ANALISADOS PELO DR. SCRATCH COMO CRITÉRIOS DE AVALIAÇÃO.....	27
QUADRO 3 - ENTENDENDO A PROBLEMÁTICA DE MIGUEL.....	84
QUADRO 4 - BLOCOS DE PROGRAMAÇÃO E EXCERTO SOBRE A SOLUÇÃO DA PROBLEMÁTICA DE MIGUEL .....	86
QUADRO 5 - REFLEXÕES DE MIGUEL.....	89
QUADRO 6 - PRIMEIRO PROGRAMA APRESENTADO POR EDUARDO .....	95
QUADRO 7 - PARTE DA SOLUÇÃO DA PROBLEMÁTICA DE EDUARDO.....	97
QUADRO 8 - TROCA DE CENÁRIOS PROGRAMADA POR EDUARDO.....	98
QUADRO 9 - OUTRO PROJETO DE EDUARDO .....	101
QUADRO 10 - SOBRE A ANIMAÇÃO UTILIZANDO UM GIF.....	103
QUADRO 11 - PRIMEIRA PROBLEMÁTICA DE MIGUEL .....	108
QUADRO 12 - SOLUÇÃO DA PRIMEIRA PROBLEMÁTICA DE MIGUEL.....	109
QUADRO 13 - REFLEXÃO EM TORNO DO PRÓPRIO EIXO VERTICAL.....	110
QUADRO 14 - POSSIBILIDADES DE MOVIMENTO NO EIXO X DO SCRATCH .....	110
QUADRO 15 - TROCA DE CENÁRIOS DO PROGRAMA DE MIGUEL .....	112
QUADRO 16 - PROBLEMÁTICA NA TROCA DE CENÁRIOS DE MIGUEL .....	113
QUADRO 17 - SOLUÇÃO DA PROBLEMÁTICA DA TROCA DE CENÁRIOS DE MIGUEL.....	113

## **LISTA DE TABELAS**

<b>TABELA 1- PRESENÇA DOS ESTUDANTES EM CADA ENCONTRO .....</b>	<b>79</b>
---	-----------

## **LISTA DE SIGLAS**

<b>BNCC</b>	<b>Base Nacional Comum Curricular</b>
<b>EF</b>	<b>Educação Financeira</b>
<b>ERE</b>	<b>Ensino Remoto Emergencial</b>
<b>GIF</b>	<b>Graphics Interchange Format</b>
<b>PC</b>	<b>Pensamento Computacional</b>
<b>RSL</b>	<b>Revisão Sistemática de Literatura</b>
<b>TD</b>	<b>Tecnologias Digitais</b>
<b>TIC</b>	<b>Tecnologias da Informação e da Comunicação</b>

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO E JUSTIFICATIVA</b> .....	<b>14</b>
<b>2</b>	<b>TRABALHOS CORRELATOS</b> .....	<b>20</b>
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>34</b>
3.1.	TECNOLOGIAS DIGITAIS NA EDUCAÇÃO MATEMÁTICA.....	34
3.2.	CONSTRUCIONISMO COMO BASE PARA O ENSINO E PARA A APRENDIZAGEM DE MATEMÁTICA POR MEIO DE PROJETOS.....	36
3.3.	PENSAMENTO COMPUTACIONAL.....	41
3.3.1.	PENSAMENTO COMPUTACIONAL COM A LINGUAGEM <i>SCRATCH</i> .....	47
3.4.	A ESCOLA E O FUTURO.....	49
<b>4</b>	<b>METODOLOGIA DE PESQUISA</b> .....	<b>52</b>
4.1.	PESQUISA QUALITATIVA.....	52
4.2.	PARTICIPANTES DA PESQUISA.....	54
4.3.	RECURSOS UTILIZADOS PARA A PRODUÇÃO DE DADOS.....	56
4.4.	PLANEJAMENTO DAS ATIVIDADES.....	66
4.4.1.	<i>A proposta de Aprendizagem Criativa e a sua relação com o Scratch</i> .....	66
4.4.2.	<i>Planejamento dos sete encontros</i> .....	72
4.5.	DADOS PRODUZIDOS.....	78
<b>5</b>	<b>ANÁLISE DOS DADOS</b> .....	<b>82</b>
5.1.	EPISÓDIO 1: MIGUEL E A DECOMPOSIÇÃO NÃO LINEAR.....	83
5.1.1.	<i>Primeiro momento: entendendo o problema</i> .....	83
5.1.2.	<i>Segundo momento: problema resolvido</i> .....	86
5.1.3.	<i>Terceiro momento: reflexões com base na solução do problema</i> .....	89
5.1.4.	<i>Como aspectos Matemáticos e do Pensamento Computacional emergiram no projeto desenvolvido por Miguel</i> .....	91
5.2.	EPISÓDIO 2: EDUARDO RECONHECE PADRÕES E OS DUPLICA.....	94
5.2.1.	<i>Apresentando a primeira ideia de Eduardo</i> .....	95
5.2.2.	<i>Problemáticas diferentes, soluções similares</i> .....	96
5.2.3.	<i>Novas construções de Eduardo a partir dos problemas solucionados</i> .....	101
5.2.4.	<i>Como aspectos matemáticos e do Pensamento Computacional emergiram nos projetos desenvolvidos por Eduardo</i> .....	104
5.3.	EPISÓDIO 3: PAULO E O PROCESSO DE DEPURAÇÃO.....	106
5.3.1.	<i>Movimento de Rotação e Reflexão</i> .....	107

5.3.2.	<i>Correção de bugs em pares</i> .....	111
5.3.3.	<i>Como aspectos matemáticos e do Pensamento Computacional emergiram nos projetos desenvolvidos por Paulo</i> .....	115
5.4.	<i>Sistematização da Análise</i> .....	116
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>119</b>
<b>7</b>	<b>REFERÊNCIAS</b> .....	<b>123</b>
	<b>APÊNDICE I: CARTA DE PEDIDO DE AUTORIZAÇÃO À ESCOLA</b> .....	<b>128</b>
	<b>APÊNDICE II: TERMOS DE ASSENTIMENTO E CONSENTIMENTO</b> .....	<b>129</b>
	<b>APÊNDICE III: PLANEJAMENTO DOS ENCONTROS</b> .....	<b>134</b>

## 1 Introdução e Justificativa

Esta pesquisa começou a ser elaborada no ano de 2019. A minha ideia<sup>1</sup> principal ao ingressar no mestrado já era pesquisar sobre o ensino e a aprendizagem de matemática por meio de programação, com foco no desenvolvimento do Pensamento Computacional dos participantes, que seriam os alunos dos anos finais do ensino fundamental que estudam na escola onde sou professor de matemática. Fui provocado pelo meu orientador com a ideia de que, além da programação, deveríamos trazer o Aprendizado de Máquina para a pesquisa, algo que, apesar de conhecer o termo, eu não estava inteirado do assunto.

No âmbito do Programa de Pós-Graduação em Ensino de Matemática, já pesquisando sobre trabalhos que envolviam tecnologias digitais no ensino, tive a oportunidade de conhecer estudos como os de Gayeski (2019), Dal Agnol (2019) e Dalla Vecchia (2018) que faziam uso do *Big Data*<sup>2</sup> junto de algum recurso tecnológico para ensinar matemática e desenvolver outras competências, tanto para alunos de ensino básico quanto de ensino superior. A partir disso, fui pesquisar sobre Inteligência Artificial, mais precisamente sobre uma de suas áreas de estudo denominada Aprendizado de Máquina.

O Aprendizado de Máquina pode ser visto como um conceito que compreende várias técnicas que visam ao desenvolvimento de programas de computador que tenham a capacidade de, a partir de suas próprias experiências, melhorar o desempenho na realização de alguma tarefa (FACELI *et al.*, 2011; CERRI, CARVALHO, 2017). Fiquei surpreso, pois, nas buscas feitas naquele momento, não encontrei trabalhos que utilizaram alguma técnica dentro da área de Aprendizado de Máquina para ensinar algum conteúdo no ensino básico, apenas elas sendo aplicadas por profissionais para analisar algo referente à Economia, Medicina, Miologia, entre outros, e, quando era relacionado à Educação, as pesquisas procuravam usar o Aprendizado de Máquina como um recurso de avaliação do ensino já realizado. Com isso, surgiu a seguinte questão: por que não ensinamos aos alunos como utilizar alguma técnica envolvendo conceitos de Aprendizado de Máquina para que eles resolvam problemas, ao invés de só usarmos essas técnicas para solucionar problemas referentes a esses estudantes?

---

<sup>1</sup> Parte desse capítulo é escrito em primeira pessoa do singular devido ao fato de apresentar a motivação e parte do percurso do autor.

<sup>2</sup> “O termo *Big Data* está relacionado ao tratamento de bases de dados muito grandes e que muitas vezes exigem técnicas e recursos diferentes dos que normalmente são utilizados para trabalhar dados” (DALLA VECCHIA, 2018, p. 2).

Após algum tempo de estudos sobre algumas técnicas que envolviam a área de Aprendizado de Máquina e a compreensão da falta de pesquisas que a relacionavam à Educação Matemática, eu trouxe a ideia de utilizarmos alguns algoritmos compreendidos neste campo com a linguagem de programação *Python*, mas, logo depois, por exemplos de algoritmos utilizados pelo orientador, mudamos para a linguagem *R*.

Começamos a pensar em atividades para serem elaboradas com os alunos programando em *R* e notei uma certa dificuldade em planejar algo que dialogasse com o nosso referencial teórico, principalmente com Papert (1986, 1988, 2008). Isso se dava pelo fato de que as atividades com a linguagem de programação *R* iriam depender de muito tempo com o professor dando instruções em como programar e criar algoritmos, ou seja, o foco ficaria no que Papert (1988) define como Instrucionismo e não com o que queríamos, o Construcionismo. Nesse mesmo período conheci o ambiente de programação *Scratch* na disciplina Tecnologias Digitais na Educação Matemática do Programa de Pós-Graduação, no contexto em que esta pesquisa foi desenvolvida e está vinculada, e o utilizei com meus alunos dos anos finais do ensino fundamental.

O *Scratch* é um ambiente de programação visual criado em 2006, que funciona por meio de blocos. Foi desenvolvido pelo *Lifelong Kindergarten Group* do *MIT Media Lab* e é destinado a jovens entre 8 e 16 anos, apesar de também ser utilizado por pessoas de outras idades. Com ele, os usuários podem programar suas próprias histórias, jogos, animações interativas, músicas, entre outros projetos, podendo também compartilhá-los entre si. Após experimentar alguns recursos, descobri o *Machine Learning For Kids*<sup>3</sup>, site que apresenta uma abstração de como o Aprendizado de Máquina pode ser utilizado na construção de algoritmos que buscam classificar texto, imagens ou sons em categorias predefinidas, tanto no *Scratch* como em outras linguagens de programação. Assim, depois de uma revisão de literatura sobre a utilização do *Scratch* no ensino de matemática, a compreensão de algumas potencialidades para o desenvolvimento do Pensamento Computacional e a possibilidade de utilizar o *Machine Learning For Kids*, decidimos utilizar o *Scratch* como recurso principal para o desenvolvimento de habilidades relacionadas ao Pensamento Computacional e à matemática.

Neste mesmo período, trabalhei como voluntário na tradução do *Scratch Jr*<sup>4</sup>, uma linguagem de programação criada para que crianças entre cinco e sete anos programem e, nesse

---

<sup>3</sup> Disponível em: <https://machinelearningforkids.co.uk/>.

<sup>4</sup> Disponível em: <http://www.Scratchjr.org/>.



processo, resolvam problemas, criem projetos e se expressem criativamente. Fazendo parte da equipe de tradução, conheci a proposta de Aprendizagem Criativa de Resnick (2020), um conjunto de princípios que visa orientar ações em sala de aula que possibilitem o desenvolvimento e o refinamento de habilidades relacionadas ao pensamento criativo, para que os estudantes possam desenvolver suas próprias ideias, testá-las, experimentar alternativas, procurar opiniões de outras pessoas e desenvolver ideias de acordo com suas experiências. Ao entender que essa proposta se relacionava com o desenvolvimento do Pensamento Computacional e observar a falta de pesquisas na área de Educação Matemática junto a ela, decidimos utilizá-la como um princípio norteador para o planejamento das atividades que seriam desenvolvidas com os estudantes na produção de dados.

Com o planejamento praticamente pronto, visando à utilização do Laboratório de Informática da escola e com o objetivo de aplicá-lo no início do ano letivo de 2020, ou seja, março/abril, tivemos um acontecimento que nos obrigou a pensar novamente o nosso planejamento: iniciou-se a pandemia do Coronavírus SARS-CoV-2 e todas as escolas estaduais foram fechadas por determinação do Decreto nº 55.118, de 16 de março de 2020, que dizia, em seu Artigo 5º:

Ficam suspensas, a contar de 19 de março de 2020, pelo prazo de quinze dias, prorrogáveis, as aulas presenciais no âmbito do Sistema Estadual de Ensino, devendo a Secretaria da Educação estabelecer plano de ensino e adotar as medidas necessárias para o cumprimento das medidas de prevenção da transmissão do COVID-19 (novo Coronavírus) determinadas neste Decreto (RIO GRANDE DO SUL, 2020a).

No primeiro mês, continuamos com o mesmo planejamento, com a esperança de que as aulas presenciais voltassem logo, mas, já no segundo mês, com as notícias sobre o crescimento da pandemia, notamos que precisaríamos adiar por mais tempo, principalmente quando o Conselho Estadual de Educação (CEEEd) emitiu um parecer afirmando que estávamos em uma “situação emergencial para o momento atual e que as alternativas possíveis, para validação do ano letivo 2020, podem ser por meio de atividades domiciliares e/ou de reorganização do Calendário Escolar [...]” (RIO GRANDE DO SUL, 2020b), desde que assegurada a carga horária mínima de 800 horas, distribuídas em 200 dias letivos determinados pela Lei de Diretrizes e Bases da Educação Nacional (BRASIL, 1996). Com isso, toda a rede pública estadual de ensino do Rio Grande do Sul e diversas instituições de ensino privado adotaram um modelo de ensino e de aprendizagem não presencial, denominado, em algumas instituições, de Aulas Remotas ou Ensino Remoto Emergencial (ERE) (HODGES *et al.*, 2020).

Não há um consenso sobre a definição de Aulas Remotas ou ERE. Entretanto, destacamos que o ERE é uma mudança temporária das atividades de diferentes instituições de

ensino devido à crise vivenciada, no qual o objetivo principal não é recriar todo um sistema educacional, mas fornecer, temporariamente, acesso às atividades e suporte aos envolvidos nesse processo durante a atual emergência (HODGES *et al.*, 2020). Por exemplo, entre março e abril de 2020, os alunos vinculados às escolas estaduais se envolveram com a realização de atividades domiciliares via material impresso, *WhatsApp* ou outros recursos organizados pelas instituições. Em maio deste mesmo ano, foi anunciado que seriam implantadas, em toda a rede estadual de ensino, as Aulas Remotas por meio do *Google Classroom*, uma plataforma *on-line* voltada ao ensino e aprendizagem que oportuniza aos educadores o gerenciamento das suas atividades e a realização delas pelos seus estudantes.

Segundo a Secretaria de Educação do Estado do Rio Grande do Sul, o objetivo do modelo de Aulas Remotas foi, “[...] por meio da plataforma *Google Classroom*, a conexão entre professores e alunos para a realização de aulas no ambiente virtual” (RIO GRANDE DO SUL, 2020c). As aulas nesse ambiente começaram a ser desenvolvidas no final de junho e, ainda sem previsão para o retorno presencial, constatamos que tínhamos duas possibilidades: adiar a aplicação das atividades para a produção de dados para o final de 2020 ou até para o próximo ano, ou pensar em outro meio para essa produção, e até mesmo de pesquisa, que fosse vinculada a esse modelo de Ensino Remoto Emergencial. Após algum tempo, optamos pela segunda opção por meio de atividades remotas síncronas envolvendo o *Google Meet*, ou seja, encontros virtuais por videochamada, além utilização do *WhatsApp* e de um site criado por nós com algumas instruções. Na escola, chamamos esse projeto de “Programação com *Scratch*”.

Assim, a partir dessa trajetória, planejamos novas atividades para serem ministradas por meio de atividades remotas síncronas utilizando, principalmente, o recurso de vídeo chamadas do *Google Meet*. Os sete encontros aconteceram semanalmente entre agosto e setembro de 2020 e, com eles, realizamos a produção de dados para que pudéssemos responder à pergunta diretriz que tínhamos naquele momento: quais as potencialidades para o campo da Educação Matemática do uso de Aprendizado de Máquina na construção de jogos eletrônicos?

Porém, considerando que “nenhuma pesquisa é totalmente controlável, com início, meio e fim previsíveis” (GOLDENBERG, 2014, p. 13), tivemos que reorientar o nosso estudo novamente após as considerações realizadas pelos membros da Banca de Qualificação que ocorreu no dia 4 de março de 2021. Os dados que produzimos não respondiam à nossa questão inicial, pois os estudantes não se envolveram com o recurso *Machine Learning For Kids* ao ponto de termos evidências para respondê-la. Ainda assim, os membros da banca destacaram que o que tínhamos produzido mostrava outros aspectos que relacionavam o Pensamento

Computacional e a matemática por meio de atividades remotas de forma síncrona. Assim, após complementarmos o nosso referencial teórico sobre o PC e a realização de uma nova busca por trabalhos correlatos, analisamos os nossos dados com o intuito de responder à nova questão: **como o Pensamento Computacional e a matemática emergem de projetos desenvolvidos com o ambiente de programação *Scratch* em atividades remotas?**

Observar como o PC e a matemática emergem por meio de projetos também é relevante, pois a Base Nacional Comum Curricular (BNCC) traz o Pensamento Computacional como uma das habilidades e competências que devem ser desenvolvidas nos alunos do ensino fundamental e médio. Além disso, ela aparece estritamente ligada ao Ensino de Matemática, área que “centra-se na compreensão de conceitos e procedimentos em seus diferentes campos e no desenvolvimento do pensamento computacional, visando à resolução e formulação de problemas em contextos diversos” (BRASIL, 2018, p. 471).

A BNCC define o PC como “as capacidades de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio do desenvolvimento de algoritmos” (BRASIL, 2018, p. 474). Neste mesmo documento, o Pensamento Computacional é relacionado ao que é chamado de Letramento Matemático, definido como:

[...] as competências e habilidades de raciocinar, representar, comunicar e argumentar matematicamente, de modo a favorecer o estabelecimento de conjecturas, a formulação e a resolução de problemas em uma variedade de contextos, utilizando conceitos, procedimentos, fatos e ferramentas matemáticas. É também o letramento matemático que assegura aos alunos reconhecer que os conhecimentos matemáticos são fundamentais para a compreensão e a atuação no mundo e perceber o caráter de jogo intelectual da matemática, como aspecto que favorece o desenvolvimento do raciocínio lógico e crítico, estimula a investigação e pode ser prazeroso (BRASIL, 2018, p. 266).

De acordo com o documento, essas competências e habilidades são desenvolvidas a partir dos processos matemáticos utilizados para a resolução de problemas, investigação matemática, desenvolvimento de projetos e modelagem, procedimentos que são “potencialmente ricos para o desenvolvimento de competências fundamentais para o letramento matemático (raciocínio, representação, comunicação e argumentação) e para o desenvolvimento do pensamento computacional” (BRASIL, 2018, p. 268).

A BNCC, na sua definição de Pensamento Computacional, já o associa à utilização de algoritmos, definindo como “a decomposição de um procedimento complexo em suas partes mais simples, relacionando-as e ordenando-as” (BRASIL, 2018, p. 271), e, especificamente na álgebra do ensino fundamental, o reconhecimento de padrões também se torna um dos pilares

do PC e vem “para se estabelecer generalizações, propriedades e algoritmos” (BRASIL, 2018, p. 271).

Além da BNCC, o Ministério da Educação (MEC), com o intuito de trazer uma formação aos educadores sobre o Pensamento Computacional, disponibiliza um curso de trinta horas, em um Ambiente Virtual de Aprendizagem (AVA), intitulado Introdução ao Pensamento Computacional<sup>5</sup>, no qual, além de trazer características dos quatro pilares que o sustentam (Abstração, Decomposição, Reconhecimento de Padrões e Algoritmos), também o associa à interdisciplinaridade e à cooperação dentro da sala de aula.

Considerando a BNCC como um documento orientador para as atividades desenvolvidas nas escolas brasileiras, cabe aos interessados, assim como foi feito nesta pesquisa, elaborarem estudos de como isso pode acontecer.

Essa dissertação segue estruturada em seis capítulos. No capítulo 2, apresentamos as pesquisas selecionadas no Banco de Dissertações e Teses da CAPES com relação com alguns aspectos da nossa pesquisa, além dos critérios de seleção delas.

No Capítulo 3, tratamos sobre as principais teorias que sustentaram esta pesquisa. Abordamos alguns possíveis impactos que o uso das Tecnologias Digitais causa na Educação Matemática, principalmente a partir do Construcionismo de Papert (1986, 1988, 2008), complementado por Maltempo (2005, 2012). Também apresentamos a conceituação do que é o Pensamento Computacional com base nas ideias de Wing (2006, 2008, 2011), e de outros pesquisadores, principal suporte de análise nesta pesquisa, além do seu envolvimento com o ambiente de programação *Scratch*.

No Capítulo 4, abordamos a nossa metodologia de pesquisa que foi seguida durante todo o processo de desenvolvimento deste trabalho, iniciando com o que entendemos sobre Pesquisa Qualitativa, apresentando os participantes da pesquisa, os recursos utilizados para a produção de dados, o planejamento dos encontros com base na Aprendizagem Criativa de Resnick (2020) e como foi a organização desse material.

No Capítulo 5, temos a descrição e a análise dos dados produzidos, que ocorreu a partir da criação de episódios que contam parte da trajetória de cada um dos participantes que se envolveu com pelo menos um projeto. Por fim, no Capítulo 6, apresentamos as nossas considerações finais, respondendo à nossa pergunta de pesquisa e fazendo algumas reflexões sobre o que foi desenvolvido durante a elaboração deste trabalho.

---

<sup>5</sup> Disponível em: <http://avamec.mec.gov.br/#/>.

## 2 Trabalhos Correlatos

Neste capítulo apresentamos trabalhos correlatos à nossa pesquisa, ou seja, que envolvessem o desenvolvimento do Pensamento Computacional junto à matemática. No catálogo de Teses e Dissertações da CAPES, buscamos pela expressão “Pensamento Computacional” e fizemos um primeiro filtro para trabalhos publicados entre 2020 e 2021, já que queríamos explorar pesquisas que tivessem sido publicadas durante a pandemia de Covid-19, com o intuito de que alguns pesquisadores tivessem produzido dados de forma remota. Além disso, para refinar ainda mais essa busca, filtramos a Área de Concentração para todas aquelas que possuíam a palavra “matemática”, que são: Ensino de Ciências e Matemática; Ensino de Ciências, Matemática e Tecnologias; Ensino de Matemática; Ensino e Aprendizagem de Ciências e Matemática; Ensino de Ciências e Educação Matemática e Matemática. Assim, tivemos 28 trabalhos previamente selecionados.

Buscamos, no título e no resumo desses trabalhos, por quatro palavras-chave: *Scratch*, Aulas Remotas, Ensino Remoto e “Remot”. Nosso intuito foi selecionar aqueles que pudessem estar mais próximos da nossa pesquisa, ou seja, que também tivessem utilizado o *Scratch* ou que tivessem realizado a produção de dados por meio de atividades remotas. Por fim, selecionamos os que tinham pelo menos uma das palavras-chave, que estavam disponíveis e que tinham produzido dados para a elaboração da pesquisa, resultando, assim, nos seis trabalhos apresentados neste capítulo.

A pesquisa de Corrêa (2021), intitulada “Programando com *Scratch* no Ensino Fundamental: uma possibilidade para a construção de conceitos matemáticos”, coletou dados por meio de uma oficina presencial oferecida de abril a julho de 2019 para estudantes do oitavo e do nono ano do ensino fundamental, onde, organizados em grupos, foram desafiados a programar um jogo utilizando a versão *on-line* do *Scratch*. Tendo como pergunta diretriz “quais as contribuições do uso da programação no *software Scratch* no desenvolvimento de conceitos e de elementos do Pensamento Computacional com estudantes de Ensino Fundamental?” (CORRÊA, 2021, p. 54), o pesquisador tinha como um dos objetivos “[...] investigar a conduta dos estudantes perante [...] o computador e as ações de programação” (CORRÊA, 2021, p. 58) e, para isso, utilizou-se do Método Clínico de Piaget.

A oficina, que foi organizada em dez encontros e que contou com a presença de oito estudantes, aconteceu na sala de informática de uma escola da rede privada no município de Porto Alegre, no Rio Grande do Sul, onde o pesquisador era professor, e foi dividida em duas

etapas: os cinco primeiros encontros foram para que os estudantes se familiarizassem com o *software* e com a programação em blocos, além da apresentação de recursos do *Scratch* que poderiam ser interessantes à construção dos programas, o que aconteceria na segunda etapa. Nesta primeira etapa, os estudantes foram divididos em dois grupos e cada grupo teve uma hora disponível com o professor em cada encontro. Já os últimos cinco encontros, que faziam parte da segunda etapa, tiveram como objetivo geral “[...] observar e analisar os participantes na programação de seus projetos, investigando as ações dos participantes na tentativa de compreender os conteúdos de seus pensamentos no processo de suas programações” (CORRÊA, 2021, p. 60-61). O pesquisador relata que, nesta segunda etapa, o segundo grupo produziu um trabalho que foi muito semelhante ao que foi desenvolvido nos encontros iniciais e, por este motivo, optou por analisar o desenvolvimento dos estudantes do primeiro grupo, com o intuito de responder à sua pergunta diretriz.

Sustentando o seu trabalho com base na perspectiva construcionista de Papert (1988, 2008), nos quatro pilares do Pensamento Computacional a partir das ideias de Wing (2006, 2008) e Resnick (2007), e a análise com base na Teoria do Desenvolvimento Cognitivo (PIAGET, 1978) que “[...] busca compreender de que forma a criança se desenvolve e como o conhecimento progride da passagem da ação física para o pensamento (ação mental) que é efetuado a partir das tomadas de consciência” (CORRÊA, 2021, p. 29), o pesquisador relata que “foi possível observar o pensamento matemático em diferentes momentos da programação das ações dos personagens” (CORRÊA, 2021, p. 140). O autor destaca a construção de algoritmos para solucionar situações específicas do programa, o uso de números positivos e negativos para que os atores pudessem se locomover tanto na horizontal quanto na vertical, coordenadas no Plano Cartesiano e proporcionalidade por meio de grandezas diretamente ou inversamente proporcionais.

Além disso, Corrêa (2021) sugere alguns conceitos matemáticos que podem ser construídos durante a programação no *Scratch*, evidenciando que pode servir “[...] como ponto de partida a aprendizagem do conceito de coordenadas cartesianas, o que foi de fundamental importância para os participantes compreenderem o movimento e a localização dos atores no palco” (CORRÊA, 2021, p. 141). Além dos já mencionados, outros conceitos tiveram destaque na pesquisa, como o uso de variáveis e de condicional na lógica matemática, este último podendo ser trabalhado com operadores de igualdade e desigualdade, “oportunizando na prática o uso desses conceitos que, geralmente, são utilizados em equações e inequações matemáticas” (CORRÊA, 2021, p. 141).

Analisando como o Pensamento Computacional pode ter contribuído, no processo de programação, para o desenvolvimento de conceitos matemáticos, Corrêa (2021) realça que o PC está ligado à habilidade de resolver problemas e que “[...] muitos desses problemas foram solucionados a partir do uso de conceitos matemáticos” (CORRÊA, 2021, p. 142). Ele também relata que o PC pode, além de desenvolver, também auxiliar na consolidação de conceitos matemáticos durante o processo de programação. Por fim, o autor conclui que o *Scratch* pode contribuir para o desenvolvimento, tanto diretamente como indiretamente, de habilidades matemáticas junto ao Pensamento Computacional, “possibilitando a construção de novos conceitos assim como o uso de conceitos já conhecidos, tanto para contribuir no ato de programar quanto para serem consolidados em situações práticas” (CORRÊA, 2021, p. 143).

Buscando responder “Como se apresenta a produção acadêmica (dissertações e teses) voltada aos estudos sobre o uso do ambiente de programação *Scratch* no processo de ensino e aprendizagem de Geometria desenvolvido no Brasil?”, Oliveira (2021) realizou uma Revisão Sistemática de Literatura (RSL), em agosto de 2020, para a sua pesquisa “*Scratch* no Ensino e Aprendizagem de Geometria: um Panorama de Pesquisas Brasileiras”.

Oliveira (2021) procurou trabalhos compreendidos na Biblioteca Digital Brasileira de Teses e Dissertações (BDTD), no Catálogo de Teses e Dissertações da CAPES e no portal do Mestrado Profissional em Matemática em Rede Nacional (PROFMAT). Seguindo os Critérios de seleção das produções conforme o Quadro 1, a pesquisadora encontrou 15 trabalhos compreendidos entre 2015 e 2021, nos quais pôde evidenciar quatro principais perspectivas do uso do *Scratch* no processo de ensino e aprendizagem de Geometria desenvolvido no Brasil.

**Quadro 1- Critérios de seleção das produções de Oliveira (2021)**

Critérios de inclusão	Critérios de Exclusão
<ul style="list-style-type: none"> <li>• O estudo deveria indicar alguma relação entre o <i>Scratch</i> e alguma competência, habilidade ou tópico relacionado à Matemática.</li> <li>• A palavra <i>Scratch</i> deve aparecer no texto da produção.</li> <li>• A palavra Geometria deve aparecer mais de 30 vezes no texto da produção.</li> </ul>	<ul style="list-style-type: none"> <li>• Produções caracterizadas como “estado da arte”, revisão sistemática da literatura e mapeamento sistemático.</li> <li>• Produções que não exploraram conceitos geométricos.</li> <li>• Pesquisas não disponíveis para acesso integral.</li> </ul>

Fonte: Adaptado de Oliveira (2021).

A primeira perspectiva “[...] refere-se as produções que descrevem a elaboração, o desenvolvimento e a análise de propostas didáticas, direcionadas a estudantes da Educação Básica, voltadas à aprendizagem de conteúdos/conceitos geométricos com auxílio do *Scratch*” (OLIVEIRA, 2021, p. 112). A segunda, similar à primeira, foca na aprendizagem do ambiente

de programação e, a partir disso, explora conteúdos/conceitos de geometria. A terceira é a respeito de propostas didáticas com o intuito de auxiliar professores no ensino e na aprendizagem de geometria com o auxílio do *software* e, por último, a quarta refere-se a pesquisas que elaboram e desenvolvem propostas didáticas dirigidas à formação de professores.

A pesquisadora constatou que as ideias construcionistas de Papert (1988, 2008) apareceram como principal referencial teórico nos trabalhos analisados. Além disso, o foco dos trabalhos estava nos Anos Finais do Ensino Fundamental, no qual “uma interpretação para este resultado pode estar no fato de haver um maior número de habilidades voltadas ao desenvolvimento do PC na BNCC na parte específica da matemática nos Anos Finais [...]” (OLIVEIRA, 2021, p. 113). Entre as atividades que eram desenvolvidas e relatadas nas pesquisas, a pesquisadora destaca que as principais envolviam “construção de polígonos regulares, estudo de características e propriedades de triângulos, ângulos e deslocamento e localização no plano” (OLIVEIRA, 2021, p. 115), estando o conceito de ângulo presente em todas as produções.

Observando aspectos relacionados ao Pensamento Computacional, a pesquisadora destaca que a principal representação de Algoritmos estava diretamente relacionada aos códigos construídos no *Scratch*. Entretanto, Oliveira (2021, p. 115) entende que “[...] a construção do algoritmo em linguagem natural pode auxiliar na resolução do problema e conseqüentemente na elaboração de um código mais eficiente, em especial, na identificação de padrões e generalização”, que são essenciais ao desenvolvimento do Pensamento Computacional e matemático.

Ao realçar algumas limitações presentes nas pesquisas, Oliveira (2021) destaca que a maioria dos trabalhos analisados contava com um número reduzido de participantes, destoando da realidade da educação brasileira. Além disso, limitações a alguns conteúdos/conceitos geométricos também foram encontrados, principalmente nas propostas que envolviam “[...] o estudo de semelhança de triângulos, relações métricas no triângulo retângulo e transformações geométricas (simetria)” (OLIVEIRA, 2021, p. 115).

Por fim, Oliveira (2021) conclui que o Pensamento Computacional pode sim contribuir para o desenvolvimento do Pensamento Matemático. Além disso, a pesquisadora compreende que “[...] a construção de jogos pelos próprios educandos, potencializa atribuir significados aos conceitos matemáticos e computacionais, por ser uma produção do estudante e fortalecendo o senso crítico, pelas várias etapas necessárias à sua construção” (OLIVEIRA, 2021, p. 116).



Podendo servir de contraponto às limitações encontradas por Oliveira (2021), o trabalho de Horbach (2020), intitulado “Semelhança de Triângulos: um estudo propositivo através do *Scratch*”, teve como questão norteadora “Quais as contribuições do *software Scratch* na aprendizagem significativa do conteúdo de semelhança de triângulos para alunos do 9º ano?” (HORBACH, 2020, p. 12). Para isso, o pesquisador realizou um estudo de caso com uma turma do 9º ano, contendo 28 estudantes, em uma escola municipal em Chapecó, Santa Catarina. Uma sequência didática para 10 encontros de 1h e 30min foi organizada e aplicada na sala de informática da escola, entre setembro e novembro de 2019, onde parte dos dados foram produzidos.

As atividades foram organizadas de forma que os alunos resolvessem exercícios predefinidos para se familiarizar com as funções do *Scratch* já envolvendo aspectos matemáticos, como a construção de quadrados e triângulos equiláteros. Depois disso, o pesquisador organizou outras atividades a serem seguidas pelos estudantes, que envolveram retângulos semelhantes e, posteriormente, o conteúdo de semelhança de triângulos. Algumas limitações apareceram, como a utilização de uma versão mais antiga do *Scratch*, já que os computadores do laboratório de informática não suportavam a versão mais recente e não acessavam a internet. Por também não ser possível salvar os arquivos com os programas nos computadores, o pesquisador solicitou que os participantes escrevessem os blocos de programação construídos para que ele e os estudantes pudessem utilizá-los posteriormente. Além da aplicação da sequência didática, a produção de dados envolveu o emprego de dois questionários, um inicial e outro final, ambos abrangendo questões diferentes sobre lógica de programação e semelhança de triângulos.

Além de procurar aporte teórico no Construcionismo de Papert (1988, 2008) e nas ideias sobre o Pensamento Computacional de Wing (2006) e Brackmann (2017), Horbach (2020) adotou a Aprendizagem Significativa de Ausubel (MOREIRA; MASINI, 1982). Segundo o pesquisador, a Aprendizagem Significativa pode ser definida como “[...] um processo por meio do qual uma nova informação relaciona-se com alguma informação já existente na estrutura cognitiva do indivíduo” (HORBACH, 2020, p. 18) e que, em contrapartida a ela, temos a Aprendizagem Mecânica, que foca a aprendizagem de novas informações tendo pouca ou nenhuma relação com conceitos já desenvolvidos.

O pesquisador pode constatar que os estudantes conseguiram compreender o conceito de Semelhança de Triângulos, “pois utilizaram corretamente as proporções entre os lados dos triângulos para encontrar os valores desconhecidos” (HORBACH, 2020, p. 46). Além disso,

ele ressalta que os participantes assimilaram os comandos básicos do *Scratch*, já que conseguiram construir os triângulos sem muitas dificuldades, e que, ao observar os códigos, notou uma desorganização, com partes das resoluções estando misturadas umas com as outras. Ainda sobre o *Scratch*, Horbach (2020, p. 47) observa que “[...] as estruturas de programação como ‘repita’, ‘espere’ e ‘variável’ se fizeram presentes, demonstrando ganhos significativos relacionados à lógica de programação e que poderão ser ampliados em trabalhos futuros”.

Refletindo sobre as dificuldades que encontrou, o autor realça que notou o interesse dos estudantes diminuindo nos últimos encontros, considerando, assim, duas possibilidades do motivo: a aplicação de uma sequência didática longa que, no final, acabou se tornando cansativa ou “[...] por se tratar de atividades que incentivavam a criatividade e autonomia dos participantes, alguns não se sentiam confortáveis com essa situação, talvez por estarem habituados com o ensino tradicional” (HORBACH, 2020, p. 49).

Enfim, o pesquisador concluiu que “[...] foi possível estimar contribuições na utilização do *software Scratch* na aprendizagem significativa do conteúdo de semelhança de triângulos” (HORBACH, 2020, p. 45-50), alcançando, assim, o objetivo do trabalho.

Outra pesquisa que utilizou o *Scratch* e respaldou suas ideias sobre o Pensamento Computacional a partir de Wing (2006) foi a de Ferreira (2020). Com o título “Educação Financeira com o *Scratch*: potencialidades e dificuldades”, a pesquisadora traz a informação de que existe um número significativo de pessoas que possuem restrição no nome e, após análise de dados disponibilizados no site do Serasa<sup>6</sup>, observa que “[...] a maioria das complicações financeiras foram causadas por compras de eletrônicos, eletrodomésticos, serviços de telefonia, roupas e afins” (FERREIRA, 2020, p. 14). Considerando que esses itens, na maioria das vezes, admitem a possibilidade de um planejamento de compra a longo prazo, a pesquisadora defende que seria possível diminuir esse índice de devedores com mais Educação Financeira (EF), que, no ensino fundamental, “serve como apoiadora ao cidadão, dando embasamento às escolhas mais efetivas sobre o uso e gerenciamento do seu dinheiro” (FERREIRA, 2020, p. 14). Além disso, a autora considera outros fatores que agravam esta situação, como a falta de estudo sobre operações referentes às variedades de opções de compra e à disponibilidade de crédito fácil.

Tudo isso corrobora para a necessidade de discutir educação financeira principalmente com as crianças e jovens em idade escolar e se justifica pois, se as relações consumeristas tornaram-se mais complexas, mais precocemente o cidadão

---

<sup>6</sup> A Serasa é uma empresa privada que analisa dados e informações relevantes para decisões de crédito. Disponível em: <https://www.serasa.com.br/>.

deve educar-se financeiramente assumindo uma postura mais crítica e consciente quando participar ativamente no mercado de trabalho (FERREIRA, 2020, p. 16).

A partir disso, o trabalho de Ferreira (2020) buscou, por meio de um viés qualitativo, responder à seguinte pergunta diretriz: “como a produção de animações com o *software Scratch* pode contribuir para a compreensão e significação da educação financeira por alunos do sétimo e oitavo ano do Ensino Fundamental?” (FERREIRA, 2020, p. 18). Para tal, a pesquisadora realizou um levantamento bibliográfico que pudesse auxiliá-la na compreensão do seu estudo e no entendimento do que seria exequível. Também foi realizado um curso de 32 horas com 15 estudantes do sétimo e do oitavo ano dos anos finais do ensino fundamental. O curso foi elaborado com base em uma sequência didática dividida em nove encontros que aconteceram entre abril e junho de 2019 no Laboratório de Informática da Universidade do Estado de Mato Grosso (UNEMAT). Com isso, Ferreira (2020) coletou dados por meio de entrevistas semiestruturadas, observações, notas de campos, fichas, registros audiovisuais e questionários.

O primeiro encontro foi destinado à apresentação do *Scratch* e de como os próximos encontros funcionariam, além de um questionário com o objetivo principal de “[...] coletar dados demográficos para caracterização dos sujeitos pesquisados, tais como, idade, sexo e renda familiar, além de informações que indicassem os conhecimentos prévios dos alunos sobre temas como gastos pessoais e familiares, despesas e economia.” (FERREIRA, 2020, p. 60). Já foi observado nas respostas deste primeiro questionário que o conhecimento dos estudantes em relação à Educação Financeira poderia ser apontado como limitado ou quase nulo, evidenciando, de acordo com a pesquisadora, a necessidade de uma abordagem de conceitos básicos ao longo do curso. Para familiarizá-los com alguns recursos do *software*, eles começaram sendo orientados para a criação de um jogo de “*ping pong*” e, em seguida, puderam manuseá-lo livremente.

