UNIVERSIDADE DE CAXIAS DO SUL

LEONARDO POLONI

APRENDIZAGEM DE PROGRAMAÇÃO MEDIADA POR UMA LINGUAGEM VISUAL: POSSIBILIDADE DE DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL

CAXIAS DO SUL

2018

LEONARDO POLONI

APRENDIZAGEM DE PROGRAMAÇÃO MEDIADA POR UMA LINGUAGEM VISUAL: POSSIBILIDADE DE DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL

Dissertação apresentada ao Programa de Pós-Graduação em Educação da Universidade de Caxias do Sul para obtenção do título de Mestre em Educação.

Orientadora: Prof.ª Dr.ª Eliana Maria do Sacramento Soares

CAXIAS DO SUL

2018

Dados Internacionais de Catalogação na Publicação (CIP)
Universidade de Caxias do Sul
UCS - BICE - Processamento Técnico

P778a Poloni, Leonardo, 1976-
Aprendizagem de programação mediada por uma linguagem visual : possibilidade de desenvolvimento do pensamento computacional / Leonardo Poloni. – 2018.
180 f. : il. ; 30 cm

Dissertação (Mestrado) – Universidade de Caxias do Sul, Programa de Pós-Graduação em Educação, 2018.
Orientação: Profa. Dra. Eliana Maria do Sacramento Soares.

1. Programação (Computadores). 2. Aprendizagem. 3. Ensino auxiliado por computador. I. Título. II. Soares, Eliana Maria do Sacramento, orient.

CDU 2. ed.: 004.43

Índice para o catálogo sistemático:

1. Programação (Computadores)	004.43
2. Aprendizagem	37.013
3. Ensino auxiliado por computador	37.091.64:004

Catalogação na fonte elaborada pela bibliotecária
Paula Fernanda Fedatto Leal – CRB 10/2291

UNIVERSIDADE DE CAXIAS DO SUL

"Aprendizagem de programação mediada por uma linguagem visual: possibilidade de desenvolvimento do Pensamento Computacional"

Leonardo Poloni

Dissertação de Mestrado submetida à Banca Examinadora designada pela Coordenação do Programa de Pós-Graduação em Educação da Universidade de Caxias do Sul, como parte dos requisitos necessários para a obtenção do título de Mestre em Educação. Linha de Pesquisa: Educação, Linguagem e Tecnologia.

Caxias do Sul, 16 de abril de 2018.

Banca Examinadora:

Dra. Eliana Maria do Sacramento Soares (presidente – UCS)

Dra. Carla Beatris Valentini (UCS)

Dra. Carine Geltrudes Webber (UCS)

Dr. Eliseo Berni Reategui (UFRGS)

Dedico essa dissertação à minha mãe Edita, *in memorian*, por ter sempre me incentivado e apoiado, não medindo esforços para proporcionar condições de avançar nos estudos.

AGRADECIMENTOS

Primeiramente a Deus, por estar sempre presente em minha vida e por ter permitido a realização desse trabalho.

À minha mãe Edita, *in memorian*, por todo amor e ensinamentos, pelo exemplo de vida, ética, bondade e profissionalismo. É uma ausência ainda muito sentida, mas a levo sempre comigo em meu coração e pensamentos.

Ao meu pai Jaime, por sempre ter sido um pai amigo e presente, batalhador, zeloso e bondoso. Um vovô que gosta de brincar e ajuda a cuidar dos netos.

À minha irmã Daniela pelo apoio, incentivo, por caminhar ao meu lado em todos os momentos. Um agradecimento especial pelas leituras e correções dos textos, foi uma ajuda muito importante para a realização desse trabalho!

À minha esposa Marisa, pelo companheirismo e por ter aberto mão de muitos momentos de lazer e convivência para possibilitar a efetivação desse trabalho.

Ao meu filho Miguel, minha razão de viver, força necessária para vencer essa etapa.

À minha orientadora, Profª. Drª. Eliana Maria do Sacramento Soares, pelo respeito, paciência e conhecimento que marcaram esse percurso de orientação, e pelo aprendizado que me proporcionou.

Aos membros da banca de qualificação, Profª. Drª Carla B. Valentini e Profª. Drª. Carine G. Webber, por suas contribuições para a realização dessa pesquisa.

Aos professores do PPGEdu, cada qual deixou sua marca e contribuiu de alguma forma para o desenvolvimento desse trabalho.

À Letícia Cancellier, secretária e colaboradora do Programa do Mestrado, que tão gentilmente me auxiliou em dúvidas, documentações e demais questões ao longo das disciplinas e eventos.

Aos amigos e colegas de mestrado, pelos momentos de diálogo, debate e apoio mútuo, contribuindo para superar os obstáculos e aprofundar os conhecimentos.

Aos alunos que participaram do piloto e da oficina, que contribuíram para a realização dessa pesquisa e possibilitaram um novo olhar, um redimensionamento sobre o fazer docente.

Enfim, a todos os que por algum motivo contribuíram para a realização dessa pesquisa.

RESUMO

O presente trabalho buscou identificar e analisar as formas de mediação possibilitadas pelo *Scratch* no processo de ensino aprendizagem de programação no Ensino Médio. Para isso, tomou como pergunta norteadora: como o *Scratch* pode mediar a aprendizagem de programação no Ensino Médio com vistas ao desenvolvimento do pensamento computacional a partir da teoria vigotskiana? O quadro teórico foi baseado na teoria sociointeracionista de Vigotski (2007), nos conceitos de pensamento computacional e de programação de computadores, aspirando a formação de cidadãos para o século XXI, apoiado pelas três dimensões do *framework* de Brennan e Resnick (2012): conceitos computacionais, práticas computacionais e perspectivas computacionais. O percurso metodológico foi delineado por um estudo de caso. Para criá-lo, foi organizada uma oficina de introdução à programação de computadores para alunos de primeiro ano do Ensino Médio Técnico de uma instituição de ensino da Serra Gaúcha. O *corpus* foi constituído pelos cadernos de reflexão dos alunos, pelos programas criados pelos estudantes para resolver cada tarefa, pelas anotações do pesquisador e pelo questionário pós-oficina. A análise do *corpus*, na busca de compreender e explicar o contexto dos alunos, atuando na oficina e utilizando o *Scratch,* seguiu as seguintes etapas: tabulação dos dados dos questionários pré-oficina; submissão dos programas criados pelos estudantes para resolver cada tarefa à análise do *Dr. Scratch*; análise por parte do pesquisador dos programas criados pelos estudantes com base no *framework* de Brennan e Resnick; e, análise conjunta dos cadernos de reflexão dos estudantes, das anotações do pesquisador e dos questionários pós-oficina, articulando com os resultados das etapas anteriores para construir a resposta à pergunta de pesquisa. Os desfechos alcançados apontam que o ambiente *Scratch* tem potencial para mediar o aprendizado de programação, proporcionando um ambiente amigável, dinâmico e motivador, com características de micromundo. Indicam ainda que o papel do professor precisa ser redimensionado. Ele precisa atuar como mediador, assim como sugere a teoria vigotskiana, criando estratégias e intervenções que tenham o potencial de levar o aluno a dar sentido às ações desenvolvidas no ambiente *Scratch*. Podemos dizer ainda, com base no estudo desenvolvido, que nesse contexto, os alunos podem desenvolver habilidades relacionadas ao pensamento computacional e interagir com autonomia.

Palavras-chave: Ambiente *Scratch*. Mediação. Pensamento Computacional. Aprendizagem de programação no Ensino Médio.

ABSTRACT

The present work sought to identify and analyze the forms of mediation made possible by Scratch in the process of teaching programming learning in High School. For this, it took as a guiding question: how does Scratch can mediate programming learning in High School with a view to the development of computational thinking from the Vygotskian theory? The theoretical framework was based on the socio-interactionist theory of Vygotsky (2007), in the concepts of computational thinking and computer programming, aspiring to the formation of citizens for the XXI century, supported by the three dimensions of the Brennan and Resnick framework (2012): computational concepts, practices, and perspectives. The methodological course was delineated by a case study. To create it, an introductory computer programming workshop was organized for first-year students of the Technical High School of a teaching institution in the state of Rio Grande do Sul. The corpus consisted of students' reflection books, the programs created by the students to solve each task, the researcher's notes and the post-workshop questionnaire. The analysis of the corpus, in the search to understand and explain the context of the students, acting in the workshop and using Scratch, performed the following steps: tabulation of the data of the pre-workshop questionnaires; submission of programs created by students to solve each task to Dr. Scratch's analysis; analysis by the researcher of the programs created by the students based on the Brennan and Resnick framework; and, joint analysis of the students' notebooks, the researcher's notes and the post-workshop questionnaires, articulating with the results of the previous steps to construct the answer to the research question. The results show that the Scratch environment has the potential to mediate programming learning, providing a friendly, dynamic and motivating environment with microworld characteristics. They also indicate that the role of the teacher needs to be resized. Teacher needs to act as a mediator, as suggested by the Vygotskian theory, creating strategies and interventions that have the potential to lead the student to give meaning to the actions developed in the Scratch environment. We can also say, based on the study developed, that in this context, students can develop skills related to computational thinking and interact with autonomy.

Keywords: Scratch Environment. Mediation. Computational Thinking. Programming learning in high school.

LISTA DE FIGURAS

Figura 1 - Quantitativo de dispositivos conectados à internet 18
Figura 2 - Exemplo de programa escrito utilizando a linguagem C#. 35
Figura 3 - Principais elementos do Scratch. 52
Figura 4 – Exemplo de programa escrito no Scratch. 53
Figura 5 - Relação do estudante com os ciclos de programação e a figura do mediador 56
Figura 6- Ambiente de realização da oficina. 63
Figura 7 - Projeto empregando o pensamento computacional em nível básico. 74
Figura 8 - Projeto empregando o pensamento computacional em nível intermediário (em desenvolvimento). 75
Figura 9 - Projeto empregando o pensamento computacional em nível avançado. ..75
Figura 10 - Projeto 1 do estudante E1 83
Figura 11- Utilização eficiente de laço de repetição. 84
Figura 12- Projeto 2 do estudante E1 85
Figura 13- Exemplo de inicialização de variáveis. 86
Figura 14- Exemplo de interação com o usuário. 86
Figura 15- Projeto 3 do estudante E1 88
Figura 16- Exemplo de utilização de funções (criação de blocos) 90
Figura 17- Exemplo de utilização do novo bloco (função) criado. 90
Figura 18- Inicialização da variável Pontuação com o valor zero. 91
Figura 19- Exemplo de incremento da variável Pontuação. 91
Figura 20- Projeto 4 do estudante E1 93
Figura 21- Exemplo de utilização de operadores lógicos e relacionais. 94
Figura 22- Exemplo de pensamento lógico condicional (se–então–senão) em conjunto com expressões lógicas. 94
Figura 23- Projeto 1 do estudante E2 97
Figura 24- Exemplo de emprego de comando desnecessário. 98
Figura 25- Projeto 2 do estudante E2 99
Figura 26- Projeto 3 do estudante E2 100
Figura 27- Exemplo de emprego de comando desnecessário. 101
Figura 28- Projeto 4 do estudante E2 102
Figura 29- Emprego de variável visível ao usuário, porém nunca utilizada. 103

Figura 30- Projeto 1 do estudante E3. 107
Figura 31- Utilização de pensamento lógico condicional. 107
Figura 32- Projeto 2 do estudante E3. 108
Figura 33- Exemplo de "código morto". 109
Figura 34- Projeto 3 do estudante E3. 110
Figura 35- Projeto 4 do estudante E3. 112
Figura 36- Regras e instruções do projeto 4. 113
Figura 37- Exemplo de utilização desnecessária de eventos. 114
Figura 38- Projeto 5 do estudante E3. 116
Figura 39- Exemplo de uso desnecessário de laço de repetição. 117
Figura 40- Exemplo de criação de blocos com nomes de comandos já existentes. 117
Figura 41- Exemplo de comandos com o mesmo nome, porém com funções diferentes. 118
Figura 42- Exemplo de utilização duplicada de evento. 118
Figura 43- Exemplo de definição de blocos não utilizados. 119
Figura 44- Projeto 1 do estudante E4. 122
Figura 45- Exemplo de comandos utilizados de maneira incorreta. 123
Figura 46- Exemplo de comandos desnecessários. 124
Figura 47- Exemplo de comandos duplicados, sem necessidade. 124
Figura 48- Projeto 2 do estudante E4. 125
Figura 49- Utilização de variável. 126
Figura 50- Alteração de fantasia que o usuário não visualiza. 126
Figura 51- Alteração de fantasia visível ao usuário. 126
Figura 52- Exemplo de comando que não tem efeito. 127
Figura 53- Espiral do Pensamento Criativo de Resnik. 130
Figura 54- Projeto 1 do estudante E5. 131
Figura 55- Projeto 3 do estudante E5. 133
Figura 56- Projeto 4 do estudante E5. 135
Figura 57- Elementos de mediação na oficina de introdução à programação. 148
Figura 58- Cenário da oficina com um exemplo de sociointeração. 152
Figura 59- Ambiente de programação padrão do Scratch. 157
Figura 60- Exemplo de emprego do Scratch como micromundo. 158
Figura 61- Exemplo de micromundo - Plano Cartesiano 159

LISTA DE QUADROS

Quadro 1 - Critérios de Inclusão e Exclusão. 28

Quadro 2 - Pesquisa realizada na base de periódicos da CAPES. 28

Quadro 3 - Resultado da pesquisa nos anais do Simpósio Brasileiro de Informática na Educação – SBIE. 29

Quadro 4 - Pesquisa nos anais do Workshop de Informática na Escola. 29

Quadro 5 - Funções Psicológicas Inferiores e Superiores 47

Quadro 6 - Relação de conceitos computacionais a serem abordados na oficina e sua respectiva descrição. 64

LISTA DE GRÁFICOS

Gráfico 1- Pontuação atribuída pelo Dr. Scratch a cada projeto analisado. 80

Gráfico 2- Importância dos elementos no aprendizado de programação e realização das tarefas..... 147

Gráfico 3- Utilização do Scratch pelos participantes fora do horário da oficina..... 149

LISTA DE ABREVIATURAS E SIGLAS

ACM	*Association for Computing Machinery*
BNCC	Base Nacional Comum Curricular
CAGR	*Compound Annual Growth Rate*
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNE	Conselho Nacional de Educação
CSTA	*Computer Science Teachers Association*
IFRS	Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
IoT	*Internet of Things*
ISTE	*International Society for Technology in Education*
LDB	Lei de Diretrizes e Bases da Educação
MEC	Ministério da Educação
MIT	*Massachusetts Institute of Technology*
NPD	Núcleo de Processamento de Dados
P21	*Partnership for 21st Century Learning*
PCN	Parâmetros Curriculares Nacionais
PCNEM	Parâmetros Curriculares Nacionais para o Ensino Médio
RFID	*Radio-Frequency IDentification*
SBC	Sociedade Brasileira de Computação
SBIE	Simpósio Brasileiro de Informática na Educação
SIGPROJ	Sistema de Informação e Gestão de Projetos do MEC
TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicação
UCS	Universidade de Caxias do Sul
WIE	Workshop de Informática na Escola
ZDP	Zona de Desenvolvimento Proximal

Sujeitos participantes da pesquisa:

E1
E2
E3
E4
E5

SUMÁRIO

1 APRESENTAÇÃO E CONTEXTUALIZAÇÃO 16

1.1 PROBLEMA DE PESQUISA 23

1.2 OBJETIVO GERAL 26

1.3 OBJETIVOS ESPECÍFICOS 26

2 QUADRO TEÓRICO 27

2.1 PENSAMENTO COMPUTACIONAL E O ENSINO DE PROGRAMAÇÃO 27

2.2 O CONCEITO DE PENSAMENTO COMPUTACIONAL 29

2.3 O CONCEITO DE PROGRAMAÇÃO 33

2.3.1 Sobre o *Scratch* 36

2.3.2 Ensino de programação e pensamento computacional no currículo da Educação Básica 39

2.4 O PENSAMENTO COMPUTACIONAL E A FORMAÇÃO DE CIDADÃOS PARA O SÉCULO XXI 41

2.5 A TEORIA SOCIOINTERACIONISTA 45

3 DELINEAMENTO METODOLÓGICO 54

3.1 ABORDAGEM TEÓRICO-METODOLÓGICA 54

3.2 PROCEDIMENTO DE GERAÇÃO DE DADOS 55

3.2.1 Estudo piloto 57

3.2.2 Realização e descrição da oficina 60

3.3 CONSTITUIÇÃO DO *CORPUS* 66

4 PROCEDIMENTO DE ANÁLISE 67

4.1 ANÁLISE DOS QUESTIONÁRIOS PRÉ-OFICINA 67

4.2 ANÁLISE DOS PROJETOS PELO DR. SCRATCH 71

4.2.1 Conhecendo o *Dr. Scratch* **71**

4.2.2 Relatórios de análise dos projetos gerados pelo *Dr. Scratch* **76**

4.3 ANÁLISE DOS PROJETOS À LUZ DOS NORTEADORES DO QUADRO TEÓRICO 80

4.3.1 Conhecendo o *framework* de Brennan e Resnick **81**

4.3.2 Análise dos projetos **82**

4.3.2.1 Análise dos projetos do participante E1 83

4.3.2.2 Análise dos projetos do participante E2 97

4.3.2.3 Análise dos projetos do participante E3 106

4.3.2.4 Análise dos projetos do participante E4 121

4.3.2.5 Análise dos projetos do participante E5 130

5 O QUE EMERGE DA ANÁLISE DO *CORPUS* **138**

6 CONSIDERAÇÕES FINAIS **151**

REFERÊNCIAS **161**

APÊNDICE A – FORMULÁRIO PESQUISA CURSO DE PROGRAMAÇÃO **168**

APÊNDICE B – FORMULÁRIO DE AVALIAÇÃO DA OFICINA **172**

APÊNDICE C – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO **176**

APÊNDICE D – TERMO DE ASSENTIMENTO LIVRE E ESCLARECIDO **179**

1 APRESENTAÇÃO E CONTEXTUALIZAÇÃO

Desde o início da graduação em Ciência da Computação, em 1996, fui movido pela vontade de conhecer e aprender a utilizar as tecnologias digitais. Comecei a estagiar junto ao Núcleo de Processamento de Dados (NPD) da Universidade de Caxias do Sul (UCS), o que fez aumentar meus conhecimentos e fascínio pela tecnologia. Estagiei por três semestres nesse setor, período em que coincidiu com o surgimento dos computadores pessoais. A própria universidade estava começando a trocar os poucos e antigos equipamentos, exclusivos até então de alguns setores, e iniciando a expansão do uso de computadores nos mais diversos centros e departamentos.

A maioria dos usuários tinha um conhecimento muito restrito quanto ao uso dos recursos tecnológicos. Cabia ao grupo de suporte do NPD – do qual eu fazia parte – instalar, configurar, fazer manutenção de *hardware* e *software*, instruir o usuário, além de outras atribuições. Todas essas atividades, em especial o auxílio ao usuário, contribuíram para aumentar meu interesse por ensinar, por ajudar os outros a utilizar os recursos tecnológicos.

Trabalhei como professor de informática em diversas escolas e instituições, onde tive a oportunidade de atuar com as mais diversas faixas etárias, com turmas englobando o Ensino Fundamental, Médio e Superior, grupos da terceira idade, além de turmas com familiares dos alunos e com professores.

Concomitantemente a atuação acadêmica também trabalhei com análise, desenvolvimento e teste de *software*. Essa atuação em empresas de desenvolvimento de *software* enriqueceu e contribuiu na minha atuação como professor. Desde o início de 2013 atuo como professor de informática no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – IFRS, ministrando disciplinas de informática para o Ensino Médio e cursos superiores.

Em 2014, ingressei no Programa Especial de Formação Pedagógica da UCS – exigência da instituição por trabalhar com alunos do Ensino Médio. Foi muito importante fazer esse curso, pois ampliou minha visão sobre a escola, sobre os alunos e principalmente sobre a didática e o fazer pedagógico. Desafiei-me a mudar minha

forma de atuar em sala de aula, busquei outras estratégias a fim de pôr em prática o que estava vendo nas disciplinas do curso.

Essa formação pedagógica contribuiu para aprimorar a forma de ver e observar as turmas. Percebi que as turmas eram muito heterogêneas, mas que a maioria dos alunos já possuía familiaridade com o uso da tecnologia. Já nasceram em outra época, na qual a tecnologia está muito mais acessível e presente em todos os ambientes. Vivemos em uma sociedade interconectada, com acesso simples e rápido aos mais diferentes conteúdos e informações, onde a grande maioria dos alunos possui seu *smartphone* (geralmente com acesso à internet), além de conhecer e ter outros recursos tecnológicos.

Essa mudança de comportamento mostra que estamos caminhando para uma nova fase, na qual a tecnologia faz parte do cotidiano, trazendo consigo inúmeras possibilidades e facilidades. Atingimos um estágio de evolução tecnológica com outro nível de maturidade, o que possibilita pensar nas mais diversas interações entre os mais variados equipamentos e objetos.

Conhecida como IoT (*Internet of Things*) ou internet das coisas, surgiu do conceito desenvolvido ao longo dos séculos de ampliar as conexões humanas. Esse conceito, segundo a Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação[1] - Brasscom, está ligado à ideia de colocar as pessoas conectadas com tudo, com todos e em qualquer lugar, possibilitando uma popularização e maior abrangência do uso de dispositivos de sensoriamento, interconectando o mundo físico e biológico ao universo digital (BRASSCOM, 2013).

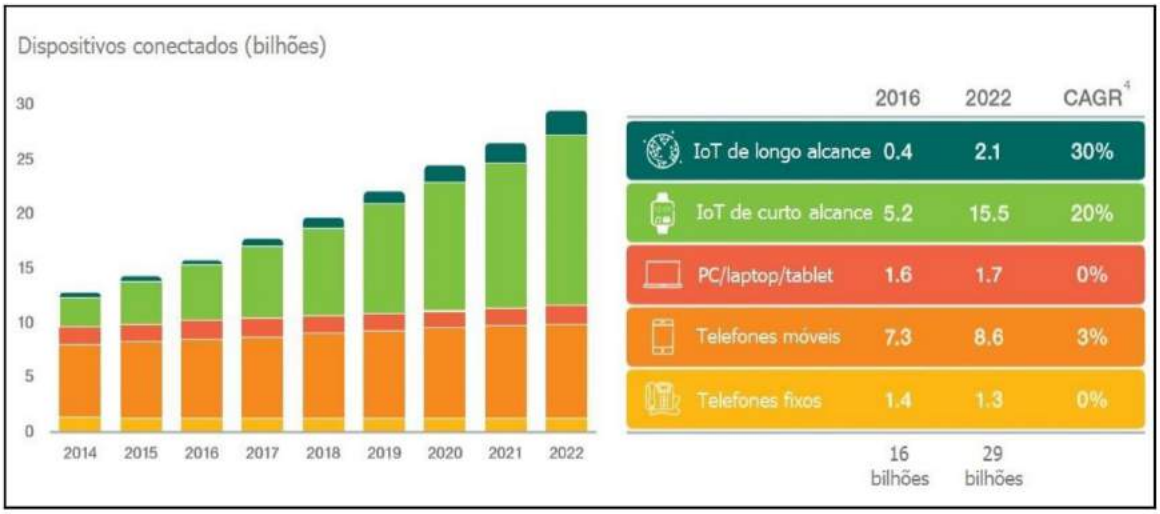
Evans (2011), fez um estudo sobre as proporções e possibilidades das conexões na internet das coisas cujo resultado sugere que a internet das coisas entrou em funcionamento quando o número de dispositivos conectados à internet ultrapassou o número de pessoas conectadas ao ciberespaço, "a IoT é o momento exato em que

[1] A Brasscom, Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação, objetiva promover o desenvolvimento da indústria de software e de serviços de tecnologia da informação e comunicação no Brasil, assim como todos os segmentos relacionados, desde a cadeia produtiva até os consumidores de bens e serviços baseados em tecnologia da informação e comunicação. A Brasscom atua como articuladora entre setores públicos e privados, nas esferas municipal, estadual e federal, discutindo temas estratégicos, como relações laborais, tributação, internacionalização, educação e governo digital, entre outros (BRASSCOM, 2016).

foram conectados à internet mais 'coisas ou objetos' do que pessoas" (EVANS, 2011, p. 2).

De acordo com o relatório de mobilidade da Ericsson (HEUVELDOP, 2017), estão previstos para o ano 2022 cerca de 29 bilhões de dispositivos[2] conectados, dos quais cerca de 18 bilhões estarão relacionados a IoT. Os dispositivos IoT conectados incluem carros, máquinas, medidores, sensores, terminais de ponto de venda, eletrônicos[3] de consumo e que podem ser vestidos. Entre os anos de 2016 e 2022, espera-se que os dispositivos IoT aumentem um CAGR[4] de 21%, impulsionados por novas possibilidades de uso, conforme ilustra a Figura 1.

Figura 1 - Quantitativo de dispositivos conectados à internet



Fonte: Adaptado de Heuveldop (2017).

Na Figura 1, a IoT é dividida em segmentos de curto e longo alcance. O segmento de curto alcance consiste principalmente em dispositivos conectados por tecnologias de rádio sem necessidade de licença com alcance de até 100 metros,

[2] Para a previsão da Ericsson, um dispositivo conectado é um objeto físico que possui um processador, permitindo a comunicação através de uma interface de rede Nota: telefones fixos tradicionais estão incluídos por motivos de legados.

[3] Incluindo: TVs inteligentes, caixas de mídia digital, leitores de Blu-Ray, consoles de jogos, receptores de áudio/vídeo (AV), etc.

[4] CAGR (*Compound Annual Growth Rate*) em português significa Taxa de Crescimento Anual Composta. Maiores informações: http://marcelotoledo.com/cagr-como-calcular-este-kpi-fundamental/

como Wi-Fi, Bluetooth e ZigBee[5]. Essa categoria também inclui dispositivos conectados através de rede de linha local fixa e tecnologias de rede elétrica. O segmento de longo alcance consiste em dispositivos que utilizam conexões celulares, bem como tecnologias de baixa potência sem licença, como Sigfox[6], LoRa[7] e RPMA[8]. Atualmente, a tecnologia dominante nesse segmento é a GSM/GPRS[9] (HEUVELDOP, 2017).

Um exemplo prático da IoT é a utilização nos automóveis de um dispositivo que permite a passagem pelas praças de pedágio sem a necessidade de parar nas cancelas para efetuar o pagamento. Ao se aproximar da cancela, esse dispositivo transmite as informações do carro e de cobrança do motorista, para o sistema de recepção instalado na guarita, liberando a passagem automática pela cancela. Outro exemplo de aplicação de tecnologia de transmissão de dados no mundo atual que descrevem os conceitos da IoT é uma pessoa sair para sua corrida e seu tênis transmitir dados do exercício para o celular que, via aplicativo, analisa o desempenho do atleta e compartilha nas redes sociais. Outros exemplos poderiam ser citados, tanto na área da agricultura e manejo de gado, como em controle de estoques, monitoramento de tráfego, na área da saúde, etc.

A IoT também está presente na área da educação, como mostram os estudos de Lemos (2012), os quais relatam uma experiência para controlar a entrada e a saída de alunos da escola Centro Municipal de Educação Professor Paulo Freire, em Vitória da Conquista – BA. Os uniformes que os estudantes recebem da Prefeitura possuem

---

[5] Zigbee designa um conjunto de especificações para a comunicação sem-fio entre dispositivos eletrônicos, com ênfase na baixa potência de operação, na baixa taxa de transmissão de dados e no baixo custo de implementação. Mariores informações em: http://www.zigbee.org/

[6] A Sigfox é uma rede global para conectar dispositivos de IoT, transmitindo dados sem a necessidade de estabelecer e manter conexões de rede. Maiores detalhes em: https://www.sigfox.com/en

[7] A rede LoRa é uma solução sem fio sub-GHz em frequência não licenciada que endereça demandas para conexão entre dispositivos para aplicações de baixo consumo, longa distância (em alguns locais é possível conseguir 15 km) e baixo custo de infra-estrutura - https://www.lora-alliance.org/technology

[8] RPMA (Random Phase Multiple Access) é a tecnologia sem fio desenvolvida pela Ingenu. É uma rede sem fio projetada exclusivamente para aplicações de máquinas para máquinas (M2M) e Internet de coisas (IoT). Mairores informações: https://www.ingenu.com/technology/rpma/

[9] As redes GSM (Sistema Global para Comunicações Móveis) foram implantadas no mundo inteiro como sistemas de comunicação móvel de segunda geração. GPRS (Serviços Gerais de Rádio por Pacote) é um serviço de dados disponível para os usuários de celulares GSM - https://en.wikipedia.org/wiki/General_Packet_Radio_Service

dispositivos integrados, os quais utilizam uma etiqueta RFID[10] (ativadas por radiofrequência). O objetivo é controlar o acesso dos alunos nas aulas e identificar suas faltas e ausências. Quando a integração com o sistema detecta mais de 3 ausências ou fuga das aulas, os pais são imediatamente chamados.

A IoT na educação pode ser pensada através de aplicações e softwares educacionais integrando dispositivos conectados ao ciberespaço e na web para práticas pedagógicas em sala de aula, para simulação de ambientes e testes. Além disso, Lopes (2016) sugere a aplicação da IoT para auxiliar alunos com necessidades especiais e também como forma de implementar a aula invertida.

Segundo a autora, alunos com necessidades especiais podem se beneficiar dos dispositivos conectados de diversas maneiras. Uma delas é a disponibilização de um cartão especial para deficientes visuais que, quando registrado em um computador, automaticamente amplia o tamanho da fonte. Em vez de precisar chamar um professor para auxiliá-lo o tempo todo, o educador e o aluno podem criar novas ideias e propor novas discussões mais produtivas, uma vez que o aluno ganha independência e autoconfiança.