Depois disso, cada encontro foi dividido em dois momentos: no primeiro foram abordados conceitos de Educação Financeira e, depois do intervalo, os estudantes se reuniam em grupos para a produção de animações no *Scratch* e a utilização de outros recursos, como planilhas eletrônicas, *e-mail*, *YouTube*<sup>7</sup>, entre outros.

Evidências de que o uso do *Scratch* favoreceu o desenvolvimento do Pensamento Computacional apareceram com a análise dos programas desenvolvidos pelo recurso Dr. *Scratch*<sup>8</sup>, que, de acordo com a pesquisadora,

---

<sup>7</sup> Disponível em: <https://www.youtube.com/>.

<sup>8</sup> Disponível em: <http://www.drscratch.org/>.

[...] tem como principal objetivo analisar, de forma gratuita e automática, a presença de blocos de códigos, o uso de *scripts* e a forma como a programação foi realizada, fornecendo um relatório sobre aspectos do desenvolvimento do pensamento computacional encontrados em cada projeto avaliado (FERREIRA, 2020, p. 49).

O programa gera uma pontuação de acordo com sete conceitos/habilidades consideradas como tendo relação com o PC: abstração, lógica, sincronização, paralelismo, controle de fluxo, interatividade com o usuário e representação de dados. Cada um dos critérios avaliados nesses itens está descrito no Quadro 2.

**Quadro 2- Conceitos/habilidades analisados pelo Dr. *Scratch* como critérios de avaliação**

Conceitos-alvo	Crítérios Avaliados
Abstração	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).
Lógica	Essas instruções definem um processo de estruturação da programação de acordo com as normas da lógica que permite chegar a uma determinada conclusão ou resolver um problema, utilizando os conectivos, se e senão, se e então, e as operações E e OU.
Sincronização	Permite que os personagens organizem a ordem em que as ações vão ocorrer, ou seja, sincronizar o comportamento com relação a tempo.
Paralelismo	É a possibilidade de que várias coisas ocorram simultaneamente, ou seja, ao mesmo tempo. Por exemplo, um personagem faz uma ação enquanto outro está fazendo outra.
Controle de Fluxo	Instruções relacionadas a noções de controle de fluxo podem controlar o comportamento dos personagens, fazendo, por exemplo, com que alguns blocos sejam repetidos um número de vezes até que uma situação ocorra. Controlando o personagem com um conjunto de blocos executados um após o outro.
Interatividade com o usuário	As instruções sobre interatividade com o usuário podem ajudar os projetos <i>Scratch</i> a ficarem mais interativos com quem os manuseia, ou seja, permitir que quem os rode possa criar situações novas. Por exemplo, poderia usar o teclado ou mouse para mover um personagem, responder perguntas ou usar a webcam, entre outras possibilidades.
Representação de dados	Conjunto de informações sobre os personagens para funcionar corretamente. Por exemplo, posição de cada personagem, a direção que está apontando, tamanho, entre outros.

Fonte: Ferreira (2020, p. 50-51).

Além disso, mesmo não sendo o objetivo do trabalho, a pesquisadora ressalva que diversos conceitos matemáticos que não estavam relacionados à Educação Financeira foram desenvolvidos na programação das animações. Entre eles, Plano Cartesiano, ângulos, números positivos e negativos, lógica, entre outros.

Enfim, focando o objetivo de “identificar as potencialidades e dificuldades do uso do *software Scratch* para a produção de animações abordando temas para educação financeira” (FERREIRA, 2020, p. 20), a autora acredita que o que foi desenvolvido pelos estudantes contribuiu para o entendimento dos conceitos e princípios da EF. Isso aconteceu pelo fato dos estudantes “[...] precisarem pesquisar para montarem e fundamentarem suas histórias, e o desejo de produzirem animação, fizeram com que investigassem e compreendessem sobre os

temas abordados” (FERREIRA, 2020, p. 89). Também é destacado que o *Scratch* promoveu a socialização e o trabalho colaborativo, já que “os alunos não somente conversavam com seus pares, mas também discutiam com outros grupos sobre determinadas funções que tinham sido utilizadas em suas animações” (FERREIRA, 2020, p. 87-88).

Em “Uso do *Scratch* para investigação matemática e os Números Mágicos de Ball”, Silva (2021) propôs oficinas para analisar como estudantes dos anos finais do ensino fundamental se envolviam em um processo de investigação matemática sobre os Números Mágicos de Ball com o uso do *Scratch*. De acordo com o pesquisador, a Investigação Matemática é um processo que pode ser dividido em quatro momentos:

O primeiro momento envolve o reconhecimento da situação, a sua exploração preliminar e a formulação de questões. O segundo refere-se ao processo de formulação de conjecturas. O terceiro inclui a realização de testes e o eventual refinamento das conjecturas. E, finalmente, o último, diz respeito à argumentação, demonstração e avaliação do trabalho realizado (PONTE, 2003, p. 7<sup>9</sup>, apud SILVA, 2021, p. 25).

Entretanto, devido às restrições impostas em decorrência da pandemia de Covid-19, o pesquisador reorganizou o seu trabalho com a disponibilização de um vídeo explicativo da tarefa, que consistia em um programa disponibilizado na comunidade *on-line* do *Scratch*. Além disso, um questionário foi enviado aos participantes com o objetivo de “[...] averiguar se o cenário elaborado no *Scratch* auxiliava no entendimento do algoritmo para gerar os números mágicos de Ball [...]” (SILVA, 2021, p. 57).

A produção de dados aconteceu em 2020 por meio desse questionário aplicado a cinco estudantes do 6º ao 8º ano da Escola Estadual Homero Honorato em Goiás e da Escola em Tempo Integral Padre Josimo Taves no Tocantins. A atividade foi executada durante quatro dias, iniciando pelo vídeo explicativo do pesquisador e auxílio dos professores regentes dos estudantes. Foi planejado que eles tivessem acesso ao programa do *Scratch* por três dias e, só depois desse tempo, o questionário que contemplava questões envolvendo os Números Mágicos de Ball foi enviado.

Dessa forma, o pesquisador destaca que o uso do *Scratch* proporciona uma alternativa ao ensino e a aprendizagem de matemática, além de

[...] uma dinâmica de possibilidades na construção do saber, pois tem características de uma nova forma de aprender, que predomina do início da elaboração e construção do objeto, acompanhada ou realizada pelo docente, e a sua execução ou utilização com estudantes (e demais usuários da plataforma *Scratch*) (SILVA, 2021, p. 62).

---

<sup>9</sup> PONTE, E. A. S. **Investigações matemática na sala de aula**. Autêntica Editora, Edição do Kindle, 2003.

Além disso, Silva (2021) afirma que a utilização da Investigação Matemática como metodologia de ensino pode despertar o interesse e a curiosidade do estudante, além do desejo de entender e resolver atividades de forma autônoma.

Outras linguagens de programação também podem favorecer o desenvolvimento do Pensamento Computacional junto ao ensino e à aprendizagem de Matemática. Graves (2021), no seu trabalho intitulado “O Pensamento Computacional na prática: uma experiência usando *Python* em aulas de matemática básica”, realizou um estudo de caso com 25 estudantes do sétimo ano que eram seus alunos em uma escola da rede privada de Cruz Alta, no Rio Grande do Sul. Pelo fato da produção de dados ter acontecido durante a pandemia de Covid-19, todo o processo ocorreu de forma remota, ou seja, sem encontros presenciais.

Com o objetivo de “[...] introduzir a programação juntamente com o pensamento computacional nas aulas de Matemática em uma turma de educação básica” (GRAVES, 2021, p. 14), o pesquisador embasou o seu trabalho nos quatro pilares do Pensamento Computacional a partir de Brackmann (2017) e de Wing (2006), alinhados à Resolução de Problemas, de acordo com Polya (1978). A produção de dados aconteceu no segundo semestre de 2020, a partir de cinco encontros de 30 minutos cada via *Google Meet*. Nesses encontros os seguintes assuntos foram abordados: “o que é a linguagem de programação *Python*; como utilizar a função *print*; efetuando operações matemáticas utilizando a programação; atribuindo valores as variáveis; resolvendo problemas usando a programação em *Python*” (GRAVES, 2021, p. 38).

Cada encontro era iniciado com uma introdução e explicação da atividade do dia por videochamada. Com isso, o pesquisador compartilhava a tela do seu computador e mostrava o *Google Colab*<sup>10</sup>, ou *Google Colaboratory*, “um serviço de nuvem gratuito hospedado pelo *Google* para incentivar a pesquisa de Aprendizado de Máquina e Inteligência Artificial” (GRAVES, 2021, p. 39). Por ser *on-line*, gratuito e possível de ser executado em computadores, *tablets* e *smartphones*, foi escolhido para que os estudantes desenvolvessem seus códigos e algoritmos. A escolha da linguagem *Python*, de acordo com o autor, foi feita pela possibilidade de utilizá-la no navegador de internet junto ao *Google Colab* e por suas características que permitem que, com poucas instruções, pequenos programas possam ser desenvolvidos e experimentos matemáticos possam ser realizados.

Depois da apresentação, era disponibilizado um *link* ao ambiente no *Google Colab*, elaborado pelo professor para que, com base no exemplo realizado na videochamada, os alunos

---

<sup>10</sup> Disponível em: <https://research.google.com/colaboratory/>.

pudessem fazer a atividade/exercício proposto. O pesquisador também sugeria aos estudantes que buscassem auxílio na internet para a resolução desses exercícios, como em vídeos no *YouTube* ou em outras fontes. Esse material compartilhado no final de cada aula foi reorganizado e disponibilizado como um Produto Educacional “[...] composto por uma série de atividades, nas quais o professor terá um aprendizado básico da linguagem de programação e posteriormente algumas sugestões de atividades para serem desenvolvidas no ensino fundamental e ensino médio” (GRAVES, 2021, p. 47).

O pesquisador observa que a utilização das técnicas de resolução de problemas sugeridas por Polya (1978) fizeram com que as atividades fossem solucionadas de forma mais organizada. Além disso, quando vinculadas ao Pensamento Computacional em um ambiente de programação, pode-se vincular a aula às TICs (Tecnologias da Informação e da Comunicação), assim amparando-a na Base Nacional Comum Curricular. Por fim, Graves (2021, p. 55) conclui que “é possível inserir a linguagem de programação em aulas de matemática de forma prática e objetiva usando a linguagem de programação *Python* e o ambiente *Google Colab*”, seja em aulas presenciais ou remotamente.

As pesquisas aqui apresentadas nos mostram diversas potencialidades da elaboração de atividades que relacionam o Pensamento Computacional junto ao ensino e à aprendizagem de matemática de diferentes formas. Assim como neste trabalho, Corrêa (2021) produziu dados a partir da construção de programas no *Scratch* que estivessem relacionados aos interesses dos estudantes, sem impor qualquer tipo de regra ou tema, característica importante em um ambiente construcionista (PAPERT, 1986). Outro ponto correlato à nossa pesquisa é o fato de o pesquisador ter analisado o processo de programar dos participantes a partir das falas e das construções, entendendo assim como o PC pode auxiliar na consolidação de conceitos matemáticos. Entre esses conceitos, destacamos aqueles que também apareceram nos nossos dados, como o Plano Cartesiano, algoritmos, Números Inteiros, variáveis e proporcionalidade. Diferente de nós, Corrêa (2021) teve a oportunidade de trabalhar com os estudantes no formato presencial no laboratório de informática e em grupos, assim como tínhamos planejado no início, aspecto importante de uma abordagem com base na Aprendizagem Criativa (RESNICK, 2020).

Apesar de ter concentrado seu estudo nas pesquisas envolvendo o ensino e a aprendizagem de geometria, a RSL de Oliveira (2021) oportunizou que conhecêssemos algumas perspectivas de como o *Scratch* vem sendo utilizado neste campo. A primeira se refere às produções voltadas à aprendizagem de conteúdos de geometria com o auxílio do *Scratch*, já a segunda, da qual a nossa mais se aproxima, tem o foco na aprendizagem do ambiente de

programação e, a partir disso, possibilita a exploração de conteúdos/conceitos relacionados à geometria. A terceira perspectiva evidencia propostas didáticas destinadas a auxiliar professores no ensino e na aprendizagem de Geometria por meio do *Scratch* e, por último, a quarta enfatiza a formação de professores relacionada ao Pensamento Computacional. Apesar de serem diferentes, grande parte dos trabalhos dentro dessas perspectivas possuem algo em comum: o Construcionismo de Papert (1986, 1988, 2008) presente como aporte teórico.

Oliveira (2021) salienta que a maioria dos trabalhos tinha como foco a construção de polígonos regulares, como quadrados e triângulos retângulos, nos dando indícios de que as atividades possuíam um viés mais fechado, não oportunizando que o estudante desenvolvesse projetos do seu interesse. Todavia, assim como Corrêa (2021), nós concordamos com a importância de um início fechado para possibilitar que o estudante consiga se habituar aos comandos do *Scratch*, mas também, seguindo a ideia de um ambiente construcionista (PAPERT, 1988, 2008) e a metodologia da Aprendizagem Criativa (RESNICK, 2020), que a proposta tenha um final aberto, permitindo que o estudante desenvolva projetos do seu interesse.

Diferente do que foi feito na nossa pesquisa e na de Corrêa (2021), Horbach (2020) relata que utilizou exercícios predefinidos na sua produção de dados, tanto para a familiarização dos participantes com o *Scratch* quanto para o desenvolvimento das atividades sobre Semelhança de Triângulos. Como evidenciado pelo pesquisador, uma proposta com esse viés pode sim desenvolver habilidades relacionadas ao PC junto a conceitos/conteúdos de matemática. Entretanto, numa perspectiva construcionista, ela pode destoar de algumas dimensões, como a Sintônica, que destaca a importância de o estudante desenvolver projetos que tenham relevância para ele, e a Social, na qual o ambiente precisa estar relacionado com as relações pessoais e com a cultura na qual o estudante faz parte.

Com interesse na construção de animações que envolvessem o tema de Educação Financeira, a pesquisa de Ferreira (2020) apresenta o *Scratch* sendo utilizado como um recurso para complementar outras atividades, como pesquisas e aulas. A pesquisadora iniciava os encontros desenvolvendo alguns conceitos do tema proposto e, posteriormente, os alunos utilizavam o que tinha sido aprendido, ou até buscavam em outras fontes, para construir animações sobre o assunto. Nesse procedimento, a pesquisadora utilizou o Dr. *Scratch* para avaliar algumas características do Pensamento Computacional, analisando assim apenas o que nós chamamos de produto do PC, e não o processo pelo qual os estudantes passaram. Com isso, o *Scratch* foi utilizado como um meio para possibilitar o aumento do interesse dos alunos sobre



a temática que estava sendo trabalhada, diferentemente do nosso, no qual conceitos e aspectos matemáticos foram emergindo a partir da programação e das discussões.

Assim como o nosso trabalho, as pesquisas de Silva (2021) e de Graves (2021) também foram afetadas pela pandemia de Covid-19 e, com isso, as atividades para a produção de dados tiveram que ser reformuladas, acontecendo, assim, remotamente. Silva (2021) utilizou a metodologia de Investigação Matemática e construiu um cenário no *Scratch* para que os alunos se envolvessem com a construção e o reconhecimento dos Números Mágicos de Ball. Diferentemente do que temos nas pesquisas comentadas até o momento e na nossa, os estudantes não utilizaram o *Scratch* como um recurso de programação, mas como algo que guiou parte da investigação proposta pelo pesquisador. Já Graves (2021), divergente dos trabalhos destacados até o momento, elaborou atividades focadas no desenvolvimento do Pensamento Computacional e no ensino e aprendizagem de matemática utilizando o *Google Colab* e a linguagem de programação *Python*. Para isso, o pesquisador também organizou encontros via *Google Meet*, entretanto, tendo como suporte teórico as ideias de Resolução de Problemas de Polya (1978) e não o Construcionismo (PAPERT, 1988, 2008), elaborou atividades fechadas organizadas na forma de exemplos e exercícios, ou seja, os estudantes não tiveram a oportunidade de construir seus próprios programas.

Em resumo, as pesquisas apresentadas nos mostram diferentes formas de desenvolver o Pensamento Computacional ao mesmo tempo que trabalham com aspectos relacionados ao ensino e à aprendizagem de matemática. Notamos que a nossa pesquisa converge com as ideias da organização de um ambiente construcionista trazida por Corrêa (2021) e com a segunda perspectiva observada por Oliveira (2021), ou seja, a utilização de atividades abertas em que os alunos possam desenvolver projetos no *Scratch* com base em seus interesses e, a partir disso, aspectos relacionados ao PC e à matemática são desenvolvidos. Entretanto, assim como Silva (2021) e Graves (2021), estávamos em um período de Ensino Remoto Emergencial, realizando, assim, as atividades para a produção de dados no formato não presencial.

Quando observamos os aspectos em que a nossa pesquisa se diferencia das que foram selecionadas para este capítulo, notamos que conseguimos transpor parte daquilo que vinha sendo feito presencialmente para um modelo de atividades remotas. Corrêa (2021) criou um ambiente construcionista para que os estudantes, em grupos, construíssem programas no *Scratch* com base em seus interesses e, a partir disso, aspectos relacionados à matemática e ao Pensamento Computacional surgiram e foram desenvolvidos. Nós planejamos as atividades para que os estudantes tivessem a mesma liberdade para projetarem a partir de seus interesses,

entretanto, devido às restrições impostas pelas autoridades para conter a pandemia de Covid-19, tivemos que repensar o ambiente construcionista de forma remota. Entretanto, conseguimos observar como os estudantes se envolveram nesse processo de construção de forma remota e como a matemática e o PC emergiram nos projetos desenvolvidos com o *Scratch*.

Ao analisarmos os trabalhos que também produziram dados por meio de atividades remotas, notamos que as atividades que foram planejadas por esses pesquisadores não envolviam o processo de construção de conhecimentos pelos estudantes a partir de seus interesses. Silva (2021) elaborou atividades preestabelecidas no *Scratch* e os dados dos estudantes foram analisados a partir de questionários, ou seja, a programação foi realizada apenas pelo professor. Já Graves (2021) criou exercícios para que os estudantes realizassem via outro *software* de programação, não oportunizando que os estudantes o utilizassem na construção de algo que partisse deles. Assim, o nosso estudo destaca que há possibilidades de desenvolver aspectos relacionados à matemática e ao PC com programas que foram criados a partir do interesse dos estudantes, mesmo de forma remota, possibilitando, ainda, observar como essas duas áreas se relacionam nesse processo.

Diferentemente de Horbach (2020), de Ferreira (2020), de Silva (2021) e de Graves (2021), as nossas atividades não tiveram como objetivo utilizar o *Scratch* para desenvolver algum conteúdo/conceito de matemática ou relacionado ao Pensamento Computacional, mas sim oportunizar que os estudantes, ao programarem, buscassem soluções para os problemas que surgissem no desenvolvimento dos seus projetos com o *Scratch*. Assim, a partir desse processo, buscamos evidências para responder à nossa pergunta diretriz: **como o Pensamento Computacional e a matemática emergem de projetos desenvolvidos com o ambiente de programação Scratch em atividades remotas?**

### 3 Fundamentação Teórica

Neste capítulo estão reunidas as principais teorias que sustentaram esta pesquisa. A primeira seção traz uma discussão sobre o impacto do uso de Tecnologias na Educação Matemática a partir das ideias construcionistas de Papert (1986, 1988, 2008) e de Maltempi (2005, 2012). Já na Seção 3.2, apresentamos a caracterização do que é o Pensamento Computacional com base em Wing (2006, 2008, 2011) e de outros pesquisadores, principal suporte de análise que tivemos nessa pesquisa, além do seu envolvimento com o *Scratch*. Por fim, trazemos algumas ideias sobre a escola e o futuro, refletindo sobre quais tipos de estudantes acreditamos que as instituições precisam formar e como isso pode ser feito.

#### 3.1. Tecnologias Digitais na Educação Matemática

Segundo Abbagnano (2007), em seu *Dicionário de Filosofia*, o termo tecnologia pode ser definido como um “estudo dos processos técnicos de determinado ramo da produção industrial ou de vários ramos” (ABBAGNANO, 2007, p. 942). Com isso, queremos ressaltar que tecnologia não precisa ser, necessariamente, uma tecnologia digital (TD). Por se tratar de um termo mais amplo, destacamos que tecnologias e TD não são sinônimas. Assim, destacamos que esta pesquisa evidenciou a relação da utilização de algumas tecnologias digitais com o desenvolvimento do Pensamento Computacional e o ensino e a aprendizagem de matemática.

Podemos considerar um recurso de Informática como uma tecnologia digital que, com o seu surgimento no século passado, as primeiras iniciativas para que fosse utilizada na educação foram visando a “transformar o computador em máquina de ensinar, ‘transmissora de conhecimentos’” (MALTEMPI, 2005, p. 1). Para Papert (1988, 2008), criador do ambiente de programação *LOGO* e um dos precursores da pesquisa do uso de tecnologias digitais na educação, esses recursos podem servir como um meio para a criação de ambientes nos quais os estudantes possam se desenvolver mediante um processo ativo de aprendizagem, proporcionando “[...] ao aprendiz a possibilidade de fazer algo e, com isso, poder construir conhecimentos a partir de suas próprias ações” (MALTEMPI, 2012, p. 289). Assim, Papert (1988, 2008) propõe a utilização da informática como um meio para que o conhecimento seja construído ativamente pelas pessoas e não transferido para elas.

Autores como Basso e Notare (2015), que buscam compreender o impacto do uso de recursos das tecnologias digitais nos processos que envolvem o conhecimento matemático, observam uma simbiose entre as tecnologias digitais e o desenvolvimento humano. Afirmam que a evolução da matemática e a da Educação Matemática foi oportunizada pelo surgimento dessas TD e dos ambientes dinâmicos, que proporcionaram que o estudante pudesse refletir ao observar e construir com as tecnologias digitais, trazendo a ideia de que “os computadores afetam as pessoas e suas formas de pensar” (BASSO; NOTARE, 2015, p. 3). Assim, essas tecnologias digitais potencializaram o que antigamente era feito somente com lápis e papel, auxiliando-nos a fazer melhor o que já fazíamos antes (PAPERT, 2008).

Destacamos que esse potencial não é visto pelos autores como forma de proporcionar mais rapidez na busca por soluções de problemas, praticidade no trabalho do professor ou tornar a aula mais interessante ao aluno, mas sim com um meio para “desencadear o pensamento matemático, a proporcionar aos alunos possibilidades para acessar e manipular objetos matemáticos até então não acessíveis” (BASSO; NOTARE, 2015, p. 3). Além disso, um dos principais impactos na educação é de ordem epistemológica e cognitiva, pois as atividades de matemática envolvendo tecnologias digitais “vão além do ‘aprender matemática’, em direção ao ‘fazer matemática’” (BASSO; NOTARE, 2015, p. 4). Com isso, elas possibilitam que o estudante compreenda que é capaz de criar e pensar com matemática, não apenas reproduzir.

Rosa (2018), por sua vez, defende que as tecnologias digitais podem servir como um meio pelo qual nós “[...] produzimos conhecimento com o mundo, com as Tecnologias Digitais que se encontram no mundo, e não sobre o mundo, sozinhos, de forma que essas tecnologias simplesmente nos auxiliam a pensar sobre algo” (ROSA, 2018, p. 257), tendo o recurso tecnológico participação na construção do conhecimento. Para o autor, as TD, no processo educacional, são vistas como ferramentas que poderiam tornar a atividade mais eficiente, ágil ou econômica. “A mídia está envolvida no próprio pensar” (ROSA, 2018, p. 260). Assim, há “uma relação de pensar-com-TD que permite a produção de conhecimento” (ROSA, 2018, p. 262.).

Com isso, inspirados na ideia de “fazer matemática” de Basso e Notare (2015) e no “pensar-com-TD” de Rosa (2018), desenvolvemos ações buscando atentar para o uso de tecnologias digitais em um processo no qual os estudantes tivessem a oportunidade de, a partir de seus interesses, envolverem-se com aspectos associados à matemática. Para isso, firmamos nossas ideias na perspectiva construcionista de Seymour Papert (1986, 1988, 2008), apresentada na próxima seção.

### 3.2. Construcionismo como base para o ensino e para a aprendizagem de matemática por meio de projetos

Papert (1988) via, na utilização do computador e do que ele denominou “geometria da Tartaruga” no ambiente *LOGO*, um modo do estudante usar, pensar e até brincar com a matemática, ou seja, expressar-se matematicamente. Isso acontecia, pois o ambiente possibilitava a visualização, fosse ela na tela do computador ou no robô tartaruga, do que ele estava fazendo durante a construção dos programas, e o que ele fazia partia das suas necessidades para o que estava construindo e não as de outra pessoa. A partir disso, o autor criou um conjunto de ideias denominado Construcionismo, na qual nega que “o caminho para uma melhor aprendizagem deve ser o aperfeiçoamento da instrução” (PAPERT, 2008, p. 134), sugerindo que a criança aprenderá melhor descobrindo e construindo o conhecimento específico que precisa.

A teoria construcionista considera que “[...] o desenvolvimento cognitivo é um processo ativo de construção e reconstrução das estruturas mentais, no qual o conhecimento não pode ser simplesmente transmitido do professor para o aluno” (MALTEMPI, 2012, p. 288). Sendo assim,

Educar, portanto, é principalmente dar condições para que os alunos construam, mas não se resume a isso. O Construcionismo postula que o aprendizado ocorre especialmente quando o aprendiz está engajado em construir um produto de significado pessoal (por exemplo, um poema, uma maquete ou um *website*), que possa ser mostrado a outras pessoas. Portanto, ao conceito de que se aprende melhor fazendo, o Construcionismo acrescenta: aprende-se melhor ainda quando se gosta, pensa e conversa sobre o que se faz (MALTEMPI, 2005, p. 3).

Apesar disso, o Construcionismo “[...] não põe em dúvida o valor da instrução como tal, pois isso seria uma tolice: mesmo a afirmativa (endossada, quando não originada, por Piaget) de que cada ato de ensino priva a criança de uma oportunidade de descoberta” (PAPERT, 2008, p. 134). A meta, segundo Papert (2008, p. 134), “é ensinar de forma a produzir a maior aprendizagem a partir do mínimo de ensino”. Essa mudança de uma perspectiva instrucionista para uma construcionista “[...] assemelha-se a um provérbio africano: se um homem tem fome, você pode dar-lhe um peixe, mas é melhor dar-lhe uma vara e ensiná-lo a pescar” (PAPERT, 2008, p. 134). Assim, o Construcionismo sugere que as crianças aprenderão melhor “pescando” o conhecimento que precisam e uma das possibilidades para que isso ocorra é utilizando computadores como “varas de pesca” e ambientes que tenham “águas férteis” para a produção desse conhecimento, ou seja, “[...] um ambiente que seja rico em materiais de

referência, que incentive a discussão e a descoberta e que respeita as características específicas de cada um” (MALTEMPI, 2012, p. 290), denominados, nesta perspectiva, micromundos.

A linguagem de programação *LOGO* é parte de um micromundo, um ambiente no qual as pessoas possam usar a matemática a partir de projetos que partam de seus interesses, que elas possam, além de usar, brincar com a matemática, possibilitando essa construção do conhecimento. A ideia de que um micromundo potencializa essa construção também está vinculada ao fato de que, nesse ambiente, “falso ou verdadeiro, certo ou errado não são os critérios decisivos” (PAPERT, 1988, p. 163). Assim, a utilização do computador junto ao *LOGO* no processo de aprendizagem possibilita que o estudante desenvolva habilidades referentes ao processo de correção de *bugs*, denominado depuração ou *debugging*, fazendo com que ele lide melhor com os seus erros:

[...] numa aula de matemática típica, a reação da criança a uma resposta errada é tentar esquecê-la o mais rápido possível. Mas no ambiente LOGO ela não é criticada por ter feito um erro ao desenhar. O processo de debugging é uma parte integrante do processo de compreensão de um programa. O programador é encorajado a estudar o bug ao invés e esquecê-lo. No contexto da Tartaruga há uma boa razão para estudá-lo: valerá a pena (PAPERT, 1988, p. 85).

Assim, as habilidades desenvolvidas nesse processo de depuração em um ambiente de programação podem potencializar outras áreas, pois é uma atividade que

[...] tem sua origem no erro, e este está intimamente relacionado com a construção de conhecimentos, pois atua como um motor que desequilibra e leva o aprendiz a procurar conceitos e estratégias para melhorar o que já conhece. Nesta busca, novas informações são processadas e agregadas ao conhecimento já existente (MALTEMPI, 2012, p. 295).

Segundo Papert (1988, p. 142), “a escola ensina que errar é mau; a última coisa que alguém deseja fazer é examinar esses erros, deter-se neles ou mesmo pensá-los”. Em um micromundo com o *LOGO* não há erros, apenas *bugs* e, com eles, o processo de depuração, no qual não se espera que o programa funcione na primeira tentativa e não se faz julgamentos do tipo “certo” ou “errado”, mas se pergunta: como eu posso corrigir isso? Assim, esse processo sugere que “erros são benéficos porque nos levam a estudar o que aconteceu, a entender o que aconteceu de errado, e, através do entendimento, a corrigi-los” (PAPERT, 1988, p. 142).

Para que possamos organizar ambientes que estejam de acordo com as ideias construcionistas, Papert (1986) sugere cinco dimensões que devem ser buscadas quando há este propósito, que são: Pragmática, Sintônica, Sintática, Semântica e Social<sup>11</sup>.

---

<sup>11</sup> Do inglês: Logo’s pragmatics, Logo’s syntonic, Logo’s syntactic, semantics and social.

A Dimensão Pragmática traz a ideia de que os estudantes não devem aprender algo com a promessa de que aquilo será importante e usual quando eles crescerem, mas que já podem usar agora. Segundo Papert (1986, p. 14, tradução nossa), “essa característica na qual a utilização é imediata é rara nas disciplinas de matemática e ciências vistas na escola”.<sup>12</sup> Assim, o “despertar para o desenvolvimento de algo útil coloca o aprendiz em contato com novos conceitos. O domínio destes conceitos traz uma sensação de praticidade e poder, incentivando cada vez mais a busca pelo saber” (MALTEMPI, 2012, p. 290).

A Dimensão Sintônica destaca a importância de o estudante desenvolver projetos que estejam em sintonia com os seus próprios interesses, sendo assim, “é importante dar ao aprendiz a oportunidade de participar da escolha do tema do projeto a ser desenvolvido” (MALTEMPI, 2012, p. 291). Diferentemente das atividades que, na maioria das vezes, são encontradas no ensino tradicional, no qual os estudantes resolvem problemas que estão completamente dissociados da realidade na qual vivem, o desenvolvimento de um projeto precisa partir do interesse do aluno, estando sim relacionado com as suas experiências.

Um projeto “[...] envolve a construção de artefatos ou objetos, que podem ser concretos ou abstratos [...]. Esses artefatos são frutos de ideias e do meio usado para expressar e materializar essas ideias [...]” (MALTEMPI, 2012, p. 292). Além desta construção, o desenvolvimento de projetos abrange outras características consideradas importantes, como planejamento, organização, estratégias, política, entre outras.

Pelo fato de que o objetivo a ser atingido em um projeto não é bem definido, determinar o problema também faz parte do trabalho de quem está desenvolvendo o projeto. Além disso, Maltempi (2012) salienta que o resultado de um projeto não é único, depende do meio e que cada projetista pode decidir soluções que sejam satisfatórias de forma diferente, variando assim de acordo com quem está projetando. Assim, “[...] o que pode ser uma ótima solução para um indivíduo, pode não satisfazer um outro” (MALTEMPI, 2012, p. 292).

Trabalhar com projetos nos possibilita a abertura de espaços para que os estudantes aprendam com base em experiências anteriores, em um processo social no qual os desenvolvedores de projetos constroem soluções diferentes para um mesmo problema e estão dispostos a discutir e aprender sobre resultados divergentes. Entretanto, para que isso ocorra, é importante o papel do professor como alguém que problematiza as questões que podem surgir, agindo como um mediador deste processo de aprendizagem.

---

<sup>12</sup> Do inglês: “This immediate utilitarian quality is rare in school mathematics and science”.

Outro fator a considerar é que não devemos limitar ou reduzir o projeto no qual o estudante está envolvido a uma sequência de passos predefinidos, pois assim poderemos restringir algumas possibilidades que tornarão o estudante o agente criador do que está desenvolvendo. Também é necessário, de acordo com Maltempi (2012, p. 293), “[...] que o aprendiz tenha tempo suficiente para se relacionar com a tarefa e, assim, executá-la”.

Sendo assim, podemos considerar que um ambiente de aprendizagem que propõe a construção do conhecimento por meio de projetos com um viés construcionista possui os seguintes aspectos:

- o aprendiz torna-se um participante ativo no processo de aprendizagem, tendo controle e responsabilidade sobre o mesmo;
- reflexão e discussão são estimuladas pela presença do artefato que está sendo desenvolvido;
- a tarefa de projetar pode ser abordada de diferentes formas, satisfazendo estilo e preferências do aprendiz. Uma vez que a dicotomia certo/errado é evitada, múltiplas estratégias e soluções são possíveis;
- as atividades de projetos geralmente são interdisciplinares;
- a relação aprendiz-artefato é facilitada e fortalecida pelo fato de o aprendiz ser o agente criador do artefato; e
- o aprendiz é estimulado a considerar a reação de outras pessoas perante o artefato que criou; (MALTEMPI, 2012, p. 293).

A terceira Dimensão, chamada de Sintática, sugere que o recurso o qual o aprendiz está usando deve possibilitar que ele, mesmo que iniciante, acesse os recursos básicos sem precisar adquirir algum conhecimento previamente, podendo assim progredir “[...] gradualmente nesse ambiente, de acordo com suas necessidades e seu próprio desenvolvimento cognitivo” (ROSA, 2004, p. 46). Essa característica também é destacada nas ideias de Aprendizagem Criativa de Resnick (2020), que utilizamos como parte do nosso referencial teórico como “pisos baixos”. Além disso, é importante que o ambiente viabilize o avanço para aspectos mais complexos, característica denominada de “tetos altos”. Rosa (2004) destaca que, caso o recurso seja um *software*, a sua interface deve ser de fácil manipulação.

A Dimensão Semântica corrobora com a ideia de que “com a Geometria da Tartaruga, nós não estamos lidando com uma estrutura matemática formal cercada por símbolos. As ações no Geometria da Tartaruga têm significado. De fato, elas têm muitos significados”<sup>13</sup> (PAPERT, 1986, p. 14, tradução nossa). Ou seja, os objetos a serem manipulados pelo estudante devem “apresentar no que se refere às situações reais do mundo, ou mesmo concretas (que façam

---

<sup>13</sup> Do inglês: “With Turtle Graphics, we are not dealing with a formalistic mathematical structure in which symbols are pushed around. Statements in Turtle Graphics have meaning. In fact, they have multiple meanings.”



sentido ao aluno) e que tenham relação com os conceitos a serem construídos” (ROSA, 2004, p. 46), indo além de um aglomerado de símbolos, letras e números.

Por fim, a Dimensão Social sustenta a ideia de que o ambiente construcionista precisa estar relacionado com as relações pessoais e com a cultura nas quais o estudante se encontra. Para Rosa (2004, p. 47), esta dimensão é de grande importância para a aprendizagem do estudante, pois, ao “[...] desenvolver um ambiente no qual o aprendiz busca relações sociais e que, conseqüentemente, revele que tal envolvimento humano, durante uma atividade, facilita muito a produção de significado para o aluno”.

Neste sentido, essas cinco dimensões

[...] indicam que o Construcionismo vai além do aspecto cognitivo, incluindo também as facetas social e afetiva da educação. Assim, ele abre espaço para o estudo das questões de tecnologia, gênero, cultura, personalidade, motivação, entre outras (MALTEMPI, 2012, p. 292).

A ideia de Papert (1988, 2008) em utilizar o *LOGO* como parte de um micromundo que possibilita a construção de conhecimentos pelos participantes por meio de projetos foi, na nossa pesquisa, proposta mediante a utilização do *Scratch*. Utilizamos esse recurso em um micromundo no qual os estudantes foram convidados a desenvolver projetos a partir de seus próprios interesses, projetos estes constituídos por programas utilizando a linguagem de programação definida, com jogos, animações, histórias interativas, entre outros. Com isso, possibilitamos que eles se relacionassem com habilidades relativas ao Pensamento Computacional e, junto disso, pudessem construir algum conhecimento envolvendo matemática ou outras áreas. Além disso, assim como no processo construcionista criado com as atividades em *LOGO*, o processo de depuração também pode acontecer no *Scratch*, podendo estabelecer uma melhor relação do estudante com o seu erro, e essa relação ser transposta para outras áreas, sendo, assim, benéfica.

Também, considerando o surgimento de outras tecnologias digitais, acreditamos que cabe a nós, professores, agirmos como o que Papert (1988) caracterizou como educador-antropólogo, no qual

[...] sua tarefa é trabalhar para entender que materiais dentre os disponíveis são relevantes para o desenvolvimento intelectual. Assim, ele deve identificar que tendências estão ocorrendo no meio em que vivemos. Uma intervenção significativa só acontece quando se trabalha de acordo com essas tendências (PAPERT, 1988, p. 50).

Com isso, a utilização do *Scratch* junto do recurso *Machine Learning for Kids* surgiu como uma possibilidade de trazer outros campos tecnológicos para a sala de aula, no nosso caso, o envolvimento com uma potencialidade do Aprendizado de Máquina. Além disso, a utilização desses recursos trouxe a oportunidade para que os estudantes se envolvessem com

aspectos relacionados ao Pensamento Computacional, principal conjunto de ideias que serviu como suporte para a análise dos dados produzidos e que é apresentado na próxima seção.

### 3.3. Pensamento Computacional

Jeannette M. Wing publicou, em 2006, um artigo intitulado “Pensamento Computacional: um conjunto de atitudes e habilidades universalmente aplicáveis, que todos, não apenas cientistas da computação, estariam ansiosos para aprender e utilizar”<sup>14</sup> (WING, 2006, p. 33, tradução nossa). O termo se relaciona com um conjunto de habilidades que seriam essenciais para a busca por soluções de problemas, a partir de conceitos fundamentais das Ciências da Computação (WING, 2006), em que “o poder das nossas ferramentas ‘mentais’ é amplificado pelo poder das nossas ferramentas metálicas”<sup>15</sup> (WING, 2008, p. 3718, tradução nossa), como os computadores. Algumas características específicas são apresentadas (WING, 2006), como a reformulação de um problema difícil em problemas menores que podem ser resolvidos com mais facilidade; pensar recursivamente; utilizar-se da abstração e decomposição ao atacar um problema, separando-o em interesses menores; a correção de erros; planejar, aprender e agendar na presença da incerteza, utilizar-se de uma quantidade imensa de dados para aumentar a velocidade de computação; entre outros.

Algumas estratégias sugeridas por Wing (2006, 2008, 2011) para a busca de solução de problemas são semelhantes às ideias de Polya (1978) sobre como resolver problemas de matemática e às ideias de Papert (1988). Em “A Arte de Resolver Problemas” (POLYA, 1978), o autor traz diversas estratégias e métodos que permitiriam potencializar a busca por soluções de problemas matemáticos, sugerindo por onde começar, a procura por problemas correlatos, a ação de decomposição e recombinação, utilização de resultados de outros problemas, características que se assemelham ao PC. Já nas ideias de Papert (1988), observamos correlações com o que o pesquisador define como Pensamento Processual, que inclui os processos de desenvolvimento, representação, teste e *debugging*, visto como um processo

---

<sup>14</sup> Do inglês: “Computational Thinking: It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use”.