A IOT também colabora para aplicação do Modelo de Aula Invertida (*Flipped Classroom*) onde o professor orienta o aluno para o estudo fora da sala de aula, usando diferentes mídias e plataformas que podem ser acessadas quando e onde o estudante quiser. Nesse modelo de aula, também são realizados encontros presenciais e virtuais para a realização de aulas práticas referentes ao conteúdo que foi indicado de início. A aprendizagem é mais rápida e significativa, com projetos colaborativos e resolução de problemas (LOPES, 2016).

Sobre a inserção das tecnologias nos processos educacionais, Santos (2013) traz que:

> [...] a Internet das Coisas é um importante marco evolutivo da Internet, uma nova forma emergente onde o homem passa a não ser o único condutor da ação, mas cria e permite que objetos mediem suas vidas. Esse processo evolutivo da Internet proporciona a sociedade maneiras inusitadas de analisar, coletar e distribuir dados, gerando novos conhecimentos, informação e sabedoria. (SANTOS, 2013, p.5)

[10] RFID - Identificação por Radiofrequência (tradução nossa), do inglês *Radio-Frequency IDentification*

Neste contexto, onde o computador e a internet estão cada vez mais ubíquos[11] na vida das pessoas, as tecnologias representam um dos principais alicerces desse ambiente. Entretanto, as áreas da Computação e da Tecnologia da Informação (TI) enfrentam grandes desafios na formação de profissionais para suprir as demandas do mercado.

Em pesquisa realizada pela Brasscom sobre o mercado de profissionais de Tecnologia da Informação e Comunicação (TIC) no Brasil, analisando dados do período de 2006 a 2013, concluiu que nos últimos anos, cresceu em mais de 70% o número de interessados em ingressar em cursos das áreas relacionadas a TI (Tecnologia da Informação) e TIC. Esse aumento pode ser em virtude do acesso e convivência diária dos jovens com as TICs gerando, por conseguinte, interesse deles por tecnologia (BRASSCOM, 2015).

Observando o período de 2010 a 2013, a pesquisa mostrou um crescimento anual de 19% dos inscritos em vestibulares, sem que houvesse alteração no número de vagas ofertadas, tornando mais acirrado o processo de ingresso dos alunos. Essa concorrência chegou ao índice de 2,4 candidatos por vaga no ano de 2013. Interessante destacar que, apesar do aumento significativo dos inscritos nos vestibulares, o número de ingressantes e, principalmente, de concluintes, mantiveram-se praticamente estáveis (*ibidem*, 2015).

Com relação ao mercado de trabalho, no período de 2006 a 2013, 313.904 pessoas foram contratadas. Já o número de alunos que se formaram no período foi de apenas 38.564, sendo que 130.340 alunos ingressaram em cursos correlatos. Chama atenção a alta taxa de evasão dos alunos, sendo que a média de concluintes no período analisado se manteve relativamente estável em 31% dos ingressantes (BRASSCOM, 2015).

Outro ponto relevante da pesquisa feita pela Brasscom (2015) é o fato de as escolas privadas terem formado 80% dos profissionais, com uma taxa de evasão em torno de 30%. Nas escolas públicas essa taxa é menor, girando em torno de 25%, e responde por apenas 20% da formação de recursos humanos. A possibilidade levantada pelos pesquisadores, quanto às causas dessa alta taxa de evasão, está

[11] O termo ubíquo refere-se ao fato de estar em todos os locais, descrever a presença constante da informática, internet e tecnologias na vida das pessoas, em suas casas e ambientes de convívio social.

relacionada ao fato dos estudantes abandonarem os cursos devido às responsabilidades profissionais. Por vezes, a carga horária de trabalho é muito extensa, atrapalhando os estudos, outras vezes, o estudante não sente a necessidade de concluir a graduação por já ter conseguido uma boa colocação no mercado de trabalho. Somam-se a esses fatores as dificuldades encontradas no curso, baixa compreensão da profissão além de questões relacionadas à valorização da profissão.

A pesquisa da Brasscom (2015) também constatou que as certificações possuem maior relevância que o diploma acadêmico para os cargos de nível técnico e de entrada no mercado de trabalho. Porquanto a certificação reconhece o domínio específico daquela tecnologia por um determinado período. As certificações são reconhecidas e valorizadas globalmente.

A titulação acadêmica é levada em consideração pelas empresas como diferencial, devido à qualidade do Ensino Médio brasileiro, no qual os estudantes, muitas vezes, não conseguem uma boa base educacional e de formação. Essa constatação da pesquisa vem reforçar a ideia de que são necessárias mudanças no Ensino Fundamental e Médio para que os alunos cheguem melhor preparados nos cursos de graduação, evitando tamanha evasão e repetência. Além disso, é preciso preparar os jovens para conseguirem dominar as novas tecnologias e a demanda por profissionais capazes de criar, inovar e se adaptar (BRASSCOM, 2015).

Nesse cenário apresentado no texto, no qual a evolução tecnológica revolucionou todas as esferas da sociedade, o professor e a própria Educação necessitam se adaptar às novas necessidades, assumindo um papel central no processo de formação dos sujeitos. É importante que o professor se mantenha atualizado, reveja suas práticas pedagógicas e procure deixar de ser apenas um transmissor de informações ou de técnicas e passe a ser um mediador do processo de construção do conhecimento.

Um dos caminhos para preparar os jovens a enfrentar as demandas da sociedade do século XXI é o desenvolvimento de habilidades e competências para buscar soluções aos mais diversos tipos de problemas e desafios, incentivando-os a pensar de forma crítica e lógica.

1.1 PROBLEMA DE PESQUISA

A sociedade contemporânea, em suas múltiplas denominações (do conhecimento, da aprendizagem colaborativa, autoral, cibercultural, neoliberal, pós-moderna, neurocultural, líquida, dentre outras), caracteriza-se por apresentar sujeitos sociais. Esses indivíduos se informam por hipertexto na Internet, divertem-se com jogos, relacionam-se por redes sociais e virtuais (Facebook, WhatsApp, Youtube, Twitter, Google+, Instagram, etc.) e podem produzir em qualquer lugar. Assim sendo, "a distância não é nada e a velocidade é tudo" (LÉVY, 1997, p. 218).

Para Lévy (1995), o hipertexto é uma rede composta de nós ligados por conexões. Os nós podem ser palavras, páginas, imagens ou partes de imagens, sequências sonoras, referência a documentos complexos que podem ser eles mesmos hipertextos. Os nós não estão ligados linearmente, como em uma corda ou como nos elos de uma corrente mas cada um deles, ou a maioria, estende suas conexões em estrela, de modo reticular. Em conjunto com o visualizador (browser) representa um tipo de sistema para a organização de conhecimentos ou dados, aquisição de informações e comunicação.

Segundo Lévy (1995, p. 28), a mente humana "pula de uma representação para outra ao longo de uma rede intrincada, desenha trilhas que se bifurcam, tece uma trama infinitamente mais complicada do que os bancos de dados de hoje". Para adentrar em outras páginas e conteúdos, o texto tem disponíveis links, nós, e/ou hiperlinks que o ligam a outros textos, a outras páginas que o complementam, podendo ser escritos, sonoros ou visuais, os quais proporcionam uma desterritorialização do texto.

Estamos vivenciando a era em que tarefas do cotidiano são desempenhadas quase sempre com a intermediação das tecnologias digitais, o que possibilita não ficarmos presos a um único local, como um computador em um escritório, apesar de ser um dos principais meios de produção. Essas tecnologia digitais, segundo Lévy (2000, p. 32), "surgiram como a infraestrutura do ciberespaço, novo espaço de comunicação, de sociabilidade, de organização e de transmissão, mas também novo mercado da informação e do conhecimento".

O jovem de hoje faz parte da primeira geração imersa quase que totalmente na tecnologia, onde percebemos os meios tecnológicos como se fossem extensões do homem. É a tecnologia acoplando-se ao indivíduo. Esses jovens constituem-se, sobretudo, como geradores de conteúdos e não apenas como consumidores, pois aprendem a usar as tecnologias por meio do desenvolvimento de diferentes tarefas, criando habilidades para a utilização de recursos digitais enquanto procuram atingir seus objetivos.

Variados recursos e ferramentas tecnológicas estão disponíveis, bastam alguns cliques, alguns toques na tela e pronto, um novo vídeo foi criado, uma foto retocada, uma charge montada e em segundos tudo é publicado, curtido, comentado. O que até alguns anos atrás era difícil, caro e geralmente só podia ser feito por um profissional, agora está disponível na maioria dos celulares e demais dispositivos tecnológicos.

Os *softwares* e *hardwares* evoluíram de forma surpreendente, reduzindo seu custo, diminuindo seu tamanho, agregando mais recursos e funcionalidades. Além disso, a interface tornou-se mais intuitiva e amigável, atraindo e conquistando cada vez mais usuários, das mais diferentes faixas etárias. Vemos crianças se apropriando naturalmente dessa tecnologia e utilizando-a de forma hábil e automática.

Outro fator que levou a popularização dos computadores, celulares e demais recursos tecnológicos foi o avanço da internet. A partir do momento em que a rede mundial ganhou mais velocidade, passou a estar disponível em mais locais, principalmente nos celulares, tornou-se indispensável, essencial para a maioria da população pós-moderna. A internet é para Lévy (1996, p. 114), "um tapete de sentido tecido por milhões de pessoas e devolvido sempre ao tear. Da permanente costura pelas pontas de milhões de universos subjetivos emerge uma memória dinâmica, comum, 'objetivada', navegável".

Em geral, observando os jovens nas escolas podemos dizer que eles não mostram interesse nas atividades escolares. Entretanto, estão sempre conectados, buscando curtir/comentar seus contatos nas redes sociais; querendo ouvir música enquanto assistem às aulas, preferindo ficar jogando no computador a focarem-se nas atividades propostas.

Por outro lado, essa facilidade em utilizar os aplicativos para comunicação e entretenimento não se estende a outros programas como, por exemplo, as planilhas eletrônicas. A maioria dos estudantes de 1º ano do Ensino Médio apresenta dificuldades em utilizar lógica, em estabelecer relações, em montar cálculos e fórmulas matemáticas.

Nessa conjuntura, podemos perceber a necessidade de novas abordagens para o ensino, que não tratem as características dessa nova geração como um empecilho ao ensino e aprendizagem, mas sim que as aproveitem de forma a aproximar o mundo da escola do mundo dos alunos. Além disso, cabe à escola preparar esses alunos para conviverem e atuarem em uma sociedade permeada pela computação. Nesse sentido, não se pode deixar de integrar aspectos básicos da Ciência da Computação na formação dos alunos (FRANÇA; AMARAL, 2013).

Alguns destes conceitos introdutórios de Computação, também denominados de Pensamento Computacional, são: aprender a construir algoritmos, estruturar o pensamento lógico, dividir problemas complexos em problemas menores, desenvolver habilidades para elaborar soluções para problemas de maneira que permita a utilização de um computador e outras ferramentas para ajudar a resolvê-los (CSTA; ISTE, 2011).

Analisando o contexto internacional destacamos que os Estados Unidos, Portugal, Espanha, Reino Unido dentre outros países europeus vêm alterando seus currículos escolares, pensando na formação de cidadãos capazes de atuar em uma sociedade tecnológica. Essa alteração dos currículos escolares da Educação Básica busca inserir o pensamento computacional de forma gradual na formação dos estudantes. No caso do Reino Unido, desde 2014, todas as escolas são obrigadas a ensinar linguagem de computação para alunos a partir dos cinco anos. Estônia, Finlândia e Austrália também já incorporaram a disciplina no currículo. Atualmente, 40% das escolas americanas ensinam programação e a meta do governo é dobrar esse número nos próximos anos. No Brasil, o aprendizado de programação está ganhando mais espaço na grade curricular ou extracurricular das escolas, porém ainda está mais restrito à rede particular (DINIZ, 2017).

Discorrendo sobre esse tema, Zorzo, (2017), Diretor de Educação da Sociedade Brasileira de Computação (SBC), enfatiza que a Computação é uma área

que permeia, atualmente, todas as demais áreas do conhecimento e as impacta profundamente. Problemas complexos de diferentes áreas da ciência estão agora sendo abordados com uma perspectiva computacional, uma vez que a Computação provê estratégias e artefatos para lidar com a complexidade, avançando na solução de problemas que há poucos anos não seriam possíveis. Como exemplo, esse autor cita o mapeamento do genoma humano, a identificação de variações de enzimas, a simulação da adaptação de seres vivos em diferentes ambientes, o desenvolvimento de medicamentos, realização de cirurgias remotas e até mesmo na simulação de previsão de tempo de contaminação por uma doença em um determinado ambiente.

Nesse contexto, emerge o problema de pesquisa dessa dissertação: **como o *Scratch* pode mediar a aprendizagem de programação no Ensino Médio com vistas ao desenvolvimento do pensamento computacional a partir da teoria vigotskiana[12]?**

1.2 OBJETIVO GERAL

Identificar e analisar as formas de mediação possibilitadas pelo *Scratch*[13] no processo de ensino aprendizagem de programação no Ensino Médio.

1.3 OBJETIVOS ESPECÍFICOS

a) Constituir o quadro teórico para fundamentar a pesquisa, em especial o conceito de mediação e aprendizagem na teoria vigotskiana.

b) Construir o *corpus* da pesquisa por meio da realização de uma oficina de programação para alunos do Ensino Médio, utilizando o *Scratch* e inspirada nos conceitos apresentados no quadro teórico.

c) Analisar o corpus da pesquisa buscando entender de que forma o Scratch foi elemento mediador do processo de aprendizagem de programação.

---

[12] Optamos por empregar a grafia vigotskiana, conforme referencial teórico adotado - Vigotski (1987; 2007).

[13] O *Scratch* é uma linguagem de programação desenvolvida pelo MIT e será detalhado na seção 4.3.1.

2 QUADRO TEÓRICO

Para desenvolver essa pesquisa propusemos um referencial teórico articulando os conceitos de pensamento computacional, programação de computadores e aprendizagem, baseado em diversos autores, destacando Seymour Papert, Jeannette Wing, Mitchel Resnick e Lev Vigotski[14]. Procuramos apresentar um panorama mundial sobre o pensamento computacional e o desenvolvimento de competências e habilidades fundamentais para o século XXI, além de sua inserção no currículo da Educação Básica.

2.1 PENSAMENTO COMPUTACIONAL E O ENSINO DE PROGRAMAÇÃO

Apresentamos algumas considerações relacionadas ao panorama atual das pesquisas relacionadas ao emprego do pensamento computacional no contexto do aprendizado de programação, bem como as pesquisas voltadas à utilização do ambiente *Scratch*.

Para tanto foi realizado um levantamento bibliográfico por meio de busca manual nos anais do Simpósio Brasileiro de Informática na Educação – SBIE, do Workshop de Informática na Escola - WIE e na base de dados da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), sem a utilização de filtro por ano.

Essa busca ocorreu em duas fases. A primeira, que estabeleceu-se em maio de 2016, foi a pré-seleção dos artigos, a qual consistiu em verificar os anais dos referidos eventos e a base da CAPES, acessando manualmente todos os artigos completos, lendo os seus títulos, resumos e palavras-chave. Na segunda fase, todos os artigos pré-selecionados foram analisados e a cada um deles foram aplicados os critérios de inclusão e exclusão, apresentados no Quadro 1.

[14] Optamos por empregar a grafia Vigotski (1987; 2007), conforme referencial teórico adotado.

Quadro 1 - Critérios de Inclusão e Exclusão.

Critérios de Inclusão	Critérios de Exclusão
- Artigos completos que abordam o desenvolvimento de habilidades do pensamento computacional no Ensino Médio; - Artigos completos que relacionam o emprego do pensamento computacional no ensino de programação; - Artigos completos que abordam o processo de ensino-aprendizagem de programação através do *Scratch*; e - Artigos completos relacionados ao ensino de programação no Ensino Médio.	- Artigos duplicados; - Artigos que abordam o desenvolvimento do pensamento computacional fora da Educação Básica; - Artigos que abordam o processo de ensino-aprendizagem de programação sem a utilização do *Scratch*; - Artigos resumidos; e - Artigos não relevantes (excluídos pelo título, resumo, palavras-chave não relacionados aos objetivos propostos).

Fonte: criado pelo pesquisador (2016).

O Quadro 2 apresenta os resultados da pesquisa na base de dados da CAPES. Na primeira coluna, estão listados os termos utilizados nas pesquisas; e, na segunda coluna, a respectiva quantidade de trabalhos disponíveis.

Quadro 2 - Pesquisa realizada na base de periódicos da CAPES.

BASE DA CAPES	
Termo pesquisado	**Quantidade**
Pensamento computacional	29
Scratch	20.639
Scratch e pensamento computacional	1
Pensamento computacional e ensino médio	0
Scratch e ensino médio	0

Fonte: criado pelo pesquisador (2016).

No Quadro 3, estão relacionados os resultados obtidos em pesquisa nos anais do Simpósio Brasileiro de Informática na Educação – SBIE; e, no Quadro 4, estão relacionados os resultados obtidos em pesquisa nos anais do Workshop de Informática na Escola. Em ambos os quadros, na primeira coluna estão listados os termos utilizados nas pesquisas e, na segunda coluna, a respectiva quantidade de trabalhos disponíveis.

Quadro 3 - Resultado da pesquisa nos anais do Simpósio Brasileiro de Informática na Educação – SBIE.

ANAIS DO SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE	
Termo pesquisado	**Quantidade**
Pensamento computacional	8
Scratch	4

Fonte: criado pelo pesquisador (2016).

Quadro 4 - Pesquisa nos anais do Workshop de Informática na Escola.

ANAIS DO WORKSHOP DE INFORMÁTICA NA ESCOLA	
Termo pesquisado	**Quantidade**
Pensamento computacional	9
Scratch	15

Fonte: criado pelo pesquisador (2016).

A partir desses dados é possível dizer que a relação entre o Pensamento Computacional e a programação ainda é pouco estudada em nosso país. Os trabalhos desenvolvidos considerados relatam, em geral, experimentos e tentativas do ensino de programação, de robótica ou outras estratégias de desenvolver habilidades e competências relacionadas ao pensamento computacional para alunos do Ensino Médio e Fundamental.

2.2 O CONCEITO DE PENSAMENTO COMPUTACIONAL

O termo Pensamento Computacional foi inicialmente definido por Jeannette Wing (2006) como um conjunto de habilidades e competências da Computação que podem ser úteis em outras áreas do conhecimento. A autora defende a resolução de problemas aplicando o pensamento computacional, através da redução de problemas grandes e aparentemente insolúveis em problemas menores e mais simples de serem resolvidos. Para isso, é necessário pensar de forma abstrata e em múltiplos níveis, não apenas em uma simples aplicação de técnicas de programação.

A autora considera o pensamento computacional como uma habilidade fundamental para possibilitar a resolução de problemas através do uso de

computadores, os quais estão presentes na maioria dos espaços da sociedade contemporânea. Logo, essa habilidade deveria ser desenvolvida por todos os estudantes. Utilizar o pensamento computacional na resolução de problemas não significa delegar aos computadores a tarefa de pensar, mas sim pensar e encontrar maneiras para que os problemas possam ser resolvidos pelos computadores. As pessoas pensam, criam e os computadores executam, ganhando, com isso, tempo e reduzindo trabalhos automáticos e repetitivos (ibid., ibidem).

Ainda conforme Wing (2006), o eixo central da utilização do pensamento computacional está na geração de ideias e na utilização dos conceitos fundamentais da Computação para resolver os mais variados problemas do cotidiano sem que para isso seja necessária a produção de hardware ou software. Essa forma de pensar e resolver problemas pode ser útil para todas as pessoas, independentemente da área de formação ou atuação.

Jeannette Wing promoveu diversas palestras, debates e conferências, levando à discussão e reflexão, a nível global, sobre a importância desse tema e a necessidade de repensar e reestruturar a educação a fim de preparar os jovens para atuarem nesta sociedade cada vez mais tecnológica. Atuação essa que, para Mitchel Resnick, vai muito além do simples navegar na internet, conversar e interagir, buscando desenvolver nas crianças e nos jovens habilidades como desenhar, criar e combinar. Propõe, para isso, um aprofundamento no conhecimento e domínio das tecnologias relacionadas à Computação, em especial a programação de computadores (Resnick et al., 2009).

O pensamento computacional reúne uma diversidade de conceitos, teorias, modelos e áreas de prática de um vasto campo que é a Ciência da Computação. Por ser um construto bastante amplo, entidades internacionais ligadas à Ciência da Computação e tecnologias na educação elaboraram algumas definições sobre o pensamento computacional a fim de facilitar e padronizar o entendimento sobre esse conceito e, com isso, possibilitar sua inserção no currículo escolar. Essas associações internacionais, com sede nos Estados Unidos, protagonizaram a elaboração de documentos e propostas de currículos para o sistema de educação americano (K-12), seguidos por iniciativas em diversos outros países (CSTA, 2011).

Para uma melhor compreensão do tema e contextualização do panorama americano, faz-se necessário contextualizar e definir alguns elementos e organizações, haja vista as diferenças nas estruturas escolares entre Brasil e Estados Unidos, além da necessidade de compreensão da importância, dimensão e respectivos papéis das organizações citadas. Assim, esclarecemos a seguir a introdução do termo K-12.

O termo K-12, segundo o departamento de educação dos Estados Unidos (EDUCATION, 2017), é utilizado na educação e na tecnologia educacional nos Estados Unidos, Canadá, Coreia do Sul, Turquia, Filipinas, Egito, Austrália e Irã. Também é ocasionalmente usado na Austrália. Essa é uma forma abreviada para os níveis escolares antes da faculdade. Esses níveis são do jardim de infância (K) e do 1º ao 12º ano (1-12) e correspondem à Educação Básica no Brasil.

A *Association for Computing Machinery*[15] - ACM (2016) é a maior sociedade educacional e científica do mundo, reunindo educadores em Computação, pesquisadores e profissionais para inspirar o diálogo, partilhar recursos e enfrentar os desafios do campo. Fundada em 1947, na Universidade de Columbia em Nova Iorque, possui mais de 100 mil membros, sendo que destes, metade reside fora dos Estados Unidos. Possui conselhos na Europa, Índia e China, promovendo uma rede mundial de contatos direcionada à Computação, visando fortalecer os laços dentro e entre países e comunidades técnicas. Suas ações visam a sensibilização para importantes questões técnicas, educacionais e sociais da Computação em todo o mundo.

Uma das ações da ACM tenciona a capacitação dos estudantes a fim de prepará-los para desempenhar plenamente suas atividades profissionais. Participam dessa ação mais de 500 faculdades e universidades em todo o mundo, melhorando a aprendizagem dos alunos através da troca de ideias com outros estudantes e profissionais estabelecidos. Além disso, a ACM oferece 1,5 milhão de dólares em bolsas de estudos para estudantes de graduação e pós-graduação. Parcerias com grandes empresas do setor, como a Microsoft, Google e IBM, possibilitam o desenvolvimento de diversas pesquisas e competições na área da Computação, além de premiarem ideias inovadoras e importantes, bem como as ideias humanitárias e interdisciplinares (ACM, 2016).

---

[15] Associação para Engenharia da Computação (tradução nossa).

A *Computer Science Teachers Association*[16] - CSTA (2011) é a maior associação profissional dos professores de Ciências da Computação a nível mundial, com mais de 21 mil membros. Fundada em 2005 e com sede em Nova Iorque, essa associação faz parte de uma organização que promove e dá suporte aos professores de Ciência da Computação e outras disciplinas ligadas à área da Computação, desde o jardim da infância até o décimo segundo nível (3 º ano do Ensino Médio).

A *International Society for Technology in Education*[17] - ISTE (2016) é uma organização sem fins lucrativos, reunindo educadores e líderes educacionais comprometidos em aprimorar o uso da tecnologia na educação. A ISTE reúne mais de 100 mil agentes de educação em todo o mundo, fornecendo recursos de tecnologia educacional no apoio à aprendizagem profissional. Possui um amplo conjunto de diretrizes voltadas para professores, administradores, palestrantes, educadores em Ciência da Computação e também para estudantes. Essas diretrizes destacam as principais habilidades e conhecimentos necessários para compreender e integrar os conceitos computacionais, em uma sociedade cada vez mais global e digital. A ISTE fornece também um conjunto de recursos de aprendizagem profissional, incluindo seminários e cursos on-line, consultoria, livros, jornais, revistas e publicações.

No ano de 2003, a ACM publicou um modelo de Currículo para a Ciência da Computação na Educação Básica (K-12), sendo que, no ano de 2006, foi lançada sua segunda versão. O referido modelo descreve uma visão para a educação em Ciência da Computação no ensino básico, separado em quatro níveis (CSTA, 2011):

Nível I – Conceitos Básicos de Ciência da Computação, recomendado ao nível K-8 (do jardim da infância ao 8º ano do Ensino Fundamental).

Nível II – A Ciência da Computação no Mundo Moderno, recomendado ao nível 9 ou 10 (9º ano do Ensino Fundamental ou 1º ano do Ensino Médio).

Nível III – A Ciência da Computação como Análise e Design, recomendado ao nível 10 ou 11 (1º ano ou 2º ano do Ensino Médio).

Nível IV – Tópicos em Ciência da Computação, recomendado ao nível 11 ou 12 (2º ano ou 3º ano do Ensino Médio).

---

[16] Associação dos Professores em Ciência da Computação (tradução nossa).
[17] Associação Internacional para Tecnologia em Educação (tradução nossa).

A ISTE e a CSTA desenvolvem projetos em conjunto com coordenadores de Ensino Superior, membros da indústria e integrantes da educação K-12. Elas criaram uma definição operacional de pensamento computacional, a qual fornece conceitos e vocabulário para o pensamento computacional, compartilhados com todos os educadores K-12 (CSTA; ISTE, 2011).

Com base nos resultados de uma pesquisa com professores de Ciência da Computação e profissionais da área, a ISTE, em conjunto com a CSTA, definiu o Pensamento Computacional como sendo um processo de resolução de problemas, que inclui:

- formulação de problemas de maneira que permita a utilização de um computador e outras ferramentas para ajudar a resolvê-los;
- organização lógica e análise de dados;
- representação de dados através de abstrações, utilizando para isso modelos e simulações;
- automatização de soluções através do pensamento algorítmico (uma série de passos ordenados de maneira lógica);
- identificação, análise e implementação de possíveis soluções com o objetivo de alcançar a mais eficiente combinação de passos e recursos; e
- generalização e transferência do processo de resolução de um problema para uma ampla variedade de problemas.

2.3 O CONCEITO DE PROGRAMAÇÃO

Segundo Lamim (2008), a programação pode ser definida como o processo de escrita, teste e manutenção de um programa de computador. Este, por sua vez, é escrito em uma linguagem de programação, pois seria muito complexo escrevê-lo diretamente em linguagem de máquina, denominada linguagem nativa do computador. Essa linguagem de máquina está baseada na lógica binária, que, por sua vez, tem seus fundamentos na álgebra booleana, definida por George Boole[18].

[18] **George Boole** foi um autodidata que, aos 20 anos de idade, fundou sua própria escola e dedicou-se ao estudo da Matemática. Em 1847 publicou *The Mathematical Analysis of Logic* no qual introduziu os conceitos de lógica simbólica demonstrando que a lógica podia ser representada por equações algébricas. Este trabalho foi fundamental para a construção e programação dos computadores eletrônicos, iniciada cerca de 100 anos mais tarde. Na Álgebra de Boole existem apenas três

Uma das principais funções das linguagens de programação, de acordo com Andrade (2015), é de servir como meio de comunicação entre humanos e computadores, pois estes precisam de instruções claras, precisas para tudo o que fazem. Por isso as linguagens de programação são baseadas em caracteres/códigos matemáticos, lógicos e computacionais, constituindo uma linguagem artificial. O que é óbvio para os humanos, certamente não é óbvio para uma máquina. E se desejamos que a máquina faça algo para nós, necessitamos uma maneira de nos comunicar com ela, ou seja, precisamos codificar algum programa que faça o que se pretende.

Uma linguagem de programação atua como um tradutor entre quem está escrevendo o programa e o computador. Em vez de aprender a linguagem nativa do computador, o programador pode fazer uso de uma linguagem de programação para instruir o computador de uma maneira mais fácil, tanto de aprender como de entender.

Para Antônio Leitão (1995), uma linguagem de programação deve primar pela simplicidade. A combinação de ideias e comandos simples deve ser capaz de formar ideias mais complexas, além de possibilitar a realização de abstrações de ideias complexas, a fim de torná-las simples.

Quem não conhece programação costuma associá-la com caracteres passando em uma tela, como ocorre no filme *Matrix*[19]. A programação é percebida como uma listagem de palavras em inglês, números e símbolos, muito difícil de ser entendida. Essa visão tem certa base na realidade, pois grande parte dos programas de computador são escritos na forma de texto, utilizando linguagens de programação como: Java, C, PHP, Python, entre outras. Mesmo linguagens visuais como o C#, da Microsoft, operam baseadas em comandos textuais, sendo que a parte visual está muito mais relacionada com a construção (*design*) de páginas para a internet ou formulários.