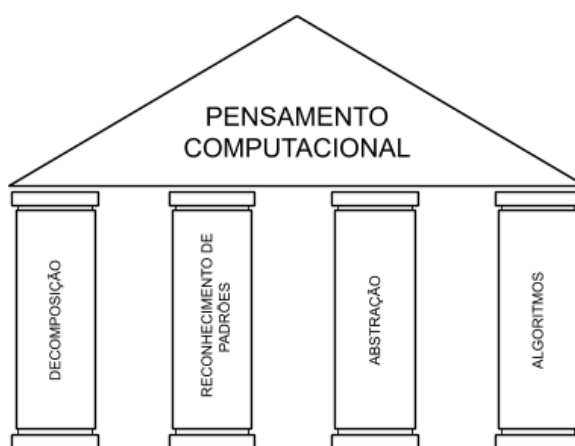
<sup>15</sup> Do inglês: “The power of our ‘mental’ tools is amplified by the power of our ‘metal’ tools”.

efetivo para a busca de resoluções de problemas. Papert (1988) também utiliza algumas estratégias sugeridas por Polya (1978) no seu conjunto de ideias construcionistas:

[...] algumas ideias usadas na geometria da Tartaruga são casos especiais das sugestões de Polya. Por exemplo, ele recomenda que sempre que abordamos um problema deveríamos percorrer uma lista mental de perguntas heurísticas, tais como: esse problema pode ser subdividido em problemas mais simples? Pode ser relacionado a outro problema que já sei como resolver? A geometria da Tartaruga presta-se a esse tipo de exercício (PAPERT, 1988, p. 88).

A partir disso, consideramos o Pensamento Computacional como sendo um conjunto de processos mentais que envolvem a formulação e a busca de soluções de problemas por meio de conceitos fundamentais da Ciência da Computação que podem ser interpretados por agentes humanos, máquinas ou, na maioria das vezes, pela combinação dos dois (WING, 2006, 2008, 2011; NRC, 2010). De forma sintetizada, o PC abrange a identificação de um problema complexo que pode ser dividido em problemas menores, aumentando assim a sua facilidade de resolução (BRACKMANN, 2017). Além disso, cada um desses problemas pode ser analisado separadamente, possibilitando a sua identificação com problemas que já foram solucionados e focando exclusivamente nos detalhes que mais importam, em oposição a informações que são irrelevantes para a solução. Por fim, um conjunto finito de passos pode ser planejado para que cada um dos subproblemas seja resolvido. Esses processos contornam algumas habilidades que são vistas como fundamentais para o desenvolvimento do Pensamento Computacional e que podem ser entendidas como sendo sustentadas por quatro pilares: Abstração, Decomposição, Reconhecimento de Padrões e Algoritmos, conforme a Figura 1.

**Figura 1 - Os quatro pilares que sustentam o Pensamento Computacional**



Fonte: Adaptada de Vicari (2018).

O pilar da Abstração, visto por Wing (2006, 2008, 2011) como a principal habilidade do PC, pode ser definido como “a ação ou o efeito de selecionar os aspectos de objetos ou de processos que devem ser considerados para satisfazer um determinado objetivo” (BRASIL,

2019). É quando classificamos partes de um problema em categorias que são consideradas mais ou menos importantes para a busca de sua solução. A Abstração também envolve a “filtragem dos dados e sua classificação, essencialmente ignorando elementos que não são necessários para que se possa concentrar nos que são relevantes” (BRACKMANN, 2017, p. 38).

Wing (2011) traz que a abstração é utilizada para definir padrões e “capturar as propriedades essenciais que são comuns em um conjunto de objetos enquanto escondemos distinções que são irrelevantes entre eles”<sup>16</sup> (WING, 2011, p. 2, tradução nossa), oportunizando-nos organizar e trabalhar com problemas complexos. A autora concebe este pilar como o principal, pois ele pode ser utilizado em diversos momentos, como na escrita de um algoritmo e suas iterações, na seleção dos dados relevantes para o problema, na elaboração de uma pergunta, entre outros aspectos (WING, 2006). Podemos citar alguns exemplos dessa habilidade em situações não relacionadas com a computação, como quando classificamos livros em uma biblioteca e os organizamos de forma que facilite a procura por algum deles; a organização das peças que estão dentro de um roupeiro que facilita a procura de alguma específica; a escrita de um problema matemático de forma descritiva para uma linguagem algébrica; entre outros.

De acordo com Ribeiro, Foss e Cavalheiro (2020), a abstração no contexto do Pensamento Computacional compreende três aspectos principais: Dados, Processos e Técnicas de construção de algoritmos. Os Dados, que são divididos em de entrada e de saída, permitem que as informações envolvidas na solução de um problema possam ser descritas. Em computação, as abstrações de dados mais importantes são mediante registros, ou seja, um conjunto de informações de um objeto, uma lista com uma sequência de dados ou por meio de grafos, estrutura que relaciona diferentes entidades associadas por vértices e arcos (RIBEIRO; FOSS; CAVALHEIRO, 2020). Os Processos são “abstrações que permitem definir algoritmos que descrevem a solução de um problema” (RIBEIRO; FOSS; CAVALHEIRO, 2020, p. 23). Já os modos que admitem a construção da solução de problemas complexos são conhecidos como Técnicas de construção de algoritmos.

Saber classificar quais dados são ou não importantes na busca pela solução de um problema também é uma característica que Polya (1978) considera importante, trazendo que “a primeira coisa a fazer com um problema é compreendê-lo bem: Quem entende mal, mal

---

<sup>16</sup> Do inglês: “It is used to capture essential properties common to a set of objects while hiding irrelevant distinctions among them”.

responde. Precisamos distinguir claramente a meta que desejamos alcançar” (POLYA, 1978, p. 140). Assim, ao evidenciar a meta, a classificação dos dados que serão ou não necessários pode facilitar a resolução do problema.

O segundo pilar, definido como o pilar da Decomposição, sustenta a habilidade de decompor um problema complexo em problemas menores de mais fácil resolução. De acordo com Brackmann (2017),

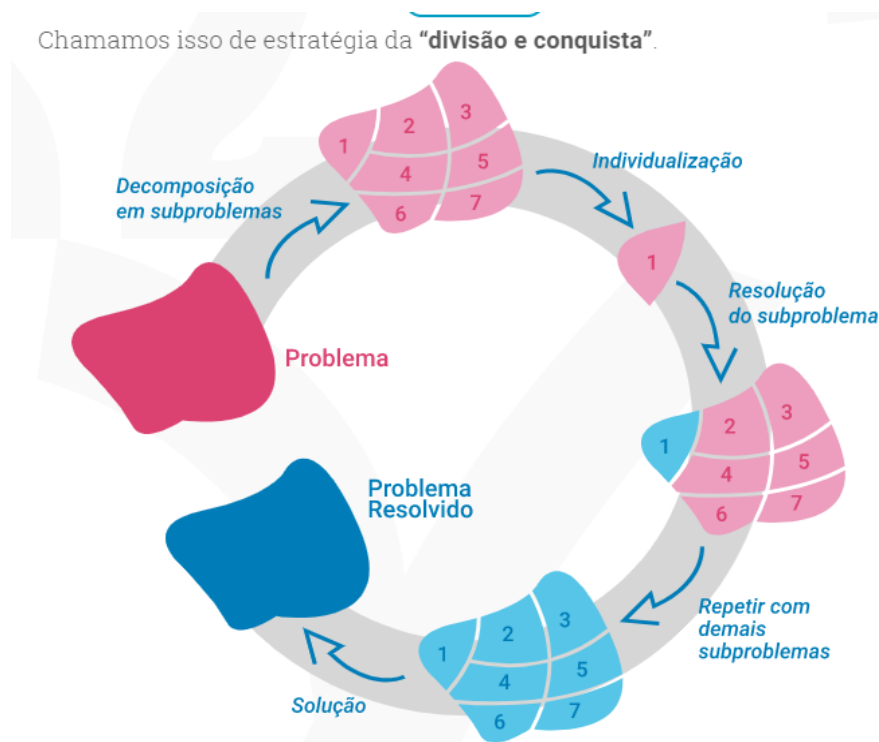
[...] trata-se de quebrar um problema ou sistema complexo em partes menores, que são mais manejáveis e mais fáceis de entender. As partes em menor tamanho podem, então, serem examinadas e resolvidas, ou concebidas individualmente, uma vez que são mais fáceis de trabalhar (BRACKMANN, 2017, p. 37).

Esse pilar pode ser visto nas ideias de Papert (2008) e de Polya (1978) como o princípio ou estratégia conhecida como “dividir para conquistar”. Segundo Papert (2008),

[...] é comum os estudantes falharem ao tentar resolver um problema, porque insistem em resolvê-lo por inteiro, de uma só vez; em muitos casos seria muito mais fácil se reconhecessem que partes do problema podem ser resolvidas separadamente e depois reunidas para lidar com o todo (PAPERT, 2008, p. 90).

Assim, essa estratégia de “dividir para conquistar” surge como uma possibilidade para facilitar o processo de resolução. O problema é decomposto em problemas menores que são resolvidos separadamente com o objetivo de encontrar a solução do principal. Essa separação em problemas menores é caracterizada pela Figura 2.

**Figura 2 - Estratégia "Dividir para Conquistar"**



Fonte: Brasil (2019).

Um processo semelhante é caracterizado por Polya (1978) como “Decomposição e Recombinação”, no qual

[...] examina-se um objeto que desperta o interesse ou provoca curiosidade [...] ou qualquer problema que se queira resolver. Tem-se uma impressão do objeto como um todo, mas esta impressão possivelmente não é bastante definida. Um detalhe sobressai e sobre ele focaliza a atenção. Em seguida, concentra-se num outro detalhe, depois ainda num outro. Diversas combinações de detalhes podem apresentar-se e, um pouco depois, considera-se novamente o objeto como um todo, mas agora ele é visto de maneira diferente. Decompõe-se o todo em suas partes e recombina-se partes num todo mais ou menos diferente (POLYA, 1978, p. 41).

Ribeiro, Foss e Cavalheiro (2020) ressaltam que definir as interfaces de entrada e saída de cada subproblema também é uma característica importante e que essa decomposição “permite que se trabalhe cooperativamente para resolver problemas de forma organizada e eficaz” (RIBEIRO; FOSS; CAVALHEIRO, 2020, p. 25).

Brackmann (2017) também destaca que esta prática pode aumentar a atenção aos detalhes. Para exemplificar, mostra que o funcionamento de uma bicicleta pode ser mais bem entendido após o desmembramento de suas partes, conforme Figura 3, no qual temos a bicicleta completa e seus mecanismos de estrutura, tração e transferência de energia, nesta ordem.

**Figura 3 - Partes da Bicicleta (Decomposição)**



Fonte: Brackmann (2017).

Outro pilar é o Reconhecimento de Padrões, que também está associado ao termo “Generalização” (BRACKMANN, 2017; RIBEIRO; FOSS; CAVALHEIRO, 2020), que é um pilar entendido como “a associação de algum objeto (ou parte dele), tangível ou conceitual, com padrões familiares que permitam identificá-lo e classificá-lo” (BRASIL, 2019). É quando, mesmo que seja uma situação nova, temos ideias de como lidar com ela por reconhecer alguns padrões que são comuns ao que já conhecemos, como quando notamos, pelo padrão das nuvens, que há possibilidade de chuva; se uma fruta está boa ou não pela configuração da casca; a

configuração da prova de um vestibular que tem um padrão no tipo de questões; entre outros. Com isso, “é possível simplificar a solução de problemas e replicar essa solução em cada um dos subproblemas, caso haja semelhança” (VICARI, 2018, p. 33). 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”. Quanto mais padrões são encontrados, “mais dinâmico e rápido a macro solução é encontrada” (BRACKMANN, 2017, p. 37).

Esse processo tem relação com a importância que Polya (1978) traz para a busca de “Problemas Correlatos”, indicando que podemos “[...] procurar um problema intimamente correlato; considere a incógnita ou procure um outro problema que já haja sido resolvido e que esteja relacionado ao problema que se apresenta por generalização, particularização ou analogia” (POLYA, 1978, p. 36).

Nomeando como Generalização, Ribeiro, Foss e Cavalheiro (2020) realçam que é uma técnica que permite a construção de uma solução, como um algoritmo, mais genérica a partir de uma preexistente, permitindo a sua utilização em outros contextos:

Muitas vezes, problemas que à primeira vista parecem totalmente diferentes podem ser solucionados pelo mesmo algoritmo, fazendo apenas pequenas modificações. Programas ou algoritmos são descrições de procedimentos, portanto, podem ser usados como dados para outros programas ou algoritmos (RIBEIRO; FOSS; CAVALHEIRO, 2020, p. 25).

Nessa etapa, algumas perguntas podem servir de apoio, como “este problema é similar a um outro que já tenhas resolvido” ou “em que pontos o atual problema é diferente de outro que já tenhas resolvido?” (VICARI, 2018, p. 32).

Uma forma de exemplificar a utilização do Reconhecimento de Padrões é por meio da análise da Sequência de Fibonacci (Figura 4), no qual o termo  $F_n$  é sempre igual à soma dos dois anteriores, ou seja,  $F_n = F_{n-1} + F_{n-2}$ .

**Figura 4 - Sequência de Fibonacci**

$$\begin{array}{l} F_0 = 0 \\ F_1 = 1 \\ F_2 = F_1 + F_0 \\ F_3 = F_2 + F_1 \\ F_4 = F_3 + F_2 \\ [ \dots ] \\ F_n = F_{(n-1)} + F_{(n-2)} \end{array}$$

Fonte: Brackmann (2017).

Considerando que “um algoritmo é composto por instruções que devem ser executadas de uma forma e na ordem definida para atingir a solução desejada” (RIBEIRO; FOSS; CAVALHEIRO, 2020, p. 24), o pilar definido como Algoritmos é caracterizado pelo desenvolvimento de uma habilidade para criar “uma sequência finita de etapas (passos), cada qual executável em um tempo finito, por um agente computacional, natural (humano) ou sintético (computador)” (BRASIL, 2019). Assim que algoritmos são construídos, “seguirão sempre os mesmos passos pré-definidos, ou seja, podem ser repetidos quantas vezes forem necessários, para a solução de um mesmo problema” (VICARI, 2018, p. 36). De acordo com Ribeiro, Foss e Cavalheiro (2020), do mesmo modo que uma prova é o produto do raciocínio lógico, o algoritmo pode ser considerado o resultado do Pensamento Computacional.

Um processo de refinamentos sucessivos também pode ser realizado para tornar o algoritmo mais rápido ao ignorar detalhes que não são relevantes para o passo a passo criado para se resolver um problema ou para lidar com certa situação. Essa característica se assemelha à estratégia do pilar da Abstração.

### **3.3.1. Pensamento Computacional com a linguagem *Scratch***

Com um viés construcionista, Brennan e Resnick (2012), ao estudarem as atividades desenvolvidas na comunidade *on-line* do *Scratch* e em *workshops* envolvendo o recurso, desenvolveram ideias relacionadas ao Pensamento Computacional que envolvem três dimensões: Conceitos Computacionais, Práticas Computacionais e Perspectivas Computacionais.

Comum em muitas linguagens de programação, os Conceitos Computacionais nos quais os estudantes se envolvem junto ao *Scratch* são, de acordo com Brennan e Resnick (2012), úteis em uma ampla gama de projetos desenvolvidos no *Scratch*. Além disso, eles podem ser transferidos para outros contextos de programação e de não programação. Os conceitos são: sequências, loops, paralelismo, eventos, condicionais, operadores e dados.

As sequências são “uma série de etapas ou instruções individuais que podem ser executadas pelo computador” <sup>17</sup> (BRENNAN; RESNICK, 2012, p. 3, tradução nossa), assim como os algoritmos. Essas sequências são construídas com a união dos diversos blocos de

---

<sup>17</sup> Do inglês: “a series of individual steps or instructions that can be executed by the computer”.



programação presentes no *Scratch*. Quando uma sequência é programada para ser repetida diversas vezes, chamamos de loops. Os eventos são os comandos que fazem com que uma sequência ou várias funcionem, como o pressionamento de uma tecla, o clique do mouse, entre outros. De acordo com Brennan e Resnick (2012), o paralelismo é quando sequências de instruções podem ser executadas ao mesmo tempo, comum na maioria das linguagens de computador modernas. Outro conceito-chave é a capacidade do computador tomar decisões com base em certas condições preestabelecidas/programadas, como uma estrutura de decisão sempre-se-então, chamadas de condicionais. Os operadores funcionam como recursos que fornecem suporte para a utilização ou comparação de expressões matemáticas, lógicas ou *strings* (sequência de caracteres, como letras). Por fim, temos os dados, que no *Scratch* “envolvem armazenamento, recuperação e atualização de valores”<sup>18</sup> (BRENNAN; RESNICK, 2012, p. 6, tradução nossa) por meio de variáveis ou de listas.

As Práticas Educacionais se relacionam diretamente com as ideias de Pensamento Computacional elaboradas por Wing (2006) e se concentram, de acordo com Brennan e Resnick (2012), no processo de pensamento e aprendizagem do estudante. Elas são divididas em quatro conjuntos principais: Ser Incremental e Iterativo, Testar e Depurar, Reutilizar e Remixar e, por último, Abstrair e Modularizar.

Ser Incremental e Iterativo traz a ideia de que o desenvolvimento de um projeto no *Scratch* não é um processo que acontece de forma linear, no qual basta seguir uma sequência de passos para que ele funcione. “É um processo adaptativo, no qual o plano pode mudar em resposta à abordagem de uma solução em pequenos passos”<sup>19</sup> (BRENNAN; RESNICK, 2012, p. 7, tradução nossa). Essa solução em pequenos passos se relaciona com o pilar da Decomposição, no qual a divisão de um problema em subproblemas pode fazer com que a sua solução seja mais facilmente encontrada.

A prática de Reutilizar e Remixar acontece quando os estudantes constroem seus programas em cima do trabalho de outras pessoas, trabalhos esses que estão disponíveis na comunidade *on-line* do *Scratch*. De acordo com Brennan e Resnick (2012), essa prática contribui para o desenvolvimento da criticidade dos estudantes em cima dos códigos de outras pessoas, possibilitando o uso deles ou até o reconhecimento de alguns padrões para certas ações,

---

<sup>18</sup> Do inglês: “Data involves storing, retrieving, and updating values”.

<sup>19</sup> Do inglês: “It is an adaptive process, one in which the plan might change in response to approaching a solution in small steps”.

característica importante do PC. Além disso, os autores declaram que essa prática pode provocar questões importantes referentes à propriedade e à autoria dos programas.

A prática criada para lidar com erros, antecipar problemas e testar os algoritmos construídos no *Scratch* é denominada Teste e Depuração. Esse processo está ligado às ideias construcionistas de Papert (1988), no qual o estudante é encorajado a estudar o erro e a procurar uma solução para tal, ao invés de desistir do que está fazendo.

Por último, a prática de Abstrair e Modular traz a ideia de que a programação dos códigos no *Scratch* pode ser potencializada quando, depois de programarmos algoritmos/seqüências menores por meio da Decomposição, reunimos essas coleções e organizamos conforme acharmos mais importante. Por exemplo, podemos criar uma primeira pilha de códigos que controlam os movimentos, outra que controla a aparência e outra com os vários eventos que acontecem, assim por diante. Essa prática pode ajudar no momento de abstrair as partes mais importantes para a solução de um problema ou até mesmo no processo de depuração.

A última dimensão do Pensamento Computacional desenvolvida por Brennan e Resnick (2012) junto ao *Scratch* é chamada de Perspectivas Computacionais e abrange a evolução dos estudantes na compreensão de si mesmos, de seus relacionamentos com os outros e com o mundo tecnológico pelo qual estão rodeados. Assim, os pesquisadores destacam como os jovens programadores conseguiram melhorar a forma como se expressam, se conectam com outras pessoas e questionam quando possuem dúvidas.

Considerando essas reflexões, buscamos, na próxima seção, discorrer sobre como as escolas estão formando os alunos para lidarem com as demandas do século XXI e qual tipo de estudante que acreditamos que elas deveriam formar, além de como essas características envolvendo o PC e o *Scratch* podem emergir como um meio para isso acontecer.

### **3.4. A escola e o futuro**

Considerando que a construção de um currículo “implica um projeto de sociedade, de ser humano, de cultura” (PONCE, 2018, p. 794), podemos fazer alguns questionamentos com a intenção de orientar esse currículo, como: que tipo de estudantes as escolas estão formando? Que conhecimentos são necessários para que as pessoas consigam viver dignamente no presente

e que possibilitem a construção de um futuro melhor? Ou seja, como gostaríamos que fosse a escola no futuro?

Resnick (2020) divide esses alunos que estão sendo formados nas instituições de ensino em dois grupos: aqueles que as escolas geralmente estão focadas em formar e aqueles em que acredita que as escolas deveriam focar. O primeiro grupo é chamado de A, ou seja, os alunos “nota A”. Esses estudantes são aqueles que são treinados para resolver questões, seguir instruções e normas e são considerados bons alunos se as suas notas em provas e trabalhos forem boas, caso contrário, muitas vezes, são caracterizados como alunos não tão bons. Os estudantes do grupo A são ótimos em resolver problemas que, em sua maioria, estão em livros didáticos ou provas, mas podem não ser tão bons com outros. Segundo Resnick (2020), esses alunos fazem parte de um sistema educacional no qual os objetivos de aprendizagem e as abordagens utilizadas foram basicamente inalterados no último século e no qual devem ser subalternos e reproduzir aquilo que os professores sugerem.

Outro grupo de estudantes, denominado “alunos X”, “são aqueles dispostos a assumir riscos e a experimentar coisas novas; são ávidos por definir os próprios problemas em vez de apenas resolver aqueles nos livros escolares” (RESNICK, 2020, p. 3). Segundo o autor, precisamos de estudantes X, pois precisaremos resolver problemas que ainda não existem, que aparecerão, muitas vezes, em situações que ainda nem imaginamos. Esses jovens que estão sendo formados em nossas instituições de ensino

[...] serão confrontados com situações novas e inesperadas durante toda a vida. Eles precisam aprender a lidar com as incertezas e mudanças usando a criatividade, não só em suas vidas profissionais, mas também nos âmbitos pessoal (como desenvolver e manter amizades em uma era de redes sociais) e cívico (como ter uma participação significativa em comunidades com limites e necessidades em constante mudança) (RESNICK, 2020, p. 4).

Portanto, precisaremos de alunos que, durante o período escolar, desenvolvam habilidades referentes à criatividade, e uma possibilidade é acrescentar a proposta de Aprendizagem Criativa em seu currículo.

Pensar em mudar o currículo que forma “alunos A” e passa a formar “alunos X” pode exigir de educadores o ato de insubordinação criativa (D’AMBROSIO; LOPES, 2015; ROSA, 2019). De acordo com D’Ambrosio e Lopes (2015), essa ação surge como um recurso perante a burocracia educacional, no qual gestores e educadores podem, ao notarem a necessidade de seguir diretrizes diferentes daquelas (im)postas por superiores, desobedecer a ordens visando à melhoria e ao bem-estar da comunidade escolar, mas ainda preservando princípios éticos, morais e de justiça social. Rosa (2019) traz alguns exemplos para pensarmos em como a área

de Educação Matemática também está subordinada e estabelece subordinação em diversos aspectos:

[...] não seria subordinação, o professor forçar o estudante a pensar matematicamente da forma que ele deseja que o estudante pense? Não seria subordinação do professor, ensinar por meio de um método retrogrado, o qual foi a ele imputado na sua formação e no seu entendimento do que é aprender matemática, por toda a vida? Não seria subordinação, não possibilitar liberdade de expressão, de pensamento e de fala aos estudantes porque a direção da escola exige que isso seja feito, uma vez que a escola se enquadra em um modelo tradicional de ensino? (ROSA, 2019, p. 2).

Com isso, Rosa (2019) utiliza a metáfora, baseando-se em Ubiratan D'Ambrosio, de que professores e alunos vivem, muitas vezes, como pássaros em gaiolas, no qual tudo o que conhecem e sentem é o que as grades da gaiola permitem. A insubordinação criativa surge como uma possibilidade para soltar esses “pássaros”, ou seja, permitir que educadores e estudantes não se limitem a viver aquilo que, muitas vezes, é designado a eles por documentos que sugerem uma Educação Matemática que se mantém praticamente a mesma desde o século passado e que não leva em conta os interesses dos sujeitos desse projeto.

Concordando que cabe a nós, educadores, “libertar nossos estudantes-pássaros ao mundo, mundo das ideias, principalmente” (ROSA, 2019, p. 3), observamos que a proposta de Aprendizagem Criativa pode surgir como um meio de “abrir a gaiola com cuidado, sem precisar destruir o objeto, ou mesmo orientar os pássaros de modo que eles mesmos criem possibilidades de serem desengaiolados” (ROSA, 2019, p. 3). Sendo assim, o planejamento das atividades teve como principal suporte teórico a proposta de Aprendizagem Criativa de Resnick (2020), apresentada junto ao planejamento das atividades no próximo capítulo.

## 4 Metodologia de Pesquisa

Neste capítulo, serão apresentados os procedimentos metodológicos que foram adotados neste estudo para que pudéssemos responder à nossa pergunta de pesquisa. Com o intuito de trazer uma melhor organização para este trabalho, dividimos esse capítulo em cinco seções. Na primeira, buscamos justificar a metodologia de produção de dados escolhida devido ao fato da nossa pesquisa possuir um viés qualitativo. Nas Seções 4.2 e 4.3, trazemos, respectivamente, informações sobre os participantes da pesquisa e sobre os recursos utilizados para o desenvolvimento das atividades e para produção de dados. Na Seção 4.4, apresentamos algumas características do planejamento utilizado para os encontros e o principal conjunto de ideias que o sustentou, a proposta de Aprendizagem Criativa. Por fim, na Seção 4.5, apresentamos como os dados produzidos foram transcritos, organizados e analisados, além de quais projetos foram desenvolvidos por cada um dos participantes.

### 4.1. Pesquisa Qualitativa

Quando pensamos em metodologia científica estamos concebendo algo que vai além de regras de como pesquisar, mas que nos auxilie a refletir e que nos propicie “um novo olhar sobre o mundo: um olhar científico, curioso, indagador e criativo” (GOLDENBERG, 2004, p. 11). Podemos diferenciar esses métodos em dois principais tipos: um que procura generalizações e regularidades a partir da análise dos dados, denominado quantitativo, no qual o objetivo está apto a ser mensurável. De acordo com Bicudo (2018), este método

[...] carrega consigo as noções próprias ao paradigma positivista, destaca como pontos importantes para a produção da ciência a razão, a objetividade, o método, a definição de conceitos, a construção de instrumentos para garantir a objetividade da pesquisa (BICUCO, 2018, p. 115).

Já o método qualitativo “engloba a ideia do subjetivo, passível de expor sensações e opiniões” (BICUDO, 2018, p. 116) e “visa à compreensão interpretativa das experiências dos indivíduos dentro do contexto em que foram vivenciadas” (GOLDENBERG, 2004, p. 19). Além disso, nesta abordagem

[...] privilegiam-se descrições de experiências, relatos de compreensões, respostas abertas a questionários, entrevista com sujeitos, relatos de observações e outros procedimentos que deem conta de dados sensíveis, de concepções, de estados mentais, de acontecimentos etc. O *rationale* subjacente a esse modo de pesquisar é dado pela intenção de atingir aspectos do humano sem passar pelos crivos da mensuração sem partir de método previamente definido e, portanto, sem ficar preso a quantificadores e aos cálculos recorrentes (BICUDO, 2018, p. 117).

De acordo com Goldenberg (2004), os pesquisadores que assumem a abordagem qualitativa em seus estudos não defendem um modelo único de pesquisa para todos os campos e não aceitam que seus resultados se transformem em leis ou explicações gerais. Sendo assim, “os métodos qualitativos enfatizam as particularidades de um fenômeno em termos de seu significado para o grupo pesquisa” (GOLDENBERG, 2004, p. 49-50).

Ainda segundo Goldenberg (2004), outra característica da pesquisa qualitativa é que não existe regra para o tempo adequado de pesquisa, podendo durar algumas semanas ou até muitos anos. Além disso,

[...] o pesquisador deve estar preparado para lidar com uma grande variedade de problemas teóricos e com descobertas inesperadas, e, também, para reorientar seu estudo. É muito frequente que surjam novos problemas que não foram previstos no início da pesquisa e que se tornam mais relevantes do que as questões iniciais (GOLDENBERG, 2004, p. 34-35).

Consideramos que essa pesquisa possui um caráter qualitativo, pois se relaciona com algumas das características que Goldenberg (2004), Bicudo (2018), Bogdan e Biklen (1994) e Borba e Araújo (2018) definem para esse campo. Entre elas, a não preocupação em fixar leis para produzir generalizações, já que neste método procura-se “observar, diretamente, como cada indivíduo, grupo ou instituição experimenta, concretamente, a realidade pesquisada” (GOLDENBERG, 2004, p. 63). Também destacamos que “a fonte direta de dados é o ambiente natural, constituindo o investigador o instrumento principal” (BOGDAN; BIKLEN, 1994, p. 47) devido ao fato de que a produção de dados foi realizada pelo contato do pesquisador com os participantes da pesquisa de forma remota nos encontros via *Google Meet*, produzindo esses dados que, em grande parte, “consistem em descrições detalhadas de situações com o objetivo de compreender os indivíduos em seus próprios termos” (GOLDENBERG, 2004, p. 53). Bogdan e Biklen (1994) apontam que a investigação qualitativa é descritiva, justificando assim os nossos dados que foram produzidos por meio das transcrições dos encontros gravados, notas de campo, capturas de tela, documentos que registraram as atividades realizadas pelos alunos, entre outros.

Foram escolhidos diferentes procedimentos para produção de dados como uma forma de dar mais credibilidade à pesquisa, pois assim teríamos maiores possibilidades de “abranger a máxima amplitude na descrição, explicação e compreensão do objeto de estudo” (GOLDENBERG, 2004, p. 63). Esse método, denominado triangulação (BORBA; ARAÚJO, 2018; GOLDENBERG, 2004), destaca a importância da utilização de vários e distintos procedimentos para a obtenção de dados. Além disso, considerando que “os investigadores qualitativos se interessam mais pelo processo do que simplesmente pelos resultados ou

produtos” (BOGDAN; BIKLEN, 1994, p. 49), os diferentes tipos de dados nos oportunizaram analisar parte do processo pelo qual os estudantes passaram durante o período em que ocorreu a produção: de 12 de agosto de 2020 até 23 de setembro do mesmo ano, totalizando sete encontros via *Google Meet*.

Por fim, concordando com Goldenberg (2014, p. 13), “nenhuma pesquisa é totalmente controlável, com início, meio e fim previsíveis” e as pesquisas até são passíveis de uma reorientação do estudo, pois “é muito frequente que surjam novos problemas que não foram previstos no início da pesquisa e que se tornam mais relevantes do que as questões iniciais” (GOLDENBERG, 2004, p. 34-35).

## 4.2. Participantes da Pesquisa

Com o surgimento da pandemia do novo Coronavírus (SARS-CoV-2), a proibição de encontros presenciais e a oferta das Aulas Remotas (ou Ensino Remoto Emergencial), tivemos a ideia de reorientar parte da produção de dados para um projeto de programação que seria ofertado para todos os alunos da Escola Estadual de Ensino Fundamental Ayrton Senna do Brasil<sup>20</sup>, instituição localizada no município de Novo Hamburgo, no Rio Grande do Sul, onde o autor desta pesquisa possui vínculo empregatício desde 2018. Os encontros seriam virtuais via *Google Meet* e aconteceriam nos mesmos horários das atividades do projeto de Xadrez que já era desenvolvido na instituição. A ideia foi acordada com a diretora e com a supervisora e, a partir disso, começamos a construir um material que serviria de divulgação do projeto aos estudantes. Assim, o planejamento das atividades envolveu a parte inicial de divulgação e o convite, além dos cinco encontros, que depois foram reorganizados em sete, pensados para a produção dos dados que seriam posteriormente analisados.

Elaboramos um panfleto, conforme pode ser visto na Figura 5, uma mensagem para ser enviada nos grupos de *WhatsApp* e um site<sup>21</sup> no qual pudéssemos divulgar o projeto e suas especificidades para os possíveis interessados e seus responsáveis. Esses materiais foram enviados no dia 5 de agosto de 2020, uma semana antes da data de início do projeto.

---

<sup>20</sup> A carta de pedido de autorização à escola consta como Apêndice I.

<sup>21</sup> Site do projeto: <https://sites.google.com/view/projeto-scratch-ayrton-senna>.

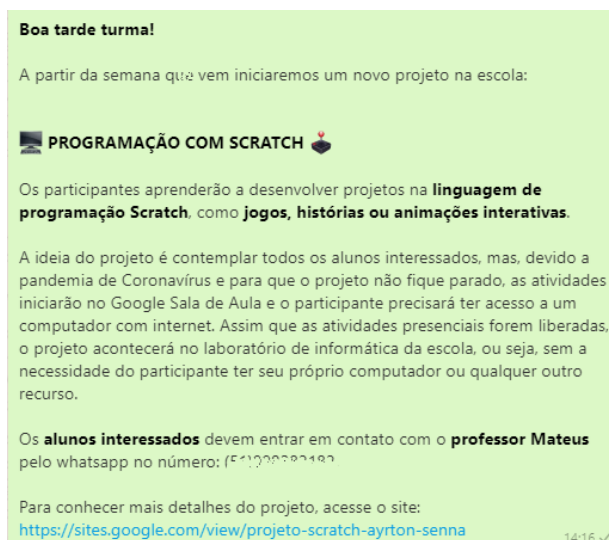
Figura 5 - Panfleto de divulgação do projeto intitulado Programação com Scratch



Fonte: Dados da Pesquisa (2020).

Com o objetivo de que o projeto precisaria oportunizar que todos os estudantes pudessem participar e, devido ao contexto da pandemia do novo Coronavírus, decidimos iniciar o curso de forma remota, buscando estar em consonância com a prática que já vinha sendo utilizada pela escola. Por isso, ao enviarmos o panfleto com o convite para o projeto nos grupos das turmas, disponíveis no *WhatsApp*, também enviamos a mensagem que pode ser vista na Figura 6.

Figura 6- Mensagem enviada via *WhatsApp* para os grupos das turmas da escola



Fonte: Dados da Pesquisa (2020).



O site elaborado para a apresentação do projeto continha informações referentes aos horários do projeto, à pesquisa do professor, ao ambiente de programação *Scratch*, programas desenvolvidos pelo professor com o intuito dos alunos experimentarem algumas possibilidades de construção no *Scratch*, um guia explicando características da interface do ambiente e um espaço destinado ao compartilhamento dos projetos construídos pelos participantes após a produção de dados.

Foram quatro os participantes que aceitaram o convite para participar das atividades de forma remota, entretanto, outros seis entraram em contato conosco para participarem quando o projeto acontecesse no laboratório de informática da escola, o que não ocorreu, pois, devido à pandemia, as restrições de atividades presenciais continuaram até após o término do projeto. Para que pudéssemos produzir e analisar os dados desses participantes, os seguintes cuidados éticos foram tomados: a manifestação por escrito, no dia 4 de agosto de 2020, pela responsável da escola para a realização das atividades, conforme o Apêndice I, e o consentimento dos alunos e dos responsáveis registrados no Termo de Assentimento Livre e Esclarecido e no Termo de Consentimento Livre e Esclarecido, que foram disponibilizados de forma *on-line* por meio do recurso *Google Forms* no primeiro encontro, dia 12 de agosto de 2020. Esses documentos estão disponíveis, respectivamente, como Apêndice II e III. Os participantes da pesquisa não tiveram seus nomes citados em nenhum momento, mas foram identificados por nomes fictícios nas transcrições dos momentos nos quais a produção dos dados estava acontecendo. Os quatro participantes eram três estudantes do nono ano (Miguel, Eduardo e Paulo), que já eram alunos do autor desta pesquisa, e uma do sétimo ano, que não era (Maria), ou seja, todos eram dos anos finais do ensino fundamental.

### **4.3. Recursos Utilizados para a produção de dados**

Devido ao fato de as atividades terem acontecido de forma remota, dividimos os recursos utilizados em dois grupos: primeiro apresentaremos os programas que foram utilizados para a comunicação e para o desenvolvimento dos encontros juntos aos participantes: *WhatsApp* e *Google Meet*. No segundo, que abordaremos com mais detalhes neste trabalho, apresentaremos o *Scratch*, um ambiente de programação visual que funciona por meio de blocos, e o *Machine Learning For Kids*, site que possibilita o treino de algoritmos via Aprendizado de Máquina e que permite adicioná-los à sua própria versão do *Scratch*. Além

disso, também apresentaremos o *Pixilart*<sup>22</sup>, recurso que foi utilizado por alguns estudantes para a construção de atores e cenários posteriormente adicionados ao *Scratch*.

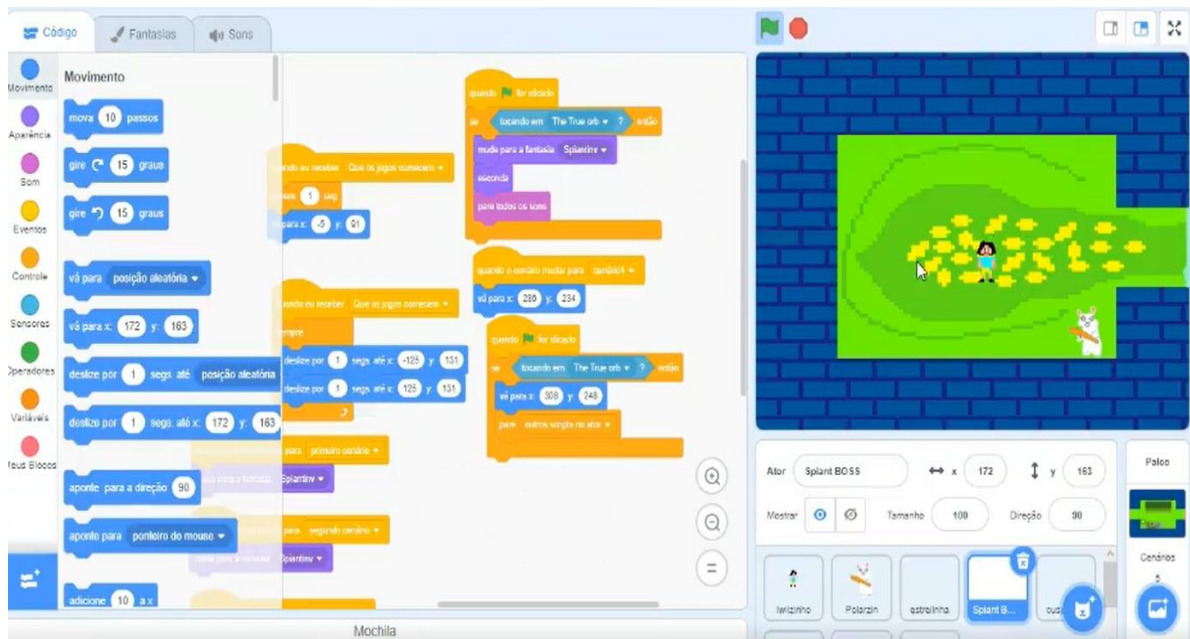
Os dois recursos utilizados para a comunicação entre o professor e os participantes da pesquisa foram o aplicativo de mensagens instantâneas *WhatsApp* e o recurso de videochamadas *Google Meet*. Utilizamos o *WhatsApp* para a divulgação do projeto que envolvia as aulas de programação e, após a demonstração de interesse dos alunos que tinham a possibilidade de trabalhar com os outros recursos via atividades *on-line*, criamos um grupo no aplicativo denominado “Projeto *Scratch – online*”, no qual os estudantes foram adicionados. Este grupo trouxe benefícios para a aprendizagem dos alunos e para a produção de dados, pois era um espaço onde eles se comunicavam a respeito dos projetos que estavam desenvolvendo.