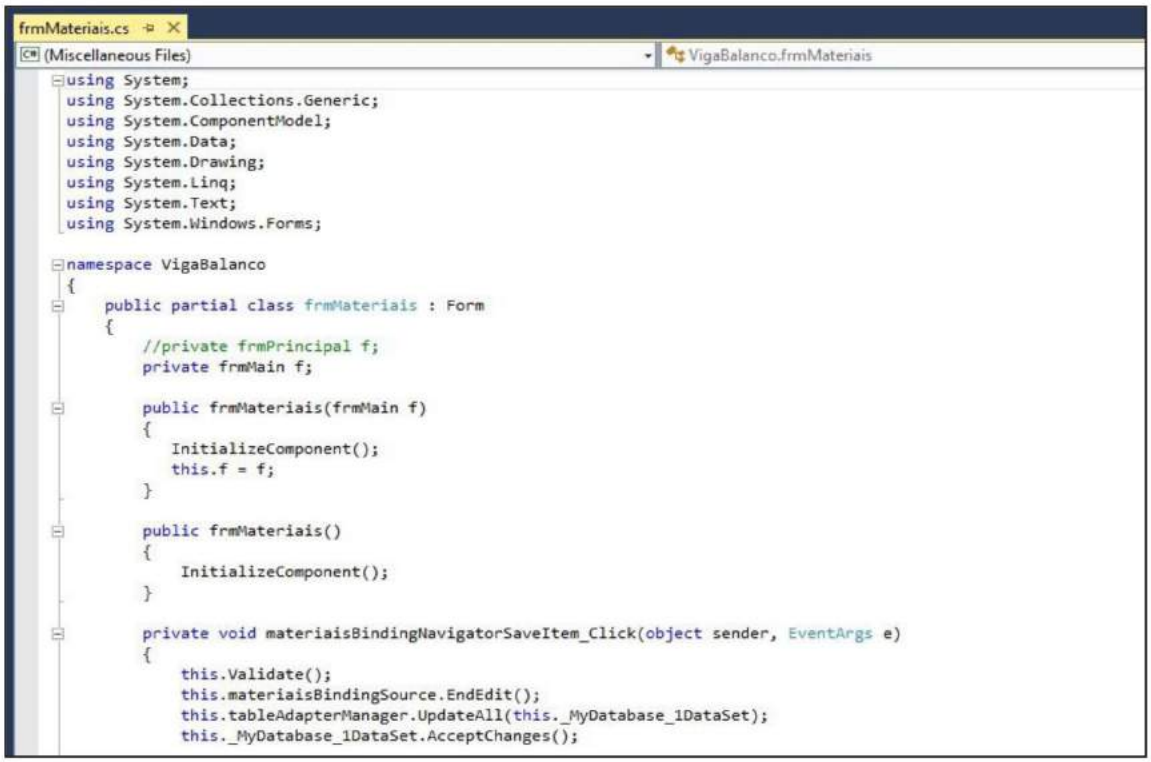
A Figura 2 ilustra parte de um programa de computador elaborado utilizando a linguagem C#.

---

operadores E, OU e NÃO (AND, OR, NOT). Estas três funções são as únicas operações necessárias para efetuar comparações ou as quatro operações aritméticas base. Ele é considerado um dos fundadores da Ciência da Computação, apesar de computadores não existirem em seus dias. Para maiores informações acesse: http://brasilescola.uol.com.br/biografia/george-boole.htm ou http://www.matematica.br/historia/boole.html

[19] Mais informações sobre o filme podem ser obtidas em: http://www.vivendonamatrix.com/filme.html

Figura 2 - Exemplo de programa escrito utilizando a linguagem C#.



Fonte: criada pelo pesquisador (2016), utilizando o *software Microsoft VisualStudio .NET*.

Há uma grande diferença entre as disciplinas de programação com relação a outras disciplinas experimentais (que envolvam a manipulação física de objetos). A maioria das linguagens de programação exige do usuário um profundo conhecimento de suas regras e sintaxe textual, as quais geralmente são rígidas e complexas. Em um levantamento realizado por Bosse e Gerosa (2015), o qual analisou as disciplinas de introdução à programação na USP, indicou que esse índice varia bastante, chegando a mais de 62%. Além disso, mais de 25% dos alunos que foram aprovados no período de 2010 a 2014 tiveram que cursar 2 ou mais vezes a disciplina de introdução à programação, sendo que esta é uma das que mais reprova entre as do primeiro semestre dos cursos de computação.

Buscando mudar essa realidade, tornando a aprendizagem de programação mais simples e interativa, alguns projetos educativos deram origem a uma nova geração de linguagens de programação, dentre as quais podemos citar o *Scratch*,

desenvolvido pelo *Massachusetts Institute of Technology*[20] (MIT), e o *Kodu*, desenvolvido pela *Microsoft Research*. O *Scratch* é muito utilizado com jovens e crianças pelo mundo todo e será detalhado a seguir.

2.3.1 Sobre o *Scratch*

A linguagem de programação LOGO, desenvolvida nos anos sessenta pelo educador matemático Seymour Papert e sua equipe do MIT, serviu de inspiração para o desenvolvimento da linguagem *Scratch* (POCRIFKA; SANTOS, 2009).

Na criação do projeto LOGO, Papert apresentou a teoria Construcionista, a qual teve origem em seu trabalho desenvolvido com Piaget na Universidade de Genebra. Piaget propôs o Construtivismo em seu trabalho sobre a gênese da inteligência humana. Esse trabalho possibilitou a Papert realizar as conexões de seus conhecimentos de matemático e pesquisador com a análise do desenvolvimento da inteligência das crianças. A partir de então começou a considerar a matemática um meio de compreender como as crianças aprendem e pensam (PRADO, 1999).

O Construcionismo seria uma extensão do Construtivismo, segundo concepção de Papert, pois a construção dos esquemas ou estruturas cognitivas seria muito mais próspera se apoiada em algo concreto, tangível, como um kit de montar da Lego, um programa de computador ou mesmo um castelo de areia na praia. Dessa maneira, o produto resultante do processo pode ser visto e exibido, externalizado e discutido, examinado e analisado. Para Papert, essa é a principal característica do Construcionismo, pois possibilita visualizar e examinar a ideia da construção mental do conhecimento (PAPERT, 2008).

Na visão de Papert, o construcionismo não é uma concepção puramente mentalista tendo em vista que agrupa o trabalho intelectual do aluno com sua externalização por intermédio de diversos recursos disponíveis. Nesse sentido, o computador e demais tecnologias associadas possuem um papel importante, pois possibilitam a expressão dos mais diversos tipos de elaboração mental. Uma característica marcante do construcionismo é o aprendizado integrado à construção

[20] O Instituto de Tecnologia de Massachusetts é uma universidade privada de pesquisa localizada em Cambridge, Massachusetts, Estados Unidos.

de algo para proporcionar a compreensão de seu funcionamento (PAPERT; HAREL, 1991).

Foi a partir de pesquisas e aperfeiçoamentos das linguagens e ambientes de programação para jovens, em especial o LOGO, que o *Lifelong Kindergarten Group* do *MIT Media Lab*, coordenado por Mitchel Resnick, desenvolveu a linguagem de programação *Scratch*. Essa linguagem possibilita a criação de histórias, animações, simulações, jogos e músicas, objetivando o desenvolvimento de competências básicas e essenciais à realidade do século XXI. O *Scratch* foi concebido para auxiliar as crianças e os jovens a pensarem de forma criativa, desenvolvendo e aprimorando o raciocínio sistematizado bem como o trabalho colaborativo.

O *Scratch* é uma linguagem de programação visual, ou seja, ao invés de usar comandos de textos, utiliza imagens. A linguagem de programação visual permite ao usuário criar programas ou animações a partir da manipulação de elementos gráficos denominados de "*sprites*" (que podem ser imagens do próprio programa ou de fontes externas) e para isso o *Scratch* utiliza uma programação orientada a eventos, os quais seguem um fluxo determinado pelo usuário (DIETRICH, 2015).

Para possibilitar a utilização da linguagem *Scratch,* os criadores dela também desenvolveram uma ferramenta[21] com o mesmo nome e que pode ser acessada gratuitamente através do site http://scratch.mit.edu. Essa ferramenta agrega blocos lógicos e itens de outras mídias, como imagens e sons, possibilitando o desenvolvimento da criatividade, a concretização da imaginação, o trabalho de maneira colaborativa, além da divulgação e compartilhamento online.

Ao se cadastrar no site, que dá acesso à ferramenta, o usuário passa a fazer parte de uma comunidade. Tudo o que é criado no *Scratch* fica inicialmente visível somente ao proprietário (quem criou). Entretanto, o usuário tem a opção de compartilhar suas criações nessa comunidade, onde todos podem visualizar, experimentar e ter acesso ao projeto[22].

---

[21] Na computação os termos *software*, programa e ambiente são utilizados como sinônimo de ferramenta.
[22] O termo projeto é utilizado para definir as criações feitas no *Scratch.* Envolve o que foi programado com os blocos de comandos do *Scratch*, além dos outros elementos gráficos (imagens, desenhos, fotos) e sonoros.

O nome *Scratch* tem origem no termo de programação chamado *scratching*, o qual significa reciclagem de código[23]. Essa reciclagem engloba combinação, compartilhamento e adaptação de código para novos cenários, sendo esta a chave do *Scratch*: utilizar projetos que já foram criados e disponibilizados, para criar outros projetos (DIETRICH, 2015).

O *Scratch* possibilita ao aluno a utilização do computador para a criação e produção de conhecimento, expressando suas ideias e pensamentos. Assim, ele passa de mero consumidor de conteúdos para a posição de criador, exercendo uma postura ativa na sociedade, com o uso da tecnologia como ferramenta.

A linguagem de programação *Scratch* foi desenvolvida para ser utilizada por crianças e jovens a partir dos 8 anos, sendo empregada também no Ensino Superior. Essa linguagem foi concebida a fim de possibilitar a criação de programas através de sequências de comandos simples. Existem diversos blocos de ações e comandos que devem ser encaixados de forma a produzirem o efeito desejado, formando assim o programa. O formato dos blocos restringe as possibilidades de uso, evitando com isso, os erros de sintaxe (comuns nas linguagens textuais).

Além das facilidades de programação oferecidas pelo *Scratch,* a possibilidade de criação de jogos torna-o muito atraente para uma geração de aprendizes que busca constantemente aplicações práticas em seus estudos. Segundo Souza e Dias, essas mídias digitais têm feito parte da *Net Generation* (ou Geração Z), grupo de pessoas nascidas a partir década de 1990, que se encontram na maior parte do tempo conectadas, e, muitas vezes, auxiliam seus professores na instalação e no uso de aparelhos eletrônicos (SOUZA; DIAS, 2013).

Ainda, segundo os autores, muitos esforços para melhorar a aprendizagem se apoiam em colaboração e troca de experiências entre os alunos. Neste sentido, os programas e jogos produzidos com o *Scratch* podem ser compartilhados com outros aprendizes através da internet, caracterizando-os como objetos de aprendizagem, os quais podem ser usados e reutilizados por outros estudantes.

[23] O termo código é utilizado como sinônimo de programa (sequência ordenada de comandos escritos numa determinada linguagem de programação).

2.3.2 Ensino de programação e pensamento computacional no currículo da Educação Básica

Existe uma defasagem entre o que é ensinado na escola e as demandas sociais, econômicas e culturais das sociedades industrializadas e desenvolvidas, nas quais o papel da tecnologia assume maior relevância, refletindo no perfil dos profissionais de diversas áreas. Essas demandas impostas pela evolução e expansão da tecnologia refletem no estímulo à introdução de programação nos currículos da Educação Básica. Mudança essa que pode ser percebida em diversos países, variando a escala e o grau de transformação.

No Brasil, ainda são poucas as iniciativas para disseminar o pensamento computacional na Educação Básica. Existem apenas algumas tentativas isoladas de trazer alguns conceitos da Computação, em especial o ensino de programação para alunos do Ensino Fundamental e Médio. Não há uma regulamentação visando implantar o ensino de fundamentos da Ciência da Computação no currículo da Educação Básica, como ocorre nos Estados Unidos e em outros países da Europa.

Israel implantou em suas escolas um currículo de Computação no ano de 1995. Currículo esse que começou a ser pensado e articulado em 1990 a partir da criação de um comitê, nomeado pelo Ministério da Educação daquele país (HAZZAN *et al.*, 2008).

No contexto da Europa, o Reino Unido foi o primeiro país a retirar a disciplina de TIC (Tecnologia da Informação e Comunicação) do currículo e a inserir uma nova disciplina, voltada a ampliação e aprofundamento dos conhecimentos e habilidades dos jovens no domínio das tecnologias no contexto da Ciência da Computação, em especial ao ensino de programação. Na sequência França, Espanha, Alemanha, Bélgica e Holanda adotaram iniciativas semelhantes. Recentemente, Portugal também anunciou uma iniciativa piloto para a introdução do ensino de programação já no primeiro ciclo (RAMOS; ESPADEIRO, 2015).

Ramos e Espadeiro (2015) também destacam que as abordagens se assemelham com relação à adoção de uma grande variedade de tecnologias e suportes lógicos, o que possibilita a criação e desenvolvimento de diversas abordagens e estratégias de desenvolvimento do pensamento computacional.

Destacam-se, nesse contexto, as linguagens de programação específicas para crianças e jovens, ampla gama de jogos educativos, trabalho com robótica etc.

Há dois anos, a Inglaterra também reformulou seu currículo, retirou a matéria baseada em TIC e inseriu a Ciência da Computação no seu quadro nacional de disciplinas, dando direito a todos os alunos de aprender Computação. Ocorre que essa matéria foi desenvolvida e implementada em um curto espaço de tempo, sem que houvesse uma introdução progressiva no ambiente existente (BERRY, 2016).

Segundo Berry, essa implementação trouxe alguns problemas, pois foi aplicado o novo currículo para todas as séries simultaneamente. Estudantes com idades entre 11 e 14 anos passaram a ter disciplinas cujos conteúdos exigiam que tivessem determinados conhecimentos prévios, constantes no currículo de alunos entre 5 e 11 anos. Como a maioria dos alunos com mais de 11 anos nunca havia estudado os conceitos relacionados à Computação, gerou grandes transtornos, obrigando os professores do ensino secundário (11 a 14 anos) a trabalhar arduamente para tentar estabelecer as bases necessárias e, consequentemente, não conseguindo atingir os objetivos esperados.

Passados dois anos da implantação, Berry destaca várias questões que tiveram de ser repensadas, como a oferta de cursos introdutórios em Computação para alunos do ensino secundário, além da formação dos professores. Grande parte nunca havia estudado ou trabalhado com os conceitos da Ciência da Computação e se depararam com a necessidade de desenvolver esses conceitos com seus estudantes. Várias entidades de ensino reuniram esforços para oferecer treinamentos e dar suporte para esses professores.

Os tropeços iniciais fizeram a Inglaterra avaliar o que funcionou e o que não deu certo, buscar soluções e continuar o trabalho de levar a Ciência da Computação para todos (BERRY, 2016).

Desde 2006, os Estados Unidos tem investido pesado na inserção do pensamento computacional em seu currículo escolar. Em 30 de janeiro de 2016, o presidente Barack Hussein Obama anunciou seu plano de possibilitar o acesso ao aprendizado da Ciência da Computação nas escolas de Educação Básica, para estudantes de todo o país, com investimentos na ordem de 4 bilhões de dólares para

estados e 100 milhões diretamente para os distritos. O foco principal do investimento foi a formação de professores com treinamentos em Ciência da Computação.

Diante dessas considerações, faz-se necessário repensar a situação das escolas brasileiras perante essa questão, que envolve a inserção da Ciência da Computação no currículo do Ensino Fundamental e Médio. Espera-se que os resultados desse estudo possam auxiliar nesse sentido, contribuindo para o avanço das iniciativas de implementar um currículo para a Educação Básica que prepare os estudantes para os desafios do século XXI.

2.4 O PENSAMENTO COMPUTACIONAL E A FORMAÇÃO DE CIDADÃOS PARA O SÉCULO XXI

Atualmente, o pensamento crítico é considerado uma das mais importantes habilidades para uma carreira de sucesso e um componente essencial para a vida na era da informação. Tanto o meio acadêmico, como o profissional, concordam com essa importância, o que tem gerado debates e suscitado mudanças na academia e nas empresas.

Nos Estados Unidos, uma parceria conhecida como P21 (*Partnership for 21st Century Learning*[24]), reúne governo, entidades empresariais e líderes da educação em torno do objetivo de preparar todos os alunos para ter sucesso em um mundo cada vez mais tecnológico e interconectado. A parceria P21 criou um *framework* (conjunto particular de ideias) para a aprendizagem no século XXI. De acordo com a P21, milhares de educadores e centenas de escolas nos Estados Unidos e no exterior estão utilizando esse *framework* como norteador para a aprendizagem (P21, 2015).

Dentre os temas abordados por esse *framework,* cabe destacar o que agrupa as habilidades de aprendizagem e inovação, por sua correlação com o objeto de pesquisa. Segundo a P21 (2015), as habilidades de aprendizagem e inovação serão responsáveis pela diferenciação entre os estudantes que estão preparados para lidar com a crescente complexidade do mundo atual. Os principais assuntos relacionados a esse tema são:

---

[24] Parceria para o aprendizado no século XXI (tradução nossa). Mais informações em: http://www.p21.org/

- Criatividade e inovação;
- Pensamento crítico e resolução de problemas;
- Comunicação;
- Colaboração.

Todos esses assuntos se relacionam com a Ciência da Computação, fazendo parte do cotidiano dos profissionais dessa área. Pelo escopo deste trabalho optamos por aprofundar o pensamento crítico e a resolução de problemas, porém sem desconsiderar os demais.

Com base no documento elaborado pela P21 (2015), as principais habilidades relacionadas ao pensamento crítico são:

- Usar de forma efetiva e apropriada os diversos tipos de raciocínio (dedutivo[25], indutivo[26], e abdutivo[27]), de acordo com cada situação.
- Usar pensamento sistêmico, ou seja, analisar, interpretar e sintetizar como as partes de um conjunto interagem entre si para gerar os resultados globais em sistemas complexos.
- Fazer análise crítica e tomada de decisões, através de uma análise fundamentada na avaliação de evidências, argumentos, afirmações e crenças. Além disso, analisar e avaliar os principais pontos de vista alternativos, sintetizando e fazendo conexões entre informações e argumentos. Complementa essa habilidade, a interpretação de

[25] Raciocínio dedutivo ou método dedutivo é um tipo de raciocínio lógico que faz uso da dedução para obter uma conclusão a respeito de determinada premissa. Ele nos permite verificar se algo é ou não verdadeiro, sendo que, o modo como se chega à prova de validade de um argumento dedutivo é verificando se a conclusão a que se chega deriva das premissas apresentadas (COCCHIERI, 2015). Ex: Todos os bois são mamíferos; todos os mamíferos tem coração; portanto todos os bois têm coração.

[26] Raciocínio indutivo ou método indutivo é um tipo de raciocínio ou argumento que parte de uma premissa particular para atingir uma conclusão universal. É o processo pelo qual, dadas diversas particularidades, chegamos a uma generalização (COCCHIERI, 2015). Ex: Este cisne é branco; o cisne que vi ontem era branco; portanto todos os cisnes são brancos.

[27] Raciocínio abdutivo é um tipo de raciocínio em que uma hipótese é adotada como uma possível explicação para um fato observado, de acordo com leis conhecidas. Ele se efetiva nas seguintes etapas: percepção de anomalia; surpresa e dúvida; abandono do hábito anterior; geração e seleção de hipóteses que poderiam solucionar o problema (COCCHIERI, 2015). Ex: Quando chove a grama fica molhada; a grama está molhada; portanto pode ter chovido.

informações com base na melhor análise para tirar conclusões, refletindo criticamente sobre as experiências e processos de aprendizagem.

Para a resolução de problemas, a P21 (2015) elenca as seguintes habilidades:

- Resolver diferentes tipos de problemas não familiares, tanto de formas convencionais como propondo soluções inovadoras.
- Identificar e fazer perguntas significativas com o objetivo de esclarecer vários pontos de vista e com isso levar às melhores soluções.

O pensamento crítico tem sido identificado como uma habilidade fundamental para promover a inovação. A pesquisa realizada pela P21 (2015) mostra que esse pensamento e a criatividade estão correlacionados, sendo que a formação do pensamento crítico está se tornando prática comum no local de trabalho, a fim de desenvolver nos funcionários a competência de inovar.

O desenvolvimento de tais habilidades na educação, junto ao emprego das tecnologias de informação e comunicação, contribui cada vez mais para deslocar o foco do ensino para a aprendizagem, influenciando diretamente a relação entre alunos e professores, muitas vezes gerando dificuldades de aprendizagem e conflitos entre as expectativas dos alunos e a prática do professor.

De acordo com a Lei de Diretrizes e Bases da Educação Nacional – LDB, a educação na sociedade tecnológica tem por base a formação de sujeitos intelectualmente autônomos, críticos pensantes, capazes de progredir a níveis mais complexos de estudos. Sujeitos com habilidades e competências que possibilitem seu ingresso no mundo do trabalho, seu aprimoramento profissional e contribuição para o desenvolvimento da sociedade na qual estão integrados.

As competências e habilidades vão sendo desenvolvidas e trabalhadas ao longo de toda a Educação Básica, em especial em sua etapa final que é o Ensino Médio. A formação do educando como produtor de conhecimento, capaz de participar do mundo do trabalho e nele evoluir, tanto em termos de conhecimentos como na parte humana, é o princípio norteador do Ensino Médio. A LDB estabelece que o Ensino Médio "deverá vincular-se ao mundo do trabalho e à prática social" (Art.1º § 2º da Lei nº 9.394/96).

No artigo 35 da LDB, que trata das finalidades do Ensino Médio, fica clara a preocupação com a continuidade do aprendizado, com o desenvolvimento de competências e habilidades que possibilitem a compreensão dos fundamentos científicos e tecnológicos, bem como a autonomia intelectual e o desenvolvimento do pensamento crítico. Essa preocupação pode ser vista em especial no inciso II desse artigo, que diz:

> II - a preparação básica para o trabalho e a cidadania do educando, para continuar aprendendo, de modo a ser capaz de se adaptar com flexibilidade a novas condições de ocupação ou aperfeiçoamento posteriores; (Art.35, Inciso II, da Lei nº 9.394/96).

Com relação às diretrizes curriculares, o artigo 36 da LDB define a integração da tecnologia com a ciência e as demais áreas, como forma de acesso ao conhecimento e exercício da cidadania. Os Parâmetros Curriculares Nacionais para o Ensino Médio (PCNEM) vem reforçar essa necessidade de preparação científica e tecnológica, propondo uma formação geral, e não específica, aliada ao "desenvolvimento de capacidades de pesquisar, buscar informações, analisá-las e selecioná-las; a capacidade de aprender, criar, formular, ao invés do simples exercício de memorização." (BRASIL, 2000, p. 5).

Nas últimas décadas, a sociedade contemporânea vivenciou uma verdadeira revolução nas áreas da tecnologia e da comunicação. Isso afetou profundamente a forma de viver e de se relacionar, trouxe novas ferramentas e recursos capazes de romper barreiras, tornando acessível e instantânea a comunicação com qualquer parte do mundo. Com isso, surgiram novas demandas e necessidade de adaptação nos mais diversos campos e setores, dentre eles o da educação.

O papel da educação, diante dessas novas demandas da sociedade tecnológica, é abordada nos Parâmetros Curriculares Nacionais para o Ensino Médio. Neles é enfatizada a necessidade de desenvolver competências cognitivas e culturais para o desenvolvimento humano global, assegurando o pleno exercício da cidadania e a possibilidade de desempenhar as mais diversas atividades profissionais.

Dentre as competências que necessitam ser desenvolvidas, segundo o PCNEM, estão:

> capacidade de abstração, do desenvolvimento do pensamento sistêmico, ao contrário da compreensão parcial e fragmentada dos fenômenos, da criatividade, da curiosidade, da capacidade de pensar múltiplas alternativas para a solução de um problema, ou seja, do desenvolvimento do pensamento

> divergente, da capacidade de trabalhar em equipe, da disposição para procurar e aceitar críticas, da disposição para o risco, do desenvolvimento do pensamento crítico, do saber comunicar-se, da capacidade de buscar conhecimento. (BRASIL, 2000, p. 11-12).

Essas competências condizem com a proposta de desenvolver o pensamento computacional, buscando a formação de sujeitos capazes de pensar de maneira crítica e sistêmica, articulando conhecimentos na abstração e resolução de problemas. O PCNEM sugere que o desenvolvimento de tais competências faça parte do currículo escolar assim como já ocorre em diversos países, conforme mencionado anteriormente no item 4.2.

Recentemente, em dezembro de 2017, o Conselho Nacional de Educação (CNE) aprovou a nova Base Nacional Comum Curricular (BNCC) para a Educação Infantil e Ensino Fundamental. Nela é ressaltada a importância do desenvolvimento do pensamento computacional por parte dos alunos a fim de torná-los "capazes de traduzir uma situação dada em outras linguagens, como transformar situações-problema, apresentadas em língua materna, em fórmulas, tabelas e gráficos e vice-versa" (BRASIL, 2017, p. 269).

O documento também salienta a importância do aprendizado de algoritmos e seus fluxogramas nas aulas de Matemática, pois o algoritmo compreende "a decomposição de um procedimento complexo em suas partes mais simples, relacionando-as e ordenando-as, e pode ser representado graficamente por um fluxograma" (*ibidem*, 2017, p.269). O pensamento computacional está relacionado à linguagem algébrica pois tem vários pontos em comum, principalmente com relação ao conceito de variável. Além disso, a identificação de padrões para estabelecer generalizações, propriedades e algoritmos é uma habilidade relativa à álgebra e que também faz parte do pensamento computacional.

2.5 A TEORIA SOCIOINTERACIONISTA

A fim de constituir o quadro teórico dessa pesquisa, tomaremos alguns conceitos da teoria sociointeracionista de Vigotski para se articularem com os conceitos já apresentados.

Lev Semyonovitch Vigotski[28] nasceu em 1896, na cidade de Orsha na Bielo–Rússia. Teve uma formação sólida desde criança, com um tutor particular até entrar no curso secundário, dedicando-se a muitas leituras desde cedo. Seu percurso acadêmico foi marcado pela multidisciplinaridade, pois enquanto estudava na Universidade de Moscou Direito e Literatura, também frequentava a Universidade Popular de Shanyavskü nos cursos de História e Filosofia. Iniciou sua carreira como psicólogo após a Revolução Russa de 1917, quando se graduou na Universidade de Moscou. Entre os anos de 1925 e 1934, trabalhou com um grupo de jovens cientistas e simultaneamente cursou medicina. Morreu precocemente de tuberculose em 1934 (VIGOTSKI, 2007).

Vigotski criou a teoria sócio-histórica, também conhecida como sociocultural ou sociointeracionismo, segundo a qual a aprendizagem e o desenvolvimento estão relacionados desde o início da vida. O processo de aprendizagem ocorre através das relações sociais que a criança estabelece externamente, em um ambiente de interação e cooperação, e que despertam processos internos de desenvolvimento.

Para Vigotski, o desenvolvimento cognitivo de um indivíduo não pode ser entendido sem que haja referência ao contexto no qual esse sujeito está inserido, considerando o meio social, histórico e cultural. Portanto, o desenvolvimento cognitivo é a conversão de relações sociais em funções mentais ou processos sociais, que dão origem às funções mentais superiores, das quais fazem parte o pensamento, a linguagem e o comportamento volitivo. "É por meio da socialização que se dá o desenvolvimento dos processos mentais superiores" (MOREIRA, 2011, p.108).

Vigotski classificou as funções psicológicas dos indivíduos em elementares e superiores. A partir das estruturas orgânicas elementares do processo de maturação é que se formam novas e complexas funções mentais que inclui também o elemento sociocultural. As funções psicológicas superiores são de origem social, porém só são possíveis porque existe uma base biológica, cerebral, que permite tais atividades.

As funções psicológicas superiores não ocorrem de forma linear e retilínea, mas a partir de uma série de transformações qualitativas e dialéticas, no qual a linguagem (primeiro a oral e depois a escrita) constitui o elemento mais significativo

[28] Optamos por empregar a grafia Vigotski (1987; 2007), conforme referencial teórico adotado.

do desenvolvimento cognitivo, representando um salto de qualidade nas funções psicológicas superiores.

O desenvolvimento dos processos psicológicos superiores ocorre através da utilização de instrumentos adquiridos culturalmente, através das interações sociais. Segundo a teoria vigotskiana, para que haja o desenvolvimento das funções mentais superiores são necessárias que ocorram interações entre o sujeito e a sociedade, a cultura e a sua história de vida, além das oportunidades e situações de aprendizagem.

No Quadro 5 estão elencados os elementos que compõem as funções psicológicas elementares e as superiores.

Quadro 5 - Funções Psicológicas Inferiores e Superiores

FUNÇÕES PSICOLÓGICAS	
ELEMENTARES **Psicofisiológicas (naturais)**	**SUPERIORES** **Neuropsicológicas (culturais)**
Sensação, atenção, integração modal, percepção, memória, audição, visão, somestesia, olfato, emoção, linguagem animal (emotiva), interocepção, propriocepção e exterocepção.	Consciência, imaginação (percepção global), cognição, memória mediada, atenção voluntária, atividade mediada (uso de instrumentos), linguagem social (racional), pensamento, leitura e escrita.

Fonte: adaptado de Moraes (2010).

Vigotski (2007, p. 56) denomina "de *internalização* a reconstrução interna de uma operação externa". No processo de aprendizagem, essa operação externa corresponde ao conjunto de interações e inter-relações do estudante com outros sujeitos e com os recursos pedagógicos, tendo o professor como mediador. O processo de internalização corresponde à formação de conceitos pelo sujeito, desenvolvendo com isso novas aprendizagens. Cabe ao professor criar situações e intervenções que levem a internalização de novos conceitos, produzindo ressignificações e possibilitando com isso o desenvolvimento de novas aprendizagens.

Ao dizer que "o desenvolvimento é visto como um processo de maturação sujeito às leis naturais; e o aprendizado, como a utilização das oportunidades criadas pelo desenvolvimento" (VIGOTSKI, 1987, p. 80-81), compreende-se que o

desenvolvimento varia conforme cada estudante, pois cada qual tem seu próprio ritmo de aprendizagem e desenvolvimento. Por isso, alguns precisam de um tempo maior e de mais explanações, incentivos, momentos de interação, de recursos pedagógicos variados para auxiliá-los nessa construção. Outros, por sua vez, conseguem alcançar os objetivos em um menor espaço de tempo e com mais facilidade.

Experiências feitas por Vigotski (2007) concluíram que a fala é tão importante quanto à ação para atingir um objetivo, pois ambas fazem parte de uma mesma função psicológica complexa, dirigida para a solução do problema em questão. Em situações obscuras, onde a solução não é direta, a fala adquire uma importância maior na concretização da ação.