Outro recurso destinado à comunicação e produção de dados foi o *Google Meet*, que possibilita a realização de videochamadas instantâneas pelo *smartphone*, *tablet* ou computador. Além disso, possui um *chat* para comunicação via mensagens digitadas durante as chamadas, além do uso de áudio e vídeo dos participantes. Outra característica importante é a opção “Apresentar Agora”, que nos possibilita, além do áudio e vídeo do participante, o compartilhamento da tela ou de alguma janela aberta do recurso o qual o usuário está utilizando. Esse compartilhamento de tela foi utilizado para que os participantes da pesquisa, tanto o professor quanto os estudantes, pudessem mostrar os projetos que estavam desenvolvendo, como pode ser visto na Figura 7. Como a pesquisa foi realizada com os alunos em um projeto de uma escola pública do Rio Grande do Sul, todos possuíam, inclusive o professor, um domínio de e-mail institucional, ou seja, vinculado à Secretaria de Educação do Estado. Graças a esse domínio, conseguimos utilizar uma versão do *Google Meet* que possibilitava a gravação de imagem, áudio, vídeo e texto de todos os encontros, dados que foram produzidos e se constituíram como a nossa principal fonte de análise.

---

<sup>22</sup> Disponível em: <https://www.pixilart.com/>.

**Figura 7 - Compartilhamento da tela do *Scratch* pelo aluno Eduardo**



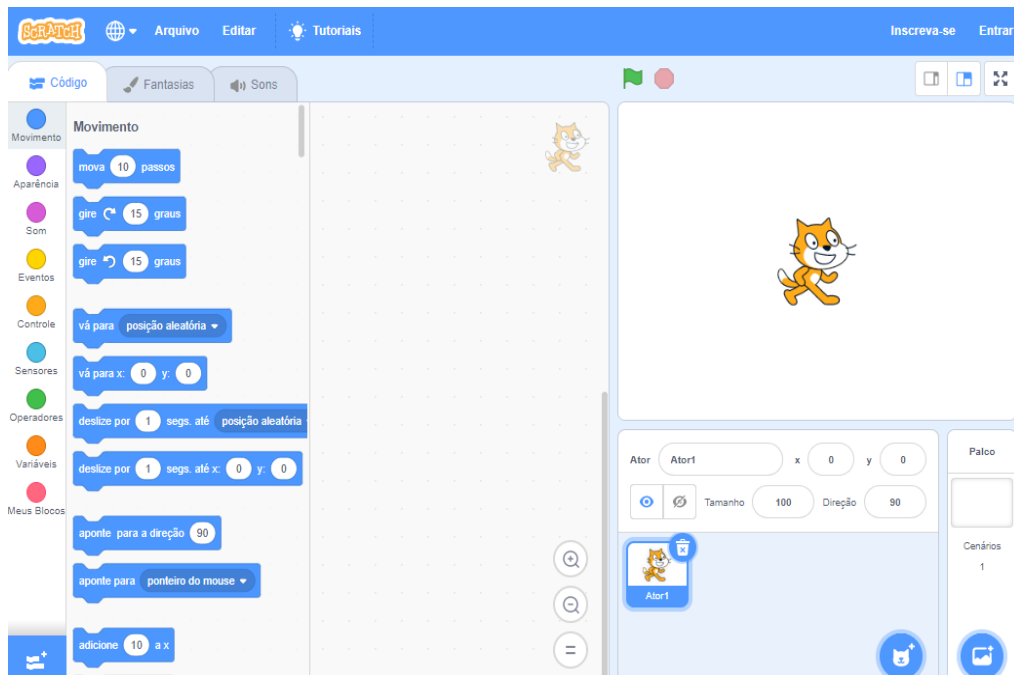
Fonte: Dados da Pesquisa (2020).

Utilizamos o *Scratch 3*, versão mais recente disponibilizada no site, disponível na versão *off-line* ao ser instalado no computador.

Ao acessar o site oficial do *Scratch*, temos uma mensagem sobre algumas possibilidades de criações e compartilhamento com esse ambiente de programação. Na aba superior, temos sete espaços clicáveis, que são: *Scratch*, Criar, Explorar, Ideias, Sobre, Busca, Inscreva-se e Entrar.

Ao clicar em *Scratch*, o site é atualizado e somos enviados à página de abertura. Em Criar, somos levados à interface da linguagem de programação do *Scratch* na sua versão 3, apresentada na Figura 8.

**Figura 8 - Interface do Scratch versão 3**



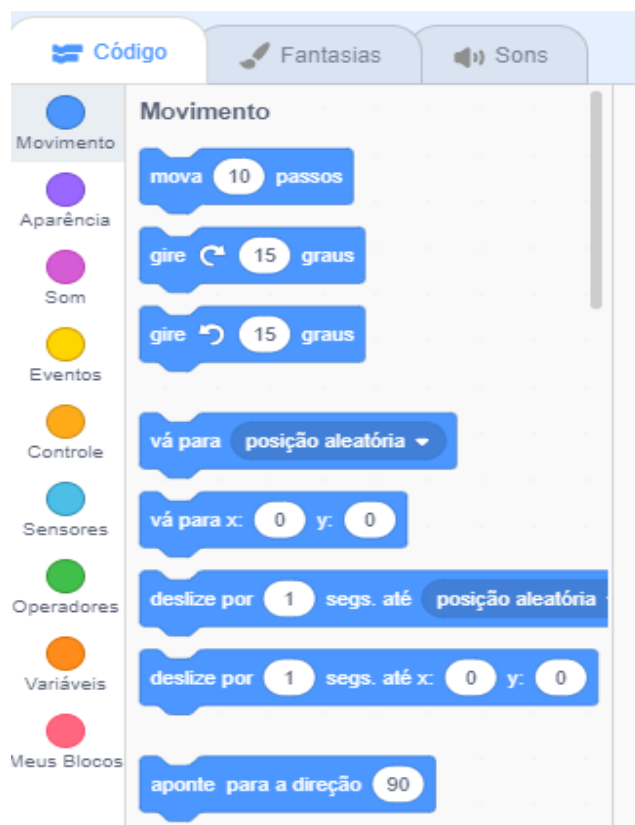
Fonte: Elaborada pelo autor.

A barra superior azul pode ser acessada para selecionar o idioma (são mais de 40 disponíveis), criar um arquivo de projeto, carregar um projeto do computador, fazer o *download* do projeto que está sendo programado para o computador, ligar o modo turbo e acessar diversos tutoriais disponibilizados pela própria equipe do *Scratch*, os quais também podem ser encontrados ao clicar em Ideias na página inicial do site.

Podemos dividir a interface da linguagem de programação do *Scratch* em quatro áreas principais: a região dos códigos com os blocos de programação separados por categorias de cores diferentes, a parte central, onde o *script* do programa é montado, o palco, que é a interface do programa que está sendo desenvolvido, e a área de seleção de atores e planos de fundo.

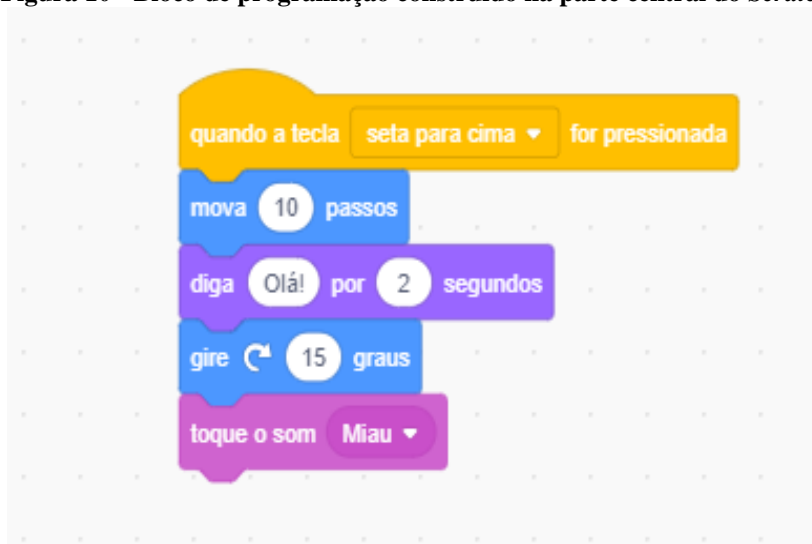
Em praticamente todo o canto esquerdo há os códigos de programação, conforme a Figura 9. Eles são divididos em nove categorias: Movimento, Aparência, Som, Eventos, Controle, Sensores, Operadores, Variáveis e Meus Blocos, mas outros tipos podem ser adicionados mediante *download*, como códigos para a detecção de vídeo, traduções, função caneta, entre outros. Os blocos disponíveis em cada uma dessas categorias podem ser selecionados e arrastados para a parte central do *software*, os quais, ao serem encaixados e formarem blocos de programação, fazem parte da construção do *script* do programa, como pode ser visto na Figura 10.

**Figura 9 - Categorias de códigos de programação do Scratch**



Fonte: Elaborada pelo autor.

**Figura 10 - Bloco de programação construído na parte central do Scratch**

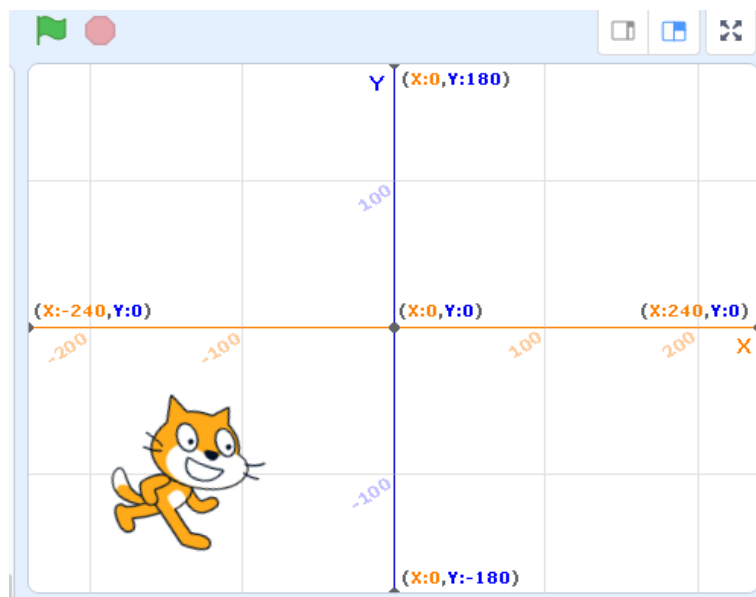


Fonte: Elaborada pelo autor.

No canto direito há o palco e os atores, ou seja, a interface do programa que está sendo construído. O palco é onde ocorrem as ações programadas na parte central, há vários cenários prontos para escolha que servem como fundo para o palco, mas todos possuem as mesmas dimensões que podem ser vistas como um Plano Cartesiano, onde o eixo X varia de -240 a 240

e o eixo Y de -180 a 180, conforme Figura 11. Também é possível enviar outras imagens e adicioná-las como plano de fundo para o palco.

**Figura 11 - Palco ou Interface do programa com seus respectivos eixos coordenados**



Fonte: Elaborada pelo autor.

Por fim, apresentaremos a área para a seleção dos atores e cenários. Podem-se utilizar os personagens (chamados de atores) e os cenários já existentes no programa, desenhar na hora com o recurso de edição do próprio programa ou fazer o envio de imagens ou *GIF (Graphics Interchange Format)* prontos. Assim que um ator ou cenário é adicionado ao programa, pode-se modificar seu nome, sua posição pelas coordenadas cartesianas, seu tamanho e sua direção, conforme visto na Figura 12. Além disso, cada personagem ou cenário tem sua área central própria para que os *scripts* a serem construídos provoquem ações apenas nos que o autor do projeto tiver interesse, ou seja, cada ator e cenário possui o seu próprio *script*.

**Figura 12 - Área para selecionar ou enviar atores e cenários**



Fonte: Elaborada pelo autor.

Ainda na página principal do site, ao clicar em Explorar, pode-se navegar pelo local onde todos os projetos que foram compartilhados pelos usuários do ambiente, chamados de

*Scratchers*, estão disponíveis. Ao clicar em um desses projetos, temos acesso ao nome/*nickname* do usuário que criou, ao nome do projeto, às instruções divulgadas pelo próprio usuário e às informações sobre notas e créditos do programa. Além disso, é possível ver o interior desses trabalhos compartilhados, ou seja, o *script* do programa, e fazer algo que o *Scratch* denomina *Remix*, ação na qual é feita uma cópia de todo o *script* do programa que foi compartilhado para o usuário que está acessando, incluindo códigos, imagens e sons. Também é possível procurar projetos específicos digitando o termo ou expressão que deseja pesquisar na área Buscar, criar uma conta e tornar-se um usuário oficial do *Scratch* clicando em Inscreva-se, acessar a sua conta clicando em Entrar e visualizar informações sobre os desenvolvedores e os objetivos do *Scratch* ao clicar em Sobre.

A interface gráfica, a programação por blocos, a capacidade de se trabalhar com o treinamento de algoritmos via Aprendizado de Máquina que será apresentada posteriormente e a possibilidade de compartilhar e acessar projetos de outros programadores foram pontos que consideramos importantes para a escolha do *Scratch* como o recurso digital que foi utilizado. Além disso, consideramos que o *Scratch* pode ajudar o usuário “a pensar de forma criativa, a raciocinar sistematicamente a trabalhar colaborativamente – competências essenciais à vida do século XXI” (LKG, 2018). Assim como Papert (1988, 2008) via o *LOGO*, vemos o *Scratch* fazendo parte de um micromundo, ou seja, um “ambiente onde falso ou verdadeiro, certo ou errado não são critérios decisivos” (PAPERT, 1988, p. 163), um local onde o estudante possa verdadeiramente usar, pensar e até brincar com a matemática. Aqui, o micromundo elaborado é visto como “águas férteis” e o *Scratch* como sendo uma “vara de pescar” (PAPERT, 2008) na qual o estudante possa descobrir e construir, por si mesmo, os conhecimentos que precisa.

O outro recurso tecnológico utilizado foi o *Machine Learning For Kids*, site que possui um recurso *on-line* de Inteligência Artificial que tem como objetivo introduzir conceitos e algoritmos de Aprendizado de Máquina em salas de aula ou em clubes de programação voltados para crianças. Ele treina modelos que são capazes de classificar textos, números, de reconhecer imagens ou sons, os quais podem ser adicionados e utilizados em versões do *Scratch* ou de outras duas linguagens de programação disponíveis no próprio site, como *Python* ou *App Inventor*.

Os projetos para o reconhecimento de textos podem ser criados sem a necessidade de um cadastro, utilizando a aba “teste sem registrar”. Clicando em “Primeiros passos” na página inicial, o recurso te guia para a criação do projeto e para a importação para a linguagem de programação que deseja, conforme Figura 13.

Figura 13- Página Inicial do recurso on-line Machine Learning for Kids

Sobre Fichas de Trabalho Noticias Ajuda Entrar

# Ensine um computador a jogar um jogo

- 1 Colete exemplos de coisas que você quer ser capaz de reconhecer
- 2 Use os exemplos para treinar o computador a ser capaz de reconhecê-los
- 3 Crie um jogo no Scratch que usa a capacidade do computador para reconhecer novos exemplos

Primeiros passos Saiba mais

Fonte: Elaborada pelo autor.

Por exemplo, para criar um modelo de um conjunto de dados em formato de texto que deve ser treinado por Aprendizado de Máquina, o programa possibilita a criação de listas rotuladas, na qual um conjunto de no mínimo cinco palavras são reconhecidas como pertencentes a cada lista, como pode ser visto na Figura 14. Esse modelo é treinado para que o computador entenda, mesmo que não seja escrito exatamente igual, a qual conjunto a palavra pertence e que tome a devida ação programada.

Figura 14 - Listas de palavras para serem treinadas com Aprendizado de Máquina

Reconhecendo **text** como **rotulo1, rotulo2 or 2 other classes**

+ Voltar para o projeto

+ Adicionar um novo rótulo

**rotulo1**

palavra1 palavras 1 frases 1

texto 1 textos 1

+ Adicionar exemplos

**rotulo2**

palavra2 palavras 2 frases 2

texto 2 textos 2

+ Adicionar exemplos

**rotulo3**

palavra3 palavras 3 frases 3

texto 3 textos 3

+ Adicionar exemplos

**rotulo4**

palavra4 palavras 4 frases 4

textos 4 texto 4

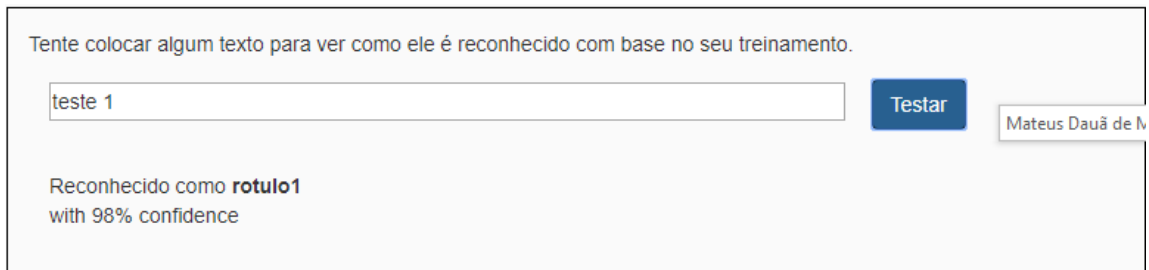
+ Adicionar exemplos

Fonte: Elaborada pelo autor.

Após treinar o modelo, o que é feito apenas com um clique e que pode demorar alguns minutos, uma caixa de testes aparece com a opção de testar o modelo antes de aplicá-lo ao programa, no qual o *input* dado pelo usuário é uma palavra ou frase escrita e o *output* gerado pelo programa é o rótulo pelo qual o modelo treinado a reconhece junto ao grau de confiança do modelo, como mostrado na Figura 15.



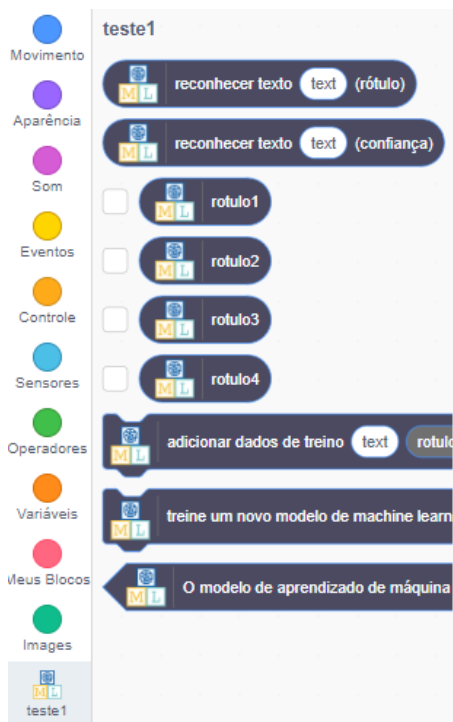
**Figura 15 - Caixa de texto para testar o modelo**



Fonte: Elaborada pelo autor.

Por último, o modelo pode ser importado para uma versão do *Scratch* ou para outras linguagens dentro do próprio site e aparece, como pode ser visto na Figura 16, como um novo bloco de códigos.

**Figura 16 – Bloco de códigos gerado no *Scratch* a partir do modelo treinado no site Machine Learning for Kids**



Fonte: Elaborada pelo autor.

Esse novo conjunto de códigos pode ser utilizado para classificar os novos dados que serão adicionados (textos, imagens ou sons) em categorias que são escolhidas pelo programador.

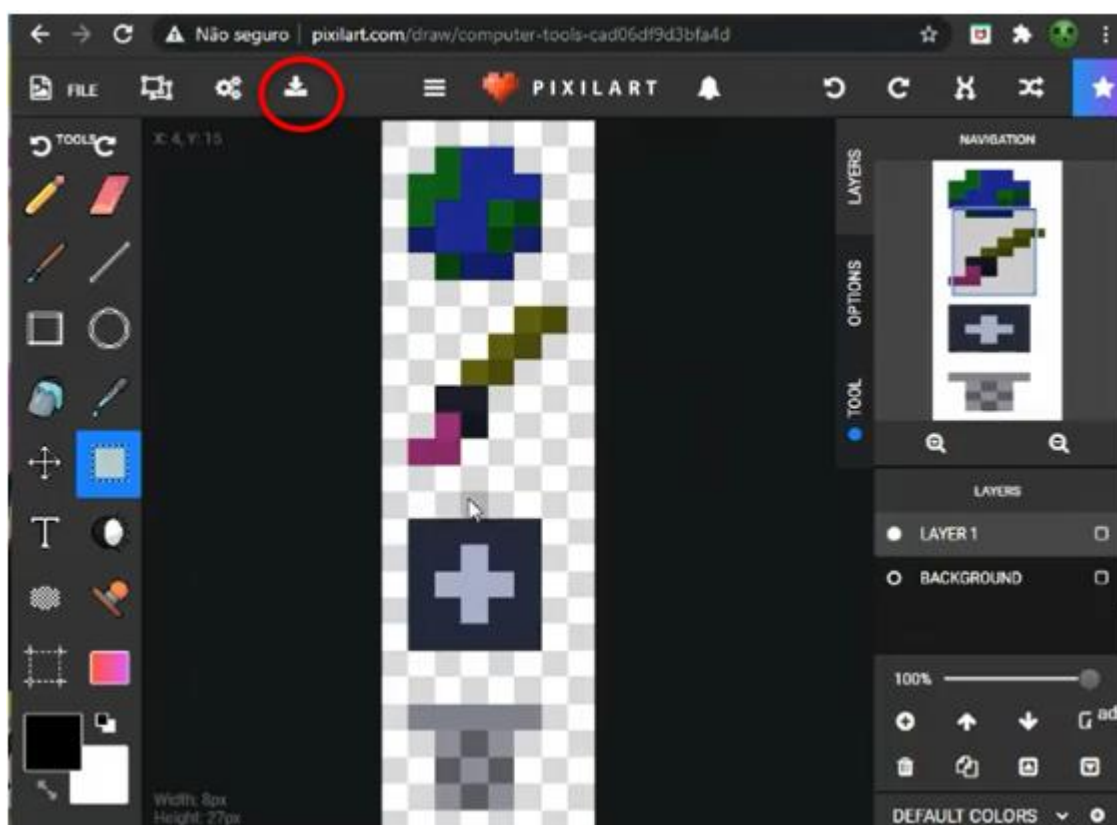
Por fim, o recurso que foi utilizado por alguns participantes para a criação de atores e cenários que, posteriormente, foram adicionados ao *Scratch* foi o *Pixilart*. Definido como uma “plataforma social para todas as atividades”<sup>23</sup> (PIXILART, 2021, tradução nossa), o *Pixilart* é

---

<sup>23</sup> Do inglês: “social platform for all ages”.

um site que possibilita a criação de desenhos a partir da pintura de pixels que são chamadas de *pixel art*. Essas pinturas podem ser compartilhadas na própria plataforma ou utilizadas junto de outros recursos a partir do *download* delas. Na Figura 17, podemos ver a interface do programa quando Miguel estava desenhando alguns atores para um dos seus projetos. Na parte esquerda da tela temos as *tools*, ferramentas utilizadas para desenhar, pintar, apagar ou inserir alguns recursos nos desenhos que são construídos na área central. Na área à direita, temos a *navigation*, região onde o estudante consegue visualizar como os seus desenhos ficarão quando compartilhados ou baixados para o computador. Após a criação do *pixel art*, o estudante pode fazer *download* clicando no ícone circulado em vermelho na Figura 17 e adicioná-lo ao *Scratch* como um ator, uma fantasia ou um cenário.

Figura 17- Interface do Pixilart



Fonte: Dados da Pesquisa (2020).

Na próxima subseção, apresentamos o planejamento das atividades que aconteceram de forma remota entre agosto e setembro de 2020.

#### 4.4. Planejamento das Atividades

As atividades desenvolvidas no projeto “Programação com *Scratch*” foram planejadas para acontecer em cinco encontros via *Google Meet*, com duração de 1 hora e 30 minutos cada, entretanto, durante a aplicação do planejamento, eles foram reorganizados em sete, conforme Apêndice III. Elas foram organizadas com base na proposta de ensino do professor Mitchel Resnick do *MIT Media Lab* que, apesar de ser recente, é, segundo ele, sustentada por várias correntes educacionais que não são de agora, como o Construcionismo de Papert. Sendo assim, iniciaremos essa seção apresentando a proposta e seguindo, logo após, com o nosso planejamento.

##### 4.4.1. A proposta de Aprendizagem Criativa e a sua relação com o *Scratch*

A Aprendizagem Criativa foi baseada no modelo de Jardim de Infância criado por Froebel, no qual a escola “passou de um modelo educacional baseado na transmissão para um modelo interativo, dando às crianças oportunidades de interagir com brinquedos, materiais de artesanato e outros objetos” (RESNICK, 2020, p. 7). O autor destaca que o modelo Jardim de Infância possibilita que as crianças passem por um processo criativo potencializado pela Espiral de Aprendizagem Criativa (Figura 18) e, ao percorrerem essa espiral, “elas desenvolvem e refinam suas habilidades como pensadoras criativas, aprendem a desenvolver as próprias ideias, testá-las, experimentar alternativas, obter as opiniões de outras pessoas e criar ideias baseadas em suas experiências” (RESNICK, 2020, p 12).

Figura 18 - A Espiral da Aprendizagem Criativa



Fonte: Resnick (2020, p. 11).

A Espiral da Aprendizagem Criativa pode ser vista como um conjunto de ações que os estudantes podem seguir para desenvolver habilidades referentes à criatividade. Primeiro é preciso que eles imaginem e, para que isso ocorra, precisamos dar espaço e deixar que a curiosidade deles cumpra esse papel. Mesmo assim, isso não impede que o professor nutra a sua criatividade para a imaginação, pois “todas as crianças nascem com a capacidade de ser criativas, mas essa criatividade não se desenvolverá, necessariamente, sozinha. Ela precisa ser nutrida, incentivada, apoiada” (RESNICK, 2020, p. 19).

O professor pode incentivá-los com ideias, exemplos, histórias ou até mesmo com a criação de um espaço que potencialize esse desenvolvimento. A partir dessa imaginação incentivada pelo professor, eles podem criar ou dar forma ao que foi pensado, ou algo semelhante, transformando as ideias em ações, e para isso alguns recursos podem ser necessários.

Será que essa criação ficou realmente como foi imaginado? Para ter certeza, é preciso testá-la e validá-la, ou, com a perspectiva de estar em um jardim de infância, brincar com ela. Depois disso, o estudante é convidado a compartilhar a sua ideia com outras pessoas, possibilitando o surgimento de outras ideias com esse compartilhamento. Essa construção feita a partir da imaginação, da experimentação ao brincar e da união de ideias com outras pessoas pode ou não sair como o esperado, por isso a importância de os estudantes refletirem sobre tudo

o que já foi feito e, a partir disso, continuarem imaginando e desenvolvendo a sua criatividade ao percorrerem a espiral.

Entretanto, ressaltamos que as ações sugeridas pela Espiral da Aprendizagem Criativa (RESNICK, 2020) não foram seguidas neste trabalho em um movimento de espiral, no qual as atividades deveriam progredir sempre em uma mesma ordem: Imaginar, Criar, Brincar, Compartilhar, Refletir e assim repetidamente. Resolvemos não utilizá-la como um modelo do que precisávamos fazer, mas de modo que as ações sugeridas fossem praticadas durante as atividades remotas síncronas conforme a necessidade do que tínhamos planejado. Além disso, ressaltamos a importância de que “[...] o/a professor/a oportunize o pensar do/a estudante e, para isso, não deve entregar ao/à estudante tudo pronto, para que ele/a apenas reproduza ou possa fazer o que o/a professor/a está limitando” (ROSA; DANTAS, 2020, p. 19). Sendo assim, observamos o cuidado que se deve ter para que a utilização de um conjunto de passos, como a Espiral da Aprendizagem Criativa, não tire a liberdade de criar dos estudantes e limite a sua criatividade.

Na perspectiva de Resnick (2020, p. 19), “a criatividade é desenvolvida a partir de um determinado tipo de esforço, que combina a exploração curiosa com a experimentação lúdica e a investigação sistemática”. São quatro os princípios orientadores que sustentam essa abordagem orientada para que os estudantes realizem as ações da Espiral da Aprendizagem Criativa e esses princípios são conhecidos, em inglês, como os quatro P: *Projects, Passion, Peers and Play*<sup>24</sup>, apresentados, em português, na Figura 19.

**Figura 19 - Os 4 princípios da Aprendizagem Criativa**



Fonte: Adaptado de Resnick (2020).

O primeiro princípio traz a ideia de que “as pessoas aprendem novas ideias, habilidades e estratégias enquanto trabalham em projetos que têm significado pessoal para elas”

---

<sup>24</sup> Nesta pesquisa, os quatro P foram traduzidos como Projetos, Paixão, Pares e Brincar.

(RESNICK, 2020, p. 32). Essa característica está de acordo com a Dimensão Sintônica do Construcionismo (PAPERT, 1986), que realça a importância de que o aluno se envolva com atividades que estejam em sintonia com seus interesses.

Ao trabalharem com projetos, os estudantes podem deixar de ser receptores passivos do saber, pois se envolvem em processos de ensino e de aprendizagem nos quais eles são construtores dos próprios conhecimentos, deixando de lado aquilo que Freire (2018) caracteriza como Educação Bancária, na qual

[...] em lugar de comunicar-se, o educador faz ‘comunicados’ e depósitos que os educandos, meras incidências, recebem pacientemente, memorizam e repetem [...] a única margem de ação que se oferece aos educandos é a de receberem os depósitos, guardá-los e arquivá-los (FREIRE, 2018, p. 80-81).

Este processo da elaboração de projetos para a construção de conhecimentos a partir da criação dos próprios estudantes é sustentado pelo Construcionismo de Papert (1988, 2008). Algumas perspectivas educacionais já trazem características da teoria construcionista sustentada por projetos para dentro das escolas, umas delas é o movimento *maker*, no qual acredita-se que “as experiências de aprendizagem mais valiosas ocorrem quando você está ativamente envolvido no desenvolvimento, na construção ou na criação de algo — quando você aprende criando” (RESNICK, 2020, p. 34).

Além disso, é importante que a metodologia utilizada pelo educador para que seus estudantes trabalhem com projetos ofereça “mais estrutura ou suporte no início de um projeto, mas sem restringir os estudantes a buscarem os próprios interesses, ideias e metas ao longo do tempo” (RESNICK, 2020, p. 75), podendo ser caracterizado como um “início fechado”. O educador começa orientando os estudantes para uma direção, mas com um “final aberto”, possibilitando que os alunos trabalhem diretamente com seus interesses. A importância de um “final aberto”, possibilitando que os estudantes se envolvam com projetos de seus interesses, também é reconhecida por Fagundes, Sato e Maçada (1999, p. 16), pois os autores consideram “fundamental que a questão a ser pesquisada parta da curiosidade, das indagações do aluno, ou dos alunos, e não imposta pelo professor”.

O ambiente de programação *Scratch* tem como foco o desenvolvimento de projetos envolvendo a construção de jogos, animações, histórias interativas e outras possibilidades, sendo considerado até como “uma versão da linguagem de programação Logo, de Seymour Papert, do século XXI” (RESNICK, 2020, p. 41). Segundo o autor, as atividades desenvolvidas pelo *MIT Media Lab* com o *Scratch* incentivam os participantes a imaginarem e a terem ideias, depois a transformarem essas ideias em projetos que serão construídos no *Scratch* e poderão ser compartilhados com outras pessoas pela internet.

Trabalhar com projetos pode não ser algo fácil, muitos problemas podem surgir no desenvolvimento, muitas horas de dedicação podem ser necessárias e a não relevância no que está sendo construído pode implicar na desistência do estudante. Sendo assim, o segundo princípio está relacionado aos interesses do estudante, a sua Paixão em fazer algo: “A paixão é o combustível que impulsiona o ciclo de imersão-reflexão, e isso se aplica aos estudantes de todas as idades [...] A única maneira de persistir e perseverar frente a todos os desafios é trabalhando em assuntos pelos quais sejam realmente apaixonados” (RESNICK, 2020, p. 66).

Para possibilitar que os estudantes se envolvam em algum projeto no qual eles tenham interesse, os recursos que serão utilizados precisam proporcionar a eles “paredes amplas”, ou seja, instrumentos que possam apoiar e serem utilizados para uma ampla variedade de projetos. Além disso, uma característica que Papert (1986) já enfatizava é a importância desses recursos possuírem “pisos baixos”, a Dimensão Sintática do Construcionismo (PAPERT, 1986), e “tetos altos”, pois “para que uma tecnologia seja eficaz, ela deve proporcionar maneiras fáceis para os iniciantes darem os primeiros passos, mas também maneiras de trabalhar em projetos cada vez mais sofisticados ao longo do tempo” (RESNICK, 2020, p. 60).

O *Scratch* foi projetado “explicitamente para incentivar explorações. É fácil montar e desmontar blocos de programação gráfico no *Scratch*, assim como peças de Lego” (RESNICK, 2020, p. 127), pois foi desenvolvido para que pudesse oferecer pisos baixos, tetos altos e paredes amplas. Por se tratar de um ambiente de programação por blocos gráficos, possibilita que os primeiros *scripts* sejam escritos de forma intuitiva ao encaixar diferentes códigos uns aos outros (pisos baixos).

Dada a variedade de blocos de programação disponíveis, a possibilidade de inúmeras combinações entre eles, a capacidade de acrescentar extensões ao ambiente e o recurso que possibilita que o usuário adicione atores, sons ou cenários próprios para o seu *script*, temos o que é caracterizado como “pisos altos”. Além disso, considerando que “crianças diferentes têm interesses e paixões diferentes, portanto, vão querer trabalhar em projetos diferentes [...]” (RESNICK, 2020, p. 128), o *Scratch* pode funcionar com essas “paredes amplas”. Isso acontece, pois ele consegue apoiar muitas ideias ao possibilitar a criação de jogos, animações, histórias interativas, músicas, arte, entre outras possibilidades.

O terceiro princípio ressalta a importância de os estudantes trabalharem com seus projetos em pares, ou seja, não sozinhos. Isso acontece, pois,

[...] muitas vezes, o pensamento é integrado ao fazer no contexto de interagir, brincar, criar coisas, e a maioria dos pensamentos é feita em conexão com outras pessoas, compartilhamos ideias, obtemos reações, complementamos as ideias delas (RESNICK, 2020, p. 86).

Observamos as ideias de Resnick (2020) em consonância com as de Amabile (1983, 2012), que destaca a importância do desenvolvimento da criatividade em um ambiente social no qual os estudantes possam trabalhar colaborando com outras pessoas. Algumas ações podem ser tomadas para nutrir essa colaboração entre os estudantes, como a organização da sala de aula de forma que ela ofereça e busque estimulá-los a se movimentarem por ela e facilite que eles colaborem entre si. Também locais que possibilitem a novos integrantes interagir com os recursos por meio de exemplos, o compartilhamento de projetos via internet para que estudantes colaborem entre si, mesmo não estando lado a lado, entre outros. Essas características estão de acordo com a importância que Stenberg (2006) sugere para o ambiente no desenvolvimento da criatividade, ou seja, espaços que valorizem e incentivem os estudantes a trabalhar e a aprender juntos.

Além das possibilidades da criação de projetos sustentados pelos interesses dos alunos, o *Scratch* também oferece uma comunidade *on-line* na qual, segundo Resnick (2020, p. 89), “os jovens estão constantemente inventando e explorando novas maneiras de colaborar”. Os programas construídos com esse ambiente podem ser compartilhados no site para que outros programadores possam ver e ter ideias para os seus próprios programas. A comunidade *Scratch* também oferece um recurso chamado “remixar” no qual, como já tratado anteriormente, os usuários podem copiar trabalhos de outros e programar como se fossem seus, ou seja, podem compartilhar a solução, ação que, segundo Amabile (1982, 2012), pode ser relevante no processo criativo.

Por último, temos o Brincar, princípio que nos diz que o desenvolvimento de habilidades relacionadas à criatividade pode ser potencializado quando o aluno está brincando com o que projetou, mais precisamente quando ele está fazendo uma exploração lúdica que, segundo Resnick (2020), está no ponto de encontro entre o brincar e o fazer. Esse processo acontece quando os alunos estão ao mesmo tempo usando o que foi construído (brincando) e pensando no que ainda podem fazer. Eles fazem “ajustes e revisam os planos, normalmente seguindo um caminho sinuoso e indireto até a solução. Mas o que perdem em eficiência ganham em criatividade e agilidade” (RESNICK, 2020, p. 125). Assim, ao brincarem, coisas que nem imaginavam podem acontecer, possibilitando o desenvolvimento de habilidades em lidar com situações inesperadas.

Quando os participantes estão brincando ou construindo seus projetos no *Scratch*, erros inesperados podem aparecer em seus programas, essas falhas, em programação, são chamadas de *bugs*. Pelo fato de esses erros fazerem parte desse processo de programação, seja com o



*Scratch* ou em outro ambiente, “precisamos criar ambientes em que elas se sintam confortáveis para errar, em que possam aprender com seus erros” (RESNICK, 2020, p. 137). Para isso, eles exercitarão o que é conhecido como depuração, ou seja, a ação de identificar o *bug*, isolá-lo e pensar quais são as alterações necessárias naquele *script* para corrigi-lo. Este erro “está intimamente relacionado com a construção do conhecimento, pois atua como um motor que desequilibra e leva o aprendiz a procurar conceitos e estratégias para melhorar o que já acontece” (MALTEMPI, 2005, p. 8). Esse processo não é exclusivo da programação, mas “quando as crianças aprendem a programar, as estratégias que aprendem para depurar seus programas são úteis [podem ser úteis] para todos os tipos de resolução de problemas e atividades de criação” (RESNICK, 2020, p. 134, comentário nosso). Assim, a programação pode criar um ambiente interessante para o desenvolvimento desse processo de depuração, podendo ser transferido, posteriormente, para a resolução de outros tipos de problemas ou até influenciando na personalidade do estudante referente a outros obstáculos.

Deste modo, consideramos que o ambiente de programação visual do *Scratch* possibilita o trabalho com todos os princípios que fundamentam a Aprendizagem Criativa, no qual os estudantes possam passar por todas as ações sugeridas pela Espiral da Aprendizagem Criativa, não necessariamente na ordem sugerida. Também observamos o ensino de programação com esse recurso como uma oportunidade para a formação de alunos do grupo X, ou seja, “aqueles dispostos a assumir riscos e a experimentar coisas novas; são ávidos por definir os próprios problemas em vez de apenas resolver aqueles nos livros escolares” (RESNICK, 2020, p. 3).

Além disso, quando se aprende a programar, pode-se tornar “um pensador melhor, [...] você aprende a dividir problemas complexos em partes mais simples e a depurá-los, e a refinar e melhorar os projetos por meio de repetições ao longo do tempo” (RESNICK, 2020, p. 45). Essas habilidades estão relacionadas ao desenvolvimento do Pensamento Computacional, um dos principais objetos de estudo desta pesquisa.

#### 4.4.2. Planejamento dos sete encontros.

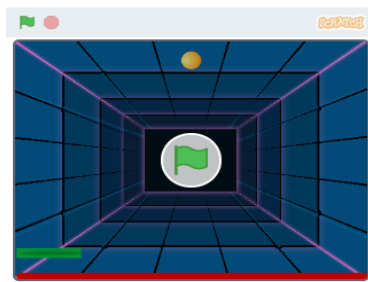
Os encontros foram pensados para que os estudantes pudessem se relacionar com todas as ações da Espiral de Aprendizagem Criativa apresentada na Figura 18, porém não necessariamente seguindo a ordem sugerida nela. Além disso, visando a que os quatro princípios da Aprendizagem Criativa também estivessem integrados às atividades, na maior

parte das vezes os participantes tiveram liberdade para trabalhar em projetos do seu interesse, ou seja, movidos por suas paixões.

Os estudantes foram incentivados a compartilhar ideias entre si ou com a própria comunidade do *Scratch* para que, ao interagirem com os seus próprios programas ou de outros, pudessem refletir sobre aquilo ao mesmo tempo em que se divertiam, contemplando assim o último princípio da proposta de Aprendizagem Criativa: brincar. Devido a isso e ao fato de que a pesquisa também é sustentada por ideias construcionistas (PAPERT, 1988; PAPERT, 2008; MALTEMPI, 2005; MALTEMPI, 2012), os participantes tiveram liberdade para seguir ou não as recomendações do professor, sendo incentivados a trabalhar em situações que realmente tinham interesse e, a partir delas, possibilitar o surgimento de discussões e reflexões sobre programação, matemática e outros assuntos que pudessem aparecer. Por isso, havia a possibilidade de o planejamento das atividades sofrer alterações durante a sua aplicação, o que aconteceu. O detalhamento desse planejamento pode ser visto no Apêndice III, aqui trazemos um breve resumo de quais foram as atividades desenvolvidas em cada um dos sete encontros que utilizamos para produção de dados, de acordo com o nosso Diário de Campo.

O primeiro encontro, realizado no dia 12 de agosto de 2020, contou com a participação de três estudantes: Maria, Miguel e Paulo. Foi pensado para que os participantes pudessem iniciar o envolvimento com as ações da Espiral da Aprendizagem Criativa pelos componentes Imaginar e Brincar. Depois da apresentação da proposta do projeto “Programação com *Scratch*” e de uma conversa sobre a experiência dos participantes com programação, os estudantes foram convidados a acessar o site do projeto criado na escola e ir até a aba Possibilidades. Neste espaço havia alguns programas construídos pelo professor para que eles pudessem conhecer algumas das possibilidades de criação com *Scratch*, duas dessas podem ser vistas na Figura 20. A primeira, denominada “PONG 2”, era um jogo no qual o usuário controlava o retângulo verde na parte inferior com o mouse e precisava movimentá-lo para não deixar que o círculo amarelo tocasse na linha vermelha. O segundo, chamado “APRESENTAÇÃO”, consistia em atores que não faziam parte do *Scratch* e que foram adicionados para interagir com o usuário. Conforme o participante clicava em um dos atores, uma ação diferente acontecia. A intenção foi mostrar a possibilidade de adicionar imagens próprias para fazer parte dos projetos desenvolvidos com a linguagem de programação que iríamos trabalhar junto ao *Scratch*.

Figura 20 - Imagem contendo dois programas disponíveis no site do projeto



## PONG 2

Clique na **Bandeira Verde** para iniciar o game!

Movimente o **cursor do mouse** para movimentar a barra verde.

Não deixe a bolinha amarela tocar a linha vermelha.

Quantos pontos você consegue fazer? Comente!

[Clique aqui para jogar em tela cheia.](#)

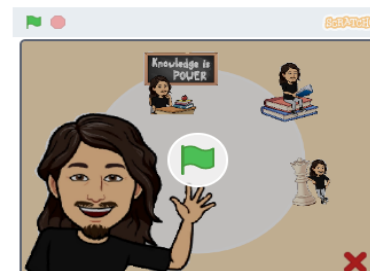
Esse jogo foi criado com base no tutorial Pong do Scratch disponível no link [scratch.mit.edu/projects/editor?tutorial=pong](https://scratch.mit.edu/projects/editor?tutorial=pong)

## APRESENTAÇÃO

No Scratch você pode fazer **animações interativas** adicionando suas próprias imagens e sons! Viram? Adicionei uma música que gosto e minhas próprias imagens!

Siga as instruções dos atores e utilize o mouse.

[Clique aqui para ver a animação em tela cheia.](#)



Fonte: Dados da Pesquisa (2020).

Além disso, este primeiro encontro também teve como objetivo a navegação no site do *Scratch* para que os participantes conhecessem um pouco de sua interface e para a criação de um usuário e senha, mas o foco principal foi na aba Explorar, possibilitando que os participantes pesquisassem projetos que tivessem relação com seus interesses e como eles estão sendo desenvolvidos por outros *Scratchers* pelo mundo. O professor sugeriu que os estudantes explorassem os trabalhos disponíveis na aba Explorar do *Scratch* durante a semana e selecionassem alguns que tivessem gostado para mostrar no próximo encontro.

Também foi feita a leitura de uma lista disponível no site do projeto sobre dez boas práticas para programar com o *Scratch*:

- Comece simples: você não precisa começar programando algo muito complexo, inicia com algo simples, adicionando alguns movimentos, sons etc. Conforme vai aprendendo, pode ir deixando o seu programa mais complexo;
- Trabalhe em algo que goste: desenvolva um programa sobre algo que você goste. Sabe aqueles personagens do jogo, filme ou anime que você assiste? Use eles!
- Se não tiver ideia do que fazer, explore o mundo: o *Scratch* possui uma biblioteca com milhares de projetos de outros "*Scratchers*", explore o site clicando aqui e se inspire.
- Não tenha medo de experimentar: você pode seguir as instruções do professor para criar algo, mas tem toda a liberdade para experimentar e construir *scripts* do jeito que quiser;

- Encontre um amigo para trabalhar e compartilhar ideias: não precisa ser no mesmo projeto, mas conversar com alguém que também está programando pode ser muito útil para compartilhar problemas e resolvê-los juntos;
- Não há nada de errado em copiar: está sem ideias do que fazer? Explore os projetos na biblioteca do *Scratch* e tente fazer algo parecido. Você também pode abrir esses projetos e clicar em "Remix", assim uma cópia do *script* criado será feita para você. Não esqueça de dar o devido crédito à pessoa em que você se inspirou na descrição do programa.
- Guarde suas ideias em um caderno de rascunhos: anote suas ideias, talvez você não consiga construí-las no momento, mas nada impede que você, depois de ter mais experiência, faça depois;
- Monte, desmonte e monte de novo: o *script* não funcionou? Desmonte e monte o código com cuidado para encontrar o bug!
- Muitas coisas podem dar errado, prenda-se a isso: programar pode não ser fácil. Está muito tempo em um projeto e não consegue sair do lugar? Descanse um pouco, procure exemplos na biblioteca do *Scratch* e depois volte ao programa.
- Crie suas próprias dicas de aprendizagem: você se conhece e provavelmente sabe qual a sua melhor maneira de aprender, use-a também para programar!

Essa lista foi organizada com base em dez sugestões trazidas por Resnick (2020) com estratégias para se trabalhar com projetos criativos, a lista original foi criada por crianças que participaram de um workshop no *Museum of Sciences* em Boston, nos Estados Unidos, e a lista organizada por Bakhiar Mikhak.

No segundo encontro, que aconteceu no dia 19 de agosto de 2020 com os quatro participantes presentes, e a partir dele, sempre buscamos iniciar as ações que compõem a Espiral da Aprendizagem Criativa pelo componente Refletir. A ideia era sempre começar atividade discutindo sobre o que foi feito na semana anterior e sobre o que eles construíram no *Scratch* durante a semana, já que foram orientados a explorar as possibilidades no final do primeiro encontro. Além disso, aqui também começamos a Criar, para isso, o professor explanou sobre o ambiente de programação *Scratch* passando por todas as abas e categorias de códigos. Para mostrar como juntar os blocos de programação na construção de um *script*, o professor construiu um “jogo de Labirinto” compartilhando a tela do seu computador com os participantes.

Nesse encontro, os participantes apresentaram alguns projetos explorados na comunidade *on-line* do *Scratch* que acharam interessantes e alguns já mostraram a construção

de alguns programas com a linguagem. Maria enviou o *link* de um programa que tinha gostado, intitulado “Bottle Flipe – are you Pro?”. Miguel apresentou o início da construção do programa apresentado no Episódio 1, contendo, segundo ele, alguns blocos de programação criados por ele e outros reproduzidos de vídeos da internet. Paulo, além de compartilhar o *link* de um programa que tinha gostado ao explorar a comunidade *on-line*, no final nos mostrou que já estava programando um jogo de labirinto parecido com o que tinha sido programado pelo professor. Eduardo também compartilhou a sua tela para mostrar a programação de um ator que tinha sido construída por ele.

Além de refletir sobre o que foi feito na aula anterior e na semana, no dia 26 de agosto de 2020 tivemos o terceiro encontro, este com três estudantes presentes: Eduardo, Miguel e Paulo. Foi pensado para que os participantes compartilhassem seus projetos no site do *Scratch* e experienciassem o programa um do outro, podendo trazer *feedbacks* sobre o projeto de cada um. Além disso, a ideia era que o professor apresentasse um programa que trabalhasse com perguntas feitas pelo computador, no qual as respostas são adicionadas pelo usuário e as ações são realizadas pelo programa a partir dessa resposta. Esse programa apresentaria algumas limitações no reconhecimento das respostas, o que auxiliaria na apresentação da importância de se utilizar um recurso de Inteligência Artificial junto ao *Scratch*. Reflexões sobre os programas dos participantes sucederam por quase todo o encontro, com os estudantes se auxiliando e sugerindo modificações nos programas uns dos outros, e por isso alguns objetivos do encontro não foram cumpridos. No final, o professor apresentou rapidamente o programa de perguntas e respostas, que seria retomado de forma mais detalhada na próxima semana.

No início da quarta aula, que ocorreu no dia 2 de setembro de 2020, tivemos apenas a presença de Miguel, que apresentou ao professor um dos recursos por ele utilizado para a criação de atores e cenários no *Scratch*, o *Pixilart*. Após isso, ficamos refletindo sobre o seu programa no *Scratch*, discutindo sobre possibilidades de melhorias e dúvidas que tinham ficado do encontro anterior. Assim que Eduardo chegou à aula nos últimos minutos, conversamos sobre Inteligência Artificial. Depois de serem questionados sobre o que eles entendiam sobre esse conceito, o professor apresentou algumas definições de Inteligência Artificial aos dois, o site *Machine Learning For Kids* e um programa construído com os algoritmos deste último recurso. Também foi apresentada uma definição para Aprendizado de Máquina (ou *Machine Learning*) e como um algoritmo como ele se diferencia de um algoritmo tradicional. No final, foi sugerido que eles entrassem no site e o explorassem durante a semana, principalmente a aba

Fichas de Trabalhos, que funcionava como guia para criar programas no *Scratch* com o recurso Aprendizado de Máquina.

No quinto encontro, ocorrido no dia 9 de setembro de 2020, o professor pediu sugestões aos três participantes que estavam presentes (Miguel, Eduardo e Paulo) para que pudessem construir, juntos, um algoritmo no *Scratch* que reconhecesse palavras ou frases e, a partir disso, executasse alguma ação. Com o programa funcionando, os participantes observaram algumas limitações com o reconhecimento de palavras pelo *Scratch*, pois, quando a frase ou palavra não era escrita exatamente igual, o programa não efetuava a ação desejada. Depois disso, montamos o mesmo programa no site *Machine Learning For Kids* utilizando os dados treinados com os algoritmos de Aprendizado de Máquina que, quando comparado ao primeiro, exercia as ações mais eficientemente. No final, eles foram apresentados a alguns exemplos de programas que poderiam ser criados com o recurso da aba Fichas de Trabalho e foi sugerido que escolhessem alguma das alternativas para programar e apresentar na próxima semana.

Eduardo, Maria e Miguel estavam presentes no sexto encontro, que aconteceu no dia 16 de setembro de 2020. Após pedir para que os participantes explicassem o que tínhamos feito em relação ao uso de *Machine Learning* no *Scratch*, o professor apresentou um programa criado no site *Machine Learning For Kids* que responde perguntas referentes ao Sistema Solar e que sempre reconhece o que o usuário escreve. Após indicar algumas limitações, ele pediu sugestões aos participantes com exemplos para melhorar os dados treinados. Nenhum dos estudantes que estava no encontro anterior criou algum programa utilizando esse recurso durante a semana, o professor sugeriu, novamente, para que tentassem fazer algo até o próximo encontro com esse recurso. Começamos a falar sobre as ideias dos estudantes em fazer diversos programas em um “mesmo universo”, pois um deles criou um estúdio no *Scratch* para isso, um local no próprio ambiente onde os usuários podem colocar vários projetos acessíveis apenas aos que estejam dentro deste local. Além disso, alguns alunos apresentaram novos projetos que estavam desenvolvendo com o *Scratch*. Um deles mostrou que usou arquivos em formato *GIF* no seu programa para fazer a animação dos seus atores. Por sugestão do professor, ele nos ensinou como fazer isso e mostrou como programou o seu último projeto utilizando esse novo recurso.

O último encontro que utilizamos para a produção de dados aconteceu no dia 23 de setembro de 2020 e contou com a presença de dois estudantes, Eduardo e Miguel. Novamente nenhum dos participantes tinha criado algo novo durante a semana. Com isso, seguimos com o planejamento que seria utilizado também nos próximos encontros com os participantes do

projeto “Programação com *Scratch*”: cada um programando do seu computador, enquanto compartilhávamos a tela e íamos discutindo e refletindo sobre as ideias e sobre os códigos utilizados. Nesse encontro, refletimos mais sobre a inserção de arquivos do tipo *GIF* e o uso do *Pixilart* para desenhar atores e adicioná-los ao *Scratch*.

#### **4.5. Dados produzidos**

Os dados produzidos foram obtidos por meio de quatro fontes de registro: um diário de campo, no qual fomos registrando informações por meio de escritas ou capturas de telas que considerávamos relevantes durante ou depois das aulas. Mensagens no aplicativo *WhatsApp* que eram escritas no grupo destinado aos participantes do projeto. Gravações de todos os encontros na plataforma *Google Meet* que continham áudio e vídeo, tanto dos participantes quanto do compartilhamento da tela, além do registro escrito do *chat* junto ao *Google Meet*. Também foi feito o *download* dos programas que eram compartilhados pelos estudantes diretamente no site do *Scratch* em todos os encontros, possibilitando um olhar mais atento à programação desenvolvida por cada um.

A organização dos dados produzidos foi feita da seguinte forma: começamos transcrevendo todos os encontros gravados pelo *Google Meet*. Registramos acontecimentos, falas dos participantes e imagens que foram compartilhadas. Após essa transcrição, separamos esse material em quatro categorias, cada categoria referente à trajetória dos participantes do projeto durante os encontros que cada um deles esteve presente, destacados com um “X” na Tabela 1, com dados das transcrições de todos os encontros e dos outros instrumentos utilizados para a produção dos dados, visando a uma triangulação desses componentes. A partir disso, tivemos como resultado a trajetória de cada um dos participantes por meio de episódios que, segundo Fiorentini e Lorenzato (2006, p. 143), são “depoimentos ou partes de texto que tenham relação explícita ou implícita com a questão investigativa”. Por fim, esses episódios foram escritos de forma mais detalhada e analisados com base no referencial teórico utilizado nesta pesquisa. Com isso, conseguimos evidenciar aspectos que nos auxiliaram nas respostas para a nossa pergunta de pesquisa.

**Tabela 1- Presença dos estudantes em cada encontro**

	Encontro 1	Encontro 2	Encontro 3	Encontro 4	Encontro 5	Encontro 6	Encontro 7
Eduardo		X	X	X	X	X	X
Maria	X	X				X	
Miguel	X	X	X	X	X	X	X
Paulo	X	X	X		X		

Fonte: Dados da Pesquisa (2020).

Apresentamos, em cada um dos episódios, pelo menos um projeto que estava sendo desenvolvido pelos estudantes durante a produção dos dados. O jogo intitulado “*Extreme-Burn*” foi um dos projetos que Miguel desenvolveu e que foi analisado por nós. Consistia em um cenário desenhado pelo próprio estudante dentro do *Scratch* e a programação de três atores. Um deles era controlado pelo usuário e se movimentava para a direita e para a esquerda, além de saltar. Outros dois foram programados para se moverem sozinhos por partes específicas do cenário e, quando alguns deles tivesse contato com o ator controlado pelo usuário, uma ação específica era gerada pelo programa e o jogo precisava ser reiniciado para continuar.

Eduardo desenvolveu dois projetos durante os encontros e parte de ambos foi trazido para a análise dos dados. O primeiro consistia em um jogo no qual o usuário controlava um ator principal que poderia se mover tanto no eixo X quanto no eixo Y do cenário e tinha como título “*Tales of SadBoy (In-dev)*”. Neste projeto, cenário e atores foram todos desenhados no *Pixilart* e, além da movimentação do ator principal, Eduardo programou algumas trocas de cenários, movimentação de outros atores que dependiam de algumas ações do ator principal, sons e algumas interações com partes do cenário envolvendo falas e movimentação. O outro trabalho que foi projetado por Eduardo tinha como título “*Fan-made | Deltarune Raisei vs Kris*”. Nesse o estudante utilizou diversos GIF para a animação dos seus atores, adicionou músicas específicas e, com base em outros trabalhos, construiu um jogo que era, segundo ele, semelhante a outro já existente.

O projeto desenvolvido por Paulo tinha algumas características semelhantes tanto ao projeto de Miguel quanto ao de Eduardo e foi intitulado como “*Marquinhos*”. O estudante programou um ator que, assim como fez Eduardo, foi criado no *Pixilart*. Este ator era controlado pelo usuário e se movimentava pelo cenário da mesma forma que o ator programado por Miguel. Além disso, Paulo também programou a troca de cenários que foram desenhados por ele. Alguns algoritmos foram construídos pelo próprio, já outros, como o que controlava a gravidade, foi, de acordo com o estudante, construído a partir do auxílio de um vídeo do *YouTube*.



Dos quatro estudantes que participaram dos encontros, Maria foi a única que não era aluna do professor responsável e que também não era do nono ano. No segundo encontro, ocorrido no dia 19 de agosto de 2020, os estudantes, que tinham sido convidados a explorar os programas já criados na comunidade *on-line* do *Scratch*, apresentaram aqueles que tiveram interesse. Maria trouxe o jogo “*Bottle Flipe – are you pro?*”, conforme podemos ver na Figura 21. O professor comentou que era uma versão virtual de uma brincadeira que acontecia, às vezes, na escola: lançar a garrafa com um pouco de água até conseguir que, quando ela caísse, ficasse em pé. Além disso, ele fez o seguinte comentário diretamente para Maria: “Maria, muito legal! Podemos fazer algo bem parecido com isso e pode ficar muito bom!”.

**Figura 21 - Programa selecionado por Maria**



Fonte: Dados da Pesquisa (2020).

Apesar da aluna ter feito a tarefa sugerida de explorar os projetos desenvolvidos na comunidade *on-line* do *Scratch*, ela não desenvolveu nenhuma outra atividade durante os encontros, inclusive não participou de todos. Em alguns, Maria falou que não conseguiria estar presente, pois teria que emprestar o seu telefone para outra pessoa.

Diversos aspectos podem ter influenciado a aluna a não elaborar projetos no *Scratch*. A única evidência que temos da sua não participação é a falta do recurso tecnológico necessário, algo que também teve destaque durante o período de Ensino Remoto Emergencial com parte dos professores e estudantes de outras instituições, pois, de acordo com Oliveira (2020a; 2020b), a disponibilidade de recursos tecnológicos para a realização das atividades remotas era fundamental. Entretanto, outros fatores podem ter causado isso, como a fato de ela ser a única estudante que não era do nono ano, que não conhecia o professor, ser a única menina nos encontros ou até mesmo algumas falhas que tivemos como mediadores dessa aprendizagem ao


não observarmos esses ou outros fatores que podem ter influenciado a aluna para não desenvolver projetos durante os encontros que utilizamos para a produção de dados.

No próximo capítulo trazemos a análise dos dados realizada de acordo com o que foi relatado anteriormente. Buscamos, a partir de evidências encontradas com o refinamento das categorias e com sustentação no nosso referencial teórico, responder à pergunta de pesquisa: **como o Pensamento Computacional e a matemática emergem de projetos desenvolvidos com o ambiente de programação Scratch em atividades remotas?**

## 5 Análise dos Dados

A análise dos dados foi realizada a partir de episódios que se referem à parte do percurso dos participantes do projeto “Programação com *Scratch*”. A partir dos dados obtidos por meio da gravação dos encontros via *Google Meet*, dos arquivos com os programas do *Scratch*, das imagens retiradas do grupo do *WhatsApp* e do diário de campo, organizamos parte da trajetória de cada um dos quatro participantes de forma que pudéssemos responder à nossa pergunta de pesquisa.

Consideramos, algumas vezes, adicionar comentários às falas que possam ajudar na compreensão do texto. Esses comentários estão entre parênteses junto aos excertos. Além disso, alguns excertos possuem cortes para indicar a exclusão de algumas falas, algumas por não serem relevantes para a pesquisa e outras pela impossibilidade de entendê-las nas gravações. Esses cortes foram identificados por três pontos entre colchetes ([...]). Também, quando houver a presença de três pontos sem os colchetes (...), estamos indicando uma pausa na fala do participante.

Os códigos utilizados na construção dos programas no *Scratch* estão representados, quando necessário, ao longo do texto entre colchetes. Por exemplo, o código  é denotado como [quando bandeira verde for clicado].

Assim, no caso específico desta proposta investigativa, debruçamo-nos a buscar alguns momentos nos quais o Pensamento Computacional se manifesta nas ações dos participantes por meio de relações com os quatro pilares ou com outras características do PC. Buscamos também compreender não apenas como o Pensamento Computacional se manifesta nas construções feitas, como também buscar indícios de como a consolidação desse pensamento emergiu com a construção ou explicitação das ideias matemáticas. Sendo assim, procuramos entender estas manifestações de modo mais profundo, ou seja, como o Pensamento Computacional e a matemática podem emergir juntos, destacando algumas complementaridades e singularidades que apareceram na análise dos dados produzidos. Na busca para evidenciar esses aspectos, dividimos este capítulo em três episódios, cada um relacionado com parte da trajetória dos estudantes que participaram desta pesquisa e que se envolveram com o desenvolvimento de um projeto com o *Scratch*, além de uma sistematização do que foi encontrado em cada um desses episódios.

### 5.1. Episódio 1: Miguel e a decomposição não linear

Miguel foi um dos estudantes que participou do projeto “Programação com *Scratch*” por meio de atividades remotas desenvolvidas com o *Google Meet* e com o *WhatsApp*. Durante os sete encontros, o estudante desenvolveu três programas que envolveram a resolução de diversos problemas que incluíram aspectos matemáticos e de programação. Para apresentar estes aspectos, trazemos o caso no qual Miguel tinha como problema principal a programação de um algoritmo que fizesse com que um projétil partisse de um dos atores (circulado em vermelho na Figura 22) e fosse em direção à área na qual o ator principal se movimentava (circulado em verde na Figura 22).

**Figura 22 - Interface do programa que estava sendo programado por Miguel**



Fonte: Dados da Pesquisa (2020).


O processo de construção desse algoritmo não se mostrou uma tarefa simples e se desmembrou em alguns procedimentos. Para facilitar a explanação, apresentaremos a resolução do problema em três momentos, mostrando como foi a evolução do processo para Miguel até encontrar a solução do seu problema.

#### 5.1.1. Primeiro momento: entendendo o problema

O primeiro momento aconteceu no final do terceiro encontro síncrono, ocorrido dia 26 de agosto de 2020, no qual deixamos um dos problemas de Miguel em aberto, pois a atividade

já havia terminado. O problema, como explicado anteriormente, consistia em como programar um projétil que surgisse junto a um dos atores e fosse em direção à área na qual o ator principal se movimentava. Neste terceiro encontro, Miguel já havia apresentado dois blocos de programação na busca da solução deste problema, que podem ser vistos no Quadro 3. O primeiro bloco iniciava com o código cabeça [quando ‘bandeira verde’ for clicado], seguido do [espere 13 seg] e [crie um clone de sun-ray]. Este bloco, assim que o programa iniciava ao clique da bandeira verde, criava um clone de um “raio de sol” após 13 segundos junto ao ator do sol, circulado em vermelho na Figura 22. O segundo tinha como código cabeça o [quando eu começar com um clone], seguido de [vá para x: número aleatório entre -180 e 180 y: 161], [mostre], [espere 2 seg] e uma estrutura de decisão que iniciava com [se posição x > -179 então] seguida do código [sempre]. Apesar do problema não ter sido resolvido com esses blocos, Miguel explicou o motivo de ter escolhido o intervalo de [-180 e 180], como podemos ler no excerto no Quadro 3.

**Quadro 3 - Entendendo a problemática de Miguel**

<b>Excerto referente aos códigos no bloco abaixo</b>	
<b>Professor:</b>	por que você colocou -180 a 180? (questionamento referente ao bloco contendo [número aleatório entre -180 e 180])
<b>Miguel:</b>	porque se eu botasse de 0 até 180 ele ia aparecer só daqui até aqui... então eu tenho que botar de -180 até 180 positivo que é para ser os dois lados. (Miguel fala isso enquanto mostra, com o mouse, as posições que está falando na interface do programa criado no Scratch)
<b>Professor:</b>	e esse 180 está variando no eixo X ou no eixo Y?
<b>Miguel:</b>	no eixo X, o Y é... (gravação curta)
<b>Professor:</b>	isso, o Y é a altura, o eixo X, como é pra ir de um lado para o outro, você fez isso.
<b>Blocos de programação apresentados por Miguel</b>	
	

Fonte: Dados da Pesquisa (2020).

Nos dados apresentados no Quadro 3, temos, além de reflexões sobre pontos do Plano Cartesiano, a ideia de utilizar um número aleatório dentro de um certo intervalo para a posição de um dos atores, que podemos ver no bloco [vá para x: número aleatório entre -180 e 180 y: 161] e a utilização da relação maior/menor no bloco [se posição x > -179 então], relações com conceitos matemáticos que não tinham sido vistos nos encontros anteriores.

Observamos que, desde este primeiro momento, já houve a ação de transformar o problema inicial (construção de um Algoritmo para a ação desejada) em problemas menores, pois Miguel tinha iniciado com dois blocos de programação separados, um para a criação do clone (bloco à esquerda no Quadro 3) e outro para a movimentação desse clone (bloco à direita no Quadro 3). Assim, a ideia de fazer a reformulação de um problema em problemas menores, destacada por Wing (2006), já começa a ser desenvolvida

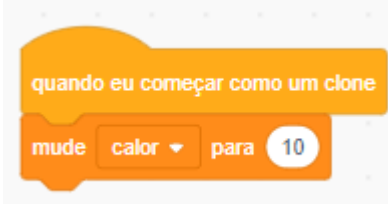
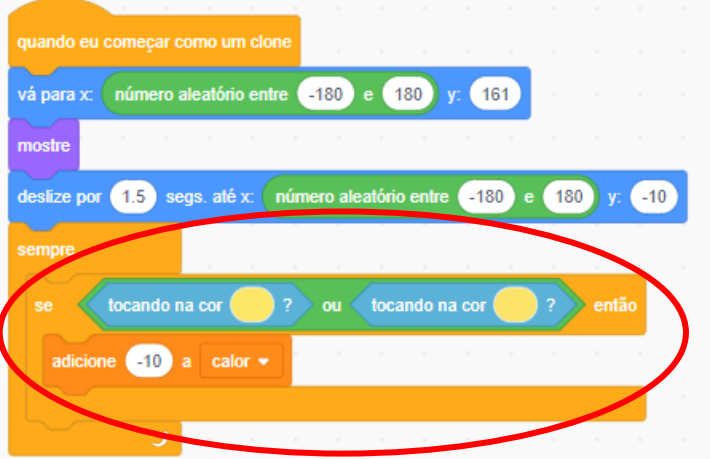
Todavia, Papert (1988) e Maltempo (2012) mostram que não existe caminho errado quando estamos em um ambiente construcionista, pois os estudantes passam a enxergar os erros como *bugs* e, a partir disso, envolvem-se com o processo de depuração. Sendo assim, na busca pela solução do problema, o estudante é encorajado a estudar o *bug*, ao invés de esquecê-lo, e, neste processo, novos conhecimentos podem ser construídos e agregados aos já existentes.

Consideramos importante destacar o entrelaçamento das ações com conceitos matemáticos, a saber o de aleatoriedade dentro de um certo intervalo, os pontos no Plano Cartesiano e o uso da relação maior/menor nos códigos. Ao observar os dados apresentados no segundo momento, referente ao quarto encontro, notamos que a matemática emergiu como um dos meios que impulsionou a correção desse *bug*, característica relevante para o desenvolvimento do Pensamento Computacional, segundo Brennan e Resnick (2012). Miguel foi tentando solucionar o problema com um “*monte de coisa aleatória*” e encontra a solução após criar uma variável para o seu código e substituir no que trazia a relação maior/menor por outro. Porém, apenas dividir um problema em subproblemas nem sempre é garantia de que o algoritmo irá ser o ideal. Isso deve fazer sentido para a problemática e tanto a matemática quanto a situação em si podem ajudar a dar esse sentido, ou seja, direcionado para a solução do problema. Assim, observamos que, no caso que se apresenta, o problema não foi resolvido neste primeiro momento.

### 5.1.2. Segundo momento: problema resolvido

No quarto encontro, ocorrido no dia 02 de setembro de 2020, Miguel voltou ao assunto sobre a resolução do problema do encontro anterior e explicou que tinha conseguido melhorar a sua busca pela solução. Ao compartilhar a tela do seu computador pelo *Google Meet*, o estudante novamente apresentou dois blocos de programação, mas com algumas modificações em relação aos anteriores. O primeiro consistia em apenas dois códigos: [quando eu começar como um clone] e [mude calor para 10]. Diferentemente do bloco do encontro anterior para a criação do clone, Miguel utilizou uma variável denominada [calor] para controlar o surgimento e o desaparecimento deste ator/clone. Já o segundo era parecido com o apresentado no encontro anterior, porém, depois de código [mostre], Miguel fez algumas modificações e adicionou o código [deslize por 1.5 segs. Até x: número aleatório entre -180 e 180 y: -10], seguido da estrutura de decisão [sempre], [se [tocando na cor 'amarela1'] ou [tocando na cor 'amarela2'] então] [adicione -10 a calor]. No Quadro 4, podemos ver, além dos blocos de programação, um excerto da conversa do professor com Miguel quando foi pedido para que ele explicasse como tinha conseguido resolver o problema que estava em aberto.

**Quadro 4 - Blocos de programação e excerto sobre a solução da problemática de Miguel**

<b>Blocos de programação referentes à programação do projétil</b>	
	
<b>Excerto referente aos blocos acima</b>	
<p><b>Miguel:</b> eu fiquei tentando um monte de coisa aleatória, eu não sabia o que tinha dado errado, o porquê de não estar funcionando. Eu só fui tentando juntar, misturar... e eu juntei o... lembra que eu não estava conseguindo fazer ele, quando tocasse na areia, sumir? (ele estava se referindo ao projétil que, ao tocar em uma área do cenário, continuava lá, sendo que ele queria que sumisse).</p> <p><b>Professor:</b> Sim, o projétil, isso?</p>	

**Miguel:** Sim, eu criei uma variável. Eu fiz uma gambiarra. Eu criei uma variável e chamei ela de calor. Eu dei um valor de 10 para ela. Eu botei pra quando começar como um clone, que é para todos os clones que criar, ele começar com calor 10 (bloco à esquerda neste quadro).

**Professor:** ok.

**Miguel:** eu tive que, pra fazer ele sumir, eu tive que juntar o *script* de colisão, que é para quando ele tocar na areia, com o *script* dele cair. Eu não consegui fazer ele cair reto. Eu botei ele pra cair de uma posição aleatória pra outra posição aleatória. As vezes ele vai muito radical, as vezes ele quase não se mexe.

**Professor:** e tu entendeu o porquê disso? Que as vezes ele vai muito rápido e as vezes ele vai muito devagar? (a minha intenção foi descobrir se ele tinha descoberto a relação proporcional da velocidade e da distância com o tempo fixado).

**Miguel:** não na verdade.

**Professor:** eu vou te explicar então.

**Miguel:** ah, eu acho que já sei. Porque como vai para posições aleatórias, e eu botei um certo tempo para ele percorrer essa distância. (entendo que ele quis dizer que o tempo é sempre o mesmo, e como a distância muda, a velocidade também vai mudar)

**Professor:** perfeito. Se for uma posição mais longe, ele vai mais rápido ou mais devagar?

**Miguel:** mais rápido.

**Professor:** isso, porque é o mesmo tempo. Se for uma posição mais pertinho?

**Miguel:** ele vai mais devagar.

[...]

**Professor:** ali você usou duas cores como comentamos na aula passada para parar o problema. O que é aquele adicione -10 a calor? (meu questionamento foi referente à parte circulada na imagem à direita neste quadro).

**Miguel:** quando calor igual a zero então apaga este clone. Eu não consegui fazer aquela tua ideia de fazer o ator só ficar indo e voltando. Eu achei muito mais fácil fazer o clone.

**Professor:** perfeito, é a tua ideia e o teu programa, se você achou fácil assim. Agora você já sabe um *script* para fazer um projétil, pode usar sempre o mesmo.

Fonte: Dados da Pesquisa (2020).

No excerto em que Miguel diz “*eu fiquei tentando um monte de coisa aleatória, eu não sabia o que tinha dado errado, o porquê de não estar funcionando. Eu só fui tentando juntar, misturar... e eu juntei o... lembra que eu não estava conseguindo fazer ele, quando tocasse na areia, sumir?*”, observamos que o estudante tinha notado que o *bug* que estava tentando resolver no seu subproblema do projétil estava relacionado a um dos atores/clones que desaparecia da tela interface do programa. Ele tinha programado os blocos de forma separada na busca pela solução e, apesar disso, o problema ainda não era solucionado. O problema foi resolvido quando o aluno decidiu juntar. Isso pode ser observado quando o estudante diz “*eu tive que, pra fazer ele sumir, eu tive que juntar o script de colisão, que é para quando ele tocar na areia, com o script dele cair*”. Aqui temos evidências de que a reformulação de problemas em problemas menores, pilar da Decomposição do Pensamento Computacional, não é um movimento que precisa acontecer de forma linear em um só sentido. Miguel tinha construído seus algoritmos separados e, para que eles funcionassem e fizessem sentido para a problemática,



precisou uni-los em um mesmo bloco. Além disso, nota-se que o uso da estratégia de Decomposição em um fluxo contrário aconteceu junto ao uso da matemática para a criação da variável [calor] pois, no excerto do Quadro 4, Miguel explica o seguinte: “[...] eu criei uma variável. Eu fiz uma gambiarra. Eu criei uma variável e chamei ela de calor. Eu dei um valor de 10 para ela. Eu botei pra quando começar como um clone, que é para todos os clones que criar, ele começa com calor 10”.

Com isso, temos evidências de que o processo de construção de conhecimento do aluno envolvendo o Pensamento Computacional e a matemática não se dava mediante uma relação de erro ou acerto, mas em um processo que envolvia o programar e a depuração. De acordo com Papert (1988), na escola o estudante é ensinado a lidar com o erro como algo negativo e, dificilmente, é incentivado a refletir ou pensar sobre, ignorando assim as possibilidades de aprender a partir dele. Sendo assim, um ambiente construcionista pode possibilitar que o aluno desenvolva habilidades referentes a esse processo de correção de *bugs* e lide melhor com seus erros. Wing (2006) também traz a correção de erros como característica importante no desenvolvimento do Pensamento Computacional, além da habilidade de decompor um problema em problemas menores de mais fácil resolução, ação realizada por Miguel e que se relaciona com o pilar da Decomposição. Brennan e Resnick (2012) nomeiam essa prática como Teste e Depuração, importante para o desenvolvimento do Pensamento Computacional e um modo de encorajar o estudante a estudar o erro e a procurar uma solução para tal, ao invés de desistir do que está sendo desenvolvido.

Outra característica importante do Construcionismo é a Dimensão Semântica, que traz a ideia de que objetos que são manipulados pelos estudantes precisam ir além de uma estrutura matemática formal, um aglomerado de símbolos e números. Para Papert (1986), eles precisam ter algum significado para o estudante e para a solução do problema. Analisando o Quadro 4, notamos diversos conceitos matemáticos por trás da construção e das falas que fazem sentido para o estudante por estarem relacionados a aspectos do seu projeto e para a solução do problema, pois os códigos utilizados relacionavam-se à ação que Miguel deseja programar para o projétil.

Temos a variável [calor] criada pelo próprio que comandava algumas ações do projétil que era um raio de sol, no qual ele programou para receber diferentes valores inteiros dependendo da situação, como em [mude calor para 10] e [adicione -10 a calor]. Relacionado com a variável [calor], o estudante programou uma estrutura de decisão sempre-se-então com os códigos [tocando na cor amarela] e o operador matemático [ou], relacionado à ação do ator

em realmente tocar parte do cenário para que a ação programada ocorresse, ou seja, parte do problema que ele não tinha solucionado anteriormente

Nos dados do Quadro 4, também temos a ideia de grandezas proporcionais aparecendo nas falas do aluno ao discutir, junto ao professor, a velocidade do projétil:

**Professor:** *e tu entendeu o porquê isso? Que as vezes ele vai muito rápido e as vezes ele vai muito devagar? (a minha intenção foi descobrir se ele tinha descoberto a relação proporcional da velocidade e da distância com o tempo fixado).*

**Miguel:** *não na verdade.*

**Professor:** *eu vou te explicar então.*

**Miguel:** *ah, eu acho que já sei. Porque como vai para posições aleatórias, e eu botei um certo tempo para ele percorrer essa distância. (entendo que ele quis dizer que o tempo é sempre o mesmo, e como a distância muda, a velocidade também vai mudar).*

**Professor:** *perfeito. Se for uma posição mais longe, ele vai mais rápido ou mais devagar?*

**Miguel:** *mais rápido.*

**Professor:** *isso, porque é o mesmo tempo. Se for uma posição mais pertinho?*

**Miguel:** *ele vai mais devagar.*

Evidenciando, assim, uma mobilização ou até mesmo uma possível construção do conhecimento relacionado a grandezas proporcionais a partir da experimentação feita com o programa ao longo da depuração e do uso da Decomposição, além da reflexão orientada pelo professor.

### 5.1.3. Terceiro momento: reflexões com base na solução do problema

No terceiro momento, destacamos quando o aluno Eduardo entrou no encontro pelo *Google Meet* alguns minutos atrasado. Com isso, o professor sugeriu que Miguel explicasse a ideia do clone para ele, pois era um problema que tinha ficado sem solução na semana anterior. A explicação de Miguel, os questionamentos do professor e o bloco de programação evidenciado pelo aluno estão apresentados no Quadro 5.

**Quadro 5 - Reflexões de Miguel**

Excerto
<p><b>Miguel:</b> bom, inicialmente eu estava querendo... eu tive aquela ideia que provavelmente todo mundo já teve, que é ficar criando clones e clones. Então o sor me deu uma ideia e eu não consegui, então voltei para a ideia inicial. Eu tava quebrando cabeça na anterior, então eu... junto com o <i>script</i> de movimento, que é o <i>script</i> que faz ele ir pra baixo. Pra ele agir como um projétil.</p> <p><b>Professor:</b> e esse é o scrip do projétil? (eu estava me referindo ao código na qual ele estava com o mouse em cima, apresentado neste Quadro).</p>

**Miguel:** esse é o *script* que faz ele se mover (nesse momento, Miguel divide o *script* em dois, para explicar o que cada parte faz, conforme pode ser visto na imagem abaixo. Está se referindo primeiro ao código acima da linha vermelha) e esse aqui era o *script* que tava separado dele que fazia que, quando ele tocasse aqui, ele sumisse (código abaixo da linha vermelha na imagem abaixo). Eu juntei com esse aqui e criei uma variável chamada calor que só se aplica ao ator do projétil. Quando o calor for 0, então o projétil se some. Seria a mesma coisa que programa colisão.

**Professor:** por que tem aquele número aleatório entre -180 e 180?

**Miguel:** é, esse se aplica ao X do projétil, que vai ser aqui em cima (ele mostra, com o mouse, de onde até onde o X de -180 até 180 vai, sendo a movimentação do Sol), mas a altura vai ser sempre a mesma.

#### Bloco de códigos que Miguel se refere no excerto acima



Fonte: Dados da Pesquisa (2020).