Nesse sentido, pesquisadores japoneses liderados por Ayanori Sat comprovaram que a técnica de apontar e falar ("*shisa kanko*") ajuda a melhorar o desempenho em tarefas que requerem atenção. Essa técnica surgiu nas linhas férreas japonesas no início do século XX e tinha por objetivo verificar se tudo estava em ordem antes da chegada do trem na estação, averiguar o embarque e desembarque e, por fim, conferir se tudo estava em conformidade para a partida do trem (GOMES, 2016).

Segundo Ayanori Sat, quando se aponta na direção da tarefa que está sendo executada, volta-se o foco para essa atividade, aumentando a concentração. Ao falar o que se está executando, auxilia na memorização e facilita a detecção de problemas. Combinando o apontar com o falar, reduzem a ocorrência de erros em quase 70% (GOMES, 2016).

As conclusões dos pesquisadores contribuem com a ideia de Vigotski que:

> O momento de maior significado no curso do desenvolvimento intelectual, que dá origem às formas puramente humanas de inteligência prática e abstrata, acontece quando a fala e a atividade prática, então duas linhas completamente independentes de desenvolvimento, convergem. (VIGOTSKI, 2007, p. 11-12)

Na visão de Vigotski, existem três níveis de desenvolvimento que podem ser considerados nos processos educativos. O primeiro, denominado de desenvolvimento real, é aquele no qual os processos de desenvolvimento já se encontram consolidados, ou seja, é o que a pessoa já internalizou e consegue realizar sem a necessidade de influência externa. O segundo, denominado de desenvolvimento potencial, remete para tudo aquilo que o indivíduo consegue realizar somente com o auxílio de alguém mais capaz ou experiente. Já o terceiro, denominado de zona de

desenvolvimento proximal (ZPD), caracteriza-se pelo espaço ou distância entre os dois estágios anteriores e é potencializado pela interação social.

A aprendizagem é uma experiência social, mediada pela interação entre a linguagem e a ação. Para que ela ocorra, a interação social deve acontecer dentro da ZDP, que é definida por Vigotski como:

> [...] a distância entre o nível de desenvolvimento real, que se costuma determinar através da solução independente de problemas, e o nível de desenvolvimento potencial, determinado através da solução de problemas sob a orientação de um adulto ou em colaboração com companheiros mais capazes (VIGOTSKI, 2007, p. 97).

Saber identificar o que o sujeito consegue fazer sozinho e o que ele está perto de conseguir fazer sozinho e trabalhar o percurso de cada aluno entre ambas são as duas principais habilidades que um professor precisa ter. A identificação da ZDP do educando possibilita ao professor a intervenção mais adequada na construção de novas e complexas estruturas, incentivando o aluno a estabelecer conexões entre o novo conhecimento em construção, com os conceitos que já tem domínio. Quando o sujeito alcança o conhecimento potencial, esse passa a ser o conhecimento real e a ZDP redefine-se a partir do que seria o novo potencial.

O nível de desenvolvimento real é dinâmico, variando de acordo com os movimentos do processo dialético de aprendizagem. O desenvolvimento potencial é determinado pelas habilidades que o indivíduo já construiu, porém encontra-se em processo. Ou seja, a aprendizagem, além de produzir o desenvolvimento real, gerou também habilidades que se encontram em um nível menos elaborado que o já consolidado. Nessa concepção, as interações têm um papel determinante, atuando na construção do desenvolvimento potencial do sujeito. O processo dialético de aprendizagem tem como fundamento o estudo de algo historicamente, ou seja, estudá-lo no processo de mudança (VIGOTSKI, 2007).

A conversão de relações sociais em funções mentais superiores não é direta, é mediada. Essa mediação corresponde a um estímulo incorporado ao impulso direto de modo a facilitar a complementação da operação. Segundo Vigotski, a mediação pode ser semiótica ou através da interação com o outro, a interação social, na qual as palavras são empregadas como meio de comunicação ou de interação. Essa mediação pela interação com o outro é denominada por Vigotski de sócio-interacionismo, onde a ação se dá numa interação sócio-histórica ou histórico-cultural.

A aprendizagem mediada atribui um papel privilegiado ao professor, pois ele torna-se o grande orquestrador de todo o processo. O professor – sujeito mais experiente – identifica a ZDP de cada sujeito, planeja e atua com intencionalidade educativa a fim de possibilitar a internalização dos conhecimentos por parte dos sujeitos envolvidos.

Ao atuar como mediador na ZDP, o professor pode fazer uso de instrumentos e signos, além da interação social. Tal utilização foi conceituada por Vigotski como mediação simbólica, sendo que os signos também podem ser associados à mediação semiótica. A interação social, por sua vez, é conceituada como mediação humana, da pessoa. De forma simplificada, podemos dizer que o instrumento é algum recurso que pode ser empregado para executar determinada ação. Já, o signo é algo que significa alguma coisa.

"O uso de signos conduz os seres humanos a uma estrutura específica de comportamento que se destaca do desenvolvimento biológico e cria novas formas de processo psicológicos enraizados na cultura" (VIGOTSKI, 2007, p. 30). Os signos podem ser classificados em três categorias: indicadores, icônicos e simbólicos.

Indicadores: são os signos que se relacionam com seu objeto, cuja relação existe naturalmente no mundo real. Os indicadores os representam em virtude de serem diretamente afetados por eles. Resultam de uma relação de associação ou referência, evidenciada pelos vestígios, pelos indícios. Por exemplo: um campo molhado é um indicador de que choveu, pois existe uma conexão natural entre a chuva e o campo molhado. No *Scratch*, a bandeirinha verde ativa indica que o programa está em execução, ou seja, o botão da bandeira já foi clicado, dando início ao programa.

Icônicos: resultam de uma relação de semelhança/analogia entre o signo e o objeto que ele substitui, podendo ser confundidos com ele de alguma forma. Assim, dizemos que um ícone apresenta alguma semelhança com o objeto representado. Exemplo: os ícones presentes nos computadores e smartphones.

Simbólicos: resultam de uma relação estabelecida por convenção. A relação entre o signo e o objeto que ele representa é arbitrária, legitimada por regras. Um símbolo não se encontra diretamente afetado pelo objeto, como no caso dos Indicadores, mas se conecta a este somente por força de uma convenção ou lei que o faz arbitrariamente. Sendo assim, os símbolos são os tipos de signos mais

poderosos, pois sua conexão entre signo e objeto não é natural, mas sim artificialmente criada, somente para os propósitos de representação. Exemplos de símbolos: qualquer palavra da nossa língua, as logomarcas de empresas.

"Palavras, por exemplo, são signos linguísticos, números são signos matemáticos; a linguagem, falada e escrita, e a matemática são sistemas de signos" (MOREIRA, 2011, p. 109). Moreira também afirma que signos e instrumentos são construções sócio-históricas e culturais e é através da apropriação e internalização dessas construções que as pessoas se desenvolvem congnitivamente. A interação social exerce papel fundamental na reconstrução de signos e instrumentos, em especial na aquisição de significados dos signos simbólicos.

Moreira usa as palavras para exemplificar a importância do papel da sociedade nessa ação, pois para ele as palavras significam aquilo que um grupo social aceita, compartilha. Na programação de computadores, por exemplo, código é utilizado para definir as instruções escritas em uma determinada linguagem de programação, enquanto que para a maioria das pessoas, que não é da área da informática, pode significar senha, chave, segredo, lei, norma, regulamento, entre outros.

A mediação simbólica através de instrumentos surgiu da necessidade de intervenção do homem no mundo. Para executar determinadas ações, o homem, além de buscar objetos existentes, também criou e desenvolveu ferramentas que possibilitassem a execução dessas ações ou que servissem como facilitadoras. A maioria dessas criações humanas foi fruto de uma necessidade, influenciadas pelo ambiente sociocultural em que o indivíduo estava inserido.

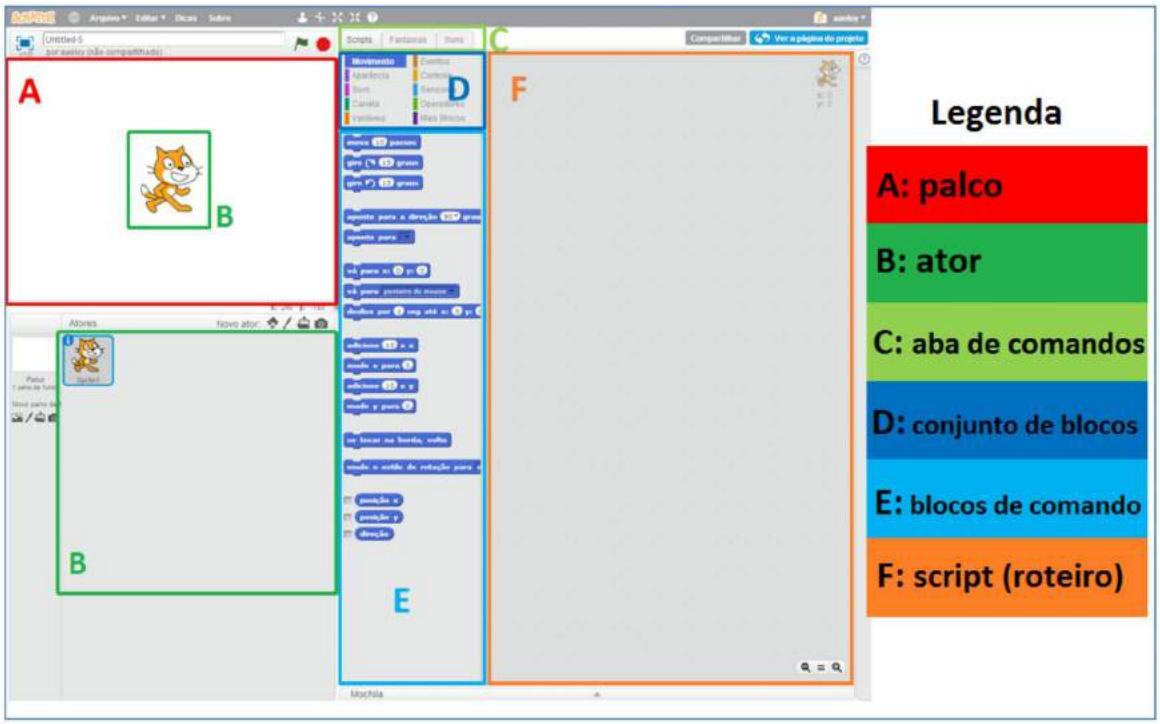
Voltando ao conceito de mediação, ela pode ser entendida como a relação estabelecida entre os sujeitos envolvidos, podendo ser: professor-aluno, aluno-aluno e também entre sujeito e instrumento ou signo. Segundo Vigotski (2007), a semelhança entre signos e instrumentos está na função mediadora, sendo que o instrumento tem a função de "servir como um condutor da influência humana sobre o objeto da atividade; ele é orientado *externamente*; deve necessariamente levar a mudanças nos objetos" (*ibidem*, p. 40).

Entende-se que o *Scratch* tem potencial para ser considerado um instrumento mediador no processo pedagógico entre o sujeito e o objeto de conhecimento (no

caso, a aprendizagem de programação). A ferramenta de programação *Scratch* é dinâmica e interativa, possibilitando que o usuário realize, a qualquer momento, testes dos comandos que utilizou a fim de validar suas hipóteses, encontrar erros de lógica e explorar os recursos de programação oferecidos. Além disso, é possível parar a execução do programa a qualquer momento, fazer modificações e retomar sua execução.

Ao utilizar o *Scratch* para construir seus programas, o usuário não necessita se preocupar ou perder tempo corrigindo erros de sintaxe. Com isso, além de simplificar a programação, possibilita dedicar mais tempo à estruturação lógica e criativa. A Figura 3 apresenta os principais elementos que fazem parte desse ambiente de programação.

Figura 3 - Principais elementos do *Scratch*.



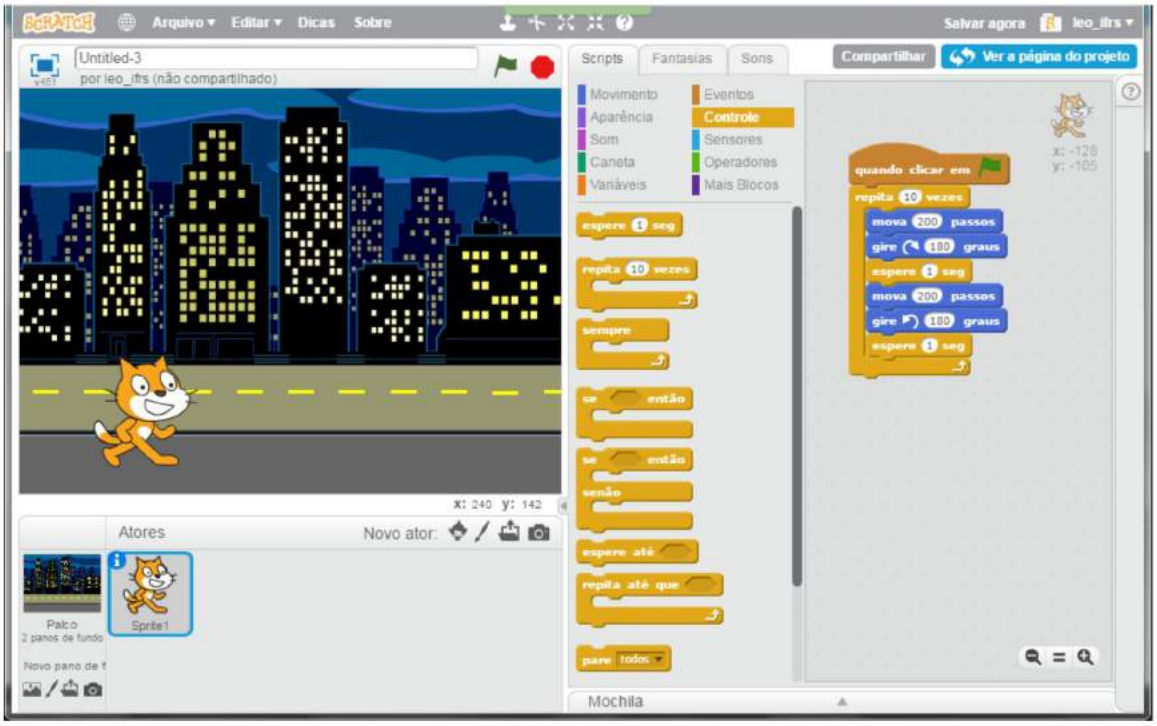
Fonte: adaptado (2016) de http://programae.org.br/

A região **F,** exibida na Figura 3, é o local onde o usuário construirá a lógica de seu programa, escolhendo na região **D** qual conjunto de blocos deseja utilizar. Ao realizar a escolha, na região **E** é exibida uma relação de comandos disponíveis. O

usuário escolhe o comando desejado na região **E**, o arrasta até a região **F** e o encaixa no local desejado (desde que as regras da linguagem assim o permita).

A Figura 4 ilustra a criação de um programa no qual alguns comandos foram inseridos para o ator 1 (gato). Esses comandos são executados quando o usuário clicar na bandeira cinza. Ao iniciar a execução dos comandos, o botão passa a ser uma bandeira verde, indicando que o programa está rodando (está em execução).

Figura 4 – Exemplo de programa escrito no *Scratch*.



Fonte: interface do *software Scratch*. Disponível em: https://scratch.mit.edu/

O *Scratch* apresenta um conjunto de instruções e comandos acessível, permitindo que os usuários programem seus próprios jogos, montem suas histórias, criem suas animações utilizando recursos multimídia (textos, imagens, sons, filmes) e compartilhem com outros usuários. Essa simplicidade e facilidade de utilização gera algumas restrições e limitações, especialmente para o desenvolvimento de aplicações mais complexas e específicas. No entanto, para os objetivos propostos e para o público alvo ela é adequada. A possibilidade de interação com outros usuários, através do compartilhamento dos programas produzidos, caracteriza outra forma de mediação e amplia as possibilidades de utilização desse recurso pedagógico.

3 DELINEAMENTO METODOLÓGICO

3.1 ABORDAGEM TEÓRICO-METODOLÓGICA

O método de pesquisa, conforme Wazlawick, é a "sequência de passos necessários para demonstrar que o objetivo proposto foi atingido, ou seja, se os passos definidos no método forem executados, os resultados obtidos deverão ser convincentes" (WAZLAWICK, 2009, p. 41).

Optamos por desenvolver um estudo de caso para gerar dados a fim de responder a pergunta problema de nosso projeto de pesquisa: como o *Scratch* pode mediar a aprendizagem de programação no Ensino Médio com vistas ao desenvolvimento do pensamento computacional a partir da teoria vigotskiana?

Segundo Yin, essa estratégia de pesquisa pode ser empregada em diversas situações, como "quando o foco se encontra em fenômenos contemporâneos inseridos em algum contexto da vida real" (YIN, 2005, p. 19). O estudo de caso contribui ainda com "o conhecimento que temos dos fenômenos individuais, organizacionais, sociais, políticos e de grupo, além de outros fenômenos relacionados" (*ibidem*, p. 20).

Além do mais, o estudo de caso possibilita uma investigação buscando compreender os fenômenos estudados de maneira holística, ou seja, na sua totalidade, levando em conta as partes e suas inter-relações, além de permitir uma investigação significativa dos acontecimentos do mundo real. Yin apresenta a seguinte definição para esse método de pesquisa:

> Um estudo de caso é uma investigação empírica que investiga um fenômeno contemporâneo em profundidade e em seu contexto de vida real, especialmente quando os limites entre o fenômeno e o contexto não estão claramente definidos (YIN, 2005, p. 32).

Segundo Oliveira (2016), esse método consiste em um estudo empírico que visa testar ou determinar uma teoria, sendo de grande utilidade quando o fenômeno é amplo, complexo e não possibilita ser estudado fora do contexto onde ocorre naturalmente.

Segundo Wazlawick (2009), não é o princípio da autoridade, mas sim a pesquisa de base empírica, cujos dados são obtidos através da experiência e tornam-

se o *corpus* do objeto de estudo, que fornece fundamentos para as pesquisas em Ciência da Computação. Na maioria das vezes, a opinião de uma ou outra celebridade da área tem pouca importância, sendo muito mais valorizadas as conclusões objetivas resultantes de estudos empíricos.

3.2 PROCEDIMENTO DE GERAÇÃO DE DADOS

Para criar o caso investigado foi desenvolvida uma oficina de introdução à programação de computadores para alunos de primeiro ano do Ensino Médio Técnico de uma instituição de ensino público da Serra Gaúcha. Essa instituição autorizou a pesquisa, conforme respectivo termo encaminhado ao comitê de ética. Nenhum dos cursos de Ensino Médio Técnico dessa instituição possui disciplina relacionada à programação de computadores. Esse foi um critério de escolha dos participantes da oficina, pois era relevante que eles não tivessem conhecimento prévio de programação. Foram ofertadas dez vagas para a oficina a alunos dessa instituição.

Para viabilizar a realização da oficina foi necessário elaborar um projeto de ensino, o qual foi cadastrado no SIGPROJ (Sistema de Informação e Gestão de Projetos do MEC) e posteriormente avaliado por uma comissão, a qual autorizou sua execução. Além disso, o projeto foi cadastrado no Comitê de Ética em Pesquisa – Plataforma Brasil, sendo aprovado em 12 de maio de 2017.

A proposta da oficina foi apresentar os conceitos básicos de programação através da resolução de situações problemas e criação de jogos digitais, utilizando para isso o *Scratch*[29], descrito na seção 2.3.1. Assim, buscou-se observar a capacidade de exploração, interação e reflexão dos estudantes, bem como avaliar as potencialidades e deficiências do *Scratch* como instrumento mediador.

O *Scratch* foi escolhido para essa oficina por ser uma ferramenta de programação educativa gratuita e possuir uma versão que não necessita ser instalada no computador, bastando acessá-la no navegador da internet. A possibilidade de aprender a programar em um ambiente multimídia, onde a programação é feita arrastando-se blocos de comandos, os quais devem ser encaixados uns aos outros, proporciona uma aprendizagem lúdica da programação de computadores. O principal

[29] Site oficial: https://scratch.mit.edu/. Site em português: http://www.scratchbrasil.net.br

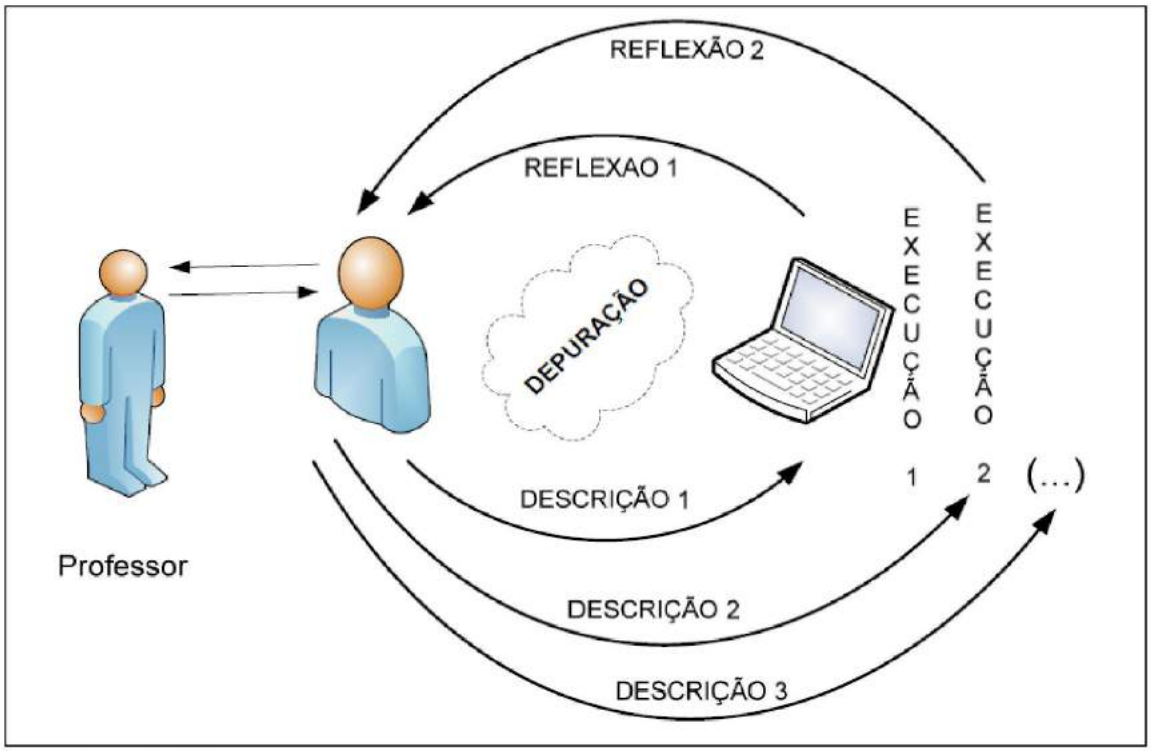
objetivo do *Scratch* é introduzir de forma simplificada a programação para quem não tem experiência no assunto.

Na visão de Lima (2009), a relação entre o computador e o estudante através de um *software* com a mediação do professor, proposta pelo construcionismo, pode levar à construção de conhecimentos por intermédio da atividade de programação de computadores e se assemelha a uma espiral. Nesse sentido, Valente afirma que:

> As ações podem ser cíclicas e repetitivas, mas a cada realização de um ciclo, as construções são sempre crescentes. Mesmo errando e não atingindo um resultado de sucesso, o aprendiz está obtendo informações que são úteis na construção do conhecimento. Na verdade, terminando um ciclo, o pensamento nunca é exatamente igual ao que se encontrava no início de sua realização. Assim a ideia mais adequada para explicar o processo mental dessa aprendizagem é a de uma espiral (VALENTE, 2005, p.222).

Na Figura 5, Lima ilustrou a relação entre o estudante e as etapas de desenvolvimento de um programa computacional, destacando a função de mediação do professor.

Figura 5 - Relação do estudante com os ciclos de programação e a figura do mediador



Fonte: Lima (2009).

A análise da Figura 5 descreve a proposta do trabalho de pesquisa, permitindo identificar as etapas de desenvolvimento de um programa computacional. Essas etapas envolvem a codificação do problema proposto para uma linguagem de programação; execução do código elaborado; identificação de erros ou necessidades de alterações com a análise e reflexão sobre os resultados gerados; e, retomada do processo fazendo as devidas adequações do código para corrigir as falhas detectadas ou melhorar a solução. Em todas as etapas, o aluno tem o professor como um mediador para indicar caminhos, problematizar e acompanhar seu percurso.

Na organização e implementação da oficina buscamos propiciar esse cenário aos participantes, dando-lhes liberdade para interagir com o *Scratch*, com os colegas, com materiais de apoio e com o professor. Nosso objetivo foi verificar como os estudantes reagiam às tarefas propostas, como se organizavam para resolvê-las, quais estratégias utilizavam para descobrir as funcionalidades do ambiente e como lidavam com os erros e com as dificuldades.

No estudo de caso, o pesquisador assumiu a condição de sujeito implicado, desempenhando papel ativo e consciente junto aos alunos participantes do estudo. Tanto no piloto como na oficina, o pesquisador esteve no papel de professor, buscando proporcionar liberdade e autonomia aos participantes.

3.2.1 Estudo piloto

Durante a preparação da oficina surgiram algumas dúvidas e questionamentos que nos levaram a escolher pela realização de um estudo piloto com alunos que não fariam parte do estudo de caso. Objetivamos com isso avaliar e adequar a estrutura, o conteúdo e a metodologia da oficina. Inicialmente optamos por realizar o estudo piloto com alunos de outra instituição de ensino na qual os alunos, a partir do segundo ano, têm a opção de escolher qual curso técnico desejam cursar, sendo um deles o de informática.

Por questões de prazo, horário das aulas e demais atividades, decidimos montar esse piloto nas férias, um pouco antes do início do ano letivo. Tivemos o apoio da direção da instituição, a qual forneceu o contato dos alunos que se encaixavam no perfil desse piloto. Entramos em contato com esses estudantes, porém não obtivemos

o número mínimo de participantes para montá-lo. A grande maioria dos alunos não aceitou participar por estar em viagem de férias ou já ter outros compromissos. Em alguns casos os estudantes não mostraram interesse pois não pretendiam seguir na área de informática.

Após o início do ano letivo ficou mais complicado conciliar os horários dos estudantes dessa instituição com os horários do pesquisador. Como alternativa optamos por desenvolver o piloto na mesma instituição de ensino na qual seria realizada a oficina, porém com outros alunos. Inicialmente foi preciso aguardar um período de adaptação deles, pois os estudantes de primeiro ano de Ensino Médio, na sua grande maioria, não se conhecem e tão pouco a instituição de ensino. Os alunos chegam de diferentes escolas locais e de municípios vizinhos, tanto públicas como privadas.

Na organização do piloto, deparamo-nos com mais um obstáculo, que foi a escassez de horários disponíveis nos laboratórios de informática. Decidimos então fazer o piloto com uma turma inteira, tendo em vista que o pesquisador leciona para seis turmas de primeiro ano de Ensino Médio dessa instituição. Escolhemos a turma com menor número de alunos para montar o piloto. O pesquisador expos para a turma a proposta de realizar esse piloto, com seus objetivos e finalidades, bem como se comprometeu a organizar os demais conteúdos programáticos da disciplina para não prejudicar os estudantes. Após essa conversa, os estudantes foram consultados sobre seu interesse e consentimento em participar desse estudo piloto. A turma de 31 alunos foi unânime em aceitar a participação, mostrando interesse pelo tema proposto.

Antes de iniciar o piloto, os estudantes participantes responderam a um questionário[30] envolvendo perguntas de identificação, questões referentes aos acessos que eles dispõem à tecnologia de informação e comunicação e como utilizam esses recursos. Com isso buscamos obter um perfil desses estudantes, em especial seus interesses relacionados a jogos digitais e programação de computadores. Ao responderem esse questionário foi possível avaliar o piloto, buscando identificar necessidades de ajustes, pois ele seria aplicado aos participantes da oficina.

[30] Link para questionário on-line: https://goo.gl/forms/BY9blNYIN17ICMG83

O piloto teve início no dia 17 de abril de 2017, no turno da manhã. Cada encontro teve duração de 1 hora e 40 minutos (dois períodos) e foram realizados no laboratório de informática (local habitual das aulas do pesquisador com a referida turma). No primeiro encontro, o pesquisador explicou a metodologia das aulas e apresentou o *Scratch* aos estudantes, mostrando dois pequenos vídeos para que eles tivessem uma visão geral do programa, tanto da versão instalada nos computadores como da versão disponível na internet.

Durante a apresentação dos vídeos, em especial no segundo vídeo, percebemos que muitos estudantes não estavam prestando atenção na projeção. Circulando pelo laboratório percebemos que os alunos não estavam dispersos, mas sim já estavam explorando o *Scratch*, buscando conhecê-lo, querendo entender como funcionava. Com isso percebemos que era preciso fazer adequações para a oficina.

Nos dois primeiros encontros, os estudantes trabalharam de forma mais individual, explorando os recursos básicos do programa, cada qual buscando criar alguma animação ou tentando descobrir as possibilidades do programa. Na terceira aula, ficou evidente a formação de pequenos grupos (duplas, trios) para troca de experiências, buscando esclarecer dúvidas/problemas enfrentados na resolução da atividade proposta. Aumentou consideravelmente a interação entre os estudantes, pois nas duas primeiras aulas eles trabalharam de maneira mais individual, interagindo apenas com o *Scratch* e, algumas vezes, com o professor. Uma possível explicação para esse comportamento está no caráter mais exploratório proposto nessas aulas iniciais.

O desenvolvimento desse piloto trouxe muitas surpresas, motivando ainda mais a realização da oficina. Alguns alunos surpreenderam pela rápida compreensão do programa e pelo eficiente emprego de lógica e criatividade. Por outro lado, também verificamos que alguns alunos apresentaram maior dificuldade para entender o programa e conseguir desenvolver as atividades, causados principalmente pela dificuldade de estabelecer relações lógicas.

Ao iniciar o piloto cogitamos a realização da oficina com uma outra turma inteira; porém, com o decorrer dos encontros, essa ideia mostrou-se inviável. Verificamos que o pesquisador sozinho não conseguiria atender a todos os estudantes e, ao mesmo tempo, acompanhar e registrar toda a riqueza de informações e detalhes

que aconteciam, tanto de forma individual como coletiva. Concluímos que em torno de dez participantes seria o ideal para montar a oficina, de forma que o pesquisador pudesse acompanhar e registrar os principais acontecimentos.

Tendo em vista o interesse e participação da turma, bem como por uma questão de respeito e valorização aos estudantes, o piloto foi mantido mesmo após o início da oficina. Com isso, objetivamos proporcionar aos estudantes um maior conhecimento do *Scratch*, possibilitando-os explorar mais sua criatividade e gosto pela programação. O piloto encerrou no dia 3 de julho de 2017, com a apresentação dos projetos finais, totalizando 10 encontros.

3.2.2 Realização e descrição da oficina

As observações feitas no decorrer do estudo piloto indicaram a necessidade de algumas adequações no planejamento da oficina. Feitas essas readequações no planejamento, iniciamos o processo de inscrição de alunos para participação na oficina. Levando em consideração os horários disponíveis do pesquisador e os horários livres dos laboratórios de informática chegamos a apenas uma possibilidades de dia/turno para a realização da oficina. Cabe ressaltar que, para conseguir esse turno, foi necessário autorização da equipe de tecnologia da informação da instituição, pois esse horário estava reservado para a manutenção semanal do laboratório de informática. Dentro desse turno foram montadas as possíveis combinações de horários e disponibilizadas em uma planilha compartilhada.

A participação na oficina foi então ofertada para as turmas de primeiros anos do Ensino Médio do contraturno da oficina. Foram disponibilizadas dez vagas para a oficina, a fim de permitir um acompanhamento mais individualizado e viabilizar a análise de todos os códigos[31] produzidos. Os estudantes interessados preencheram a planilha compartilhada, indicando os horários que poderiam participar. Por fim, chegamos no dia e horário com maior número de estudantes inscritos, totalizando nove interessados.

[31] Na Computação utilizamos a palavra código como referência aos programas de computador. São as instruções, os passos que o usuário cria para que o computador execute determinada ação ou tarefa.

Agendamos o início da oficina para o dia 24 de maio de 2017, com encontros de 2 horas cada. Dois dias antes, um dos inscritos avisou pessoalmente ao pesquisador que não possuía mais interesse em participar da oficina. Um dia antes do início da oficina outro estudante enviou e-mail avisando de sua desistência por estar com muitas outras demandas. Optamos por manter a programação da oficina, mesmo com dois participantes a menos.

Antes de iniciar a oficina, os estudantes responderam ao mesmo questionário[32] aplicado aos participantes do piloto, envolvendo basicamente questões referentes a quais são e como utilizam os recursos tecnológicos que têm acesso. Buscamos também identificar seu perfil e interesses relacionados a jogos digitais e programação de computadores.

No primeiro encontro da oficina dois alunos não compareceram, ficando apenas com cinco alunos. Os alunos ausentes estavam em atendimento com o professor de matemática. Feita a explanação sobre a oficina e seu funcionamento, o pesquisador apresentou o *Scratch* aos participantes, orientou sobre o cadastro para utilização do programa e fez um breve exemplo prático, motivando os alunos a explorar o ambiente do *Scratch*.

O pesquisador teve uma grata surpresa no dia seguinte, durante a aula regular da disciplina, ao observar que, terminada a tarefa da aula, alguns alunos que haviam participado da oficina estavam utilizando o *Scratch*, dando sequência na exploração do ambiente, buscando avançar no que haviam iniciado no curso. Os alunos que não participavam da oficina ficaram interessados, querendo ver o que os colegas estavam criando, sendo que mostraram interesse em querer utilizar e também lamentaram o fato de não conseguirem frequentar a oficina.

No segundo encontro vieram seis estudantes, dentre os quais os dois alunos que não haviam participado do encontro anterior. Em conversa posterior com o estudante que faltou, esse relatou que estava passando por problemas de saúde na família, razão pela qual não conseguiria mais frequentar a oficina. Nesse segundo encontro foi necessário uma retomada do que havia sido feito, apresentando o ambiente aos dois participantes que não estavam no primeiro dia.

---

[32] Link para questionário on-line: https://goo.gl/forms/BY9blNYIN17ICMG83

O pesquisador propôs uma atividade baseada no desenho de figuras geométricas para direcionar um pouco mais a exploração do ambiente e motivar os estudantes na busca pela solução da tarefa. A maioria dos participantes preferiu dar continuidade ao desenvolvimento do projeto exploratório iniciado no primeiro encontro. No final da aula, o pesquisador sugeriu que os participantes pensassem em como resolver a atividade proposta, avisando que ela seria retomada e desenvolvida na semana seguinte.

O terceiro encontro iniciou com a presença de seis estudantes, entretanto um deles veio apenas para avisar que não poderia mais participar da oficina, pois, devido às baixas notas, necessitava frequentar os estudos orientados de matemática cujo horário colidia com o da oficina. Portanto restaram apenas cinco alunos participando da oficina. Nesse encontro foram finalizadas as atividades propostas anteriormente e uma nova tarefa foi sugerida, a qual baseava-se na construção de um labirinto.

Alguns eventos escolares agendados para a quarta semana de oficina acabaram por contribuir para que apenas três estudantes participassem desse encontro. Nessa semana teve dois dias de conselho de classe, além de um feriado e um dia com palestras. No quarto encontro os participantes deram prosseguimento nos projetos já iniciados, buscando solucionar alguns problemas e melhorando o que haviam construído. Chamou a atenção, nesse encontro, o projeto desenvolvido por um dos alunos, o qual não tinha relação com as atividades propostas, porém era muito criativo e mostrava o potencial do *Scratch* como instrumento de criação autoral.

No quinto encontro vieram os cinco participantes, os quais trabalharam basicamente na tarefa do labirinto, finalizando ou buscando agregar outros elementos ao projeto, explorando com isso mais recursos do *Scratch*. O pesquisador propôs aos estudantes que pensassem em um projeto final, com tema livre e que fosse de seu interesse. Também foi solicitado que cada participante compartilhasse seu projeto do labirinto com os demais colegas a fim de que todos pudessem ver e testar o que os outros membros da oficina haviam pensado e programado.

O sexto encontro coincidiu com a data da prova de matemática da turma no contraturno, por isso somente dois estudantes compareceram. Os demais ficaram estudando para a prova e foram nos estudos orientados do professor da disciplina. Os dois participantes que vieram na oficina pesquisaram sobre o que iriam construir no

projeto final, exploraram mais recursos do *Scratch* e buscaram colocar em prática as descobertas pesquisadas.

No sétimo e último encontro vieram quatro estudantes, pois um dos participantes avisou que não viria por motivos de saúde. Os estudantes presentes utilizaram esse encontro para finalizar seus projetos (alguns deles haviam trabalhado em casa durante a semana e vieram esclarecer suas dúvidas e buscar orientações para superar obstáculos ou melhorar o que haviam construído). Um dos participantes ainda não havia iniciado seu projeto, somente tinha definido o que desejava fazer. Mostrou empenho e habilidade, conseguindo implementar seu projeto durante esse encontro. Ao final, o pesquisador convidou os participantes a responderem um questionário avaliando a oficina.

Os cursos tradicionais de introdução à programação geralmente utilizam aulas expositivas, tendo o professor como figura central. Entretanto, para a oficina montamos um cenário diferente, cujo planejamento e execução buscou minimizar a interferência do pesquisador. Procuramos dar preferência à livre exploração por parte dos estudantes e sua interação com o *Scratch*, com os colegas e outros materiais disponíveis na internet, conforme ilustrado na Figura 6.

Figura 6- Ambiente de realização da oficina.

Fonte: criada pelo autor (2017).

Ao iniciar a oficina, os participantes foram informados de que se tratava de uma pesquisa científica e que os programas por eles desenvolvidos fariam parte dela,

sem que houvesse uma identificação do estudante. Seus responsáveis assinaram o termo de consentimento livre e esclarecido e os participantes assinaram o termo de assentimento livre e esclarecido, conforme determinação da Plataforma Brasil[33].

O planejamento da oficina foi inspirada no guia curricular *introdução à computação criativa com o Scratch*[34], desenvolvido pelo MIT. A carga horária da oficina foi de quatorze horas, distribuídas em sete encontros de duas horas cada. Os principais conceitos computacionais que buscamos desenvolver com a oficina estão descritos no Quadro 6.

Quadro 6 - Relação de conceitos computacionais a serem abordados na oficina e sua respectiva descrição.

Conceito	Descrição
sequência	identificar uma série de etapas de uma tarefa
ciclos	executar a mesma sequência várias vezes
execução em paralelo	fazer as ações decorrerem ao mesmo tempo
eventos	fazer um acontecimento causar outro acontecimento
condições	tomar decisões com base em condições
operadores	expressar operações matemáticas e lógicas
dados	armazenar, recuperar e atualizar valores

Fonte: guia curricular introdução à computação criativa com o Scratch (2011).

Na parte inicial da oficina os estudantes foram informados sobre a estrutura, objetivos e sua respectiva programação. Informamos também sobre a possibilidade de se organizarem da forma que julgassem mais conveniente para a realização das tarefas propostas, sendo que o objetivo era que encontrassem estratégias para desenvolver as atividades.

---

[33] A Plataforma Brasil é um sistema eletrônico criado pelo Governo Federal para sistematizar o recebimento dos projetos de pesquisa que envolvam seres humanos nos Comitês de Ética em todo o país. É uma base nacional e unificada de registros de pesquisas envolvendo seres humanos, permitindo acompanhamento das pesquisas em seus diferentes estágios.

[34] Este guia foi compartilhado sob uma licença *Creative Commons Attribution-ShareAlike*, o que significa que qualquer pessoa é completamente livre para usar, modificar e partilhar este trabalho, desde que explicite adequadamente a autoria e dê aos outros um acesso semelhante ao de quaisquer trabalhos deles derivados. Esse guia está disponível em: http://projectos.ese.ips.pt/cctic/wp-content/uploads/2011/10/Guia-Curricular-ScratchMIT-EduScratchLPpdf.pdf

Pelo fato dos estudantes da escola já estarem habituados com a utilização do Moodle[35], criamos um curso nessa plataforma onde foram disponibilizadas instruções, materiais e tarefas. Além disso, o pesquisador criou uma pasta compartilhada com o grupo para envio das atividades desenvolvidas.

Ao programar um computador os conceitos de variável, laços[36], condicionais, eventos, execução em paralelo, entre outros, dificilmente aparecem de forma isolada. Diante dessa realidade da programação de computadores, resolvemos adotar uma prática de programação baseada em exemplos (ou situações-problema), propondo a criação de programas com aumento gradual de tamanho e complexidade. As situações-problema foram apenas proposições, pois o objetivo principal era que os participantes explorassem o *Scratch,* a fim de buscar elementos que indicassem possíveis respostas ao problema de pesquisa.

Ao iniciar a oficina, o pesquisador apresentou o *Scratch*, detalhando sua estrutura e possibilidades de utilização. Desenvolveu alguns exemplos com diferentes níveis de complexidade, oportunizando aos estudantes a experiência de utilização prática desse programa. Em seguida, o pesquisador detalhou e orientou sobre o uso do *Caderno de Reflexões*, que consistiu em um arquivo de texto que foi compartilhado pelo pesquisador com cada um dos participantes, os quais tiveram acesso a um arquivo que foi utilizado para registrar suas percepções sobre a oficina. Em cada encontro o pesquisador propôs algumas questões para serem respondidas pelos estudantes e registradas nesse arquivo. As questões visavam coletar observações e percepções quanto ao uso do *Scratch*, dificuldades encontradas, possibilidades e descobertas, além de sistematizarem o planejamento das atividades – um ensaio de como tentaram resolver as atividades propostas.

Durante a execução das tarefas, o pesquisador fez anotações sobre a forma de organização dos alunos, suas interações, atividades desenvolvidas, o tempo utilizado, bem como quais participantes estavam presentes e ausentes em cada

---

[35] Moodle (***M**odular **O**bject **O**riented **D**istance **LE**arning*) é um sistema gerenciamento para criação de curso online. Esses sistemas são também chamados de Ambientes Virtuais de Aprendizagem (AVA) ou de *Learning Management System* (LMS). O moodle é um software livre de apoio à aprendizagem.

[36] Laços ou *loops* são conjuntos de instruções que um programa de computador percorre e repete um significativo número de vezes, até que sejam alcançadas as condições desejadas.

encontro. Essas anotações foram utilizadas na etapa de análise da oficina, possibilitando o resgate posterior das observações feitas durante as aulas.

Alguns encontros foram filmados pelo pesquisador para auxiliar no resgate de dados relevantes para a análise da oficina. Os primeiros, entretanto, não foram filmados, pois os participantes se mostraram desconfortáveis com a ideia. Com o passar dos encontros, foi aumentando o entrosamento entre os participantes, o que diminuiu a timidez e desconforto diante da filmagem. Os estudantes foram informados e consentiram com a utilização desse recurso.

A dinâmica da oficina seguiu a proposta de Vigotski, explorando a criatividade e autonomia dos estudantes, propondo desafios com o aumento gradual da complexidade e dos conceitos envolvidos. No decorrer da oficina, os alunos produziram em média quatro programas que foram postados na pasta compartilhada, seguindo as respectivas instruções.

Os conceitos vigotskianos de mediação, zona de desenvolvimento proximal, sociointeração e aprendizagem foram importantes norteadores da oficina. Por isso, ao mediá-la, o pesquisador buscou motivar os estudantes a explorarem os recursos do ambiente *Scratch*, fazendo questionamentos e dando sugestões, a fim de mobilizá-los a se engajarem nas tarefas e conseguirem avançar na utilização do ambiente. Os participantes compreenderam com certa facilidade o ambiente do *Scratch*, ficando empolgados e motivados a explorar seus recursos e a tentar resolver as tarefas. Por esse motivo, não foi necessário fazer uso da abordagem alternativa prevista no projeto de pesquisa, ou seja, não foi preciso utilizar a *Hora do Código*, proposta pelo Code.org.

3.3 CONSTITUIÇÃO DO *CORPUS*

Destacamos os seguintes elementos como constituintes do *corpus* da pesquisa:

- questionários aplicados antes e após a oficina;
- programas criados pelos estudantes para resolver cada tarefa;
- anotações do pesquisador; e
- anotações dos estudantes (*cadernos de reflexão*).

4 PROCEDIMENTO DE ANÁLISE

A partir do referencial teórico dessa pesquisa, optamos por utilizar os seguintes norteadores para realizar a análise dos dados gerados pela oficina:

- zona de desenvolvimento proximal (ZDP);
- mediação;
- sociointeração;
- aprendizagem;
- pensamento computacional; e
- programação de computadores.

A fim de manter o anonimato dos participantes da oficina, conforme assegurado nos termos de consentimento e assentimento, eles foram referenciados por E1, E2, E3, E4 e E5. Todos eles eram estudantes de primeiro ano do Ensino Médio Técnico, com idades variando entre 15 e 16 anos.

A análise do *corpus*, na busca de compreender e explicar o contexto dos alunos, atuando na oficina e utilizando o *Scratch,* seguiu as seguintes etapas: tabulação dos dados dos questionários pré-oficina; submissão dos programas criados pelos estudantes para resolver cada tarefa à análise do *Dr. Scratch*; análise por parte do pesquisador dos programas criados pelos estudantes com base no *framework* de Brennan e Resnick, considerando também os relatórios produzidos pelo *Dr. Scratch*; e, análise conjunta dos cadernos de reflexão dos estudantes, das anotações do pesquisador e dos questionários pós-oficina, articulando com os resultados das etapas anteriores, para construir a resposta à pergunta de pesquisa.

4.1 ANÁLISE DOS QUESTIONÁRIOS PRÉ-OFICINA

Iniciamos a análise do *corpus* a partir dos questionários pré-oficina. Com base nas respostas dadas, montamos os quadros a seguir, os quais serviram para contextualizar os participantes com relação a seus usos e acessos à tecnologia, bem como seus interesses por jogos digitais.

Nos dois primeiros quadros buscamos identificar a que tipo de equipamentos tecnológicos os participantes tinham acesso fora do ambiente escolar. No Quadro 1,

estabelecemos a correlação entre o acesso que os participantes possuíam em casa a computador, notebook e tablet. Já, no Quadro 2, relacionamos o acesso de cada participante a smartphone e, aos que possuíam, o tipo de conexão com a internet.

Quadro 1 - Relação entre os membros da oficina e seu acesso a computador em casa.

Relação entre os membros da oficina e seu acesso a computador em casa	
Possui computador de mesa (desktop), notebook e tablet	0%
Possui computador de mesa (desktop) e notebook	40%
Possui computador de mesa (desktop) e tablet	20%
Possui notebook e tablet	0%
Possui apenas computador de mesa (desktop)	0%
Possui apenas notebook	20%
Não possui computador em casa	20%

Fonte: criado pelo autor (2017).

Quadro 2 - Relação entre os membros da oficina e seu acesso a smartphone.

Relação entre os membros da oficina e seu acesso a smartphone	
Possui smartphone com acesso à internet	60%
Possui smartphone sem acesso à internet (somente através de Wi-Fi)	20%
Não possui smartphone	20%

Fonte: criado pelo autor (2017).

No Quadro 3 buscamos identificar os principais usos da internet por parte dos estudantes participantes da oficina. E, no Quadro 4 correlacionamos as principais categorias de jogos e sua respectiva afeição por parte dos estudantes buscando com isso conhecer seus interesses por jogos digitais.

Quadro 3 - Relação entre os membros da oficina e seu uso da internet.

Relação entre os membros da oficina e seu uso da internet	
Acessar as redes sociais	100%
Assistir filmes	80%
Ouvir música	80%
Ficar informado sobre notícias (jornais, revistas, ...)	60%
Jogar (jogos individuais)	60%
Jogar (jogos em rede)	60%
Fazer pesquisas e trabalhos escolares	80%
Diversão e passatempo (piadas, memes, ...)	80%
Pesquisas de preço/compras de produtos	40%
Estudar ou conhecer sobre algum assunto de seu interesse	60%

Fonte: criado pelo autor (2017).

Quadro 4 - Relação entre os membros da oficina e seu interesse por jogos digitais.

Relação entre os membros da oficina e seu interesse por jogos digitais	
Ação	60%
Shooter (atirador)	60%
Aventura	60%
Construção/gerenciamento	40%
Simulação de vida	60%
Música/ritmo	60%
Festa	20%
Quebra-cabeças	20%
Esportes	40%
Estratégia	40%
RPG	40%
Simulação de veículo	20%

Fonte: criado pelo autor (2017).

O Quadro 5 retrata a opinião dos participantes quanto a importância do aprendizado de programação por parte dos alunos do Ensino Médio. As respostas apontam para uma percepção dos estudantes com relação à importância desse conteúdo fazer parte do currículo, pois 80% dos participantes afirmam que todos os estudantes do Ensino Médio deveriam ter uma disciplina de programação.

Quadro 5 - Opinião dos membros da oficina sobre a importância do aprendizado de programação para alunos do Ensino Médio.

Opinião dos membros da oficina sobre a importância do aprendizado de programação para alunos do Ensino Médio	
Todos deveriam aprender	80%
Uma disciplina optativa (não obrigatória)	20%
Não no Ensino Médio, só na graduação	0%
Não considera importante	0%

Fonte: criado pelo autor (2017).

A análise dos quadros revelou que nem todos os estudantes possuíam acesso a computador em casa, um deles não tinha smartphone e outro possuía apenas smartphone com acesso à internet restrito à redes Wi-Fi. A partir desses dados, buscamos saber dos participantes mais detalhes sobre seus acessos à essas tecnologias. Constatamos que dois deles não conseguiriam utilizar o *Scratch* fora do ambiente escolar.

Quando analisamos a utilização da internet pelos participantes, identificamos que todos gostam de jogar, alguns preferindo jogos individuais e outros em rede. Já, o interesse pelas categorias de jogos variou bastante, sendo que cada participante escolheu pelo menos três, alguns chegando a optar por sete categorias diferentes. Essa relação entre os membros da oficina e seu interesse por jogos digitais não permitiu identificar alguém com propensão à utilização de lógica computacional.

Por fim, o Quadro 5 revelou o interesse dos participantes pelo aprendizado de programação. No questionário pós-oficina[37], essa questão foi retomada e os

[37] Conforme questões 4 e 5 do questionário: https://goo.gl/forms/Yt6YXQislmPzYJQJ3

participantes mantiveram seus posicionamentos, sendo que a grande maioria demonstrou desejo por aprofundar seus estudos na área de programação de computadores.

4.2 ANÁLISE DOS PROJETOS PELO DR. SCRATCH

Completada a etapa de elaboração e análise dos quadros referentes ao questionário pré-oficina, iniciamos a avaliação dos programas criados pelos participantes. Para essa tarefa utilizamos a ferramenta *Dr. Scratch*[38].

4.2.1 Conhecendo o *Dr. Scratch*

O *Dr. Scratch* é uma ferramenta gratuita, de código aberto, disponível na internet que permite analisar com facilidade projetos criados no *Scratch*. Foi desenvolvida por uma equipe de professores, programadores e pesquisadores da Universidad Rey Juan Carlos, em Madri, Espanha, com o objetivo de avaliar os projetos criados no *Scratch*.

Essa ferramenta permite ao usuário descobrir em que nível se encontram seus projetos e fornece informações para auxiliá-lo a incrementar suas habilidades de programação. Pode ser utilizada de duas maneiras: informando o endereço (a url) do projeto ou enviando o arquivo correspondente.

O *Dr. Scratch* analisa o projeto sob a perspectiva do desenvolvimento do pensamento computacional, identificando níveis e atribuindo pontuação quanto ao emprego de abstração, pensamento lógico, sincronização, paralelismo, controle de fluxo, interatividade com o usuário e representação dos dados. Ao final da análise, informa ao usuário o grau de desenvolvimento do pensamento computacional indicado no projeto que foi analisado e atribui uma pontuação relacionada a isso.

O Quadro 6 apresenta os 7 conceitos analisados pelo *Dr. Scratch* e quais são os elementos de programação característicos em cada um dos níveis (básico, em desenvolvimento e proficiente). Esse quadro foi elaborado por Moreno-León e Robles

---

[38] *Dr. Scratch* está disponível em: http://drscratch.org/

(2015) e traduzido por Webber et al. (2016). Ao avaliar os projetos, essa ferramenta atribui uma nota de 0 (não está presente) a 3 (proficiente) para cada conceito, considerando a qualidade do projeto de acordo com a pontuação feita (até um total de 21 pontos).

Quadro 6 - Níveis de desenvolvimento para cada conceito

Conceitos	Básico	Em Desenvolvimento	Proficiente
Abstração e decomposição de problemas	Mais que uma programação e mais de um objeto	Definição de blocos	Uso de clones
Paralelismo	Dois programas são executados quando a bandeira verde é pressionada	Dois programas são executados quando uma tecla é pressionada ou quando um objeto é clicado	Dois programas são executados na verificação da ocorrência de um evento (exemplo, ao criar clone, ao mudar fundo, etc)
Pensamento Lógico Condicional	Se	Se, senão	Uso de expressões lógicas
Sincronização	Esperar	Anunciar, receber mensagem, parar tudo, parar programa, parar programa do objeto	Esperar até, quando o plano de fundo mudar para, anunciar e esperar
Controle de Fluxo	Sequência de blocos	Repetir, sempre	Repetir até
Interatividade com o Usuário	Bandeira verde	Tecla pressionada, objeto clicado, pergunte e espere, blocos de mouse	Quando %s é > %s, vídeo, áudio
Representação de Dados	Modificador de propriedades de objetos	Operações com variáveis	Operações em listas

Fonte: (WEBBER et al., 2016)

Para uma melhor compreensão do quadro, detalharemos a seguir os conceitos e os elementos de programação nele representados com uma breve descrição, baseados em Brennan e Resnick (2012):

a) abstração e decomposição de problemas: a habilidade de abstrair e decompor problemas auxilia a quebrar um problema em partes menores, mais fáceis de entender, a programar e depurar (corrigir os erros).

b) paralelismo: é a possibilidade de que várias coisas ocorram simultaneamente, ou seja, ao mesmo tempo. Por exemplo, dois ou mais personagens executando ações diferentes simultaneamente.

c) pensamento lógico condicional: são instruções que auxiliam os projetos a se tornarem dinâmicos, de modo a se comportarem de forma diferente dependendo da situação. Essas instruções não são tão importantes em histórias, porque essas geralmente tem uma estrutura linear, ou seja, geralmente desejamos que sempre sejam executadas do mesmo jeito. Mas em outros projetos, como jogos, as instruções de lógica são essenciais para criar ações diferentes dependendo das situações.

d) sincronização: permite organizar e definir em que ordem e em que momento cada ação irá ocorrer.

e) controle de fluxo: permite controlar o comportamento dos personagens, fazendo, por exemplo, alguns blocos serem repetidos um determinado número de vezes ou até que uma situação ocorra.

f) interatividade com o usuário: tornam os projetos mais dinâmicos, permitindo que o usuário possa criar situações novas cada vez que executar o projeto. Por exemplo, possibilitar que o usuário utilize o teclado ou o mouse para mover um personagem, que responda perguntas ou use a webcam.

g) representação de dados: cada projeto precisa de um conjunto de informações sobre os personagens para funcionar corretamente. Por exemplo, a posição de cada personagem, a direção que está apontando, tamanho, etc. Além disso, ao criarmos um projeto podemos criar novos repositórios de informação para armazenar dados como pontuação, o tempo decorrido, a classificação, as recompensas coletadas, etc.

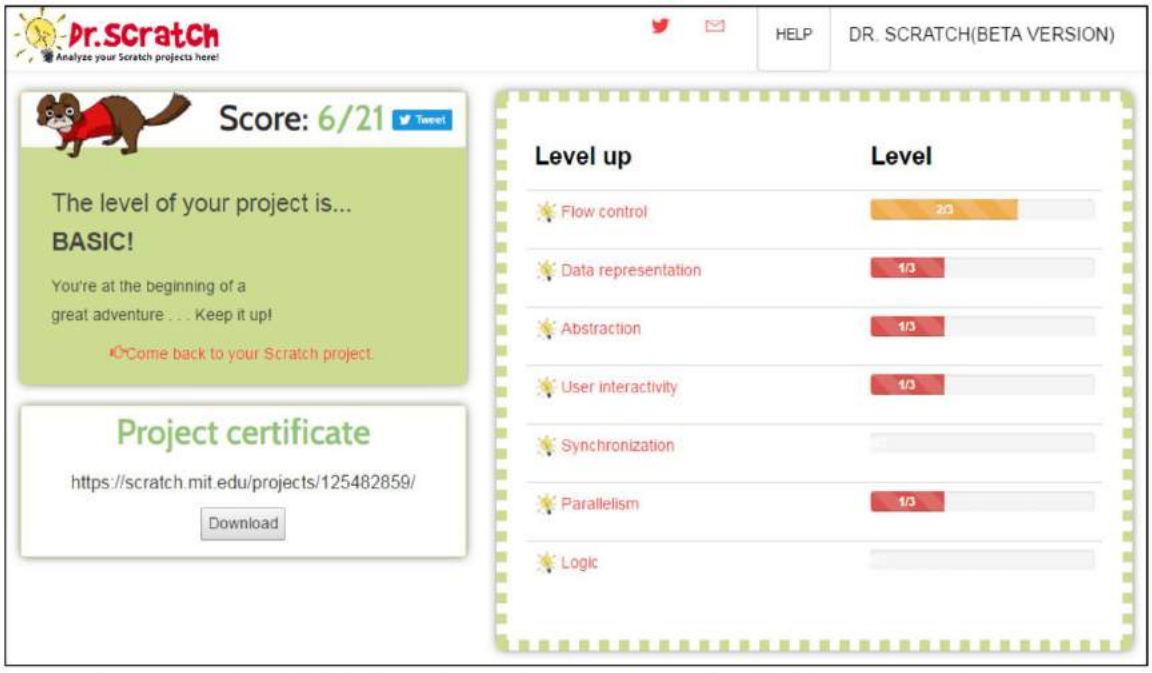
h) elementos de programação: em cada nível encontramos diferentes comandos, ações, estruturas lógicas ou relacionais. Cada um desses elementos possui uma complexidade de compreensão e de emprego, por isso cada nível tem um determinado conjunto de elementos. Conforme o usuário for avançando no seu aprendizado, for desenvolvendo seu pensamento lógico e relacional, irá aprimorar seus projetos, inserindo novos elementos, e assim esses tenderão a aumentar de complexidade.

Além dessas informações relacionadas ao pensamento computacional, a ferramenta *Dr. Scratch* também verifica e detecta várias questões como, por exemplo,

código que nunca é executado (código morto), código repetido, objetos com nome padrão, ou atributos de atores que podem ser modificados, porém não foram corretamente inicializados.