Além da matemática envolvendo o Plano Cartesiano, os Números Inteiros, uso de números aleatórios dentro do intervalo  $[-180, 180]$  e o operador [ou], já destacados anteriormente, os dados no Quadro 5 nos apresentam a estratégia que Miguel utilizou para a construção do seu *script* e como isso fez com que ele se envolvesse com os diversos aspectos da matemática. Ou seja, o ato de projetar um programa no *Scratch*, do seu interesse e sozinho, fez com que o estudante tivesse a oportunidade de investigar e experimentar a matemática naquele momento. Estes aspectos podem ser observados no início da explicação de como ele iniciou o processo ao dizer “*bom, inicialmente eu estava querendo... eu tive aquela ideia que provavelmente todo mundo já teve, que é ficar criando clones e clones*”, seguindo de quando o estudante conta que não seguiu a ideia do professor: “[...] *Eu tava quebrando cabeça na anterior, então eu... junto com o script de movimento, que é o script que faz ele ir pra baixo. Pra ele agir como um projétil [...]*”, além da explicação de como construiu o seu algoritmo: “*esse é o script que faz ele se mover [...]* e esse aqui era o script que tava separado dele que fazia que, quando ele tocasse aqui, ele sumisse [...]. Eu juntei com esse aqui e criei uma

*variável chamada calor que só se aplica ao ator do projétil. Quando o calor for 0, então o projétil se some. Seria a mesma coisa que programa colisão*”. Estas características são fundamentais em um ambiente construcionista por estarem relacionadas com a Dimensão Pragmática, pois a matemática que emergiu nesse processo de investigação e experimentação foi importante para aquele momento, ou seja, não precisou da ideia de que aquilo que foi aprendido pode ser relevante no futuro.

#### 5.1.4. Como aspectos Matemáticos e do Pensamento Computacional emergiram no projeto desenvolvido por Miguel

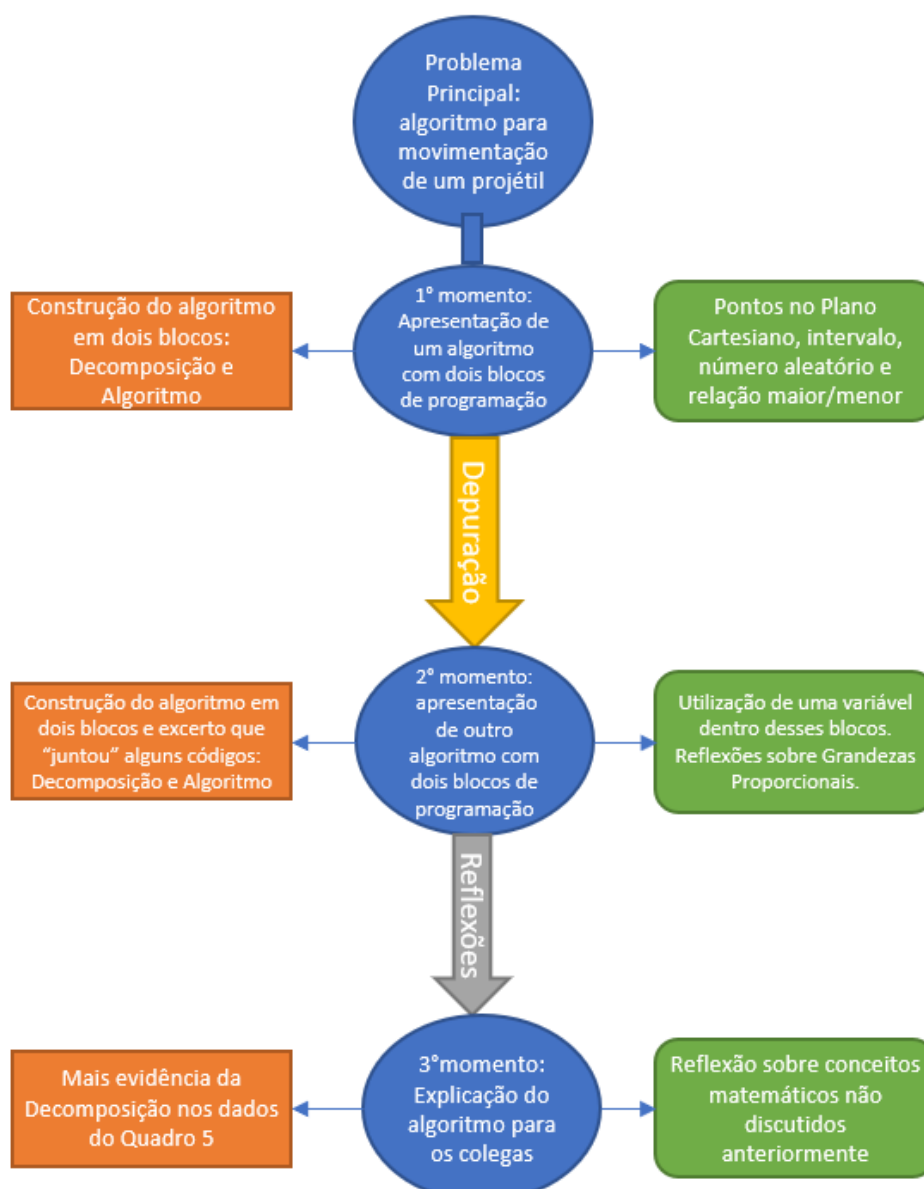
A análise apresentada até o momento nos traz evidências de que o estudante, tanto para a construção do seu programa quanto nas ações para a busca da solução do problema apresentado, utilizou-se de diversos aspectos que estão relacionados à matemática, como o Plano Cartesiano, Números Inteiros, intervalos, conectivos lógicos, variáveis, grandezas proporcionais, entre outros. Além disso, Miguel se envolveu com principalmente dois dos pilares do Pensamento Computacional: Decomposição e Algoritmos.

No fluxograma apresentado na Figura 23, trazemos um resumo de como esses aspectos se relacionaram ao problema principal, que consistia na construção de um algoritmo para a movimentação de um projétil, durante os três momentos destacados nesta análise.

No primeiro momento, temos a apresentação de um algoritmo envolvendo dois blocos de programação no qual já observamos a relação de Miguel com pontos no Plano Cartesiano, número aleatório em um intervalo e o uso da relação maior/menor. Neste mesmo momento, a construção do Algoritmo já tinha sido feita de forma fragmentada, evidenciando aspectos do Pensamento Computacional. Já no segundo momento, temos a apresentação de outro algoritmo também com dois blocos de programação que foi criado a partir do processo de depuração. Diferentemente do anterior, nele tínhamos uma variável criada pelo estudante e uma estrutura de decisão sempre-se-então junto ao operador [ou]. Além disso, reflexões sobre grandezas proporcionais foram feitas. É neste segundo momento que notamos a não linearidade do movimento de decomposição ligado ao Pensamento Computacional, pois, apesar do aluno ter iniciado a construção do algoritmo fragmentando o código em alguns blocos, o problema só é solucionado ao juntá-los. Por fim, conseguimos notar o surgimento de outros aspectos

matemáticos e relacionados ao Pensamento Computacional nas reflexões sobre o algoritmo criado que aconteceu no terceiro momento destacado nesta análise.

**Figura 23 - Fluxograma do entrelaçamento da Matemática com o Pensamento Computacional no Episódio 1**



Fonte: Elaborada pelo autor.

Ponto no qual podemos observar como a matemática e os pilares da Decomposição e Algoritmos estiveram envolvidos na solução do problema de Miguel podem ser evidenciados no terceiro momento. No excerto do Quadro 5, Miguel explica como tinha construído um bloco para o movimento e outro para que ele desaparecesse. O estudante complementa: *“Eu juntei com esse aqui e criei uma variável chamada calor que só se aplica ao ator do projétil. Quando o calor for 0, então o projétil se some. Seria a mesma coisa que programa colisão”*. Assim,

após juntar os dois blocos, ação apresentada na figura do Quadro 5, e utilizar também o bloco apresentado à esquerda no Quadro 4, o problema foi resolvido com a construção de um Algoritmo que funcionava como esperado e envolvia diversos aspectos matemáticos.

A construção de um algoritmo que representa a solução de um problema que pode ter sido desenvolvido junto à Decomposição é uma das competências fundamentais do Pensamento Computacional (CSTA, 2011), além de ser considerada o resultado dele (RIBEIRO; FOSS; CAVALHEIRO, 2020). A articulação de símbolos e códigos, no qual o aluno possa traduzir uma certa situação de uma linguagem a outra, é uma das competências esperadas no ensino de matemática (BARCELOS; SILVEIRA, 2012), na qual o estudante possa “expressar suas respostas e sintetizar conclusões, utilizando diferentes registros e linguagens” (BRASIL, 2018, p. 267). Quando Miguel é questionado pelo professor sobre o motivo de ter escolhido o intervalo  $[-180, 180]$  no Quadro 3, ele explica que não funcionaria caso utilizasse o intervalo  $[0, 180]$ , utiliza a expressão “*daqui até aqui*” e utiliza o mouse para destacar as posições. Estas ações evidenciam que a linguagem visual do aluno foi traduzida para um algoritmo na linguagem do *Scratch* que funcionou por meio do uso da matemática.

Outra evidência dessa característica está na fala de Miguel no excerto do Quadro 4: “[...] eu tive que juntar o script de colisão, que é para quando ele tocar na areia, como o script dele cair. Eu não consegui falar ele cair reto. Eu botei ele pra cair de uma posição aleatória para outra [...]”. Aqui podemos notar que a linguagem utilizada por Miguel para colisão do seu ator principal com a areia foi traduzida em um algoritmo no *Scratch* utilizando, junto da estrutura de decisão sempre-se-então, o operador matemático [ou]. Já para a queda do objeto o aluno usou pontos no Plano Cartesiano no qual um deles era um número aleatório entre  $[-180, 180]$ , presentes do código [deslize por 1.5 segs. Até x: número aleatório entre -180 e 180 y: -10].

Entretanto, não podemos apenas pensar em uma relação de causa e consequência. Não é apenas o Pensamento Computacional que gera a necessidade do uso da matemática. O uso e a manipulação dos conceitos matemáticos também geram *insights* para que a Decomposição do problema evolua constantemente na construção de um Algoritmo que seja eficiente para o propósito desejado. Como quando olhamos para os blocos de programação no Quadro 3 e notamos que a Decomposição do Algoritmo na busca da solução do problema já tinha acontecido, porém ele ainda não tinha sido resolvido. Já no Quadro 4, vemos que há poucas diferenças entre esses blocos e os apresentados anteriormente, todavia, o problema é resolvido quando Miguel cria e adiciona ao seu Algoritmo a variável [calor] e programa, de forma

separada, os códigos [sempre], [se tocando na cor ‘amarela1’ ou tocando na cor ‘amarela1’ então] [adicione -10 a calor] que, como mostrado no bloco de programação do Quadro 5, foi adicionado posteriormente junto ao bloco anterior. Assim, a matemática foi utilizada para fazer com que o Algoritmo, construído a partir da Decomposição, pudesse funcionar e solucionar o problema.

É importante notar que, no caso das ações de Miguel, o Pensamento Computacional e a matemática parecem atuar como movimentos complementares na condução e determinação da solução. Enquanto o PC aparece como uma possibilidade para resolver o problema, a matemática, em alguns casos, surge como um problema a ser resolvido, seja para traduzir alguma ação desejada de uma linguagem para outra ou até mesmo para programar algum aspecto relacionado com o algoritmo que está sendo construído. Também notamos que a Decomposição de problemas não é algo direto, fixo ou linear, pois, apesar de Miguel ter programado em blocos diferentes e depois ter reunido eles para a solução, construindo assim um Algoritmo, não conseguimos afirmar que essa era a intenção do aluno desde o começo. Notamos que as habilidades relacionadas ao Pensamento Computacional são processos que se dão ao longo de muitas mudanças e que a matemática, nesse caso, pode contribuir para auxiliar no processo de determinação do encaminhamento da solução.

Na próxima seção apresentaremos o Episódio 2, mostrando parte da trajetória de Eduardo na construção de seus programas e como esse processo fez com que ele se envolvesse com aspectos ligados ao Pensamento Computacional e à matemática em um meio construcionista.

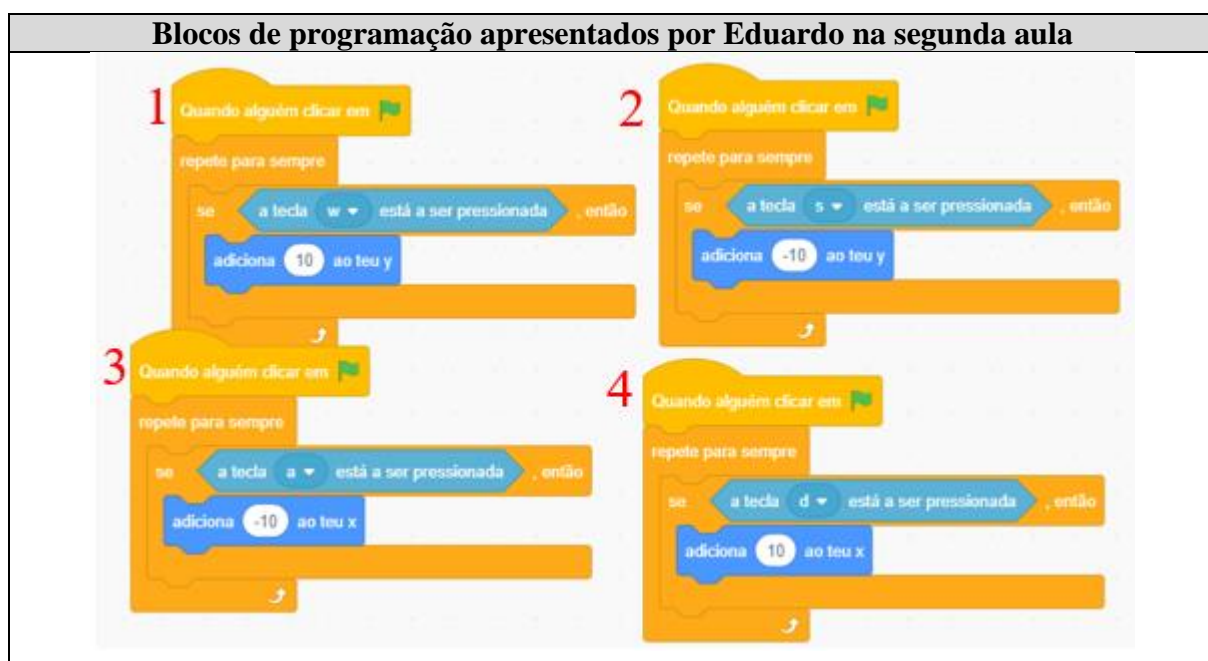
## **5.2. Episódio 2: Eduardo reconhece padrões e os duplica**

Nesta seção vamos apresentar parte da trajetória de Eduardo, que, mesmo não participando do primeiro encontro, envolveu-se com dois projetos no *Scratch* durante o período da produção de dados. No primeiro, intitulado “Tales of SadBoy (In-dev)”, o estudante programou um jogo no qual um ator principal podia ser movimentado por diferentes cenários e interagir com alguns ambientes lá presentes. Já o segundo foi construído com base em um jogo já existente e teve a utilização de GIF presentes na sua programação, algo que o professor não sabia que poderia ser feito.

### 5.2.1. Apresentando a primeira ideia de Eduardo

No dia 18 de agosto de 2020, durante o segundo encontro, Eduardo, que não tinha microfone, fez o seguinte comentário no chat do *Google Meet*: “*sor eu já tive uma ideia de fazer um jogo no estilo bullethell igual undertale*” e, logo depois, apresentou parte do programa que estava desenvolvendo utilizando o *Scratch* em Português de Portugal, conforme podemos verificar no Quadro 6. Apesar de o estudante já ter programado quatro blocos que seguem o mesmo padrão — iniciavam com o código cabeça [quando alguém clicar em ‘bandeira verde’] e seguiam com uma estrutura de decisão [repita para sempre] [se] [a tecla w está a ser pressionada], então [adiciona 10 ao teu y], com alterações nas teclas e nos valores adicionados aos eixos no outro — o programa não estava completo. O professor notou que o ator principal, circulado em vermelho no quadro abaixo, fazia a movimentação conforme o programa, mas se mantinha sempre com o mesmo esboço. A partir disso, o professor comentou que já tinha visto projetos no *Scratch* que usavam várias fantasias para dar a impressão de animação e que elas mudavam conforme a tecla pressionada. Com isso, Eduardo escreveu no chat do *Meet* “*era isso que eu ia perguntar sor*”, mostrando que esse já era um possível problema a ser solucionado. Além disso, outro problema também ficou em aberto neste encontro: a programação para a troca de cenários, ou seja, como fazer a interface do programa mudar no momento desejado pelo programador.

Quadro 6 - Primeiro programa apresentado por Eduardo







Fonte: Dados da Pesquisa (2020).

Nos dados apresentados no Quadro 6, notamos que o estudante já se envolveu com alguns aspectos relacionados à matemática, como o Plano Cartesiano e a soma de inteiros ao usar o bloco [adicione -10 ao teu y]. Além disso, o aluno já construiu um Algoritmo para a ação de movimentação do seu ator, dividindo-o em quatro blocos de programação (blocos 1, 2, 3 e 4 no Quadro 6) que seguem um padrão, características importantes para o início do desenvolvimento dos pilares Algoritmos, Reconhecimento de Padrões e Decomposição do Pensamento Computacional.

Considerando que em um ambiente construcionista o estudante precisa estar envolvido com projetos ou solucionar problemas que sejam do seu interesse, a frase “*sor eu já tive uma ideia de fazer um jogo no estilo bullethell igual undertale*” evidenciou esta característica. A Dimensão Sintônica do Construcionismo destaca a importância em desenvolver atividades que estejam associadas à realidade ou aos interesses dos estudantes, oportunizando, assim, que o aluno participe da escolha do tema do projeto que será desenvolvido (MALTEMPI, 2012). Eduardo continuou com o desenvolvimento do seu projeto referente ao tema escolhido e de outros durante os encontros e pôde, a partir disso, abranger diversos aspectos relacionados ao Pensamento Computacional e à matemática, conforme destacados nas próximas subseções.


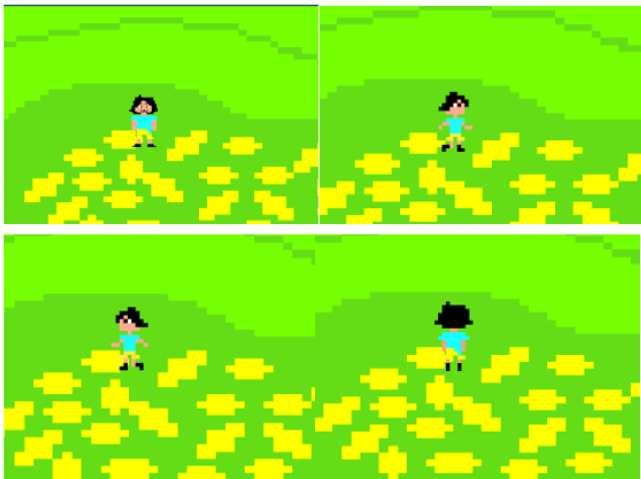
### 5.2.2. Problemáticas diferentes, soluções similares

No terceiro encontro já foram apresentadas partes da solução de Eduardo para os problemas relacionados à movimentação do ator que foram vistos anteriormente. O primeiro,

que consistia na criação de uma animação para a movimentação do ator principal, foi resolvido a partir de um incremento dos códigos que já tinham sido construídos, como podemos ver no Quadro 7.

Eduardo utilizou o mesmo algoritmo para todos: [quando bandeira verde for clicado], [sempre], [se [tecla a pressionada] então], [mude para fantasia \_\_\_], [adicione \_\_\_ a x] ou [adicione \_\_\_ a y], fazendo algumas modificações no código [mude para fantasia] e no de movimento pelo Plano Cartesiano. Além disso, podemos ver na interface do programa como o ator principal se comportava a partir do que foi programado pelo Eduardo, também no Quadro 7.

**Quadro 7 - Parte da solução da problemática de Eduardo**

Blocos de programação apresentados por Eduardo na terceira aula	
	
Interface do programa do Eduardo na terceira aula para cada um dos movimentos do seu ator principal	
	

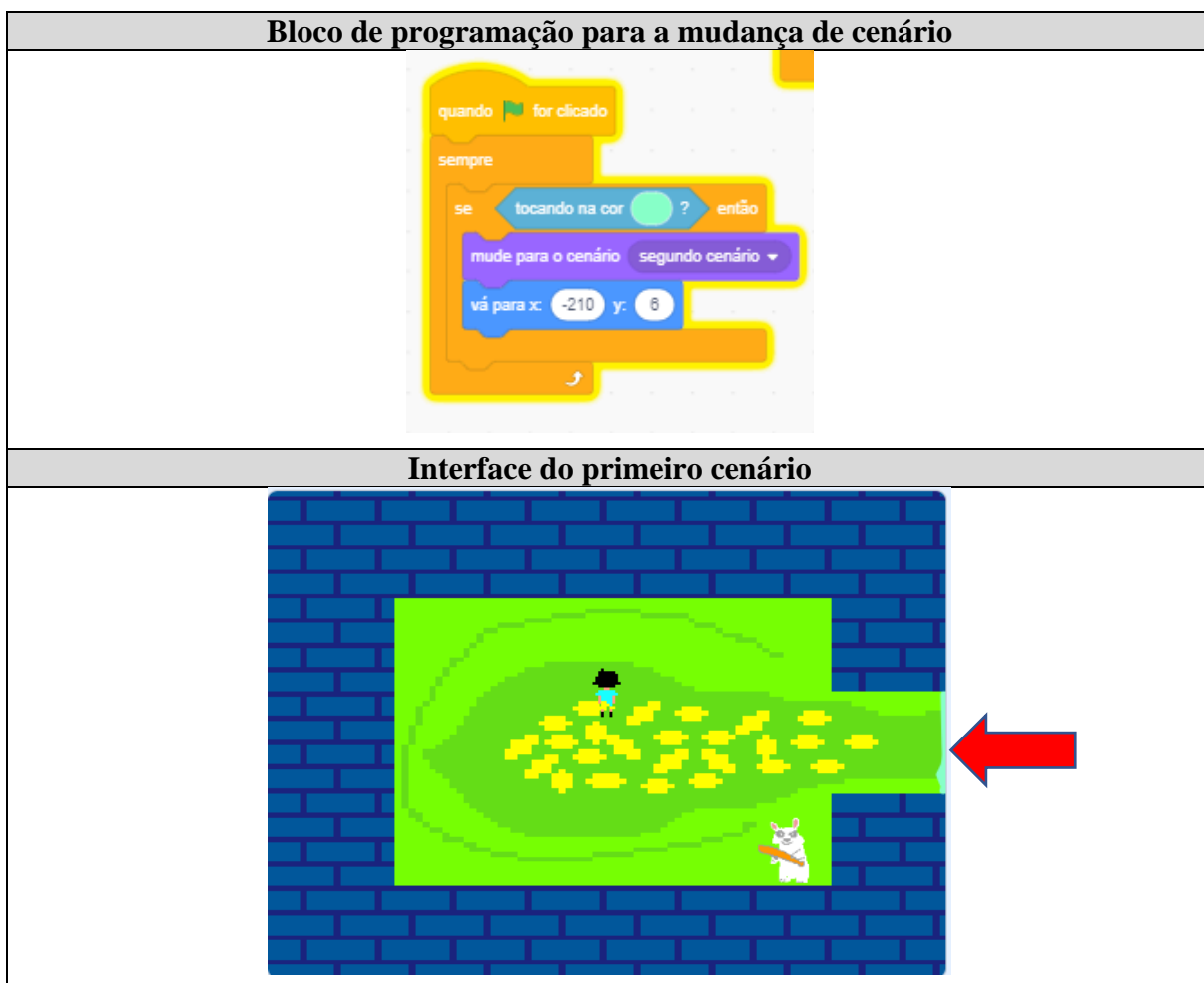
Fonte: Dados da Pesquisa (2020).

Quando questionado sobre como programou cada um dos blocos, Eduardo comentou que usou a função “duplicar” do *Scratch* após programar um dos blocos e ver que funcionava

como desejado, isto é, repetiu o mesmo algoritmo para as outras movimentações, apenas modificando alguns detalhes.

No Quadro 8, podemos conferir detalhes do algoritmo criado por Eduardo para a mudança de cenário. O bloco construído, assim como os outros, também inicia com [quando bandeira verde for clicado] e segue para uma estrutura de decisão sempre-se-então. Diferentemente do bloco da movimentação, aqui temos, depois do [se], o código [tocando na cor \_\_\_?], que funciona como um sensor para que a ação logo depois do [então] aconteça somente quando o ator principal tocar na cor definida. Ao invés do [mude para fantasia \_\_\_], contamos com o código [mude para o cenário \_\_\_], pois a modificação não era no ator. Por fim, temos um código de movimentação também diferente do da movimentação: [vá para x: -210 y: 6]. Na interface do primeiro cenário, podemos notar que a cor utilizada no sensor está à direita do cenário, destacada, no Quadro 8, por uma seta.

**Quadro 8 - Troca de cenários programada por Eduardo**



Fonte: Dados da Pesquisa (2020).

Visando destacar os aspectos matemáticos que podem ser evidenciados nos dados do Quadro 7 e do Quadro 8, olhamos para os diferentes modos que Eduardo usou o Plano Cartesiano para movimentar o seu ator pelos cenários. Nos blocos de programação do Quadro 7, Eduardo programou a movimentação do seu ator principal na estrutura de decisão utilizando os códigos [adicione \_\_\_ a x] e [adicione \_\_\_ a y], ou seja, valores inteiros escolhidos pelo estudante faziam com que o ator se movimentasse por um eixo ou outro, dependendo da tecla pressionada, fazendo com que a nova posição fosse a posição anterior mais o incremento escolhido. Exemplificando, caso o ator estivesse na posição (10,10) e o usuário pressionasse a tecla 'w', o bloco com [adicione 5 a y] era ativado e a nova posição seria (10, 15). Já no bloco apresentado no Quadro 8, a movimentação do ator, que também parte de uma estrutura de decisão, é ativada a partir de um sensor [tocando na cor \_\_\_], porém ela não varia dependendo da posição anterior do ator, pois o código [vá para x: \_\_\_ y:\_\_\_] transporta o ator imediatamente para as coordenadas (x,y) programadas.

Além da matemática utilizada no desenvolvimento do programa, temos indícios de que, a partir da solução do primeiro problema com a estrutura de decisão e o uso da movimentação no Plano Cartesiano por meio do código [adicione \_\_\_ a x], Eduardo aplicou ideias do Reconhecimento de Padrões para solucionar outro problema. De acordo com Brackmann (2017), este pilar traz a ideia de que novos problemas podem ser resolvidos mais rapidamente com base na solução e na experiência de outros já solucionados.

O primeiro problema foi referente à animação do ator principal. Nos dados apresentados no Quadro 7, podemos verificar que o estudante usou o mesmo algoritmo com pequenas modificações para cada um dos movimentos: estrutura de decisão sempre-se-então, seguida dos códigos [tecla \_\_\_ for pressionada], [mude para a fantasia \_\_\_] e [adicione \_\_\_ a x] ou [adicione \_\_\_ a y]. Ou seja, dentro de um mesmo problema, tivemos um padrão para a solução com pequenas modificações.

Nomeando o Reconhecimento de Padrões como Generalização, Ribeiro, Foss e Cavalheiro (2020) trazem a ideia de que, muitas vezes, um problema novo pode parecer totalmente diferente de qualquer outro que o estudante já tenha solucionado, entretanto, em alguns casos, ele pode ser solucionado a partir de pequenas modificações em alguma solução já existente, como um algoritmo que já foi programado para outro problema. O segundo problema de Eduardo consistiu na programação da mudança de cenário da interface do seu programa. Aparentemente diferente do primeiro, que era referente à movimentação do ator principal, ele foi solucionado a partir de algumas modificações no algoritmo criado para o

anterior. No Quadro 8, podemos verificar que o estudante utilizou novamente a estrutura de decisão sempre-se-então, porém os códigos que faziam parte desta estrutura eram outros: [tocando na cor \_\_\_], [mude para cenário \_\_\_] e [vá para x: \_\_\_ y:\_\_\_]. Além disso, a solução que pode ter sido criada a partir do primeiro problema também foi modificada para as diferentes trocas de cenário, como podemos verificar na Figura 24, com blocos de programação que seguem o mesmo padrão, apenas com algumas alterações nas cores, nos nomes dos cenários e nas coordenadas cartesianas.

**Figura 24 - Bloco de Programação para as diferentes trocas de cenário**



Fonte: Dados da Pesquisa (2020).

É importante destacar que os dois problemas solucionados a partir do Reconhecimento de Padrões envolveram aspectos matemáticos nas suas construções. A estrutura de decisão sempre-se-então pode ser transposta para a seleção de qual fórmula/ algoritmo deve ser utilizado na resolução de algum exercício ou problema. Exemplificando, sempre que uma equação for do tipo  $ax^2 + bx + c = 0$ , com  $a, b, c \in R$  e  $a \neq 0$ , podemos utilizar a Equação Resolvente do Segundo Grau<sup>25</sup> para encontrar o conjunto solução. Além disso, o estudante utilizou a movimentação no Plano Cartesiano de duas formas. No primeiro problema foi a partir da incrementação de valores ao eixo das abscissas com os códigos [adiciona 10 ao teu x] e [adiciona -10 ao teu x] e ao eixo das ordenadas com [adiciona 10 ao teu y] e [adiciona -10 ao teu y]. Porém, já no outro problema, a movimentação no plano foi feita pelos códigos contendo pontos na forma (x,y) por meio do código [vá para x:\_\_\_ y:\_\_\_].

Na próxima subseção, veremos outro programa construído por Eduardo no qual os padrões encontrados nos dois problemas apresentados até o momento foram utilizados na construção de outros algoritmos.

<sup>25</sup> Equação Resolvente do Segundo Grau:  $x = \frac{-b \pm \sqrt{(b^2 - 4ac)}}{2a}$ .

### 5.2.3. Novas construções de Eduardo a partir dos problemas solucionados

No sexto encontro, ocorrido no dia 16 de setembro de 2020, Miguel comentou que viu um programa construído por Eduardo no estúdio criado na comunidade *on-line* do *Scratch* para o projeto. A partir disso, o professor pediu para que Eduardo apresentasse para o grupo o que tinha desenvolvido.

O novo projeto de Eduardo, intitulado “Fan-made | Deltarune Ralsei vs Kris”, mostrava ações que ainda não tinham sido programadas nas construções dos estudantes até o momento. Conforme pode ser visto no Quadro 9, Eduardo foi questionado pelo professor sobre como ele tinha programado alguns aspectos, além de terem conversado sobre a utilização de *Graphics Interchange Format*<sup>26</sup>(GIF) no programa. Sobre a construção, Eduardo disse “*eu fiz inteiro.... só a colisão com a barra branca eu tive que pesquisar... a vida eu peguei pronta do jogo em si... eu não terminei ainda*”. Além disso, quando questionado sobre a “*vida*” e o “*movimento de vento*”, Miguel comentou que achava que era um GIF e o professor pensava que não existia a possibilidade de utilizar esse recurso no *Scratch*, entretanto, Eduardo confirmou que foi isso mesmo que fez: “*se botar um GIF... ele bota todas as fantasias*”.

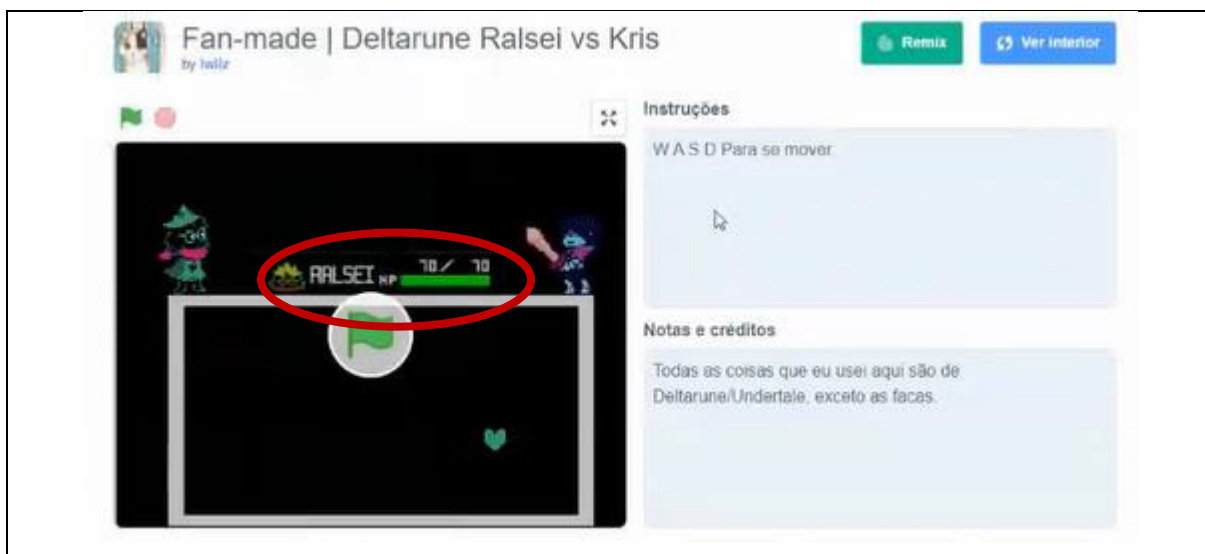
**Quadro 9 - Outro projeto de Eduardo**

<b>Excerto referente a construção do novo programa</b>
<b>Professor:</b> como você programou isso? Fez tudo sozinho ou usou programas já criados?
<b>Eduardo:</b> eu fiz inteiro... só a colisão com a barra branca eu tive que pesquisar... a vida eu peguei pronta do jogo em si... eu não terminei ainda.
<b>Professor:</b> Muito legal! Agora eu quero saber várias coisas. Como você fez a tua ‘vida’? (circulado em vermelho na interface do novo programa)... Esse movimento de ‘vento’ nos teus personagens, como vocês fez?
<b>Miguel:</b> Eu acho que são ‘gifs’ sor.
<b>Professor:</b> Eu acho que não tem como colocar gif no <i>Scratch</i> .
<b>Miguel:</b> Não tem?
<b>Professor:</b> Acho que não. Acredito que ela tenha programado esse efeito
<b>Eduardo:</b> se botar um gif... ele bota todas as fantasias.

#### **Interface do novo programa**

---

<sup>26</sup> “De modo geral, o GIF é um formato de imagem, assim como JPG e PNG. Na prática, ele pode ser utilizado tanto para imagens estáticas quanto para imagens animadas” (GAIATO, 2021).




Fonte: Dados da Pesquisa (2020).

Entre as Práticas Educacionais relacionadas ao Pensamento Computacional elaboradas por Brennan e Resnick (2012), a ação de Reutilizar e Remixar destaca a importância do processo de construir programas a partir dos projetos compartilhados na comunidade *on-line* do *Scratch*. Brennan e Resnick (2012) ressaltam que, além de desenvolver a criticidade dos estudantes, este processo pode possibilitar o reconhecimento de alguns padrões na programação de certas ações, aspecto importante para o desenvolvimento do Pensamento Computacional.

Destacamos também o momento no qual Eduardo ensinou ao professor como utilizar GIF no *Scratch*. Observamos que este aspecto está relacionado ao ambiente construcionista no qual os encontros estavam acontecendo, pois o estudante, como um participante ativo do processo de aprendizagem, pode trazer suas ideias e compartilhá-las com os outros. Além disso, notamos que a atitude de Eduardo está relacionada à formação de um aluno “X” (RESNICK, 2020), pois ele esteve disposto a experimentar algo que não sabia se funcionaria e assumir o risco nesse processo, sem considerar as “grades das gaiolas” que poderiam existir naquele momento (ROSA, 2019).

No Quadro 10, podemos ver um dos blocos de programação para a animação do “vento” destacada no excerto anterior. Eduardo iniciou com [quando bandeira for clicado], seguindo com [sempre] e a dupla [mude para a fantasia \_\_\_] e [espero 0.1 seg] diversas vezes. No excerto presente no mesmo quadro, verificamos que Eduardo construiu o algoritmo aos poucos, testando as possibilidades para ir corrigindo os bugs: “*eu fui testando [...] aos poucos, porque se não aparecia muitos bugs*”.

### Quadro 10 - Sobre a animação utilizando um GIF

Excerto referente a construção do novo programa

Interface do novo programa
<p><b>Professor:</b> Eduardo, sobre os personagens, você só os programou se movendo? É só a animação que está programada?</p> <p><b>Eduardo:</b> é</p> <p><b>Professor:</b> Legal. Vi que usou o quando, sempre, espera... você foi testando o tempo ou colocou direto 0,1 segundo?</p> <p><b>Eduardo:</b> Eu fui testando.</p> <p><b>Professor:</b> Legal, isso é fantástico em programação. É uma coisa que a gente vai fazer, é testar, botar, testar e botar. Teve algum bug que teve que arrumar depois? Fez tudo de uma vez ou aos poucos e foi testado?</p> <p><b>Eduardo:</b> aos poucos, porque se não aparecia muitos bugs.</p>

Fonte: Dados da Pesquisa (2020).

De acordo com Vicari (2018), algumas perguntas podem servir de apoio na utilização do Reconhecimento de Padrões para solucionar um problema, como “este problema é similar a um outro que já tenhas resolvido” ou “em que pontos o atual problema é diferente do outro que já tenhas resolvido?” (VICARI, 2018, p. 32). Não temos, nos dados produzidos, evidências de que Eduardo se guiou por esses questionamentos, mas comparando com o primeiro problema apresentado na subseção anterior, podemos enxergar a similaridade no problema e na solução.

No problema da animação do movimento do ator principal, Eduardo usou a estrutura de decisão sempre-se-então, além de outros códigos, inclusive envolvendo matemática, para encontrar a solução. Aqui o estudante fez algo parecido: programar a animação do ator sem depender da ação do usuário, pois seria o “vento” dentro do programa. Assim, Eduardo também



iniciou com o código da bandeira verde e o [sempre], mas, diferentemente do primeiro problema, programou a troca de fantasias automaticamente por meio da contagem do tempo repetindo os códigos [mude para fantasia \_\_] e [espere \_\_ seg].

Além disso, no processo de usar o mesmo padrão para solucionar outro problema, Eduardo se envolveu com outra Prática Educacional ressaltada por Brennan e Resnick (2012), que se relaciona tanto com o Pensamento Computacional quanto com o ambiente construcionista: Teste e Depuração. Este processo, associado às ideias de Papert (1988), realça a importância de o estudante estudar o erro e procurar uma solução para tal a partir de tentativas, ao invés de desistir do que está fazendo. Evidências de que Eduardo passou por esse processo podem ser vistas nas falas “*eu fui testando*” e “*aos poucos, porque se não aparecia muitos bugs*”, no excerto do Quadro 10.

#### 5.2.4. Como aspectos matemáticos e do Pensamento Computacional emergiram nos projetos desenvolvidos por Eduardo

A partir da análise realizada nas subseções anteriores que se referem ao estudante Eduardo, temos evidências de que o Reconhecimento de Padrões e o Algoritmo, dois pilares do Pensamento Computacional, estiveram presentes no processo de construção de programas no *Scratch*. Além disso, o principal aspecto matemático destacado nos algoritmos construídos foi o Plano Cartesiano, utilizado de formas diferentes conforme a necessidade do problema.