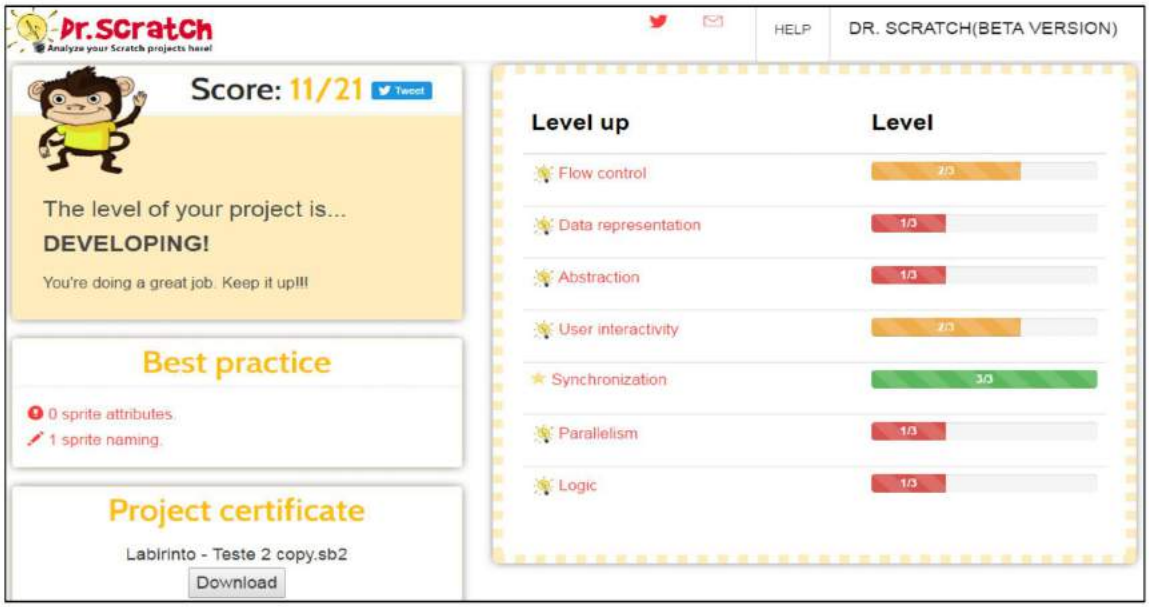
A Figura 7 ilustra a análise de um projeto de um usuário iniciante, inexperiente, fazendo uso superficial do pensamento computacional. A Figura 8, por sua vez, exemplifica a análise de um projeto cujo usuário já passou a ser categorizado como em desenvolvimento, fazendo uso mais aprofundado de alguns elementos relacionados ao pensamento computacional.

Figura 7 - Projeto empregando o pensamento computacional em nível básico.



Fonte: criada pelo autor (2017) utilizando o *software Dr. Scratch* - http://www.drscratch.org

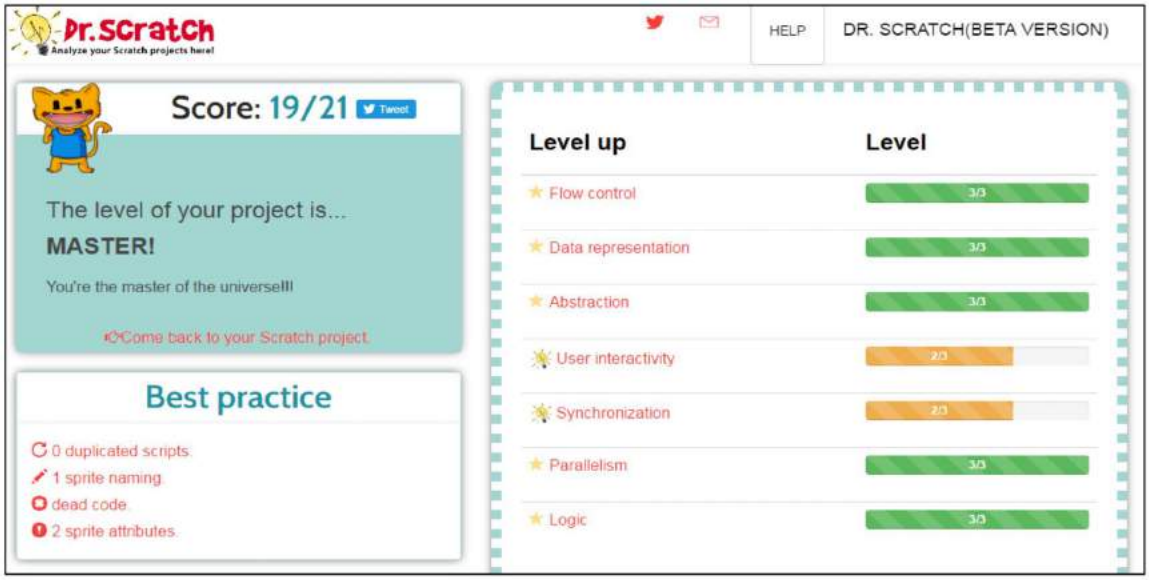
Figura 8 - Projeto empregando o pensamento computacional em nível intermediário (em desenvolvimento).



Fonte: criada pelo autor (2017) utilizando o *software Dr. Scratch* - http://www.drscratch.org

Na Figura 9, o projeto analisado pelo *Dr. Scratch* apresenta características de um usuário experiente, explorando com maior habilidade e competência as técnicas e boas práticas de programação, indicando possuir um nível avançado com relação ao pensamento computacional.

Figura 9 - Projeto empregando o pensamento computacional em nível avançado.



Fonte: criada pelo autor (2017) utilizando o *software Dr. Scratch* - http://www.drscratch.org

4.2.2 Relatórios de análise dos projetos gerados pelo *Dr. Scratch*

Os projetos criados pelos participantes da oficina foram submetidos à análise do *Dr. Scratch*, o qual gerou um relatório para cada um dos trabalhos. O número de projetos desenvolvidos variou conforme o participante, totalizando dezenove analisados. Os relatórios produzidos foram sistematizados e serviram para a elaboração dos quadros a seguir.

Cada quadro agrupa os projetos desenvolvidos por um dos participantes da oficina, relacionando a pontuação atribuída pelo *Dr. Scratch* aos sete conceitos analisados: lógica, paralelismo, interatividade com o usuário, representação de dados, controle de fluxo, sincronização e abstração. O quadro também apresenta a pontuação total atribuída a cada projeto e indica em qual nível ele se encontra, com base no desenvolvimento do pensamento computacional. No final também apresenta as observações destacadas pela ferramenta.

No Quadro 7, apresentamos os quatro projetos desenvolvidos no decorrer da oficina pelo estudante E1, com suas respectivas pontuações, nível em que se encontra, além das observações destacadas. A seguir, no Quadro 8, apresentamos os quatro projetos desenvolvidos no decorrer da oficina pelo estudante E2, relacionando a pontuação atribuída pelo *Dr. Scratch* aos sete conceitos analisados. Apresentamos também o nível atribuído a cada projeto e as respectivas observações destacadas pela ferramenta.

Quadro 7 - Análise dos projetos criados pelo estudante E1.

(continua)

Estudante E1				
Conceitos Analisados	**Projeto 1**	**Projeto 2**	**Projeto 3**	**Projeto 4**
Lógica	1	0	1	3
Paralelismo	3	1	1	3
Interatividade com usuário	2	1	2	2
Representação de dados	1	1	2	2
Controle de fluxo	2	2	1	2
Sincronização	3	1	2	3
Abstração	1	1	2	1
Pontuação (Máximo 21)	**13**	**7**	**11**	**16**
Nível	**Em Desenvolvimento**	**Básico**	**Em Desenvolvimento**	**Proficiente**

(conclusão)

Observações	1 atributo de personagem	-	0 atributo de personagem	0 scripts duplicados
	0 nomeação de personagens	-	0 nomeação de personagens	0 nomeação de personagens
	-	-	-	0 código morto
	-	-	-	2 atributos de personagem

Fonte: criado pelo autor (2017).

Quadro 8 - Análise dos projetos criados pelo estudante E2.

Estudante E2				
Conceitos Analisados	**Projeto 1**	**Projeto 2**	**Projeto 3**	**Projeto 4**
Lógica	0	0	1	1
Paralelismo	0	1	1	3
Interatividade com usuário	1	1	2	2
Representação de dados	1	1	1	1
Controle de fluxo	2	2	1	1
Sincronização	1	1	1	3
Abstração	0	1	1	1
Pontuação (Máximo 21)	**5**	**7**	**8**	**12**
Nível	**Básico**	**Básico**	**Em Desenvolvimento**	**Em Desenvolvimento**
Observações	-	-	1 atributo de personagem	1 atributo de personagem
	-	-	0 nomeação de personagens	0 nomeação de personagens

Fonte: criado pelo autor (2017).

No Quadro 9, apresentamos os projetos desenvolvidos pelo estudante E3, destacando que esse participante da oficina desenvolveu cinco projetos. Nesse quadro estão as respectivas pontuações para cada conceito analisado, a pontuação total de cada trabalho, bem como o nível em que se encontram. Ao final desse quadro estão elencadas as observações destacadas pelo *Dr. Scratch*.

Quadro 9 - Análise dos projetos criados pelo estudante E3.

Estudante E3					
Conceitos Analisados	**Projeto 1**	**Projeto 2**	**Projeto 3**	**Projeto 4**	**Projeto 5**
Lógica	1	0	0	1	1
Paralelismo	0	3	3	3	3
Interatividade com usuário	2	2	2	2	2
Representação de dados	1	1	1	1	2
Controle de fluxo	1	1	1	2	2
Sincronização	0	3	3	3	3
Abstração	1	1	1	1	2
Pontuação (Máximo 21)	**6**	**11**	**11**	**13**	**15**
Nível	**Básico**	**Em Desen-volvimento**	**Em Desen-volvimento**	**Em Desen-volvimento**	**Proficiente**
Observações	-	3 atributos de personagem	5 atributos de personagem	17 atributos personagem	0 scripts duplicados
	-	0 nomeação personagens	0 nomeação personagens	0 nomeação personagens	0 nomeação personagens
	-	-	-	-	0 código morto
	-	-	-	-	7 atributos de personagem

Fonte: criado pelo autor (2017).

No Quadro 10, apresentamos os dois projetos desenvolvidos no decorrer da oficina pelo estudante E4. Nesse quadro estão relacionadas as pontuações atribuídas a cada um dos sete conceitos analisados e as respectivas observações destacadas pela ferramenta *Dr. Scratch*.

Quadro 10 - Análise dos projetos criados pelo estudante E4.

(continua)

Estudante E4		
Conceitos Analisados	**Projeto 1**	**Projeto 2**
Lógica	1	0
Paralelismo	1	0
Interatividade com o usuário	2	2
Representação de dados	1	1
Controle de fluxo	2	1
Sincronização	3	1
Abstração	1	1
Pontuação (Máximo 21)	**11**	**6**

(conclusão)

Nível	Em Desenvolvimento	Básico
Observações	1 atributo de personagem	-
	0 nomeação de personagens	-

Fonte: criado pelo autor (2017).

O Quadro 11 apresenta os quatro projetos desenvolvidos no decorrer da oficina pelo estudante E5, com suas respectivas pontuações e nível em que cada um se encontra. Os projetos desenvolvidos por esse participante não obtiveram nenhuma observação destacada pelo *Dr. Scratch*.

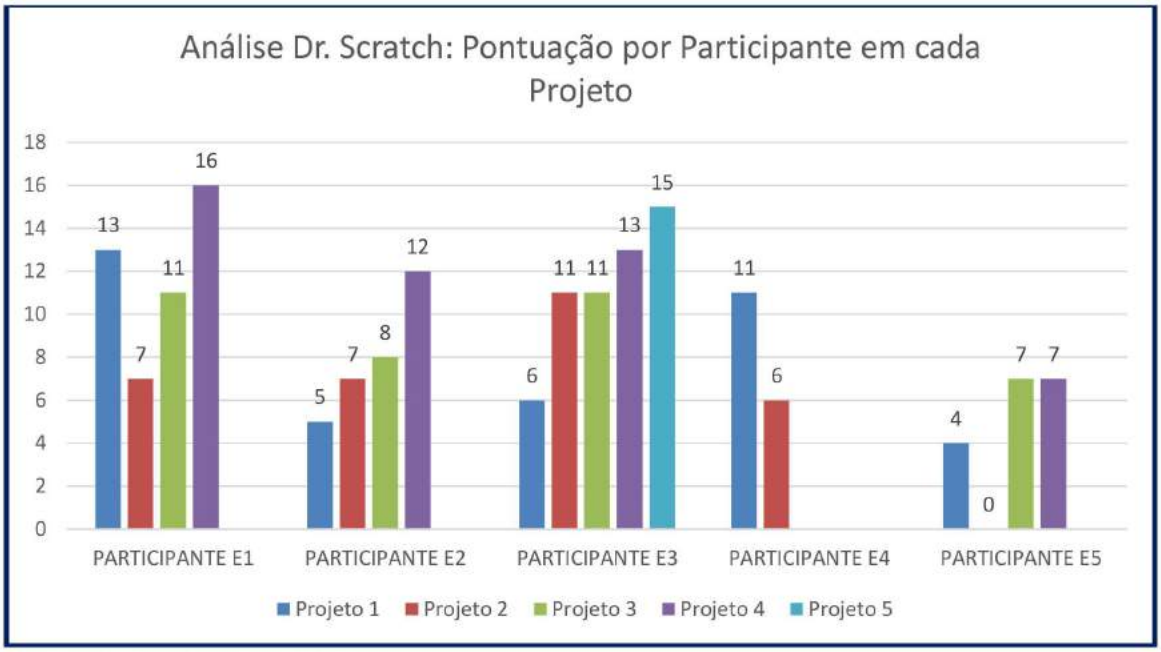
Quadro 11 - Análise dos projetos criados pelo estudante E5.

Estudante E5				
Conceitos Analisados	**Projeto 1**	**Projeto 2**	**Projeto 3**	**Projeto 4**
Lógica	0	0	0	0
Paralelismo	0	0	1	1
Interatividade com usuário	1	0	1	1
Representação de dados	1	0	1	1
Controle de fluxo	1	0	2	2
Sincronização	1	0	1	1
Abstração	0	0	1	1
Pontuação (Máximo 21)	**4**	**0**	**7**	**7**
Nível	**Básico**	**Básico**	**Básico**	**Básico**
Observações	-	-	-	-
	-	-	-	-

Fonte: criado pelo autor (2017).

No Gráfico 1 estão representadas as pontuações atribuídas pelo *Dr. Scratch* a cada um dos dezenove projetos analisados, agrupados por participante. Nele podemos observar que a maioria dos participantes apresentou uma evolução crescente em relação à pontuação obtida em cada projeto. Ressalta-se a pontuação obtida pelo Estudante E1 no seu Projeto 1, o qual atingiu 13 pontos – quase o dobro da pontuação atingida no seu Projeto 2. Outro aspecto relevante é que o participante E4 só entregou dois projetos, os quais apresentaram um declínio na pontuação. Na seção 4.3.2 detalharemos essas situações, bem como o motivo pelo qual o estudante E5 obteve 0 pontos em seu Projeto 2.

Gráfico 1- Pontuação atribuída pelo *Dr. Scratch* a cada projeto analisado.



Fonte: criado pelo autor (2017).

4.3 ANÁLISE DOS PROJETOS À LUZ DOS NORTEADORES DO QUADRO TEÓRICO

Para realizar a análise dos projetos optamos por utilizar os seguintes norteadores teóricos: zona de desenvolvimento proximal (ZDP); mediação; sociointeração; aprendizagem; pensamento computacional; e, programação de computadores. Além disso, optamos também por considerar alguns norteadores teóricos do pensamento computacional e de programação advindos do *framework*[39] de Brennan e Resnick (2012).

Enfatizamos que, ao analisar os programas, o *Dr. Scratch* não leva em consideração o enunciado da tarefa ou o que está sendo proposto desenvolver com o projeto. Por esse motivo, optamos por realizar outra análise, avaliando e testando cada projeto desenvolvido pelos participantes, na busca por reconhecer como foi estruturada a lógica do estudante na elaboração da solução para cada tarefa. Ou seja,

[39] Um *framework* pode ser definido como um conjunto de conceitos usado para resolver um problema de um domínio específico. Um *framework* conceitual é um modelo de dados para um domínio (SAUVÉ, 2017).

identificando que conceitos de programação foram empregados, quais funcionalidades foram implementadas, além de verificar erros e dificuldades do aluno no desenvolvimento dos projetos.

Empregamos os conceitos computacionais e seus respectivos níveis de desenvolvimento elaborados por Moreno-León e Robles (2015) e descritos na seção 4.2.1. Além disso, também foram utilizadas, nessa análise, anotações realizadas pelo pesquisador ao longo do desenvolvimento das oficinas, bem como os registros realizados pelos participantes em seus cadernos de reflexões e as respostas ao questionário pós-oficina.

4.3.1 Conhecendo o *framework* de Brennan e Resnick

O *framework* foi desenvolvido com o objetivo de permitir compreender quais competências do pensamento computacional podem ser exploradas com o uso de uma linguagem de introdução à programação como o *Scratch*. Esse *framework* está dividido em três dimensões: conceitos computacionais, práticas computacionais e perspectivas computacionais (BRENNAN; RESNICK, 2012; MORENO-LEÓN; ROBLES, 2015).

A primeira dimensão especificada no *framework* refere-se aos conceitos computacionais empregados na programação:

- a sequência expressa uma série de procedimentos a serem realizados, os quais são acionados passo a passo, uma linha por vez;

- as estruturas de repetição permitem que um conjunto de passos seja realizado repetidamente;

- os eventos são mecanismos programados para serem acionados a partir da ocorrência de uma situação especifica, assim quando um evento é ativado certas instruções de código serão executadas;

- o paralelismo, por sua vez, é empregado para possibilitar a execução de duas ou mais sequências de instruções ao mesmo tempo, em paralelo;

- os condicionais permitem criar condições para a execução de ações;

- os operadores podem ser compostos por estruturas da matemática, da lógica booleana (E, OU, NÃO) e expressões de comparação; e

- os dados referem-se a armazenamento, recuperação e atualização de valores que são guardados em variáveis.

A outra dimensão, denominada práticas computacionais, agrupa os processos de pensar e de aprender durante as atividades de programação e possui os seguintes grupos:

- iterativo e incremental o qual representa o processo sequencial de planejamento e execução para codificação;

- teste e depuração que envolve as etapas de análise e reflexão detalhada do código (instruções), para verificar seu funcionamento e descobrir possíveis erros e a sua origem;

- reuso e reformulação o qual aborda as práticas de utilizar trechos próprios de códigos desenvolvidos em trabalhos anteriores ou fazer uso de trechos de terceiros para ajustes e adaptações; e

- abstração e modularização que busca desconsiderar informações irrelevantes para focar apenas nas informações importantes, além de procurar dividir o problema em pequenas partes para auxiliar a propor uma solução.

A terceira dimensão, chamada de perspectivas computacionais, aborda as atitudes desempenhadas pelos alunos durante a programação (expressar-se, conectar-se e questionar-se). Essas atitudes estão relacionadas ao modo de se expressar, de se conectar com outros alunos e com a comunidade, de se questionar e de agir colaborativamente.

4.3.2 Análise dos projetos

Na análise dos projetos procuramos identificar, primeiramente, quais comandos e recursos do *Scratch* que os estudantes exploraram e utilizaram na construção de seus trabalhos. Em um segundo momento, a análise visou compreender como os participantes estruturaram seus projetos a fim de criar uma solução para a tarefa proposta e verificar se essa solução estava coerente com o que foi proposto.

Buscamos, nessa análise, trazer os principais resultados da avaliação feita pelo *Dr. Scratch*, relacionando com a apreciação do pesquisador e os demais

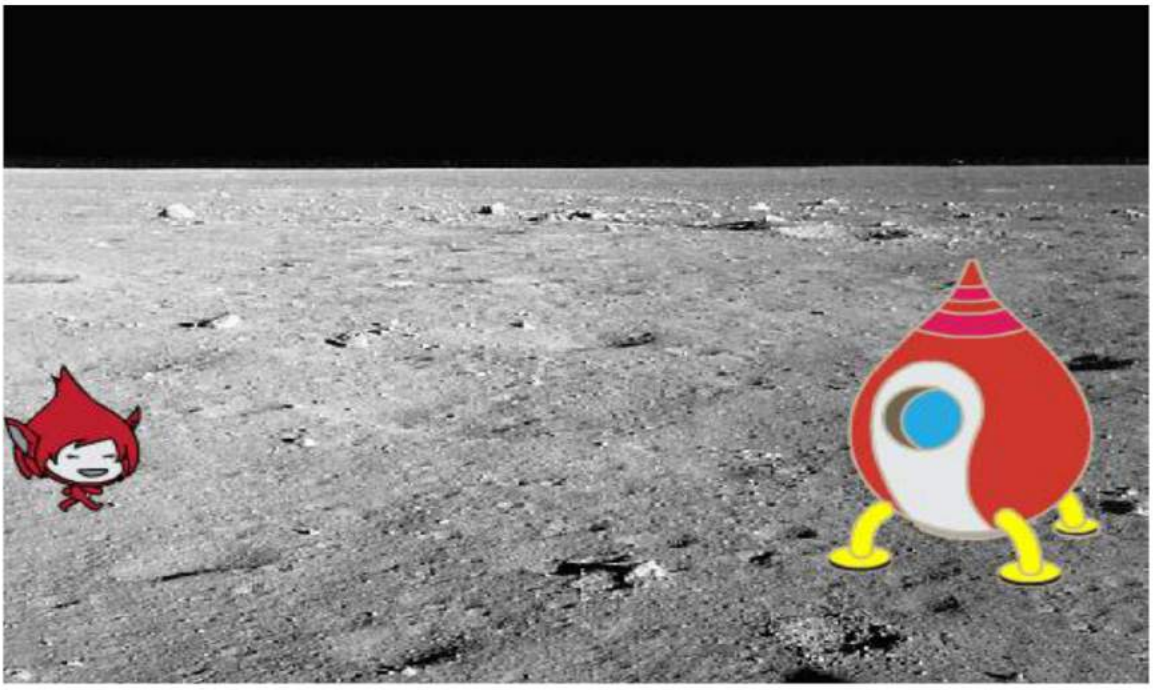
constituintes do *corpus* (cadernos de reflexão dos estudantes, anotações do pesquisador e questionário pós-oficina). Essas relações foram norteadas pela teoria vigotskiana, por isso procuramos identificar possíveis evidências relacionadas aos conceitos dessa teoria. A análise dos projetos está organizada da seguinte maneira: identificação do participante; análise individual de cada um de seus projetos; e, articulações sobre a análise dos projetos desse participante.

4.3.2.1 Análise dos projetos do participante E1

a) **Projeto 1**

O primeiro projeto do participante E1 foi de caráter exploratório, no qual o estudante utilizou três atores para simular uma viagem da lua para a terra, conforme ilustra a Figura 10. Para isso, empregou controle de movimentação, troca de fantasias combinado com execução de som, mudança de palco e controle de eventos.

Figura 10 - Projeto 1 do estudante E1.



Fonte: imagem criada no *software Scratch* (2017). Disponível em: https://scratch.mit.edu/

Ele pode ser considerado um projeto simples, pois empregou poucos comandos e ações. Entretanto, o estudante mostrou a utilização de diversos recursos

relacionados ao pensamento computacional, como a inicialização de algumas variáveis, utilização de condicional, emprego de laço de repetição, execução em paralelo e sincronização. Para esse primeiro projeto, o participante superou as expectativas, pois em alguns conceitos foi além do básico; e, nos conceitos de paralelismo e interatividade com o usuário, atingiu a pontuação máxima.

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 13 pontos, em uma escala que vai até 21, e considerou que o programa atingiu o segundo nível, denominado "em desenvolvimento". Os conceitos de paralelismo e sincronização atingiram a pontuação máxima, seguidos por interatividade com o usuário e controle de fluxo, com 2 pontos cada. Os conceitos de lógica, representação de dados e abstração atingiram 1 ponto cada.

Ao analisarmos o programa levando em consideração as ideias e propostas do estudante para essa tarefa, podemos concluir que o participante criou um projeto coerente com sua proposta, indicando ter explorado e utilizado com sucesso diversos comandos do *Scratch*. Com relação aos conceitos computacionais, podemos destacar o emprego eficiente de laço de repetição, conforme ilustrado na Figura 11. Esse laço é repetido oito vezes, fazendo com que o ator se desloque gradativamente para uma determinada posição, dando a sensação de movimento.

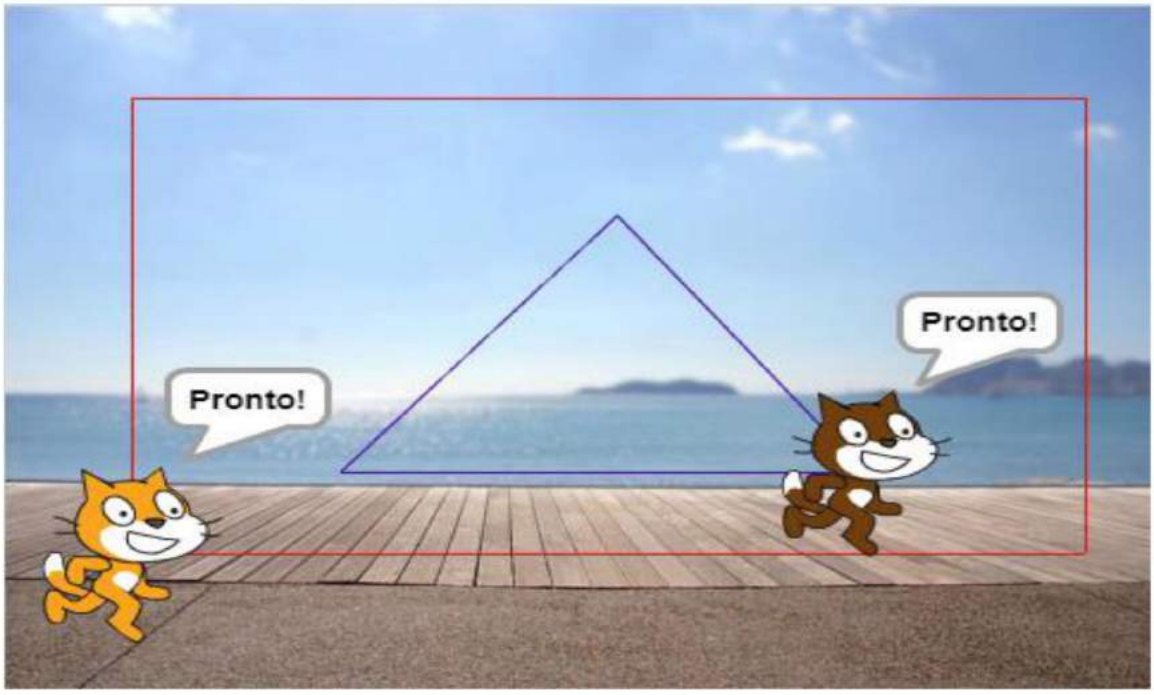
Figura 11- Utilização eficiente de laço de repetição.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

b) **Projeto 2**

No segundo projeto, foi proposto a criação de, pelo menos, duas formas geométricas. Para desenvolver essa tarefa o estudante utilizou dois atores, cada qual desenha uma forma geométrica de maneira simultânea, conforme ilustrado na Figura 12. O programa executa sempre as mesmas ações, ou seja, o usuário não controla os atores.

Figura 12- Projeto 2 do estudante E1.



Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 7 pontos, em uma escala que vai até 21 e considerou que o programa atingiu apenas o primeiro nível, denominado de "básico". Atribuiu maior pontuação ao controle de fluxo e não pontuou o conceito de lógica.

Ao compararmos com o primeiro projeto, a avaliação feita pelo *Dr. Scratch* mostra uma queda considerável na pontuação resultando na queda de nível também. Por outro lado, ao analisarmos o projeto levando em consideração o problema proposto, concluímos que ele ficou coerente com o que foi sugerido, sendo que o estudante evidenciou a exploração de outros recursos do *Scratch,* bem como teve um cuidado com a inicialização das variáveis, o que foi pouco observado no primeiro projeto. Um exemplo de inicialização de variáveis utilizado pelo estudante nesse projeto pode ser visualizado na Figura 13. Nela podemos observar que quando o evento de início do programa é ativado, o ator é posicionado em determinada coordenada do cenário, olhando para a posição 90º (olhando para a direita) e qualquer desenho que tenha sido feito no cenário é apagado.

Figura 13- Exemplo de inicialização de variáveis.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

O estudante também se preocupou em interagir com o usuário, conforme ilustra a Figura 14, na qual o ator lhe diz duas frases antes de iniciar o desenho da figura geométrica.

Figura 14- Exemplo de interação com o usuário.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Para analisar os projetos é importante levar em consideração o enunciado da questão proposta, o tempo que o estudante teve para desenvolver a atividade, seus conhecimentos prévios, bem como seu percurso exploratório do *Scratch*. Com base nisso, é possível afirmar que o estudante foi criativo na solução apresentada, mostrando bom emprego dos recursos do *Scratch* e foi coerente com a questão proposta. Entretanto, a solução apresentada poderia ser aprimorada, adicionando outros elementos e possibilitando uma maior interação com o usuário.

Segundo o estudante E1, o desenvolvimento desse projeto "*foi bem divertido, complicado também. Achei mais fácil fazer direto no Scratch, pois consegui executar para ver se estava correto e qual erros cometi*" (**E1**). Nessa colocação, o participante enfatiza sua preferência por desenvolver seus projetos diretamente no *Scratch,* sem planejar ou rascunhar primeiramente no papel, pela simplicidade e facilidade de utilização do ambiente. Ressaltou também a importância da ferramenta possibilitar testar imediatamente o que é programado, permitindo a identificação e correção dos erros cometidos.

Nesse sentido, o *Scratch,* como instrumento mediador, possibilita interação com o usuário, onde os comandos inseridos podem ser testados e validados de maneira prática e intuitiva, contribuindo para a aprendizagem de programação. Essas características também são ressaltadas por Vigotski (2007, p. 55), quando diz que "a função do instrumento é servir como um condutor da influência humana sobre o objeto da atividade; ele é orientado externamente; deve necessariamente levar a mudanças nos objetos".

O estudante **E1** afirmou, em seu caderno de reflexões, que "*utilizei os recursos que eu já conhecia no Scratch, como o adicionar um valor a y e x, pesquisei no Youtube alguns vídeos para entender melhor cada comando".* Com essa declaração, o participante indicou que alguns comandos explorados no projeto 1 foram significados e com isso foi possível relacioná-los à tarefa proposta, empregando-os no projeto atual. Além disso, revelou algumas de suas estratégias empregadas para resolver as tarefas, pesquisando suas dúvidas de maneira autônoma e testando as descobertas no ambiente.

Quanto as dificuldades e/ou facilidades encontradas na realização dessa tarefa, o estudante relatou que *"tive dificuldades em montar os comandos para fazer os personagens iniciarem em um lugar e para apagar. Mas depois que aprendi achei bem fácil"* (**E1**). Essa afirmação, além de pontuar alguns dos desafios enfrentados pelo participante E1 na realização do projeto 2, também indicou seu empenho em encontrar respostas para as dificuldades encontradas. Suas pesquisas possibilitaram elaborar hipóteses para a solução, as quais foram testadas e, ao produzirem o resultado esperado, fizeram o estudante perceber que a solução era simples. Nesse sentido, destacamos a importância do *Scratch* em fornecer possibilidades de teste de hipóteses, servindo como mediador desse processo. Mediação essa que, para Vigotski (2007), é um processo de intervenção de um elemento em uma relação objeto-sujeito.

c) **Projeto 3**

O projeto 3 foi desenvolvido em duas fases. Na primeira, foi proposta a criação de um labirinto no qual um ator deveria ter um ponto de partida e outro de chegada. O usuário deveria controlar o deslocamento desse ator e o programa não poderia

permitir que o ator ultrapassasse as linhas divisórias do labirinto. Na segunda fase, foi proposto que os estudantes aperfeiçoassem o projeto, tornando-o mais atrativo e controlando a pontuação.

Para desenvolver essa tarefa o estudante utilizou diversos atores, conforme ilustrado na Figura 15. Um, simbolizando um filho, era controlado pelo usuário e deveria chegar até o ator mãe, o qual estava posicionado no ponto de chegada. Os outros atores, representando tesouros, estavam espalhados pelo labirinto. Conforme o ator filho atingisse um desse tesouros recebia uma pontuação, a qual era exibida na parte superior da tela. Ao chegar no ator mãe um som era executado, indicando o final da tarefa.

Figura 15- Projeto 3 do estudante E1.

Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 11 pontos, em uma escala que vai até 21 e considerou que o programa atingiu o segundo nível, ou seja,

"em desenvolvimento". Interatividade com o usuário, representação de dados, sincronização e abstração foram os conceitos com maior pontuação, atingindo 2 pontos. Os demais conceitos também pontuaram, obtendo 1 ponto cada.

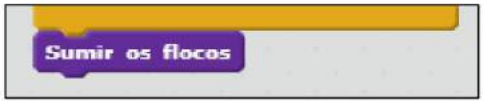
Ao analisarmos o programa levando em consideração o problema apresentado, podemos concluir que o estudante conseguiu criar um projeto coerente com o problema proposto, evidenciando um avanço significativo com relação ao pensamento computacional, principalmente pelo uso de funções para reduzir e simplificar o código. Percebemos que o uso de funções ainda precisa ser melhor compreendido e trabalhado, mas foi uma experiência que contribuiu para reduzir consideravelmente o tamanho do código. O uso de funções faz parte do grupo abstração e modularização, dentro da dimensão de práticas computacionais, do *framework* de Brennan e Resnick.

Um exemplo de aperfeiçoamento a ser implementado na utilização de funções é ilustrado na Figura 16, na qual é criado um bloco (ou função) denominado "Sumir os Flocos". Nele foram adicionados dezesseis comandos. Após a criação desse bloco, em quatro pontos do programa foram retiradas essas dezesseis linhas e inserido apenas um único comando no local, conforme ilustra a Figura 17. Com isso, várias linhas de código puderam ser suprimidas do programa.

Figura 16- Exemplo de utilização de funções (criação de blocos).

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Figura 17- Exemplo de utilização do novo bloco (função) criado.



Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Outro aspecto que merece uma atenção maior é quanto a finalização do jogo. O estudante definiu que quando o ator filho encosta no ator mãe um som é executado. Porém, se o usuário estiver com o controle de volume baixo ou desligado, não saberá que o jogo acabou, pois nenhuma mensagem é exibida e o ator continua podendo se locomover pelo labirinto.

Por fim, cabe ressaltar o uso bem sucedido de uma variável para controlar a pontuação do jogador. Na Figura 18 podemos observar a inicialização da variável *Pontuação* com o valor zero quando o jogo é iniciado (clicado na bandeira verde). Já na Figura 19 está representado um exemplo de incremento da variável *Pontuação* com mais dez, caso o ator tenha tocado na cor azul, indicando o toque em um dos flocos de neve. As variáveis fazem parte dos dados, na dimensão conceitos computacionais, do *framework* de Brennan e Resnick.

Figura 18- Inicialização da variável Pontuação com o valor zero.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Figura 19- Exemplo de incremento da variável Pontuação.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Ao avaliar essa tarefa, o estudante **E1** enfatizou que *"[...] a tarefa é um pouco complexa, em uma nota seria 7/10"*. A complexidade desafiou o participante a buscar outras formas de resolver a tarefa. Nesse sentido, ele destacou que "*aprendi a utilizar outros recursos. Tive dificuldades em fazer funcionar algumas coisas, mas como dá pra ir testando os comandos e se não dá certo é só trocar por outro e ver o que acontece, daí acabei achando uma solução"* (**E1**). Essas considerações do participante nos permitem concluir que, apesar dos desafios e dificuldades encontrados para resolver a tarefa, o *Scratch* indicou ser um eficiente instrumento mediador na construção da solução, especialmente por possibilitar testar de maneira simples e dinâmica os comandos e suas consequências na execução do projeto.

Sobre as dificuldades e estratégias utilizadas para a realização da tarefa, o estudante **E1** afirmou que "*Tive dificuldades em fazer os atores se movimentarem no*

labirinto, os flocos sumirem ao serem tocados. Optei por fazer direto no Scratch, assisti um vídeo no Youtube para poder entender os códigos utilizados para montar o labirinto". Essa declaração traz indícios do percurso percorrido pelo participante para desenvolver seu projeto 3 e quais foram as principais dificuldades encontradas.

Para realizar esse projeto o estudante manteve as mesmas estratégias empregadas no projeto 2, buscando conhecer comandos em vídeos disponíveis na internet. Além disso, ao afirmar que "*montei na cabeça os passos que teria que seguir e fui resolvendo a tarefa*" (**E1**), indica que implementou suas ideias diretamente no *Scratch*, dispensando o planejamento prévio no papel ou o emprego de outro recurso. As características do ambiente, como simplicidade de uso, interface intuitiva e amigável, e interação com o usuário, podem ter sido determinantes para que participante montasse seu projeto diretamente no ambiente.

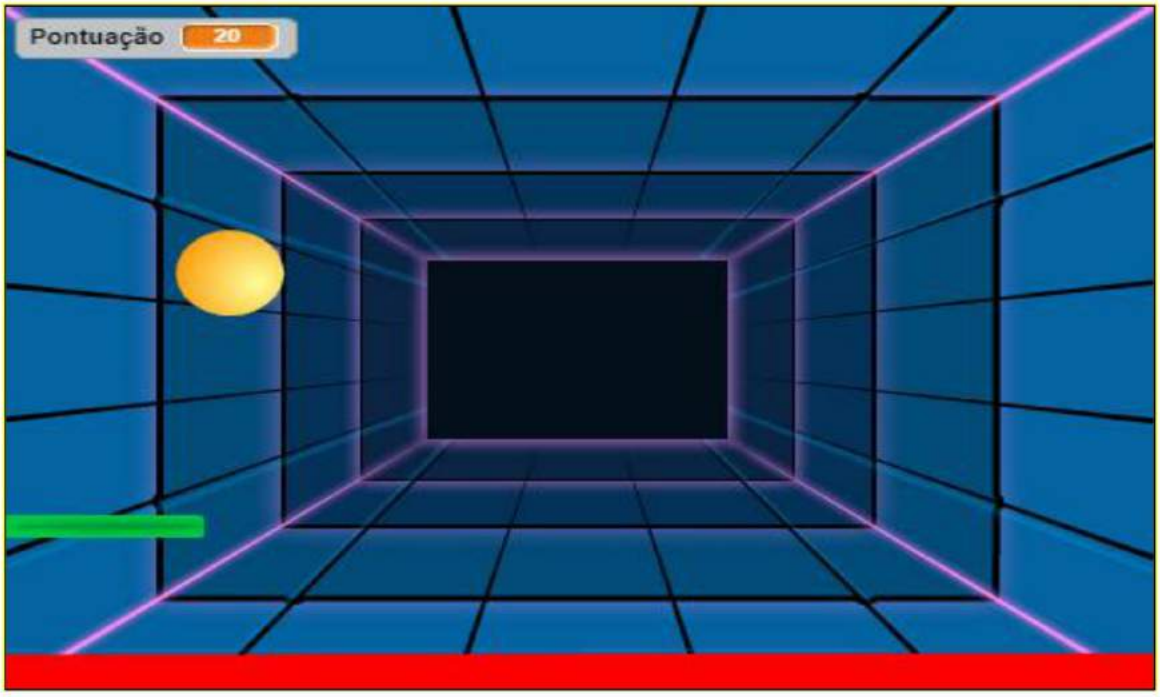
d) **Projeto 4**

Para o projeto final, de tema livre, o estudante buscou inspiração em outros trabalhos compartilhados na comunidade do *Scratch*. O estudante optou por criar um jogo no qual o usuário controla uma barra que se desloca apenas horizontalmente, a fim de rebater uma bola que se movimenta de maneira aleatória, conforme ilustra a Figura 20. Cabe ao jogador impedir que a bola toque na parte inferior do cenário, o que ocasiona o final do jogo. A cada rebatida, a pontuação é incrementada e ao ultrapassar os 100 pontos ocorre uma mudança de nível. O mesmo ocorre após os 300 pontos e, ao ultrapassar os 500 pontos, encerra-se o jogo.

De acordo com o estudante, sua inspiração para a criação desse projeto "*foram alguns jogos que encontrei na comunidade Scratch e tive vontade de desenvolver um também e a escolha do tema, foi um estilo de jogo que é 'simples' mas que diverte bastante*" (**E1**). Após a escolha da temática, o participante afirmou que "*criei o projeto do zero, mas com ideias de projetos já existentes e utilizando as dicas do professor*" (**E1**).

Nesse projeto o estudante precisou lidar com diversas variáveis e condições para alcançar seus objetivos. Foi necessário combinar controles lógicos e relacionais, algumas vezes em nível complexo (diversas relações simultaneamente).

Figura 20- Projeto **4** do estudante E1.



Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 16 pontos e, com isso, considerou que o programa atingiu o terceiro e último nível, denominado "proficiente". As maiores pontuações foram nos conceitos de lógica, paralelismo e sincronização, nos quais atingiu a pontuação máxima.

Ao analisar o programa levando em consideração o problema proposto, podemos concluir que o estudante criou um projeto coerente com o que foi indicado e evidenciou um avançou de nível, especialmente no emprego de pensamento lógico. A Figura 21 apresenta a utilização de operadores lógicos e relacionais para montar o teste do comando "se". Na Figura 22 observamos o emprego do pensamento lógico condicional através da utilização do comando "se–então–senão", combinado ao uso de expressões lógicas. Esses operadores fazem parte da dimensão conceitos computacionais, do *framework* de Brennan e Resnick. A Figura 22 também exemplifica o uso de números aleatórios para indicar a direção do deslocamento da bola pelo cenário.

Figura 21- Exemplo de utilização de operadores lógicos e relacionais.

Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

Figura 22- Exemplo de pensamento lógico condicional (se–então–senão) em conjunto com expressões lógicas.

Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

As principais dificuldades encontradas na construção desse projeto foram, segundo o estudante, "*em alguns comandos, para que a bolinha não passasse direto pela plataforma e na contagem de pontos. Também tive dificuldade na parte de sensores, de utilizar as coordenadas x e y e nos operadores*" (**E1**). Para resolver alguns desses obstáculos, o participante buscou auxílio do professor, o qual buscou mediar essas questões, dando dicas e fazendo o participante pensar no que estava ocorrendo, o que estava falhando e propondo testes para verificar as hipóteses levantadas.

O jogo apresenta algumas falhas, como, por exemplo, em relação à utilização de som. O estudante inseriu um som no projeto, porém não o executa em nenhum momento. Esse som deveria ser reproduzido cada vez que a bola tocasse na barra deslizante. Caberiam alguns ajustes e melhorias, porém levando em consideração o tempo que o participante teve para desenvolver essa atividade, bem como a curta experiência com programação, podemos considerar que esse projeto ficou coerente com a pontuação atribuída pelo *Dr. Scratch*.

Articulando a análise dos projetos do participante E1

O participante E1 foi o único a estar presente nos sete encontros da oficina, o que pode ter contribuído positivamente para seu desempenho na realização das atividades, pois foi o participante que atingiu a maior pontuação em um projeto, conforme análise do *Dr. Scratch*. No último trabalho atingiu o nível "proficiente", totalizando 16 pontos. A análise dos projetos desenvolvidos por esse estudante evidenciou um crescimento na apropriação dos comandos do *Scratch* além de um aprofundamento na utilização da lógica. Percebemos um aumento gradual na complexidade dos projetos por ele desenvolvidos.

No questionário respondido após a conclusão da oficina, esse estudante afirmou ter utilizado, por diversas vezes, o *Scratch* fora do horário das aulas, evidenciando seu interesse pela ferramenta, o desejo de aprofundar seus conhecimentos e resolver as atividades propostas. No início da oficina, o estudante relatou em seu caderno de reflexões sua percepção sobre o *Scratch*: *"achei um pouco complicado, ainda não consigo utilizar muito bem, porém é bem divertido. Adorei!"* (**E1**).

Essa afirmação evidencia o caráter lúdico e atrativo dessa ferramenta, propiciado pelo uso de uma linguagem visual, e, como também destaca Martins (2012, p. 35), é um ambiente gráfico de programação que objetiva ser diferente dos demais por ser mais simples, mais fácil de usar e mais intuitivo. Ainda, segundo o autor:

> O *Scratch* é voltado ao usuário infantil e jovem. Oferece-lhe uma linguagem de programação simples, em que é possível criar projetos que lhe auxiliam a aprender e a desenvolver suas habilidades matemáticas e computacionais, de modo a complementar e enriquecer seu pensamento criativo e a lhe ensinar a trabalhar de maneira colaborativa (MARTINS, 2012, p. 44).

Esse trabalho colaborativo ocorre quando um projeto é compartilhado na comunidade do *Scratch*, ficando disponível a qualquer outro usuário. A partir disso é possível executar, visualizar o código, reutilizar e adequar o projeto às necessidades do usuário. Também existe a possibilidade de trabalhar de forma colaborativa dentro de um grupo, como, por exemplo, em uma sala de aula. Os estudantes podem compartilhar conhecimentos, códigos e estratégias a fim de resolver um problema ou desenvolver uma tarefa. Ao trabalhar de maneira colaborativa vivenciamos a sociointeração vigotskiana, a qual representa um elemento necessário ao processo de aprendizagem e de desenvolvimento do sujeito.

Um exemplo de interação sociocultural vivenciada pelo estudante E1 na oficina foi com o auxílio dos colegas à inserção de som: "*pedi ajuda dos colegas para adicionar os áudios nos projetos*" (**E1**). Podemos considerar também que houve sociointeração quando esse estudante buscou auxílios, explicações sobre comandos e inspiração no *Scratch* e em sua comunidade, conforme ressaltou: "*destaco como ponto positivo o fato de poder explorar outros projetos realizados e ver as possibilidades de criação na plataforma*" (**E1**).

O participante também salientou como facilitador na exploração do *Scratch* e na realização das tarefas o fato de poder "*ver como foram desenvolvidos outros projetos da comunidade, assim tive uma base do que foi utilizado e de como utilizar alguns comandos*" (**E1**). Essa afirmação nos levou a entender que possivelmente ocorreu mediação por parte da comunidade do *Scratch*, uma vez que ela foi utilizada para buscar conhecer comandos, entender como eles funcionam e depois aplicá-los no projeto que estava desenvolvendo.

A exploração do *Scratch* também trouxe surpresas para o estudante, entre as quais ele ressalta a dinamicidade dessa ferramenta ao afirmar que: "*fica bem mais fácil entender os programas porque dá pra testar enquanto vamos fazendo o programa. Assim a gente consegue conhecer os blocos de comando e vai testando e vendo o que acontece*" (**E1**). Essas afirmações do participante sinalizam características de mediação do ambiente, possibilitando explorar e observar na prática as implicações de cada comando. Além disso, ir testando os programas enquanto eles são criados facilitam a localização e correção dos erros.

Com relação ao *framework* de Brennan e Resnick, podemos destacar que esse participante evoluiu no emprego dos conceitos computacionais no decorrer dos projetos. Além disso, demonstrou a aplicação de práticas computacionais, como teste e depuração, modularização, reuso e reformulação - especialmente ao buscar, na comunidade do *Scratch* e em vídeos, ideias e comandos para empregar em seus projetos. Seus questionamentos, sua forma de interação com o ambiente, com o professor e demais colegas também trazem indícios do emprego de perspectivas computacionais.

4.3.2.2 Análise dos projetos do participante E2

a) **Projeto 1**

O primeiro projeto do participante E2 objetivou uma exploração inicial do *Scratch* por parte do estudante, no qual um dragão se movimenta de maneira circular ao redor do planeta, conforme ilustrado na Figura 23. Para isso empregou dois atores e alguns poucos comandos, destacando-se a utilização de laço de repetição para completar a tarefa. Apesar de utilizar poucos comandos, o programa poderia ser simplificado, pois o participante empregou uma instrução que não desempenha nenhuma função nesse cenário, como ilustrado na Figura 24. Ali observamos a utilização do comando "próxima fantasia" sendo que o ator só possui uma fantasia, ou seja, não há o que ser alterado.

Figura 23- Projeto 1 do estudante E2.

Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

Figura 24- Exemplo de emprego de comando desnecessário.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

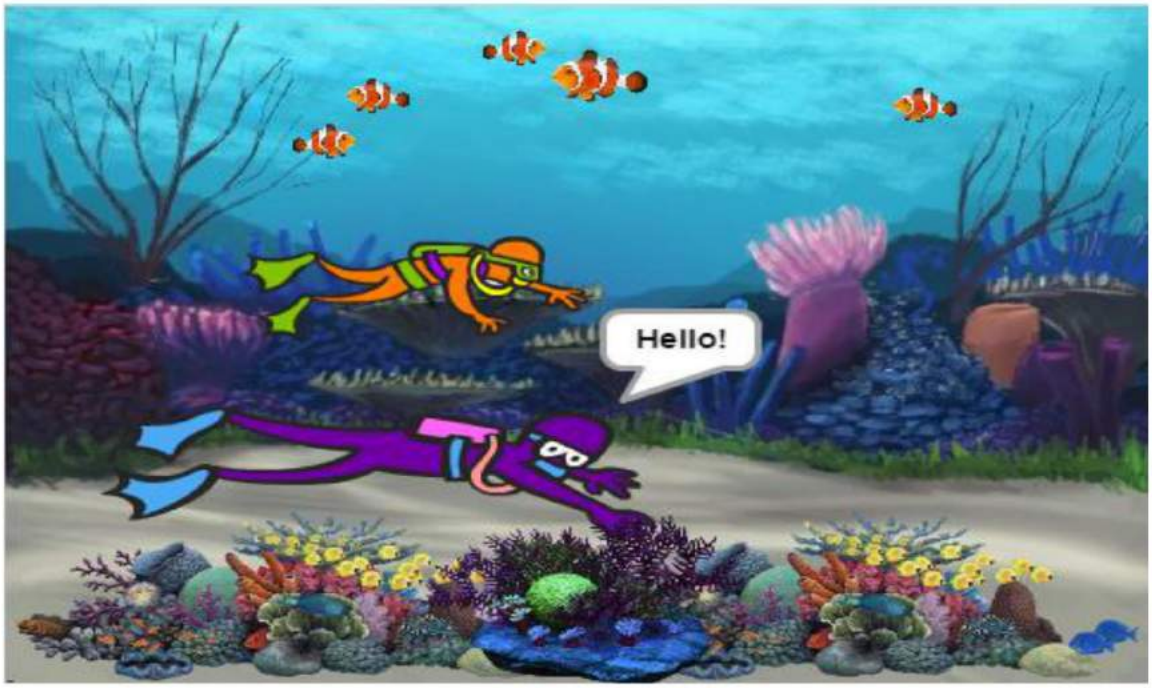
Cabe ressaltar que esse estudante não veio na primeira aula, portanto teve menos tempo para explorar o *Scratch* e desenvolver seu projeto se comparado com os demais participantes. Após o primeiro contato com o *Scratch,* ele destacou que o achou *"divertido e desafiador pois precisamos usar a lógica para desenvolver os comandos da programação"* (**E2**). Essa afirmação indica que esse participante não está familiarizado com o pensamento lógico e seu emprego na construção de soluções para problemas apresentados, o que pode ter refletido nos resultados do relatório gerado pelo *Dr. Scratch*.

A análise feita pela ferramenta *Dr. Scratch* atribuiu uma pontuação de 5 pontos, em uma escala que vai até 21 e considerou que o programa ficou no primeiro nível, denominado "básico". O controle de fluxo foi o conceito com maior pontuação, atingindo 2 pontos. Por outro lado, lógica, paralelismo e abstração não pontuaram.

b) **Projeto 2**

No projeto 2 foi proposto a criação de pelo menos duas formas geométricas. O estudante porém optou por continuar a exploração do *Scratch*. O projeto desenvolvido foi a simulação de um mergulho no mar, com dois mergulhadores e alguns peixes se movimentando pelo cenário que representa o fundo do mar, conforme ilustrado pela Figura 25. Para desenvolver essa tarefa o estudante utilizou nove atores, alguns estáticos (fixos) e outros em movimento. O programa executa sempre as mesmas ações, ou seja, o usuário não controla os atores. Um dos mergulhadores possui um balão de diálogo que fica visível o tempo inteiro com a mesma mensagem.

Figura 25- Projeto 2 do estudante E2.



Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 7 pontos, considerando que o programa atingiu apenas o nível "básico". Nesse projeto apenas o conceito de lógica não pontuou e o, controle de fluxo, novamente foi o conceito com maior pontuação, atingindo 2 pontos.

Ao compararmos com o primeiro projeto, a avaliação feita pelo *Dr. Scratch* mostra um pequeno aumento na pontuação, resultando na permanência no primeiro nível. Ao analisarmos o programa levando em consideração a proposta do estudante para essa atividade, concluímos que ele não conseguiu concluir seu projeto, sendo que algumas ações ficaram incorretas ou incompletas. O participante mostrou pouca evolução na exploração de outros recursos do *Scratch,* apenas avançou no paralelismo, com diversos atores se movimentando simultaneamente.

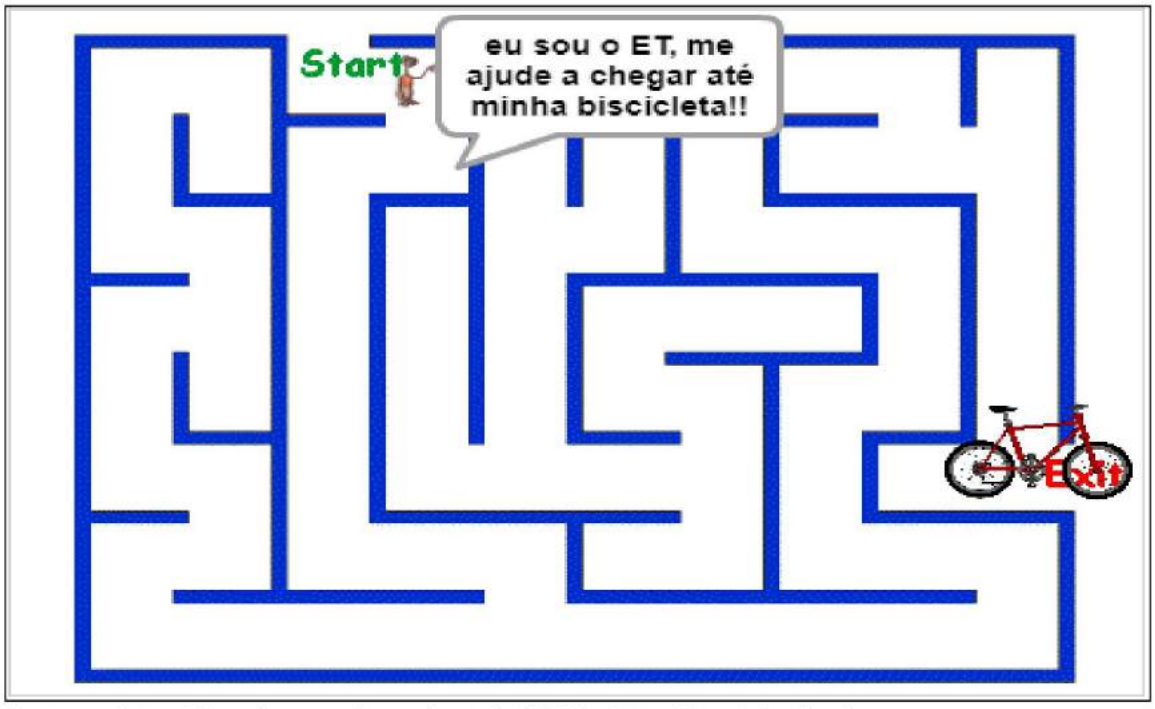
c) **Projeto 3**

O projeto 3 foi desenvolvido em duas fases. Na primeira, foi proposta a criação de um labirinto no qual um ator deveria ter um ponto de partida e outro de chegada. O usuário deveria controlar o deslocamento desse ator e o programa não poderia

permiti-lo ultrapassar as linhas divisórias do labirinto. Na segunda fase, foi sugerido que os estudantes aperfeiçoassem o projeto, tornando-o mais atrativo e controlando a pontuação.

Para desenvolver essa tarefa o estudante utilizou dois atores. Um, simbolizando um E.T., era controlado pelo usuário e deveria chegar até o outro ator - bicicleta, o qual estava posicionado no ponto de chegada, conforme ilustrado pela Figura 26. Ao alcançar a bicicleta, ocorre uma mudança no plano de fundo e é exibida uma mensagem indicando o final do jogo.

Figura 26- Projeto 3 do estudante E2.



Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu 8 pontos, dos quais a interação com o usuário foi o que mais pontuou, atingindo 2 pontos. Os demais conceitos obtiveram todos 1 ponto. Com essa pontuação total, o projeto atingiu o segundo nível, ou seja, "em desenvolvimento".

Ao analisarmos o projeto levando em consideração o problema proposto, podemos concluir que o estudante conseguiu apenas resolver o problema sugerido para a primeira fase, pois não foi implementado um controle de pontuação. Com isso

o estudante deixou de explorar e empregar as operações com variáveis, as quais fazem parte da representação de dados. Outro aspecto que poderia ter sido explorado é a criação de blocos (segundo nível do conceito abstração e decomposição de problemas), possibilitando com isso a redução de códigos. Observamos que determinadas sequências de comandos, como a ilustrada na Figura 27, são utilizadas em mais de um local, podendo assim ser substituídas por um novo bloco. Os comandos presentes nessa figura, por exemplo, estão repetidos em três locais diferentes do programa.

Figura 27- Exemplo de emprego de comando desnecessário.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

De acordo com o participante, suas estratégias empregadas para a resolução dessa tarefa foram: "*vi algumas ideias no Youtube para poder realizar a tarefa, tive muita dificuldade em achar os comandos certos para guiar o ator até o final do labirinto*" (**E2**). Ele também destaca que "*antes desta atividade não tinha muita noção de como eram realizados os jogos que utilizam o labirinto, achei muito interessante. Avalio como sendo uma atividade difícil pois temos que usar a lógica para criar os comandos certos*" (**E2**). Nessa colocação, percebemos que o estudante teve dificuldades em montar uma sequência lógica de passos para resolver essa tarefa, ou seja, criar um algoritmo para o problema.

Um algoritmo "é qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como entrada e produz algum valor ou conjunto de valores como saída" (CORMEN et al., 2002, p. 3). Nesse sentido, um algoritmo pode ser definido como uma sequência de passos computacionais que recebem uma entrada e a transformam em uma saída. Montar essa sequência de passos de forma lógica e ordenada não é tarefa trivial para quem nunca teve contato com linguagens de programação, requer prática e demanda tempo para desenvolver habilidades que possibilitem elaborar uma solução simples, correta e eficiente.

d) **Projeto 4**

Para o projeto final, de tema livre, esse estudante buscou inspiração em jogos que já conhecia, bem como explorou os projetos compartilhados na comunidade do *Scratch*. Optou por criar um jogo no qual o usuário controla um ator que se movimenta pelo cenário, desviando de obstáculos, conforme ilustrado na Figura 28. Ao atingir um determinado ator, o jogo muda de fase. No segundo estágio, há um ator escondido no cenário, que é a chave para chegar à terceira e última fase, a qual finaliza quando o ator alcança determinado ponto do cenário.

Figura 28- Projeto 4 do estudante E2.

Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 12 pontos e, com isso, considerou que o programa atingiu o segundo nível, denominado "em desenvolvimento". Todos os conceitos pontuaram, sendo que o paralelismo e a sincronização atingiram a pontuação máxima, com 3 pontos.

Ao analisarmos o programa levando em consideração a proposta do estudante, podemos concluir que ele conseguiu criar um projeto coerente com sua

proposição, indicando ter avançado em diversos aspectos como a sincronização de vários objetos e a integração de recursos multimídia. O jogo apresenta algumas falhas, como por exemplo, em relação à utilização da variável denominada "*error*", a qual fica visível ao usuário, conforme ilustrada na Figura 29, porém nunca é modificada, permanecendo sempre com o valor zero.

Figura 29- Emprego de variável visível ao usuário, porém nunca utilizada.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Caberiam ajustes e melhorias no código, inserindo também um controle de pontuação. Porém, levando em consideração o tempo que o estudante teve para desenvolvê-lo, bem como a curta experiência com programação, podemos considerar que esse projeto ficou coerente com a pontuação atribuída pelo *Dr. Scratch*.