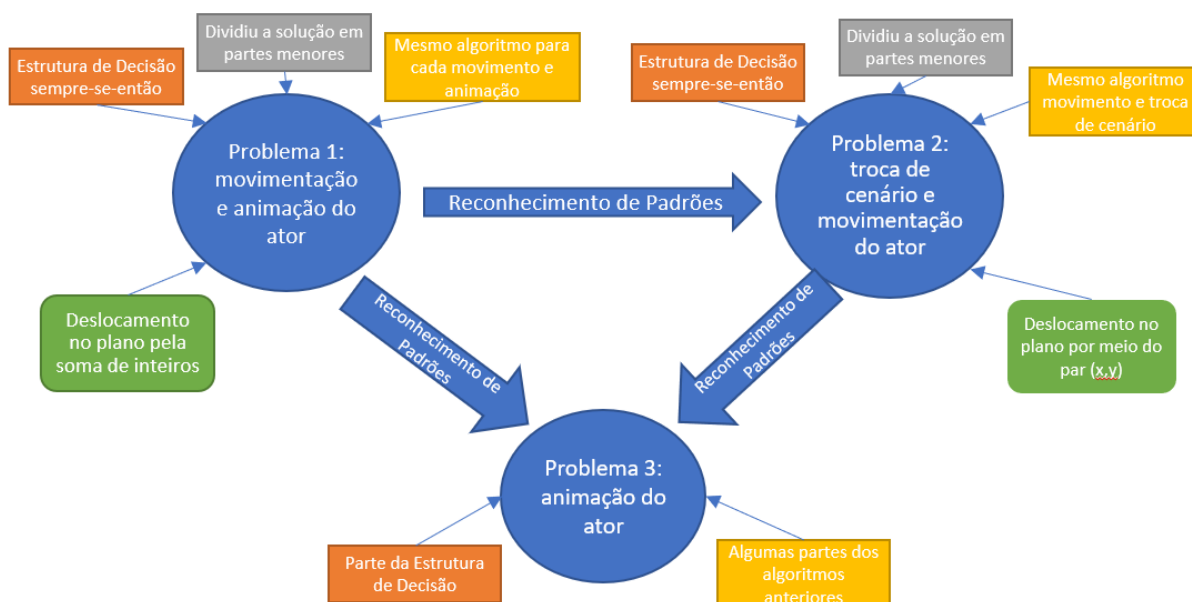
Na Figura 25, apresentamos uma condensação de como esses aspectos se relacionaram na solução dos três problemas apresentados anteriormente. O Problema 1, que se referia à movimentação e animação do ator, foi solucionado a partir da construção de blocos de programação com a estrutura de decisão sempre-se-então. Além disso, foi solucionado a partir da divisão da solução em partes menores, pois para cada sentido do movimento foi criado um bloco de programação e um mesmo algoritmo dentro de cada bloco para o movimento e a animação, no qual havia o deslocamento do ator pelo Plano Cartesiano por meio da soma de inteiros.

Já no Problema 2, tivemos uma mesma estrutura de decisão, o algoritmo dividido em partes menores, uma para cada sensor de cor, e um mesmo algoritmo em cada bloco com pequenas diferenças do que foi utilizado no Problema 1. Entre as diferenças temos o código [tecla \_\_ for pressionada], que foi substituído pelo [tocando na cor \_\_], o uso do [mude para o

cenário \_\_], ao invés do [mude para fantasia \_\_], e o deslocamento no plano que agora é pela substituição do par de coordenadas (x,y) por meio do código [vá para x:\_\_\_ y:\_\_\_], ao invés do deslocamento em cada eixo com [adicione \_\_ a x] e [adicione \_\_ a y].

Por fim, podemos notar que a solução da animação do ator no Problema 3 também envolveu aspectos dos algoritmos criados anteriormente, como a utilização de parte da estrutura de decisão e do código [mude para fantasia \_\_], utilizado no Problema 1. Entretanto, ele precisou ser modificado para que funcionasse corretamente. Ao invés de esperar uma tecla ser pressionada ou o ator tocar em alguma parte do cenário, as mudanças na fantasia foram programadas pela contagem do tempo com o código [espere \_\_ seg].

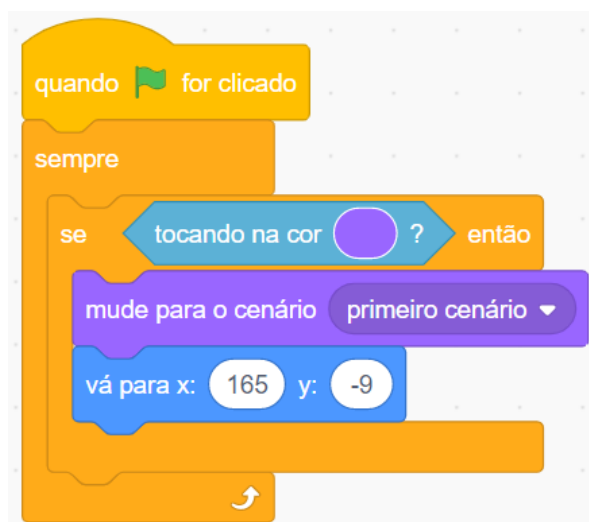
**Figura 25 - Fluxograma com aspectos envolvendo a problemática de Eduardo**



Fonte: Elaborada pelo autor.

Observamos que a matemática teve um papel fundamental na solução do Problema 2, que aconteceu por meio do Reconhecimento de Padrões relacionados ao anterior. Em ambos os problemas, tínhamos a modificação da posição do ator principal. No primeiro, o ator precisava se deslocar por meio das coordenadas cartesianas em apenas um sentido, dependendo de qual tecla fosse pressionada. Por exemplo, caso a tecla “s” fosse pressionada, o algoritmo acionava a parte da estrutura de decisão com o código [adicione -5 a y], ou seja, o deslocamento seria apenas no sentido negativo do eixo das ordenadas. Já no segundo problema, o deslocamento, acionado agora dependendo do contato do ator com uma das cores do cenário, precisava acontecer de outra forma, não dependendo da posição anterior. Exemplificando, assim que o ator principal tocasse na cor roxa, ele era enviado automaticamente para o ponto de coordenadas (165, -9), além da troca de cenário também programada, conforme Figura 26.

Figura 26 - Bloco de programação com deslocamento por pontos do Plano Cartesiano



Fonte: Dados da Pesquisa (2020).

Assim como na análise dos dados do Episódio 1, notamos que o Pensamento Computacional e a matemática agem de forma complementar. Ribeiro, Foss e Cavalheiro (2020) salientam que, a partir de algumas modificações, a busca por soluções de problemas que podem parecer totalmente diferentes pode ser realizada a partir de pequenas modificações em alguma solução que já tenha sido descoberta ou construída. Algumas das modificações feitas a partir do reconhecimento de que um problema poderia ser solucionado por meio do algoritmo criado para outro estavam diretamente ligadas ao aspecto matemático, como o deslocamento no Plano Cartesiano.

Na seção 3 abordaremos os aspectos relacionados ao Pensamento Computacional e à matemática na trajetória de Paulo, denominado, neste trabalho, Episódio 3.

### 5.3. Episódio 3: Paulo e o processo de depuração

Neste episódio, vamos descrever como o estudante Paulo lidou com alguns *bugs* que surgiram na construção do seu programa, que possuía algumas características semelhantes aos projetos de Miguel e de Eduardo. Um dos *bugs* estava relacionado à movimentação de rotação do seu ator principal, já o outro apareceu na programação da troca de cenários, no qual a solução foi encontrada com base no auxílio de outro estudante que tinha feito algo similar.

Paulo não participou dos dois últimos encontros, pois, segundo ele, estava com problemas no seu computador. Mesmo assim, o estudante conseguiu desenvolver parte de um projeto que intitulou como “Marquinhos” e, a partir dele, destacamos algumas características


envolvendo *bugs* encontrados durante os encontros e como a matemática e o Pensamento Computacional emergiram durante o desenvolvimento e o processo de depuração.

### 5.3.1. Movimento de Rotação e Reflexão

O programa no qual Paulo estava envolvido tinha algumas características que se assemelhavam ao programa de Eduardo, como a movimentação do ator pelo cenário com mudanças na fantasia conforme a tecla pressionada, onde foi encontrado o primeiro *bug*. O primeiro bloco de programação construído para a movimentação do seu ator principal pode ser visto no Quadro 11. O estudante iniciou com o código cabeça [quando a tecla d for pressionada], seguiu para o [defina o estilo de rotação para rotação completa] junto do [mude para a fantasia pixi-frame-0(4)] e [mova 5 passos]. O *bug* foi avistado, pois, dependendo da tecla que era pressionada para o ator se movimentar, ele ficava virado para baixo.

O professor observou que o erro estava no código [defina o estilo de rotação para rotação completa], pois ele fazia com que o ator girasse no sentido horário em torno do seu centro, quando o que precisava acontecer era um movimento de reflexão em torno do seu eixo vertical, como desejado. Com isso, ele sugeriu que todos olhassem para a categoria “movimentos” do *Scratch* e tentou fazer com que Paulo descobrisse qual código deveria utilizar, conforme excerto apresentado no Quadro 11. Assim, após a conversa e a sugestão do professor de que ele queria que o ator apontasse para o outro lado, Paulo modificou o seu bloco de programação, conforme circulado em verde no Quadro 11 e no Quadro 12, solucionando, assim, a problemática.


Quadro 11 - Primeira problemática de Miguel

Bloco de códigos para a movimentação do ator principal

Excerto
<p><b>Professor:</b> Paulo usou o ‘defina o estilo de rotação’, qual outro será que ele poderia usar para só virar e não acontecer isso (‘isso’ me refiro ao ator virar de cabeça para baixo)</p> <p><b>Paulo:</b> vire sei lá o que quantos graus.</p> <p><b>Professor:</b> o que você quer? Quer que ele gire ou...</p> <p><b>Paulo:</b> quero que ele mude de... eu quero que ele olhe para trás.</p> <p><b>Professor:</b> que ele na verdade só aponte para o outro lado.</p> <p><b>Paulo:</b> opaaa, já achei.</p>

Fonte: Dados da Pesquisa (2020).

Embora o erro tenha sido corrigido com a modificação do bloco de programação, conforme podemos ver no Quadro 12, outro *bug* surgiu. Paulo notou que o seu ator principal não estava mais ficando virado para baixo, porém, dependendo da situação, ele não estava se movimentando para a direção que gostaria que fosse. Por exemplo, a ação de pressionar a tecla “d” para que o ator fosse para a direita fazia com que ele fosse para a esquerda. O professor observou que o *bug* estava no código [mova -5 passos], circulado em vermelho no Quadro 12. Com isso, ele tentou direcionar o estudante para que ele mesmo encontrasse o erro e elaborasse uma possível solução, conforme pode ser visto no excerto do mesmo quadro, no qual o estudante fala que entendeu e explica qual tinha sido o *bug*: “*é que eu estava apontando para o lado errado e ele estava andando para trás pois estava negativo*”.

Quadro 12 - Solução da primeira problemática de Miguel

Bloco de códigos para a movimentação do ator principal

Excerto
<p><b>Professor:</b> Paulo, ele está apontando para a direção, quando você aperta A... (Paulo me interrompe) <b>Paulo:</b> ah, agora entendi (modificando o valor, neste instante, para 5) <b>Professor:</b> o que você entendeu? Por que estava dando errado? <b>Paulo:</b> é que eu estava apontando para o lado errado e ele estava andando para trás pois estava negativo.</p>

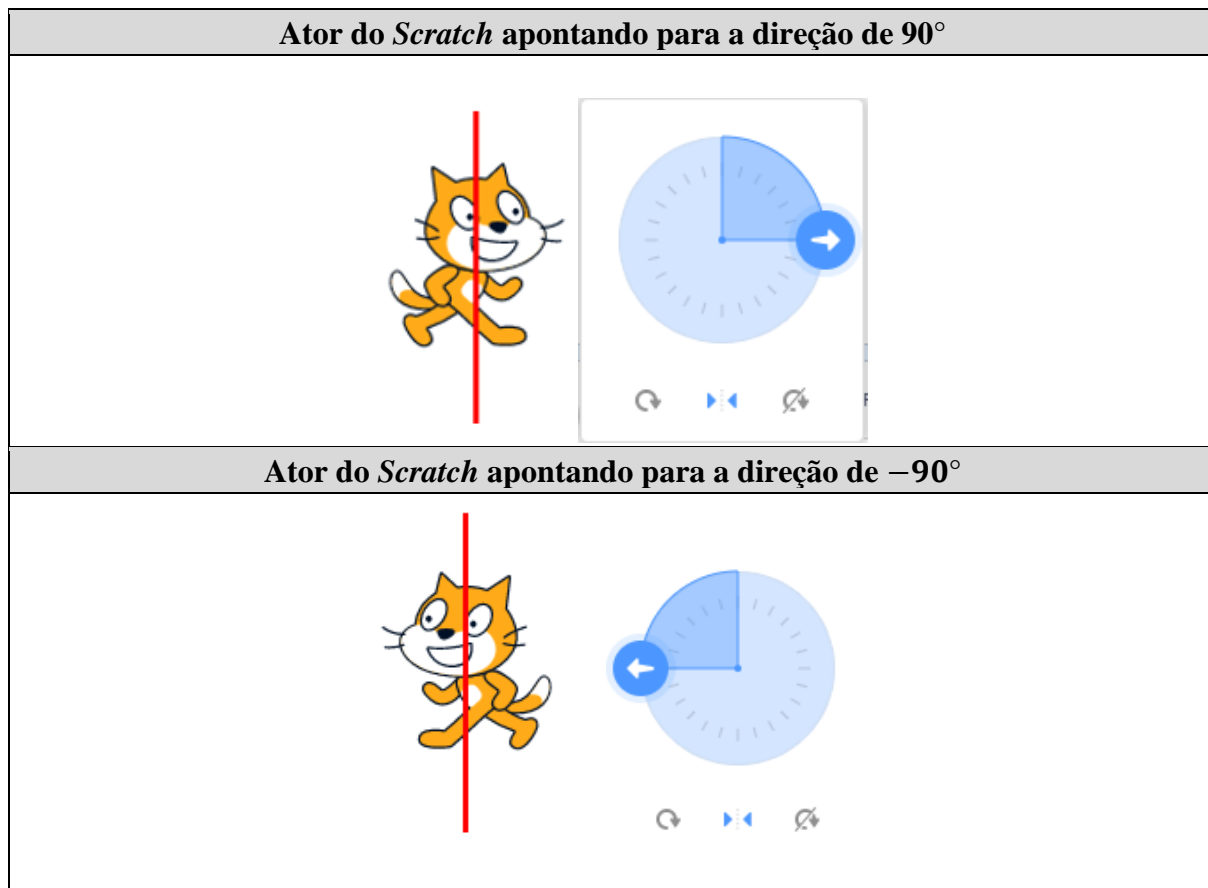
Fonte: Dados da Pesquisa (2020).

Mesmo não sendo um dos pilares, o processo de correção de erros ou *bugs* é, segundo Wing (2006), uma característica importante do Pensamento Computacional. Brennan e Resnick (2012), que nomeiam esta prática Teste e Depuração, destacam que ela pode funcionar como um convite para o estudante estudar o erro e procurar uma forma de solucioná-lo, ao invés de desistir do que está fazendo. Papert (1988) salienta que, na escola, o erro é visto como algo negativo e que o estudante não é incentivado a examinar o que aconteceu ou até mesmo a refletir sobre ele, entretanto, em um ambiente construcionista no qual o *Scratch* faz parte, os erros ou *bugs* podem guiar o estudante ao processo de depuração. Neste processo não há julgamento sobre o que é certo ou errado, pois os erros são vistos como algo que pode levar o estudante a refletir sobre o que aconteceu, entender como aconteceu e, a partir disso, corrigi-lo, ou seja, trazer benefícios (PAPERT, 1988).

Nota-se que os dois erros/*bugs* que apareceram no programa de Paulo estavam relacionados a aspectos matemáticos. O primeiro, sobre o código de movimento de rotação circulado em verde no Quadro 11, o qual a solução foi a troca por outro código, também circulado em verde, mas no Quadro 12, referem-se aos diferentes movimentos do ator principal em torno do seu centro ou dos seus eixos. O código [defina o estilo de rotação para rotação completa] fazia com que o ator principal girasse em torno do seu centro e, conforme Quadro

13, a solução foi utilizar o código [aponte para a direção -90], que fazia com que o ator apenas apontasse para a direção -90° em um movimento de reflexão em torno do seu eixo vertical.

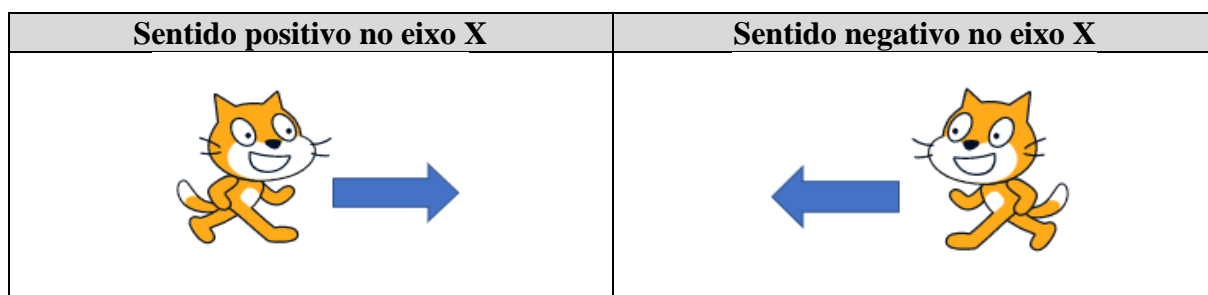
**Quadro 13 - Reflexão em torno do próprio eixo vertical**



Fonte: elaborado pelo autor.

Já o segundo estava relacionado ao movimento do ator principal no plano a partir dessa direção. O uso do código [mova \_\_ passos] faz com que o ator se movimente dependendo do sentido no qual ele está e não a sua posição. No Quadro 14, mostramos algumas possibilidades para esse movimento, no qual a seta realça o sentido do ator. Quando o ator está no sentido positivo do eixo X, o código [mova 5 passos] faz com que ele se mova 5 passos à direita, já se ele estiver no sentido negativo do eixo X, o mesmo código faz com que ele se mova à esquerda.

**Quadro 14 - Possibilidades de movimento no eixo X do Scratch**



Fonte: elaborado pelo autor.

Não houve, na solução dos dois problemas, discussões sobre os eixos de rotação ou até mesmo sobre o sentido no eixo X. Todavia, eles foram solucionados a partir do entendimento do que estava acontecendo e do que deveria acontecer. No primeiro problema, Paulo soube que tinha relação com o giro, pois ele fala “*virar sei lá quantos graus*”, apesar disso, ele consegue solucioná-lo a partir da dica do professor e, sozinho, acrescenta o código [aponte para direção -90] no seu bloco de programação. Já no segundo, Paulo traz a noção de sentido positivo ou negativo ao dizer “*é que eu estava apontando para o lado errado e ele estava andando para trás pois estava negativo*”. Estes dados evidenciam características de um ambiente construcionista por envolver a Dimensão Semântica (PAPERT, 1986), ou seja, a matemática que precisou ser manipulada teve um significado para o estudante, pois estava estritamente ligada ao problema e ao seu projeto.

Buscamos, no ambiente construcionista que montamos com o projeto “Programação com *Scratch*”, não deixar de lado os *bugs* que apareciam durante o desenvolvimento dos projetos. Os participantes foram convidados para refletir sobre os seus programas durante os encontros e, a partir da análise desses *bugs*, observamos que aspectos relacionados à matemática e ao Pensamento Computacional emergiam, confirmando assim que o processo de depuração pode ser favorável no processo da construção do conhecimento. Além disso, consideramos que esse aspecto pode possibilitar que o estudante passe a observar outros *bugs* ou erros, além dos que estão presentes na programação, de forma benéfica, de modo que, ao tentar compreendê-los ou estudá-los, possa também se beneficiar nesse processo.

Na próxima subseção apresentamos como Paulo, com o auxílio de seus colegas, solucionou um problema referente à troca de cenários no *Scratch*, destacando os aspectos relacionados ao Pensamento Computacional, à matemática e ao ambiente construcionista.

### 5.3.2. Correção de *bugs* em pares

No terceiro encontro, Eduardo explicou aos seus colegas como configurou a mudança de cenários em seu programa. Após a explicação, o professor questionou se Paulo tinha entendido e ele salienta que não. Assim, o professor sugeriu que Eduardo e Miguel explicassem para ele, conforme podemos ver as orientações no excerto abaixo:

**Miguel:** *O Eduardo, para mudar o cenário, ele botou uma cor no final do cenário que ...*

**Paulo:** *ah, boa, ta...*



**Eduardo:** *faz um ator e bota ele no final do mapa, ai bota quando ele encostar nesse tal ator vai para o próximo cenário.* (Eduardo escreve isso no chat, ao utilizar o ‘bota’, Eduardo está se referindo ao código que precisa ser programado)

A partir dessas orientações, Paulo construiu o seu bloco de programação, conforme pode ser visto no Quadro 15, iniciando com o código cabeça [quando bandeira verde for clicado] e seguindo para [se][tocando na cor roxa][mude para cenário pixi-frame-0(1)2]. No mesmo quadro, podemos ver a interface do programa de Paulo, com a faixa de cor roxa pintada à direita no cenário. Assim que Paulo terminou de montar o bloco de programação do Quadro 15, o professor comentou que faltava algo e, antes de ele fazer a correção, o estudante Miguel comentou que era o código [sempre], completando assim a estrutura de decisão sempre-se-então.

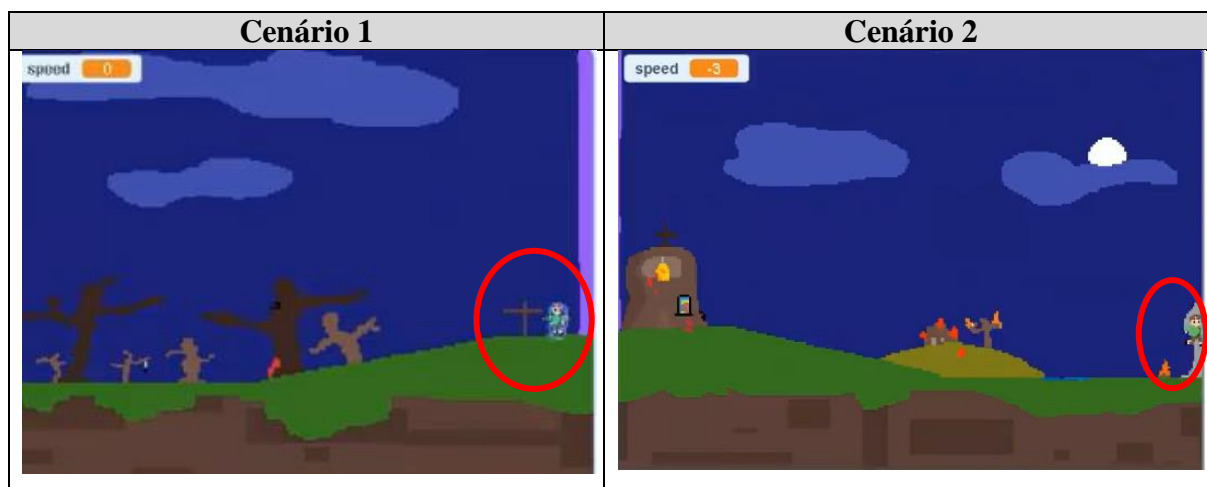
**Quadro 15 - Troca de cenários do programa de Miguel**



Fonte: Dados da Pesquisa (2020).

Mesmo com a troca de cenários funcionando, o programa de Paulo não estava operando conforme deveria. O próximo problema estava na posição do ator principal que, quando o cenário era modificado, continuava no mesmo local (com as mesmas coordenadas), conforme podemos ver nos atores circulados no Quadro 16.

Quadro 16 - Problemática na Troca de Cenários de Miguel



Fonte: Dados da Pesquisa (2020).

Após o professor ter questionado como ele poderia corrigir este problema, Paulo disse que não sabia. Entretanto, outros questionamentos foram feitos pelo professor para tentar guiá-lo na descoberta da solução, conforme podem ser vistos no excerto do Quadro 17. Quando o professor terminou de falar “*Então, além do cenário, configura a posição do ator... além de mudar o cenário, tu tens que fazer mais uma coisa*”, ele não comentou nada, porém selecionou o código [vá para x: \_\_ y: \_\_] e adicionou ao seu bloco de programação. Depois disso, Paulo ativou os códigos que mostram as posições x e y do ator, conforme circulado na imagem da interface do programa do Quadro 17. A partir disso, ele deslocou o seu ator até o local que ele gostaria que ele aparecesse depois da troca de cenários, concluindo que precisava adicionar ao código [vá para x: \_\_ y: \_\_] as coordenadas (-219, -39). No Quadro 17, podemos ver o bloco de programação completo, iniciando com o código cabeça [quando bandeira verde for clicado], seguindo para [sempre], [se [tocando na cor roxa] então] [mude para o cenário arroz doce é ruim], [vá para x: -219 y: -39]. O estudante realizou o mesmo procedimento para programar a volta para o cenário anterior, criando um bloco de programação utilizando outra cor e descobrindo o novo ponto (x, y) deslocando o ator para o local que desejava que ele aparecesse.

Quadro 17 - Solução da problemática da troca de cenários de Miguel

Excerto	
<p><b>Professor:</b> Quando vai no roxinho (cor utilizada como sensor para a troca do cenário), o que você quer que aconteça?</p> <p><b>Paulo:</b> Que ele venha pra cá. (Paulo mostra, com o mouse, a posição no canto esquerdo)</p> <p><b>Professor:</b> Então, além do cenário, configura a posição do ator... além de mudar o cenário, tu tens que fazer mais uma coisa</p>	
Bloco de programação	Interface do programa com as posições x e y



Fonte: Dados da Pesquisa (2020).

De acordo com Azevedo e Maltempo (2020), o fazer e aprender matemática, junto ao desenvolvimento do Pensamento Computacional, podem valorizar diversos aspectos e, entre eles, a instigação à tomada de decisões, sejam elas de forma individual ou coletiva. Notamos, pelos dados apresentados no Quadro 16 e no Quadro 17, que o processo de depuração aconteceu em pares, pois os estudantes, junto ao professor, auxiliaram-se para que o programa funcionasse conforme desejavam. Além disso, o estudante foi convidado a estudar o erro do seu programa e corrigi-lo, característica que, segundo Brennan e Resnick (2012), é importante para o desenvolvimento do Pensamento Computacional.

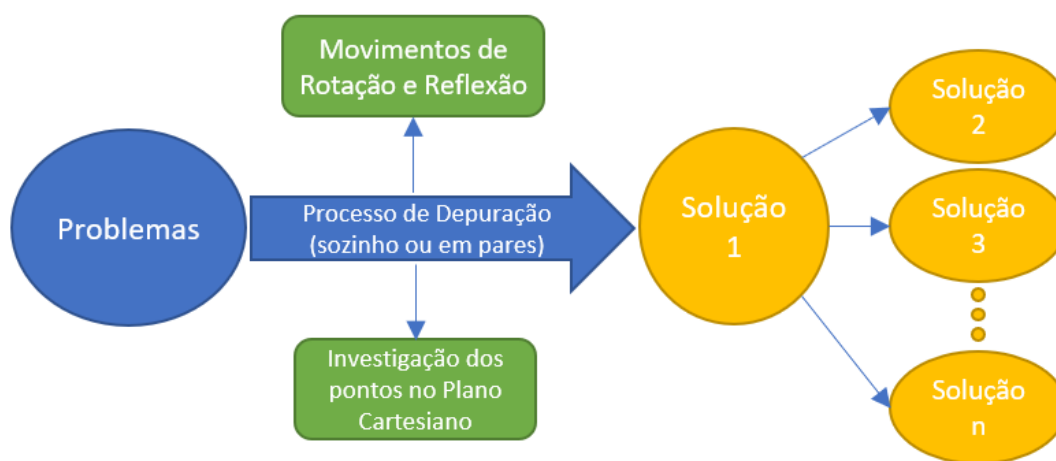
Nesse processo de depuração realizado em pares, Paulo se envolveu principalmente com a movimentação no Plano Cartesiano a partir da investigação dos pontos ao ativar o código que mostrava a posição do ator a cada instante, circulado em vermelho na interface do programa no Quadro 17. Depois do problema solucionado, ele usou a mesma estratégia para programar os outros blocos de movimentação na troca de cenários, característica relacionada ao pilar de Reconhecimento de Padrões (VICARI, 2018; BRACKMANN, 2017; RIBEIRO, FOSS, CAVALHEIRO, 2020).

Na próxima subseção, apresentamos como o Pensamento Computacional e a matemática emergiram nesses processos de depuração na trajetória de Paulo.

### 5.3.3. Como aspectos matemáticos e do Pensamento Computacional emergiram nos projetos desenvolvidos por Paulo

Com base nos dados analisados nas subseções do Episódio 3, nos quais o estudante Paulo se envolveu, podemos evidenciar que o processo de depuração do seu programa, ora realizado sozinho, ora em pares, abrangeu aspectos relacionados à matemática, como rotação, reflexão e movimentação no Plano Cartesiano. A partir disso, a solução foi encontrada e o mesmo padrão foi utilizado para problemas correlatos. Na Figura 27, podemos ver um resumo de como foi o processo de construção de Paulo.

**Figura 27 - Fluxograma com os aspectos envolvidos nas problemáticas de Paulo**



Fonte: Elaborada pelo autor.

Nota-se, a partir do primeiro problema, apresentado no Quadro 11 e no Quadro 12, que o processo de depuração aconteceu por meio da correção de um erro relacionado à rotação do ator principal. Com o código [defina o estilo de rotação para rotação completa], o ator estava girando em torno do seu centro, conforme Quadro 11. Com o processo de depuração, Paulo trocou o código anterior pelo [aponte para a direção -90], fazendo com que o ator apenas apontasse para a direção -90, ou seja, para o sentido negativo do eixo x em um movimento de reflexão em seu eixo vertical. Depois desta correção, o *bug* apareceu com o código [mova \_\_ passos], pois, diferente do [adicione \_\_ a x] ou [adicione \_\_ a y] utilizado por Eduardo e Miguel, o código escolhido dependia do sentido para o qual o ator principal estava apontando e não da sua posição (x,y) no plano.

Já no *bug* relacionado à troca de cenários, apresentado na subseção 5.3.2, além da falta do código [sempre] para completar a estrutura de decisão sempre-se-então, que foi sugerido por Miguel, o problema estava no deslocamento do ator pelo plano, que foi resolvido a partir da

investigação das coordenadas desejadas quando o código que mostrava as posições x e y do ator principal foi ativado. Depois de resolvido, a solução foi replicada para os outros deslocamentos com a troca de cenários.

Destacamos aqui que o processo de depuração do programa ocorreu por causa da matemática envolvida nos dois algoritmos, ou seja, ela foi o problema e a solução. No primeiro, ela estava no movimento em que o ator deveria fazer que, ao invés de girar em torno do seu centro, deveria refletir pelo seu eixo vertical. No segundo problema, o uso do código [mova \_\_\_ passos] considerava o sentido para o qual o ator principal estava apontando, diferenciando, assim, movimentos no sentido positivo ou negativo. Por fim, no terceiro, novamente relacionado à movimentação, o ator precisou investigar quais eram as coordenadas que seriam usadas na movimentação e fez isso utilizando o próprio *Scratch*.

Maltempo (2012) destaca que este processo age de forma a incentivar o estudante a buscar conceitos ou estratégias para que ele possa aperfeiçoar o que já conhece e que, nesta procura, novos conhecimentos podem ser agregados aos já existentes. Assim, o estudante, ao ser encorajado a estudar o seu erro ao invés de esquecê-lo, precisou construir novos conhecimentos a respeito de rotação, reflexão e da movimentação no Plano Cartesiano.

Também temos, mais uma vez, a presença da Dimensão Semântica do Construcionismo (PAPERT, 1986), pois as ações envolvendo aspectos matemáticos e computacionais tinham um significado para o que o estudante estava desenvolvendo. Além disso, após a construção da solução, Paulo replicou-a para os outros blocos de programação, fazendo apenas pequenas modificações nos pontos envolvendo o Plano Cartesiano. Essa ação está relacionada ao pilar do Reconhecimento de Padrões ou Generalização, de acordo com Ribeiro, Foss e Cavalheiro (2020).

Buscamos, na próxima seção, fazer uma sistematização de todos os episódios apresentados até o momento destacando como a matemática e o PC emergiram nos projetos desenvolvidos pelos estudantes com o *Scratch*.

#### 5.4. Sistematização da Análise

A análise dos dados realizada neste capítulo foi guiada de forma que pudéssemos responder à seguinte pergunta: **como o Pensamento Computacional e a matemática emergem de projetos desenvolvidos com o ambiente de programação Scratch em**

**atividades remotas?** Em cada um dos episódios construídos, conseguimos observar que determinados aspectos referentes ao PC e à matemática que, algumas vezes, aparecerem de forma separada, em outras, complementavam-se na busca pela solução do problema.

Evidências apresentadas no Episódio 1 nos mostraram que a Decomposição de problemas não é algo fixo, direto ou linear. Mesmo o estudante construindo seu Algoritmo programando blocos diferentes e que, ao serem reunidos, formaram a solução do problema, não temos como confirmar que essa era a intenção dele desde o início. Apesar disso, ao mesmo tempo que a utilização de uma habilidade relacionada ao PC apareceu como uma forma de solucionar o problema, a matemática, algumas vezes, mostrou-se como parte de um problema a ser resolvido. Às vezes como um aspecto relacionado ao algoritmo que estava sendo construído, outras como a ação de traduzir a ação desejada de uma linguagem para outra. Assim, observamos que a matemática e o Pensamento Computacional parecem atuar como ações que se complementam na determinação do problema e na busca pela sua solução.

No Episódio 2 observamos que os Algoritmos construídos nos três problemas de Eduardo tinham similaridades. Os dois primeiros programavam uma ação de movimentação que deveria ser executada apenas depois de algum acontecimento específico. O primeiro dependia da ação de pressionar uma tecla e o segundo, da ação do ator tocar em uma cor específica. Aspectos matemáticos relacionados à movimentação estavam presentes nos dois, porém de maneiras diferentes. Na construção do primeiro, Eduardo utilizou blocos de programação para a movimentação do ator, adicionando valores à coordenada  $x$  ou à  $y$ , já no segundo a movimentação foi por meio da mudança direta do par ordenado  $(x, y)$ . No terceiro problema, os padrões encontrados eram referentes à parte dos blocos de programação que deveriam ser utilizados, mas, ao invés da ação programada ter relação com uma tecla que deveria ser pressionada ou com um sensor no próprio ator, ela foi programada de forma automática a partir da contagem dos segundos.

Os dados também nos mostram que o processo de construção de conhecimento dos estudantes envolvendo o Pensamento Computacional e a matemática não acontecia mediante uma relação de certo ou errado, mas por meio de um processo que contornava o programar junto à depuração. Observamos que, no Episódio 3, Paulo se envolveu com o processo de depuração diversas vezes e que a matemática foi, em todos esses processos, parte do problema e da solução no Algoritmo que estava sendo construído. O primeiro estava relacionado com o movimento do ator que, ao invés de girar em volta do seu centro, deveria refletir em torno do seu eixo vertical. Em outro problema, um dos códigos utilizados diferenciava o sentido para

qual o ator estava apontando na hora do movimento e o outro não. Por último, o estudante precisou descobrir pontos específicos do Plano Cartesiano para coordenar uma certa movimentação do seu ator principal e fez isso utilizando um recurso próprio do *software*

## 6 Considerações Finais

O objetivo desta pesquisa foi entender como o Pensamento Computacional e a matemática podem emergir por meio de projetos desenvolvidos por estudantes com o ambiente de programação *Scratch* em atividades remotas. Conseguimos, a partir da produção de dados realizada em sete encontros, observar a presença de diversos aspectos do PC e da matemática nos programas construídos pelos estudantes e nas reflexões que foram realizadas por eles junto ao professor.

Observamos que o Pensamento Computacional e a matemática emergiram por meio de projetos de caráter aberto, ou seja, com o desenvolvimento de artefatos ou objetos, no nosso caso os programas no *Scratch*, que partiram das ideias dos próprios estudantes e que foram frutos do meio construcionista que foi criado para eles as materializarem. Também tivemos a proposta de Aprendizagem Criativa como base para o planejamento das atividades que foram aplicadas durante os encontros, orientando-nos para o formato dessas aulas e o modo como elas foram propostas. Destacamos a liberdade que os participantes tiveram no desenvolvimento dos seus projetos, no qual eram livres para construir o que quisessem e para buscarem informações que também estivessem fora do ambiente dos encontros. Essa característica nos trouxe evidências de que os participantes tiveram a oportunidade de pensar como alunos X, ou seja, na busca pelas soluções das situações-problemas criadas pelos próprios no desenvolvimento dos programas, eles procuraram experimentar ideias que não estavam sendo discutidas junto ao grupo. Temos como exemplos o uso de GIF, no qual o professor achava que não poderia ser utilizado, e até mesmo a construção de blocos de programação com base em outros programas ou a partir de vídeos do *YouTube*, ou seja, eles foram além daquilo que estava sendo aprendido com o professor e com os colegas.

Destacamos que, nos três episódios, observamos que a matemática e o Pensamento Computacional emergiram, de diferentes formas, na busca por soluções de problemas que apareceram no desenvolvimento dos projetos. Quando observamos a matemática presente em seus projetos, destacamos a utilização do Plano Cartesiano, seja na elaboração de algoritmos que adicionavam valores inteiros a cada uma das coordenadas com os códigos [adicione \_\_ a x] e [adicione \_\_ a y], ou até mesmo aqueles que utilizavam um ponto  $(x, y)$  específico a partir do código [vá para x: \_\_ y: \_\_], geralmente precedidos por uma estrutura de decisão sempre-então.



Além do Plano Cartesiano, destacamos outros conceitos matemáticos que apareceram nos projetos desenvolvidos pelos estudantes por meio de atividades remotas com o *Scratch*. A utilização de um número aleatório em um certo intervalo que correspondia ao valor da abscissa que seria utilizada, a relação maior/menor que apareceu dentro de uma estrutura de decisão com o código [se [posição  $x > -179$ ] então] que, posteriormente, foi trocada pelo uso do operador matemático [ou] em um algoritmo que verificava a cor na qual o ator principal tangia. Neste mesmo algoritmo, presenciamos o uso de variáveis que, a partir do que foi programado da estrutura de decisão, valores inteiros diferentes eram adicionados a ela. Outra ideia matemática discutida foi a de grandezas proporcionais que, a partir da experimentação realizada na construção do programa junto ao processo de depuração e da orientação do professor, o estudante conseguiu explicar que quanto maior a distância que se deve percorrer, maior também será a velocidade necessária para completar o trajeto em um tempo fixado.

Esses conceitos matemáticos surgiram ao mesmo tempo que os estudantes se envolviam com aspectos relacionados com o Pensamento Computacional, especificamente com o uso da Decomposição, do Reconhecimento de Padrões, na construção de Algoritmos e no processo de depuração, como relatados na análise dos dados.

Refletindo sobre as limitações que tivemos durante a realização desta pesquisa, destacamos o fato de que, devido às atividades remotas, os estudantes construíram grande parte dos seus projetos fora dos encontros, então não tivemos como afirmar que todos os conceitos observados foram realmente compreendidos pelos estudantes. Alguns, como relatado, foram mais discutidos, outros apenas apareceram nos Algoritmos criados. Entretanto, consideramos que isso não deve ser menosprezado, pois eles tiveram a oportunidade de lidar com uma matemática que teve significado para eles, um papel significativo nos projetos que eram do interesse de cada um dos participantes, características que, de acordo com Papert (1986), são importantes para a construção dos conhecimentos.

A falta de recursos tecnológicos também foi um fator que limitou o desenvolvimento desta pesquisa, tanto pelo fato de seis estudantes que estavam interessados em participar do projeto não conseguirem, pois não tinham computador com internet, tanto pela falta de algum recurso prejudicar os estudantes que participaram. Entre eles, temos o relato de Maria sobre a sua não participação, a falta de um microfone para Eduardo conversar com o grupo ou até mesmo problemas envolvendo a internet, que faziam com que o professor ou os alunos fossem desconectados do *Google Meet* em alguns momentos. Cabe ressaltar que esta limitação não foi exclusiva da nossa pesquisa, mas sim um problema que, de acordo com Oliveira (2020a;

2020b), abrangeu diversos professores e estudantes durante o período de Ensino Remoto Emergencial.

Os dados foram produzidos em um projeto que não tinha vínculo com as aulas de matemática das turmas, todavia, os diversos conceitos que se mostraram presentes na construção de programas por meio de atividades remotas apontam possibilidades para que elas possam acontecer dentro da carga horária da disciplina. Pensando em um melhor entendimento dos aspectos que aparecem nas construções, o professor pode trazer situações envolvendo as discussões, os códigos ou outros tópicos para dentro da sala de aula e, com isso, a partir do que emergiu do interesse dos estudantes e que, possivelmente, já teve algum significado para eles, desenvolver os conceitos também de outras formas. Cabe ressaltar a importância de a escola possibilitar que todos os estudantes possam se envolver com a proposta, pois, como já evidenciamos, um dos fatores que prejudicou a presença de mais alunos foi a falta de recursos tecnológicos necessários para a participação dos encontros síncronos e para o envolvimento com os projetos no *Scratch*.

Assim, consideramos que a utilização do *Scratch*, guiada pelas ideias construcionistas de Papert (1986, 1988, 2008) e auxiliadas por características da Aprendizagem Criativa (RESNICK, 2020), podem, mesmo por meio de atividades remotas, fazer com que conceitos matemáticos emergjam e sejam discutidos pelos estudantes a partir de projetos que partam de suas próprias ideias e, com isso, os façam pensar como alunos “X”. Além disso, o desenvolvimento de habilidades relacionadas ao Pensamento Computacional destacados pela BNCC (BRASIL, 2018), como a Abstração, a Decomposição, o Reconhecimento de Padrões, a utilização e construção de Algoritmos e o processo de depuração, por estarem diretamente ligadas à busca de soluções de problemas, também podem estar presentes nesse processo. Entretanto, alguns pontos que podem limitar a elaboração e a avaliação das atividades devem ser destacados, como a não possibilidade de acompanhar algumas partes que podem ser relevantes no processo do estudante e a falta de recursos tecnológicos.

Para passos futuros, queremos buscar conceitos teóricos-filosóficos que possam nos ajudar a embasar e compreender, de modo profundo, as relações entre o Pensamento Computacional e a matemática como um fluxo devir<sup>27</sup>. Além disso, exploraremos um modo de como podemos pensar em uma proposta que utilize as ideias de Aprendizagem Criativa de

---

<sup>27</sup> De acordo com Abbagnano (2007), em seu Dicionário de Filosofia, Devir ou Vir-a-ser significa “1. O mesmo que mudança. 2. Uma forma particular de mudança, a mudança absoluta ou substancial que vai do nada ao ser ou do ser ao nada [...]” (ABBAGNANO, 2007, p. 268).

Resnick (2020) e possa ser vinculada especificamente à Educação Matemática, buscando compreensões e indícios para que possamos considerar uma ideia que denotamos como Aprendizagem Matemática Criativa. Também visamos planejar atividades de forma que os conceitos envolvendo matemática possam ser trabalhados além das atividades com o *Scratch*, ou seja, possam ser discutidos por outros meios e, com isso, mais bem compreendidos pelos estudantes.

Refletimos também que diversos aspectos abordados nesta pesquisa podem servir como indicações para pesquisas futuras, sejam elas relacionadas à formação de professores para o desenvolvimento de projetos com o *Scratch* por meio de atividades remotas, à construção de um currículo que seja pensado para que os estudantes desenvolvem habilidades referentes aos alunos 'X' ou até mesmo sobre o que o professor pode aprender com os estudantes a partir do uso da proposta de Aprendizagem Criativa, entre outras.

Para finalizar, destaco a importância deste estudo na minha formação como professor. Os conhecimentos construídos durante essa trajetória, seja com o planejamento que foi modificado algumas vezes, com a aplicação das atividades que teve que ser repensada para meios remotos, seja na análise que teve seu foco alterado, ampliaram a minha compreensão de como posso ensinar, aprender e pesquisar. As potencialidades do uso do *Scratch* como um meio para ensinar e aprender Matemática e, ao mesmo tempo, desenvolver habilidades relacionadas ao Pensamento Computacional de forma remota, mostraram-me a relevância destas ideias para a área de Educação Matemática.

Sendo assim, encerramos este trabalho propondo que outros profissionais utilizem as ideias apresentadas nesta pesquisa, visando a uma Educação Matemática que esteja de acordo com as necessidades do século XXI, no qual as tecnologias digitais não sejam apenas vistas como transmissoras de conhecimentos ou facilitadoras do processo, mas sim como um meio que potencialize a construção de saberes que sejam relevantes para os estudantes e os auxiliem na busca por soluções dos diversos problemas que aparecerão em seu cotidiano.

## 7 Referências

- ABBAGNANO, N. **Dicionário de Filosofia**. São Paulo: Martins Fontes, 2007.
- AMABILE, T. M. **Componential theory of creativity**. Harvard Business School, v. 12, n. 96, p. 1-10, 2012.
- AMABILE, T. M. *The social psychology of creativity*. New York, NY: Springer, 1983.
- AZEVEDO, G. T., MALTEMPI, M. V. Processo de Aprendizagem de Matemática à Luz das Metodologias Ativas e do Pensamento Computacional. **Ciências & Educação**, Bauri, v.26, e20061, 2020.
- AZEVEDO, G. T.; MALTEMPI, M. V. Metodologias ativas de aprendizagem nas aulas de Matemática: equação da circunferência e construção criativa de pontes. **Educação Matemática Debate**, Montes Claros, v. 3, n. 9, p. 236-254, 2019. DOI: <https://doi.org/fcc4>.
- BARCELOS, T. S.; SILVEIRA, I. F. Pensamento Computacional e Educação Matemática: relações para o Ensino de Computação na Educação Básica. **XX Workshop sobre Educação em Computação**. Curitiba, 2012.
- BASSO, M. V. A.; NOTARE, M. R. Pensar-com Tecnologias Digitais de Matemática Dinâmica. **RENOTE – Novas Tecnologias na Educação**. Porto Alegre/RS. v. 13, n. 2, dez, 2015.
- BICUDO, M. A. V. Pesquisa qualitativa e pesquisa qualitativa segundo a abordagem fenomenológica. In: BORBA, M. C.; ARAUJO, J. L. (orgs.). **Pesquisa qualitativa em educação matemática**. 5. ed. Belo Horizonte: Autêntica, 2018.
- BOGDAN, R.; BIKLEN, S. K. **Investigação qualitativa em educação: uma introdução à teoria e aos métodos**. Portugal: Porto, 1994.
- BORBA, M. C.; ARAUJO, J. L. (orgs.). **Pesquisa qualitativa em educação matemática**. 5. ed. Belo Horizonte: Autêntica, 2018.
- BRACKMANN C. P. **Desenvolvimento do Pensamento Computacional Através de Atividades Desplugadas na Educação Básica**. Tese (Doutorado), Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- BRASIL. **Base Nacional Comum Curricular: Educação é a Base**. 2018. Disponível em: <<http://basenacionalcomum.mec.gov.br/a-base>>. Acesso em: 10 nov. 2019.
- BRASIL. Lei nº 9.394, de 20 de dezembro de 1996. Estabelece as diretrizes e bases da educação nacional. **Diário Oficial da União**. 23 dez. 1996.
- BRASIL. **Plataforma AVA**. 2019. Disponível em: <<http://avamec.mec.gov.br/#/instituicao/seb/curso/3801/visualizar>> Acesso em: 14 out. 2019.
- BRENNAN, K.; RESNICK, M. **New frameworks for studying and assessing the development of computational thinking**. AERA 2012, Vancouver, Canadá, 2012. Disponível em: . Acesso em: 10 out. 2021.
- CERRI, R.; CARVALHO, A. C. P. L. F. **Aprendizado de Máquina: breve introdução e aplicações**. Cadernos de Ciências & Tecnologia, Brasília, v. 34, n. 3, p. 297-313, set./dez, 2017.
- CORRÊA, B. S. **Programando com Scratch no Ensino Fundamental: uma possibilidade para a construção de conceitos matemáticos**. 171 f. Dissertação (Mestrado em Ensino de

Matemática) – Instituto de Matemática e Estatística, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2021.

**CSTA. K-12 Computer Science Standards - Revised 2011 - The CSTA Standards Task Force.** Association for Computing Machinery, 2011.

DAL AGNOL, C. **Utilizando o recurso de Big Data Gapminder:** concepções críticas e digitais no contexto de uma sala de aula de matemática. 74 f. Trabalho de Conclusão de Curso (Graduação) – Instituto de Matemática e Estatística, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2019.

DALLA VECCHIA, R. **Relações possíveis entre Big Data e Modelagem Matemática no âmbito da Educação Matemática.** In: Encontro Paranaense de Modelagem na Educação Matemática, 8, 2018, Cascavel.

D'AMBROSIO, B. S.; LOPES, C. E. Insubordinação Criativa: um convite à reinvenção do educador matemático. **Bolema**, Rio Claro, v. 29, n. 51, p. 1-17, 2015. Disponível em: [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0103-636X2015000100002&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-636X2015000100002&lng=en&nrm=iso) . Acesso em: 22 dez. 2020.

FACELI, K.; LORENA, A. C.; GAMA, J. ; CARVALHO, A. C. P. L. F. de. **Inteligência artificial:** uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, 2011.

FAGUNDES, L. C.; SATO, L. S.; MAÇADA, D. L. **Aprendizes do Futuro:** as inovações começaram. 1. ed. Brasília: PROINFO/SEED/MEC, 1999. v. 19. 95p.

FAGUNDES, L.; SATO, L. S.; MAÇADA, D. L. **Aprendizes do futuro:** As Inovações Começaram. MEC/SEED/Proinfo. Coleção: informática para a mudança em educação, 2009 Disponível em <http://repositorio.furg.br/handle/1/1130> . Acesso em 15 de out. 2020.

FERREIRA, F. P. S. S. **Educação Financeira com o Scratch:** potencialidades e dificuldades. 115 f. Dissertação (Mestrado), Universidade do Estado de Mato Grosso, Barra do Bugres, 2020.

FIORENTINI, D.; LORENZATO, S. **Investigação em Educação Matemática.** Campinas: Autores Associados, 2006.

FIORENTINI, D.; LORENZATO, S. **Investigação em educação matemática:** percursos teóricos e metodológicos. 2 ed. Campinas: Autores Associados, 2006

FREIRE, P. **Pedagogia do Oprimido.** 66 ed. Rio de Janeiro/São Paulo: Terra e Paz, 2018.

FREIRE, P. **Pedagogia do oprimido.** 66. Ed. Rio de Janeiro/São Paulo: Paz e Terra, 2018.

GAIATO, K. **GIF:** descubra o que é e como usar o formato. 2021. Disponível em: <https://www.tecmundo.com.br/internet/219225-gif-descubra-usar-o-formato.htm>. Acesso em: 19 jan. 2022

GAYESKI, R. G. **Big Data e Educação Matemática:** algumas aproximações. 135 f. Dissertação (Mestrado em Ensino de Matemática) – Instituto de Matemática e Estatística, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2019.

GOLDENBERG, M. **A arte de pesquisar:** como fazer pesquisa qualitativa em Ciências Sociais. 8 ed. Rio de Janeiro: Record, 2004.

GRAVES, L. A. S.; **O Pensamento Computacional na prática:** uma experiência usando Python em aulas de matemática básica. 64 f. Dissertação (Mestrado), Universidade Federal de Santa Maria, Santa Maria, 2021.

HODGES, C. et al. The difference Between Emergency Remote Teaching and OnlineOn-line Learning. **EDUCAUSE Review**. Mar. 2020. Disponível em: <https://er.educause.edu/articles/2020/3/the-difference-between-emergency-remote-teaching-and-online-learning> Acesso em: 18 nov. 2020.

HORBACH, I. C; **Semelhança de Triângulos**: um estudo propositivo através do *Scratch*. 70 f. Dissertação (Mestrado), Universidade Federal da Fronteira Sul, Chapecó, 2020.

LIFELONG KINDERGARTEN GROUP (LKG). *Scratch*, 2018. Sobre. Disponível em: <https://Scratch.mit.edu/about>. Acesso em: 16 de jan. de 2020.

MALTEMPI, M. V. Construcionismo: pano de fundo para pesquisas em informática aplicada à educação matemática. In: BICUDO, M. A. V.; BORBA, M. C. (org.). **Educação matemática**: pesquisa em movimento. São Paulo: Cortez, 2012. p. 287-307.

MALTEMPI, M. V. **Novas Tecnologias e Construção de Conhecimento**: Reflexões e Perspectivas. In: *V Congresso Ibero-americano de Educação Matemática (CIBEM)*. Porto, Portugal. 2005. Disponível em: <http://www.rc.unesp.br/igce/demac/maltempi/Publicacao/Maltempi-cibem.pdf>.

MOREIRA, M. A.; MASINI, E. F. S. **Aprendizagem Significativa**: a teoria de David Ausubel. São Paulo: Moraes, 1982.

NATIONAL RESEARCH COUNCIL (NRC). **Report of a Workshop on the Scope and Nature of Computational Thinking**. Washington: National Academies Press, 2010.

OLIVEIRA, C. D. S. **Scratch no Ensino e Aprendizagem de Geometria**: um panorama de pesquisas brasileiras. 166 f. Dissertação (Mestrado), Universidade Federal do Pampa, Mestrado Profissional em Matemática em Rede Nacional, Caçapava do Sul, 2021.

OLIVEIRA, D. A. (coord.). **Docência na Educação Básica Privada em Tempos de Pandemia**. Belo Horizonte: GESTRADO; 2020b. Disponível em: <https://gestrado.net.br/wp-content/uploads/2020/08/RELATORIO-COMPLETO-1011.pdf> Acesso em: 18 nov. 2020.

OLIVEIRA, D. A. (coord.). **Trabalho Docente em Tempos de Pandemia**. Belo Horizonte: GESTRADO; 2020a. Disponível em: [https://anped.org.br/sites/default/files/images/cnte\\_relatorio\\_da\\_pesquisa\\_covid\\_gestrado\\_v02.pdf](https://anped.org.br/sites/default/files/images/cnte_relatorio_da_pesquisa_covid_gestrado_v02.pdf) Acesso em: 18 nov. 2020.

PAPERT, S. **A máquina das crianças**: repensando a escola na era da informática. Ed. Ver. Porto Alegre: Artmed, 2008.

PAPERT, S. **Constructionism**: a new opportunity for elementary science education. Massachusetts Institute of Technology, 1986.

PAPERT, S. **Logo**: computadores e educação. 3. ed. São Paulo: Brasiliense, 1988.

PIAGET, J. **Fazer e Compreender**. São Paulo: Melhoramentos, Editora da Universidade de São Paulo, 1978.

PIXILART. **Pixilart**, 2021. Sobre. Disponível em: <https://www.pixilart.com/about>. Acesso em: 02 de mai. de 2022.

POLYA, G. **A Arte de Resolver Problemas**. Trad. Heitor Lisboa de Araújo. Rio de Janeiro: Editora Interciência, 1978.

PONCE, B. J. O currículo e seus desafios na escola pública brasileira: em busca da justiça curricular. **Currículo sem fronteiras**, v. 18, n. 3, p. 785-800, set./dez. 2018.

- RESNICK, M. **Jardim de Infância para a Vida Toda: Por Uma Aprendizagem Criativa, Mão na Massa e Relevante para Todos**. Edição do Kindle. Local: Porto Alegre. Penso, 2020.
- RESNICK, M. **Sowing the seeds for a more creative society**. Learning and Leading with Technology, p. 18–22, 2007.
- RIBEIRO, L.; FOSS, L.; CAVALHEIRO, S. A. C. Entendendo o pensamento computacional. In: RAABE, A.; ZORZO, A. F.; BLIKSTEIN, P. **Computação na educação básica: fundamentos e experiências**. Porto Alegre: Penso, 2020.
- RIO GRANDE DO SUL. Conselho Estadual de Educação. Comissão de Legislação e Normas. **Parecer nº 01/2020**. 2020b. Disponível em: <https://www.ceed.rs.gov.br/parecer-n-0001-2020>. Acesso em: 15 out. 2020.
- RIO GRANDE DO SUL. Decreto nº 55.118, de 16 de março de 2020. **Diário Oficial do Rio Grande do Sul**. 2020a. Disponível em: <https://estado.rs.gov.br/upload/arquivos/decreto-55118.pdf>. Acesso em: 15 out. 2020.
- RIO GRANDE DO SUL. Secretaria da Educação. **Escola RS**, 2020c. Aulas Remotas: o que é? Disponível em: <https://escola.rs.gov.br/aulas-remotas-o-que-e>. Acesso em: 15 out. 2020.
- ROSA, M. (2018). **Tessituras teórico-metodológicas em uma perspectiva investigativa na Educação Matemática**: da construção da concepção de Cyberformação com professores de matemática a futuros horizontes. In: A. M. P. Oliveira & M. I. R. Ortigão (Org.). Abordagens teóricas e metodológicas nas pesquisas em Educação Matemática. Brasília: Sociedade Brasileira de Educação Matemática. E-book. 2018; Disponível em: [http://www.sbem.com.br/files/ebook\\_.pdf](http://www.sbem.com.br/files/ebook_.pdf). Acesso em: 5 dez. 2020.
- ROSA, M. Por que Insubordinação Criativa na Educação Matemática? **RIPEM**. v. 9, n. 3, p. 1-4, 2019. Disponível em: <http://sbem.iuri0094.hospedagemdesites.ws/revista/index.php/ripem/article/view/2363>. Acesso em: 22 dez. 2020.
- ROSA, M. **Role playing game eletrônico**: uma tecnologia lúdica para aprender e ensinar matemática. 2004. ii, 170 f. Dissertação (mestrado) - Universidade Estadual Paulista, Instituto de Geociências e Ciências Exatas, 2004. Disponível em: <http://hdl.handle.net/11449/91089>.
- ROSA, M.; DANTAS, D. M. **Criatividade tecnológica**: um estudo sobre a construção de atividades-matemáticas-com-tecnologias-digitais por professores/as em cyberformação. **Zetetike**, Campinas, SP, v. 28, p. e020030, 2020. DOI: 10.20396/zet.v28i0.8654423. Disponível em: <https://periodicos.sbu.unicamp.br/ojs/index.php/zetetike/article/view/8654423>. Acesso em: 3 mai. 2022.
- SILVA, J. C; **Uso do Scratch para investigação matemática e os Números Mágicos de Ball**. Dissertação (Mestrado), Universidade Federal do Tocantins, Arraias, 2021.
- STERNBERG, R. J. The nature of creativity. **Creativity research journal**, v. 18, n. 1, p. 87-98, 2006.
- TECNOLOGIA. In: MICHAELIS **Dicionário Brasileiro da Língua Portuguesa**. Melhoramentos Ltda., 2015. Disponível em: <https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/tecnologia/>. Acesso em: 15 out. 2019.
- VICARI, R. *et al.* **Pensamento Computacional – Revisão Bibliográfica**. Monografia. Instituto de informática, UFRGS, 2018.
- WING, J. Computational thinking - What and why? **The Link Magazine**, Spring, 2011.

WING, J. Computational thinking and thinking about computing. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 366, n. 1881, p. 3717-3725, 2008.

WING, J. Computational Thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006. Disponível em: <<http://www.cs.cmu.edu/~jw/15110-s13/Wing06-ct.pdf>>. Acesso em: 7 out. 2018.



## Apêndice I: Carta de pedido de autorização à Escola



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENSINO DE MATEMÁTICA



Novo Hamburgo, 04 de agosto de 2020.

Prezada Professora SHEILA KIELING  
Diretora da E.E.E.F. Ayrton Senna do Brasil

O professor MATEUS DAUÃ DE MORAIS atualmente é mestrando regularmente matriculado no Programa de Pós-Graduação em Ensino de Matemática (PPGEMat) da Universidade Federal do Rio Grande do Sul.

Como parte das exigências do PPGEMat, o mestrando está desenvolvendo uma pesquisa na qual os objetivos podem ser resumidos em investigar que potencialidades a construção de programas com e sem Inteligência Artificial no Scratch num contexto de Aprendizagem Criativa podem trazer para o ensino e aprendizagem de Matemática e para o desenvolvimento do Pensamento Computacional nos participantes. Para esse fim, serão coletadas informações dos participantes do projeto PROGRAMAÇÃO COM SCRATCH que iniciará de forma remota via Google Meet e aceitarem os termos de assentimento e consentimento.

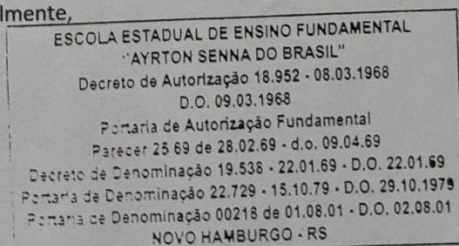
A pesquisa garante o anonimato dos participantes, evitando riscos como constrangimento, identificação, violação de confidencialidade e de privacidade, por causa do acesso e divulgação das informações dadas pelo (a) participante. Para diminuir a chance de qualquer risco, o uso de informações e dados coletados durante o projeto será apenas em trabalhos acadêmicos como: dissertação, artigos científicos, palestras, seminários etc., no qual o participante será identificado pela primeira letra do nome e a sua idade. Os dados serão registrados por meio de anotações, produções do aluno durante sua participação no projeto, registro de vídeo da tela do computador e imagens ou sons coletados nos aplicativos de conversa relacionados ao projeto), esses registros ficarão guardados por pelo menos 5 anos após o término da investigação. Além disso, o (a) participante pode deixar de participar da investigação a qualquer momento, se não se sentir bem com alguma situação.

Como benefícios, esperamos produzir informações importantes sobre contribuições do uso de tecnologias digitais na aprendizagem de matemática, no processo de aprendizagem criativa e no pensamento computacional de alunos do Ensino Fundamental. Os participantes têm o benefício de conhecer, interagir e criar programas com o software Scratch e o recurso Machine Learning For Kids, além de um possível desenvolvimento de habilidades relacionados com o Pensamento Computacional. Neste sentido, torna-se extremamente importante realizar experimentos educacionais e, por esta razão, estamos solicitando a sua autorização para que este trabalho possa ser desenvolvido na escola sob sua Direção.

Em caso de manifestação de sua concordância, por favor, registre sua ciência ao final deste documento, o qual está sendo encaminhado em duas vias.

Agradecemos a sua atenção.

Cordialmente,



Rodrigo Dalla Vecchia  
Orientador e professor do PPGEMat

Mateus Dauã de Moraes  
Pesquisador e Aluno do PPGEMat

Sheila Kieling

Diretora

16.0438/01  
D.O. 27.12.68 pag 2054

## Apêndice II: Termos de Assentimento e Consentimento

Link dos termos: <https://forms.gle/32Db4fDf2Vut9um19>



### PROJETO: Programação com Scratch

Olá! Meu nome é Mateus Dauã e eu sou um dos professores de matemática da E.E.E.F. Ayrton Senna do Brasil.

Esse formulário foi criado para coletar os dados dos alunos interessados em participar do projeto "PROGRAMAÇÃO COM SCRATCH".

Como estamos seguindo todas as orientações do Estado para o período de pandemia, o projeto inicialmente acontecerá de forma online via Google Classroom (Google Sala de Aula).

As atividades serão gratuitas e os únicos requisitos são:

- 1 - Ser aluno da E.E.E.F. Ayrton Senna do Brasil;
- 2 - Durante a pandemia, ter acesso a um computador com internet (no momento não podemos utilizar o laboratório de informática da escola);

Para conhecer mais sobre o projeto, acesso o site:

Quaisquer dúvidas, entre em contato comigo pelo telefone: (51)9 98382182.

**\*Obrigatório**

Qual é o seu nome? \*

Sua resposta

De qual turma você faz parte? \*

Sua resposta

Digite aqui o seu email @[gmail.com](mailto:) para que eu possa lhe adicionar a nossa turma no Google Sala de Aula: \*

Sua resposta

Em qual das turmas você gostaria de participar via Google Classroom? \*

- Turma MANHÃ: sexta-feira às 10h às 11h30min
- Turma TARDE: terça-feira, das 13h30min às 15h

Caso queira participar do nosso grupo no Whatsapp, digite aqui o seu número:

Sua resposta

## PROJETO: Programação com Scratch

\*Obrigatório

### Pesquisa do professor Mateus Dauã de Moraes

Esse projeto faz parte da pesquisa relacionada ao curso de mestrado vinculado ao Programa de Pós-Graduação em Ensino de Matemática da Universidade Federal do Rio Grande do Sul, no qual o professor faz parte. Com isso, o professor solicita que os dados obtidos no projeto (como os programas criados pelos participantes, dúvidas respondidas, entre outros) possam ser utilizados para fins acadêmicos, como em artigos e dissertações.

Nenhum nome ou imagem de aluno(a) será divulgado em qualquer tipo de publicação.

Os dados serão utilizados com o objetivo de verificar que potencialidades a construção de programas com e sem Inteligência Artificial no Scratch num contexto de Aprendizagem Criativa podem trazer para o ensino e aprendizagem de matemática e para o desenvolvimento do Pensamento Computacional nos participantes.

A participação na pesquisa não é obrigatória e nenhum aluno será excluído caso não concorde com os termos a seguir.

Por questões éticas, pedimos que aluno(a) e responsável leiam os termos a seguir e preencham conforme solicitado.

## TERMO ASSENTIMENTO LIVRE E ESCLARECIDO (PARA O ALUNO)

Convidamos o(a) aluno(a) escrito nesse formulário da E.E.E.F. Ayrton Senna do Brasil, para participar do projeto de pesquisa intitulado "Inteligência Artificial com Scratch" que faz parte de uma pesquisa vinculada ao Programa de Pós-Graduação em Ensino de Matemática da Universidade Federal do Rio Grande do Sul no qual o professor Mateus Dauã de Moraes faz, com orientação da professor Dr. Rodrigo Dalla Vecchia.

O participante contribuirá para o sucesso da pesquisa sem recebimento de qualquer incentivo financeiro. Os objetivos do estudo são somente acadêmicos e, em resumo, é verificar que potencialidades a construção de programas com e sem Inteligência Artificial no Scratch num contexto de Aprendizagem Criativa podem trazer para o ensino e aprendizagem de matemática e para o desenvolvimento do Pensamento Computacional nos participantes.

O uso de informações do projeto será apenas em trabalhos como: dissertação, artigos científicos, palestras, seminários etc., identificadas pela primeira letra do nome do(a) participante e sua idade. Os dados serão registrados por meio de anotações e das produções do aluno durante sua participação no projeto e ficarão guardados por pelo menos 5 anos após o término da investigação.

A participação nesta pesquisa não vai contra as normas legais e éticas, mas tem alguns riscos: constrangimento, identificação, violação de confidencialidade e de privacidade, por causa do acesso e divulgação das informações dadas pelo (a) participante aos pesquisadores. Para diminuir as chances de acontecer isso, não revelaremos o nome do(a) participante. Além disso, o (a) participante pode deixar de participar da investigação a qualquer momento, se não se sentir bem com alguma situação.

Como benefícios, esperamos produzir informações importantes sobre contribuições do uso de tecnologias digitais na aprendizagem de matemática, no processo de aprendizagem criativa e no pensamento computacional de alunos do Ensino Fundamental. Os participantes têm o benefício de conhecer, interagir e criar programas com o software Scratch e o recurso Machine Learning For Kids, além de ajudar a realizar a pesquisa.

A colaboração para esta pesquisa se inicia no momento do aceite desse Termo pelo participante, junto com o Termo de Consentimento aceito por um responsável. Se tiver alguma dúvida, ou se sentir prejudicado(a), pode falar com o pesquisador responsável ou com o professor orientador da pesquisa pelo telefone (51)9 98382182 ou pelo e-mail: [mateusdaua@gmail.com](mailto:mateusdaua@gmail.com), [rodrigovecchia@gmail.com](mailto:rodrigovecchia@gmail.com).

O(A) participante pode se retirar desta pesquisa a qualquer momento, sem sofrer quaisquer sanções ou constrangimentos.

---

Você, aluno(a), leu o termo acima? \*

- Sim, li.
- Não li.

---

Você, aluno(a), concorda com o termo de assentimento acima? \*

- Sim, concordo.
- Não concordo.

## TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (para os responsáveis)

Convidamos o(a) aluno(a) escrito nesse formulário da E.E.E.F. Ayrton Senna do Brasil, para participar do projeto de pesquisa intitulado "Inteligência Artificial com Scratch" que faz parte de uma pesquisa vinculada ao Programa de Pós-Graduação em Ensino de Matemática da Universidade Federal do Rio Grande do Sul no qual o professor Mateus Dauã de Moraes faz, com orientação da professor Dr. Rodrigo Dalla Vecchia.

O participante contribuirá para o sucesso da pesquisa sem recebimento de qualquer incentivo financeiro. Os objetivos do estudo são somente acadêmicos e, em resumo, é verificar que potencialidades a construção de programas com e sem Inteligência Artificial no Scratch num contexto de Aprendizagem Criativa podem trazer para o ensino e aprendizagem de matemática e para o desenvolvimento do Pensamento Computacional nos participantes.

O uso de informações do projeto será apenas em trabalhos como: dissertação, artigos científicos, palestras, seminários etc., identificadas pela primeira letra do nome do(a) participante e sua idade. Os dados serão registrados por meio de anotações e das produções do aluno durante sua participação no projeto e ficarão guardados por pelo menos 5 anos após o término da investigação.

A participação nesta pesquisa não vai contra as normas legais e éticas, mas tem alguns riscos: constrangimento, identificação, violação de confidencialidade e de privacidade, por causa do acesso e divulgação das informações dadas pelo (a) participante aos pesquisadores. Para diminuir as chances de acontecer isso, não revelaremos o nome do(a) participante. Além disso, o (a) participante pode deixar de participar da investigação a qualquer momento, se não se sentir bem com alguma situação.

Como benefícios, esperamos produzir informações importantes sobre contribuições do uso de tecnologias digitais na aprendizagem de matemática, no processo de aprendizagem criativa e no pensamento computacional de alunos do Ensino Fundamental. Os participantes têm o benefício de conhecer, interagir e criar programas com o software Scratch e o recurso Machine Learning For Kids, além de ajudar a realizar a pesquisa.

A colaboração para esta pesquisa se inicia no momento do aceite desse Termo pelo participante, junto com o Termo de Consentimento aceito por um responsável. Se tiver alguma dúvida, ou se sentir prejudicado(a), pode falar com o pesquisador responsável ou com o professor orientador da pesquisa pelo telefone (51)9 98382182 ou pelo e-mail: [mateusdaua@gmail.com](mailto:mateusdaua@gmail.com), [rodrigovecchia@gmail.com](mailto:rodrigovecchia@gmail.com).

O(A) participante pode se retirar desta pesquisa a qualquer momento, sem sofrer quaisquer sanções ou constrangimentos.

---

O responsável leu o termo de consentimento acima? \*

- Sim.
- Não.

---

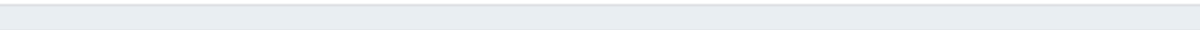
O responsável concorda com a participação do aluno na pesquisa? \*

- Sim, concordo.
- Não concordo.

Se a resposta do responsável foi SIM, preencha com o nome e o RG do responsável. Caso contrário, responda apenas com NÃO. \*

Sua resposta

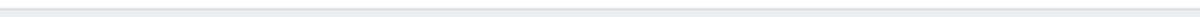
---



Comentários e/ou perguntas:

Sua resposta

---



## Apêndice III: Planejamento dos encontros

### Sequência de Atividades que foram desenvolvidas para a produção de dados:

#### Objetivo geral das atividades:

- Construção de programas no *software Scratch* com e sem a utilização de Inteligência Artificial disponibilizada pelo site *Machine Learning For Kids*, possibilitando o desenvolvimento da criatividade, de competências matemáticas e habilidades relacionadas ao Pensamento Computacional a partir de sete encontros sustentados pela proposta de Aprendizagem Criativa.

#### Objetivos específicos das atividades:

- Aprender a programar utilizando a linguagem de programação do *Scratch*;
- Interagir com o *software Scratch* por meio de programas criados com ele ou por outros usuários;
- Construir programas com o *software Scratch* utilizando algoritmos de Machine Learning disponibilizados no site <https://machinelearningforkids.co.uk>
- Desenvolver habilidades referentes a Criatividade e ao Pensamento Computacional;
- Refletir sobre a matemática que pode surgir durante a programação;

#### Público-alvo:

- Estudantes dos anos finais do ensino fundamental;

#### Descrição das atividades:

##### CONVITE

**Objetivos:** apresentar o projeto ‘Programação com *Scratch*’ e algumas possibilidades de construção com o *Scratch*;

#### Atividades:

- Enviar para todas os grupos das turmas no *WhatsApp* um convite em forma de imagem explicando o projeto;
- Enviar uma mensagem para os grupos das turmas no *WhatsApp* com o *link* do site criado para o projeto com alguns programas disponíveis para que os estudantes possam conhecer algumas possibilidades;

### ENCONTRO 1 – 12/08/2020

**Objetivos:** iniciar o percurso na Espiral da Aprendizagem Criativa pelos componentes Imaginar e Brincar, além conversar sobre algumas orientações de como o projeto funcionará;

#### Atividades:

- Apresentar a proposta do projeto vinculada ao mestrado do professor;
- Verificar quem já conhecia o *Scratch* ou outra linguagem de programação;
- Apresentar o site criado pelo pesquisador (<https://sites.google.com/view/projeto-Scratch-ayrton-senna/>);

- Mostrar a aba POSSIBILIDADES do site, onde eles poderão conhecer algumas das possibilidades de criação de jogos, histórias ou animações interativas com o *Scratch* em programas criados pelo pesquisador;
- Ler as 10 dicas para se trabalhar com programação ou atividades criativas (disponível no site);
- Apresentação do site do *Scratch*, disponível em: <https://Scratch.mit.edu/>
- Criação de um Usuário no site do *Scratch*;
- Seguir explorando o site do *Scratch*, passando pelas abas CRIAR, EXPLORAR, IDEIAS e INSCREVA-SE;
- Focar na aba EXPLORAR, mostrando que eles podem pesquisar trabalhos de “*Scratchers*” de todo o mundo e ver o *script* desses trabalhos;
- Apresentar Formulário da Pesquisa com os termos de Assentimento e Consentimento Livre e Esclarecido;
- Sugestão de atividade para a próxima semana: navegar pela aba EXPLORAR e escolher alguns projetos que tenham gostado;

## ENCONTRO 2 – 19/08/2020

**Objetivos:** focando nos componentes Imaginar e Criar da Espiral da Aprendizagem Criativa, iremos conversar sobre os projetos explorados ou iniciados pelos participantes na semana anterior e começar a construção dos programas no *Scratch*;

### Atividades:

- Conhecer e conversar sobre os programas que foram explorados ou iniciados pelos estudantes durante a semana no site do *Scratch*;
- Explicação sobre a linguagem de programação do *Scratch* passando por todas as abas e categorias;
- Construção de um “jogo de Labirinto”, para que possamos focar em alguns conjuntos de blocos que, quando unidos, formam um *script* para fazer o programa funcionar;
- Sugestão de atividade para a próxima semana: criar seu próprio jogo de labirinto ou parecido com a linguagem de programação do *Scratch*;

## ENCONTRO 3 – 26/08/2020

**Objetivos:** além de continuar o percurso pelos componentes já vistos da Espiral da Aprendizagem Criativa, nessa aula também iremos compartilhar os programas criados na comunidade *online* do *Scratch* para que cada um possa experienciar o programa do outro, podendo assim sugerir melhorias. Apresentar um programa no qual o computador pergunta, o usuário responde e, dependendo da resposta, o computador toma alguma ação.

### Atividades:

- Cada aluno fará uma breve apresentação do que já fez até o momento, mostrando como o jogo funciona, o *script* escrito e falando sobre suas principais dificuldades;
- Iremos brincar com o programa um do outro, sugerindo algumas melhorias;



- Esclarecimento de dúvidas do que já foi feito no momento;
- Apresentar um programa feito na qual o computador pergunta algo, o usuário responde e o computador, dependendo da resposta do usuário, toma alguma ação.
- Construir, junto com os estudantes, um programa parecido com o apresentado pelo professor;
- Pedir para que os participantes façam, durante a semana, algo parecido com o que foi construído em conjunto;

#### **ENCONTRO 4 – 02/09/2020**

**Objetivos:** após refletirmos sobre os programas criados na semana anterior, vamos conversar sobre as limitações do programa em reconhecer as respostas escritas pelo usuário, com isso, iniciaremos uma discussão sobre Inteligência Artificial e o programa com a utilização do site *Machine Learning For Kids* será apresentado aos participantes.

##### **Atividades:**

- Cada aluno fará uma breve apresentação do que já fez até o momento, mostrando como o jogo funciona, o *script* escrito e falando sobre suas principais dificuldades;
- Iremos brincar com o programa um do outro, sugerindo algumas melhoras;
- Esclarecimento de dúvidas do que já foi feito no momento;
- Perguntas referentes as atividades da aula passada sobre perguntas e respostas no *Scratch*;
- Testar algoritmos criados pelo professor ou pelos alunos envolvendo perguntas e respostas;
- Conversa sobre Inteligência Artificial e *Machine Learning* com a apresentação de algumas definições
- Apresentação do site *Machine Learning For Kids* e sugestão de exploração durante a semana, principalmente na aba Fichar de Trabalho;

#### **ENCONTRO 5 – 09/09/2020**

**Objetivos:** além de explorarmos os projetos que podem ter sido desenvolvidos por cada um dos estudantes envolvendo *Machine Learning* e esclarecermos as dúvidas que surgirem, iremos construir dois programas envolvendo perguntas e respostas com o intuito que alunos notem as potencialidades do *Machine Learning For Kids*;

##### **Atividades:**

- Explorar os projetos construídos envolvendo *Machine Learning*;
- Iremos brincar com o programa um do outro, sugerindo algumas melhoras;
- Esclarecimento de dúvidas do que já foi feito no momento;
- Construção de dois programas parecidos, um sem e outro com a utilização do *Machine Learning For Kids*;
- Apresentação dos exemplos disponíveis no MLFK na aba Fichar de Trabalho e

sugestão para que construíssem algo durante a semana;

### **ENCONTRO 6 – 16/09/2020**

**Objetivos:** Discutir sobre os programas desenvolvidos durante a semana envolvendo u não o MLFK. Depois disso, haverá a apresentação de um programa com IA criado pelo professor e um tempo para reflexões em cima dele. Por fim, continuação da apresentação e exploração dos projetos dos estudantes;

**Atividades:**

- Discussão sobre as atividades desenvolvidas no encontro anterior ou durante a semana;
- Apresentação de um programa envolvendo Inteligência Artificial criado pelo professor;
- Esclarecimento de dúvidas sobre o programa construído pelo professor;
- Continuação da apresentação e exploração dos projetos dos estudantes;

### **ENCONTRO 7 – 23/09/2020**

**Objetivos:** refletir sobre os programas criados utilizando *Machine Learning* indicando potencialidades e fragilidades. Além disso, também iremos começar a programar durante as aulas, para que possamos compartilhar ideias no momento da programação e continuar com o projeto desta maneira.

**Atividades:**

- Cada estudante fará uma breve apresentação do que já fez até o momento, mostrando como o jogo funciona, o *script* escrito e falando sobre suas principais dificuldades;
- Focaremos nos programas que foram construídos com o uso de *Machine Learning*, refletindo sobre as potencialidades e as fragilidades encontradas pelos participantes;
- Após as discussões e o esclarecimento de dúvidas, começaremos a programar durante o encontro, para que possamos compartilhar ideias, tirar dúvidas e conversar sobre as construções durante o processo de programação.