O estudante afirmou, em se caderno de reflexões, que sua inspiração para a escolha desse projeto final foi que "*sempre gostei do jogo Super Mário e vi no projeto final uma oportunidade de tentar programar um jogo similar ao que eu costumava jogava e tentar entender mais ou menos como funcionou o processo de criação do mesmo*" (**E2**). Com essa declaração, o participante ressalta a possibilidade de criar seu próprio jogo, de concretizar seus desejos, sua imaginação e compreender como são criados os programas, assim como Resnick (2006) destacou ser os princípios norteadores do *Scratch*.

O participante 2 também afirmou que construção do projeto 4 partiu do zero, ou seja, não readequou ou reutilizou algum projeto pré-existente. Na realização dessa tarefa enfrentou alguns obstáculos, dentre os quais destacou a "*dificuldade na hora de dar os comandos para o ator e trocar de plano de fundo, a ideia inicial era de que na hora em que ele se chocasse contra um objeto o jogo reiniciaria ou um plano de fundo escrito 'game over' apareceria na tela*" (**E2**). Apesar das dificuldades relatadas, o estudante mostrou persistência e determinação para elaborar esse projeto e, com isso, conseguiu montar o jogo desejado.

Articulando a análise dos projetos do participante E2

O participante E2 só conseguiu participar de quatro encontros da oficina, o que pode ter causado algumas das dificuldades apontadas pelo estudante no Projeto 3, além dos dois primeiros projetos terem permanecido no "nível básico", segundo avaliação do *Dr. Scratch*. Por outro lado, ao analisarmos seus projetos, seu caderno de reflexões e as anotações do pesquisador, percebemos um empenho do estudante em desenvolver as tarefas propostas, além de um avanço no conhecimento dos comandos da ferramenta e sua evolução no aprendizado de programação.

As anotações do pesquisador sobre a oficina demonstram o interesse e participação desse estudante e sua motivação em explorar o *Scratch*. Nessas anotações também estão registrados os motivos pelos quais o estudante não pode comparecer a algumas aulas e seu descontentamento com esse fato. Seu desempenho na oficina pode ter relação direta com a utilização do *Scratch* fora do horário de aula, conforme o participante relatou no questionário pós-oficina[40] e também em seu caderno de reflexões, quando declarou que *"acho que utilizar o Scratch fora do horário da disciplina possibilitou explorar melhor os comandos e repensar com mais calma a programação dos projetos que eu havia realizado"* (**E2**).

Além do mais, tais anotações revelaram que os participantes E1 e E2 interagiram em vários momentos, trocando ideias e fazendo comentários. O estudante E1, que participou de todas as aulas, auxiliou o participante E2 por diversas vezes na utilização de comandos e correção de erros. Em seu caderno de reflexões, o estudante E2 explicita essa ajuda quando diz que: "*me recordo que no início tive algumas dificuldades com a programação e obtive ajuda de minha colega, acredito que com a ajuda dela descobri alguns comandos que ainda não havia experimentado*" (**E2**).

A exploração do *Scratch* foi feita, segundo o estudante E2, diretamente na ferramenta, sem um planejamento prévio por escrito: *"Não utilizei o papel, fui vendo quais comandos funcionavam e se não, descartava e pensava em outros para comandar o ator. É bem fácil trocar os comandos e dá pra ir vendo se tá funcionado na mesma hora"* (**E2**). Tal afirmação fornece indicativos sobre o papel mediador do

[40] Conforme resposta da questão número 8 do questionário: https://goo.gl/forms/Yt6YXQisImPzYJQJ3

Scratch, pois ao desenvolver o programa, o usuário consegue visualizar a reação que cada comando causa na sua execução, possibilitando sua troca ou ajuste. Esse papel mediador também é destacado por Vigotski (2007) quando afirma que é através dos elementos (instrumentos e signos) e do processo de mediação que ocorre o desenvolvimento dos processos psicológicos superiores ou cognição.

Ao avaliar o *Scratch,* o participante ressalta, em seu caderno de reflexões, alguns pontos positivos e negativos do ambiente. Segundo ele, "*os pontos negativos são que muitos os comandos que ele oferece daí em alguns momentos fiquei em dúvida em quando utilizá-los na programação. As dicas que ele oferece auxiliam, porém em nem todos os casos*" (**E2**). Com essas colocações podemos concluir que, para iniciar a utilização do ambiente e desenvolver os primeiros projetos, seria mais adequado simplificar o número de comandos disponíveis, conduzindo o usuário a focar em apenas alguns elementos e, com isso, possibilitando um aprendizado gradual do *Scratch*. Ao afirmar que as dicas que o ambiente fornece nem sempre auxiliam ou são suficientes, percebemos que nesses momentos faz-se importante a mediação do professor e a colaboração dos colegas.

Já, sobre os pontos positivos, o participante E2 afirma que esses "*são a grande variedade de jogos que estão disponíveis para jogar ou se inspirar na hora de programação, a facilidade de colocar e mudar a ordem dos comando e também as muitas possibilidades de plano de fundo e de atores disponíveis*" (**E2**). Com base nessa declaração, podemos ressaltar que os recursos multimídia que integram esse ambiente o tornam atrativo e motivador, facilitando a elaboração dos projetos. Além disso, há a possibilidade de visualizar e testar tudo o que é disponibilizado na comunidade do *Scratch* na internet. É permitido também visualizar o código que foi empregado nesses trabalhos e utilizá-los em novos projetos, fazendo as devidas modificações e readequações.

O estudante também afirmou em seu caderno de reflexões que para desenvolver as atividades "*pesquisava alguns exemplos e via algumas aulas de programação no youtube para ter uma inspiração, e se eu não conseguisse realizar algum comando ou sentisse dificuldade perguntava ao professor e ele me orientava*" (**E2**). Essa afirmação reforça a importância do papel do professor, atuando como mediador, orientando e guiando os participantes da oficina. Com isso, buscamos

proporcionar aos estudantes meios para uma possível internalização de suas vivências com a programação de computadores. Essa internalização, segundo Vigotski (2007, p. 56), é "a reconstrução interna de uma operação externa".

Com base no *framework* de Brennan e Resnick, podemos destacar o emprego bem sucedido de alguns conceitos computacionais, porém necessitando aprofundar o estudo e exploração de outros conceitos, como, por exemplo, dados e operadores. As práticas computacionais também necessitam ser melhor trabalhadas e desenvolvidas. Percebemos que o participante buscou fazer teste e depuração, porém nem sempre conseguiu solucionar os problemas encontrados. Percebemos o emprego de reuso e reformulação nos dois últimos projetos, tanto na questão de imagens como na questão de ideias e comandos.

4.3.2.3 Análise dos projetos do participante E3

a) **Projeto 1**

Para esse primeiro projeto de caráter exploratório, o participante E3 montou o cenário ilustrado na Figura 30, solicitando que o usuário conduza o gatinho até sua casinha. Ao chegar no destino, o animal vira uma bola de fogo e é reproduzida uma frase de Fausto Silva[41], dizendo: "tá pegando fogo, bicho!".

[41] Fausto Corrêa da Silva, mais conhecido como Faustão, é um apresentador brasileiro que atualmente apresenta o programa de auditório Domingão do Faustão, da Rede Globo.

Figura 30- Projeto 1 do estudante E3.

Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

Nesse projeto o estudante utilizou poucos recursos. Destaca-se a inserção e edição de som, a utilização de testes condicionais, conforme exemplificado na Figura 31 com o uso do comando "se–então", além da troca de fantasias para completar a tarefa. Para esse primeiro projeto, chamou a atenção a sincronização dos recursos multimídias empregados. Em seu caderno de reflexões, o participante relata sua percepção da ferramenta, afirmando que *"achei o Scratch muito divertido, é uma maneira muito fácil de programar algo"* (**E3**).

Figura 31- Utilização de pensamento lógico condicional.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 6 pontos e considerou que o programa ficou no nível "básico". Não pontuou os conceitos de

paralelismo e sincronização, entretanto no conceito interatividade com o usuário atingiu 2 pontos. Nos demais, ficou com 1 ponto em cada.

Ao analisarmos o projeto levando em consideração o que foi proposto para a tarefa e o que participante E3 buscou criar, podemos concluir que o estudante implementou o que seu imaginário sugeriu, resultando em um projeto simples, exploratório, sem muitos comandos, porém coerente com sua proposta e da tarefa.

b) **Projeto 2**

O estudante desenvolveu o projeto 2 com o objetivo de avançar na exploração do *Scratch*. Criou o programa visando permitir que o usuário ouvisse o som de dois animais. No primeiro cenário elaborado era exibido um botão iniciar, o qual conduzia ao segundo cenário, ilustrado na Figura 32, onde estavam os dois animais: foca e lontra.

Figura 32- Projeto 2 do estudante E3.

Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 11 pontos e, com isso, considerou que o programa atingiu o nível "em desenvolvimento", atribuindo

pontuação máxima, 3 pontos, ao paralelismo e à sincronização. Não pontuou apenas o conceito de lógica.

Ao compararmos com o primeiro projeto, a avaliação feita pelo *Dr. Scratch* mostra um grande aumento na pontuação, resultando na mudança para o segundo nível. Já, ao analisarmos o programa levando em consideração a proposta do participante para essa atividade, concluímos que o estudante atingiu de forma básica seus objetivos, mostrando avanço na utilização de alguns recursos. Entretanto, ficou incompleto e apresentou objetos e códigos que não são utilizados para nada, também denominado de "código morto", conforme ilustrado na Figura 33.

Figura 33- Exemplo de "código morto".

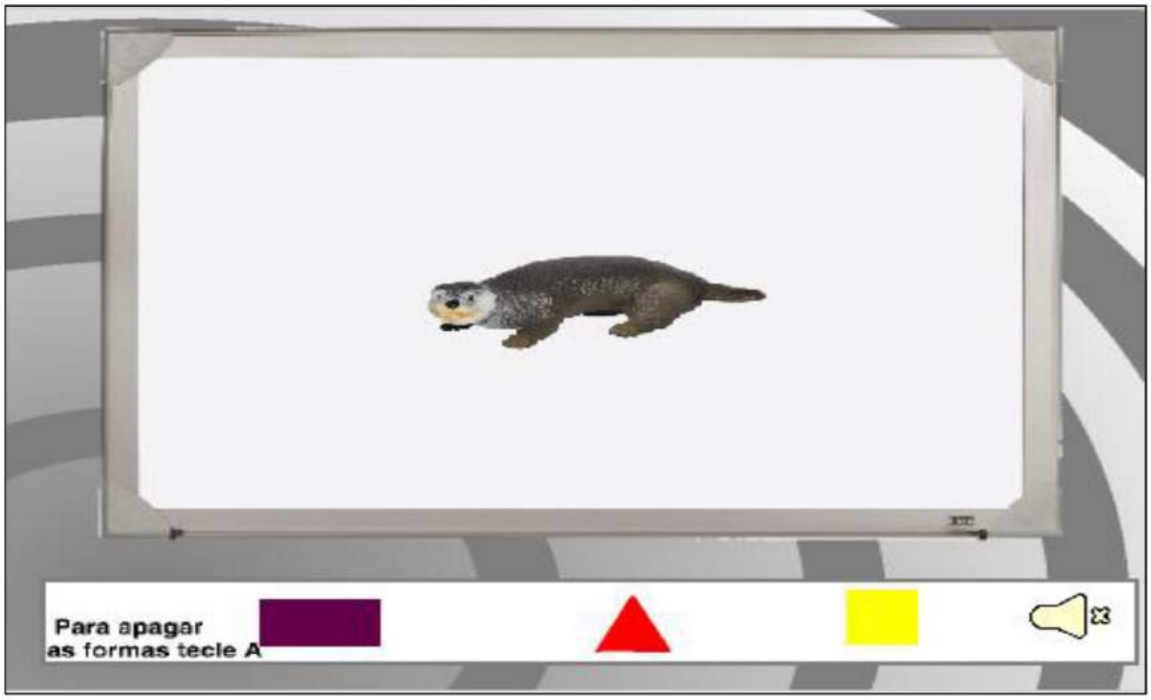
Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

Além disso, o participante poderia ter incrementado mais o projeto, principalmente informando ao usuário de que ele deveria pressionar determinada tecla para reiniciar o jogo. Concluímos, com isso, que apesar do *Dr. Scratch* ter atribuído uma boa pontuação ao projeto, ele poderia ter sido melhor desenvolvido.

c) **Projeto 3**

No projeto 3 foi proposto a criação de, pelo menos, duas formas geométricas. O estudante montou um projeto no qual o usuário pode escolher três formas geométricas para desenhar, conforme ilustrado na Figura 34. Ao clicar na forma geométrica, ela é desenhada na tela por uma caneta que assume o formato da figura escolhida, cada qual com uma cor diferente. Além disso, ao iniciar o projeto é executada uma música e, caso o usuário deseje que ela pare, basta clicar em um botão específico da tela.

Figura 34- Projeto 3 do estudante E3.



Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 11 pontos e, com isso, considerou que o programa atingiu o nível "em desenvolvimento", atribuindo pontuação máxima ao paralelismo e à sincronização. O conceito de lógica não pontuou e os demais obtiveram 1 ponto.

Ao compararmos com o projeto anterior, a avaliação feita pelo *Dr. Scratch* mostra que toda a pontuação foi exatamente igual. Entretanto, ao analisarmos o programa levando em consideração a proposta do estudante para ambas as atividades, bem como a programação desenvolvida e os recursos utilizados, podemos afirmar que esse projeto é superior ao anterior. Percebemos uma evolução do estudante, sendo que ele mostra a exploração e utilização de outros recursos do *Scratch,* resultando em um projeto mais aprofundado e detalhado. Faltaram alguns ajustes, principalmente na inicialização de variáveis, mas tendo em vista o pouco contato com a ferramenta e com a programação podemos afirmar que o projeto está coerente com problema proposto.

O participante E3 afirmou em seu caderno de reflexões que montou o projeto diretamente na ferramenta pois, *"acho melhor pensar na ideia e ficar tentando montar*

ela no Scratch, planejar num papel é mais organizado, mas eu prefiro ir direto e tentar" (**E3**). Com base nessa declaração, concluímos que o estudante optou por desenvolver as tarefas diretamente no ambiente pelas facilidades que ele oferece, tanto para testar e reorganizar os comandos, como para visualizar de maneira dinâmica as ações geradas pelos blocos de comandos empregados. Isso revela que o *Scratch* possui recursos e potencial para mediar a elaboração de soluções para os desafios propostos.

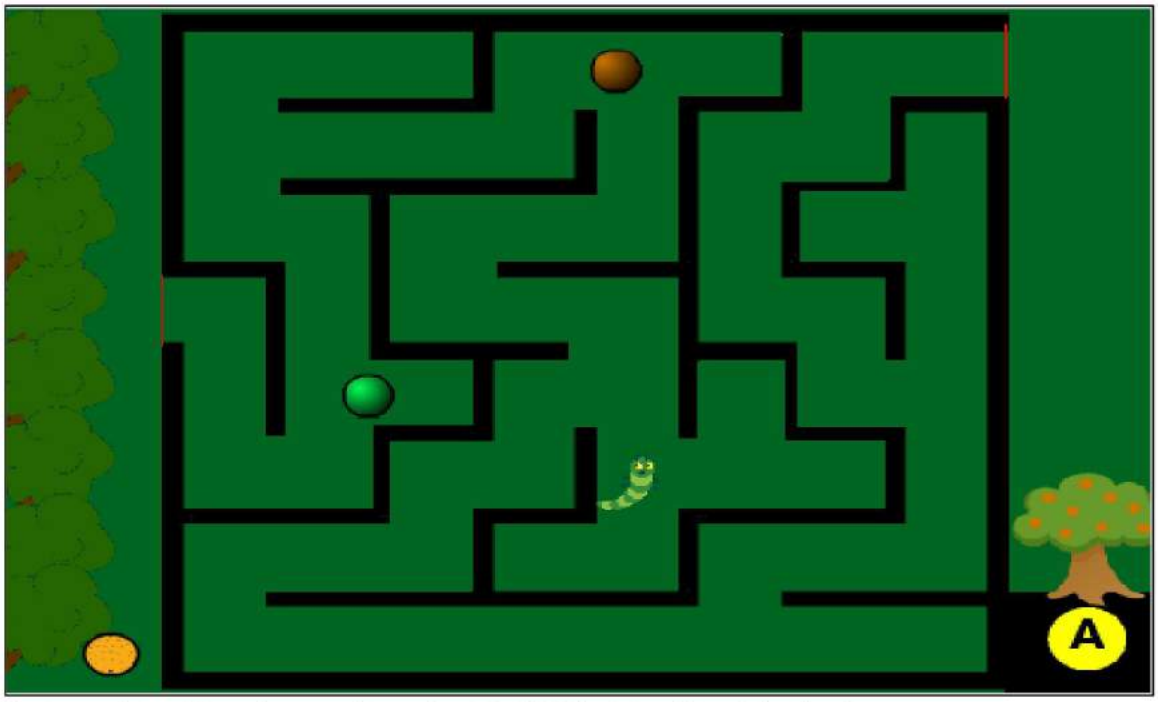
Com relação às estratégias adotadas para realizar a tarefa e as dificuldades enfrentadas, estudante destacou que *"sempre começo pelo movimento e após tento imaginar como ficaria mais agradável. Eu tenho algumas dificuldades com a ordem da movimentação dos personagens"* (**E3**). Essa colocação indica como o participante inicia a estruturação de seus projetos, ressaltando sua preocupação em montar um projeto pensando no usuário, dando-lhe instruções e procurando deixa-lo atrativo e amigável.

d) **Projeto 4**

O projeto 4 foi desenvolvido em duas fases. Na primeira, foi proposta a criação de um labirinto no qual um ator deveria ter um ponto de partida e outro de chegada. O usuário deveria controlar o deslocamento desse ator e o programa não poderia permiti-lo que ultrapassasse as linhas divisórias do labirinto. Na segunda, foi proposto que os estudantes aperfeiçoassem o projeto, tornando-o mais atrativo e controlando a pontuação.

Para desenvolver essa tarefa o participante utilizou diversos atores, conforme ilustra a Figura 35. O ator principal é uma laranja, que deve percorrer o labirinto até alcançar a árvore presente do outro lado do labirinto. Ao alcançar a árvore, ocorre uma mudança no plano de fundo, é exibida uma mensagem indicando a vitória e informa o final do jogo. Além disso, oferece a opção de jogar novamente.

Figura 35- Projeto 4 do estudante E3.

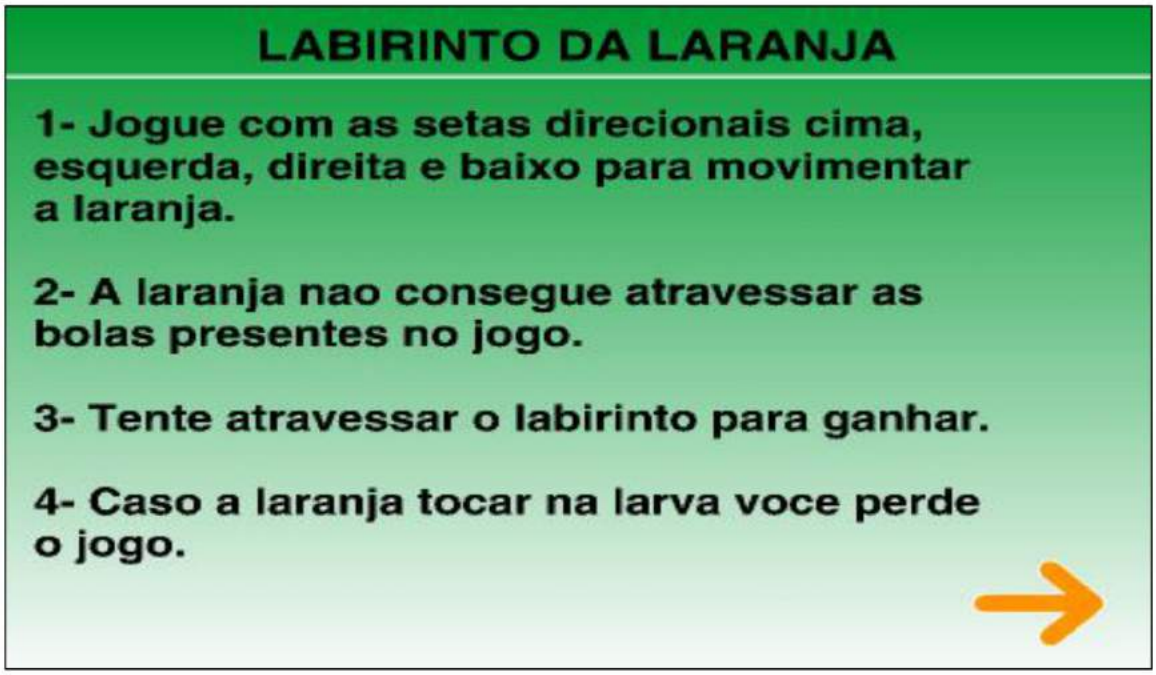


Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* conferiu uma pontuação de 13 pontos, considerando que o programa atingiu o nível "em desenvolvimento", atribuindo novamente pontuação máxima ao paralelismo e à sincronização. A interatividade com o usuário atingiu 2 pontos e todos os demais conceitos obtiveram 1 ponto.

Ao analisarmos o projeto levando em consideração o problema proposto e o código implementado, podemos concluir que o estudante criou um projeto coerente com o problema indicado para ambas as fases, faltando apenas o placar de pontuação. Destacamos a criação de diversos controles e melhorias que tornaram o jogo mais atrativo e amigável. Salientamos a inserção, no início do jogo, de uma tela com as instruções de como jogar, a fim de esclarecer ao usuário as regras e orientá-lo, conforme ilustrado na Figura 36. Esses controles e melhorias são indícios do emprego de práticas computacionais, presentes no *framework* de Brennan e Resnick.

Figura 36- Regras e instruções do projeto 4.



Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

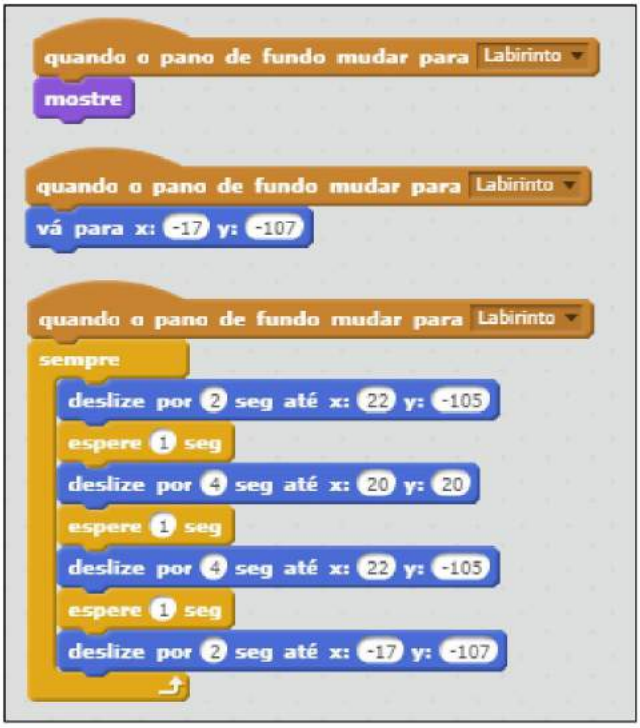
O participante também colocou alguns obstáculos que permanecem em movimento pelo labirinto e adicionou um ator que, ao ser tocado, ocasiona a perda do jogo. Outro detalhe utilizado foi marcar o caminho que é percorrido pela laranja, ou seja, é desenhada uma linha por onde a laranja passa.

Caberiam alguns ajustes e melhorias, principalmente quanto a inicialização das variáveis, que não é executada pelo tradicional botão de início do jogo, mas sim é necessário pressionar uma tecla, a qual não é informada ao usuário – somente quem analisar o código vai descobrir. Outro aspecto que poderia ter sido explorado, mas acreditamos que não houve tempo suficiente, é a criação de blocos, o que possibilitaria a redução de códigos. Observamos que determinadas sequências de comandos são utilizadas em mais de um local, podendo assim ser substituídas por um novo bloco.

Identificamos também a utilização desnecessária de eventos, conforme mostrado na Figura 37. Nela são utilizadas três vezes o evento "quando o plano de fundo mudar para Labirinto" para executar diferentes ações. O ideal é inserir apenas

uma vez cada evento e adicionar de forma sequencial e ordenada os comandos a serem executados.

Figura 37- Exemplo de utilização desnecessária de eventos.



Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Podemos concluir que há evidências que indicam que o participante E3 evoluiu com relação aos projetos anteriores, tanto em conhecimento como no cuidado com os detalhes, mostrando interesse em construir um jogo atrativo e com uma interface amigável. Em seu caderno de reflexões, o participante destacou como aprendizado dessa tarefa que *"ensinou como movimentar um personagem com alguns limites"* (**E3**). Com essa frase o estudante relata quais elementos foram mais significativos no desenvolvimento do projeto 4, onde necessitou controlar o movimento de diversos objetos, cada qual devendo respeitar seus limites.

Ao apresentar novos desafios que exigiram empenho e persistência do participante para a resolução dessa tarefa, construídas através de interações com o *Scratch*, levou-nos a acreditar que houve uma internalização do resultado dessas interações com o ambiente, assim como aponta a teoria vigotskiana. Segundo essa teoria, a partir de uma operação externa do sujeito (a qual pode ser intermediada por

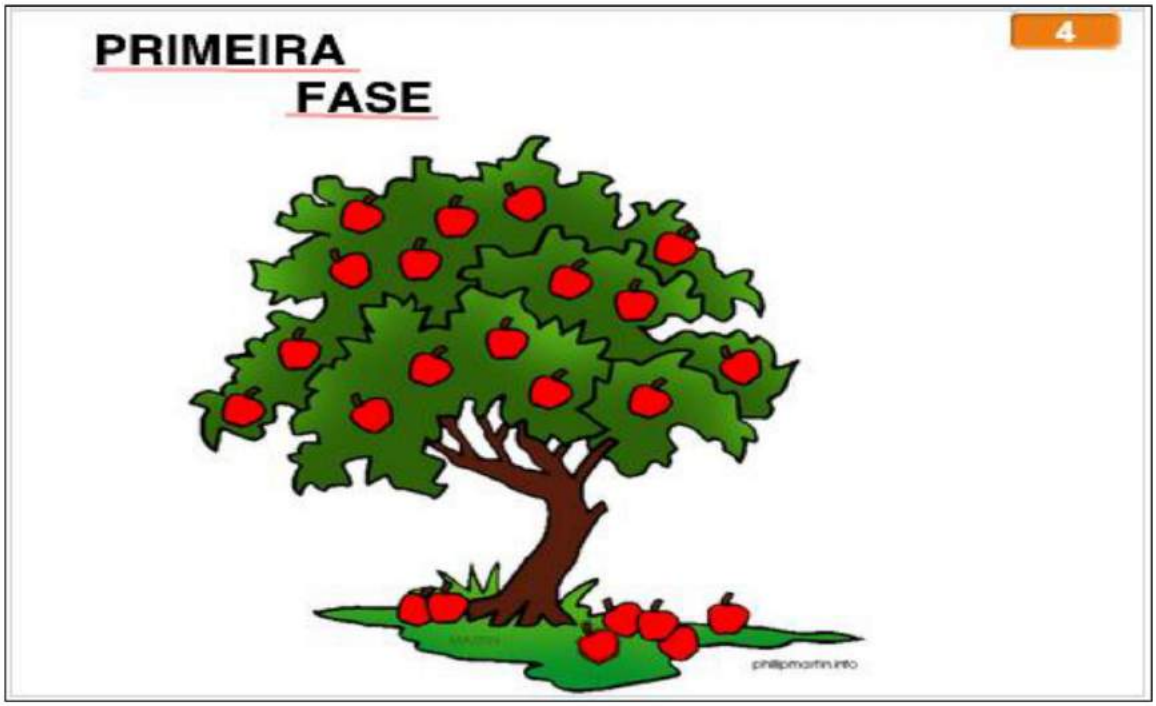
um instrumento) ocorre um processo de reconstrução interna, denominado de internalização, no qual o indivíduo significa seu aprendizado.

O participante E3 também afirmou em seu caderno de reflexões não ter montado um algoritmo ou tentado estruturar uma solução no papel previamente, porém *"pesquisei muito na internet para tentar descobrir uma melhor maneira para a montagem do labirinto e depois fui testando no Scratch. Conforme ia conseguindo avançar eu pensava em como ir melhorando o programa"* (**E3**). Esses apontamentos do participante fornecem subsídios para crermos que o *Scratch* atuou como instrumento mediador, possibilitando ao estudante transformar suas ideias em um programa computacional. Conforme o projeto era desenvolvido, testes e verificações foram executados a fim de experimentar e validar o que estava sendo criado. Outro ponto que chama a atenção é que, conforme esses testes e validações foram avançando, surgiram novas ideias para incrementar e melhorar o programa em construção.

e) **Projeto 5**

Para o projeto final, de tema livre, esse participante buscou inspiração em jogos que já conhecia, bem como explorou os projetos compartilhados na comunidade do *Scratch*. O estudante optou por criar um jogo no qual o usuário precisa encontrar o ator camuflado no cenário. São três fases, cada qual com atores e cenários diferentes. Ao iniciar o jogo duas telas com instruções são exibidas a fim de orientar o usuário sobre como jogar e as respectivas regras. Ao acessar a tela da primeira fase, ilustrada na Figura 38, um cronômetro é disparado, indicando o tempo decorrido. O usuário tem trinta segundos para completar as três fases. Caso não consiga, uma mensagem é exibida e o jogo retorna para o início.

Figura 38- Projeto 5 do estudante E3.



Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 15 pontos e com isso considerou que o programa atingiu o terceiro nível, denominado "proficiente". Atribuiu a pontuação máxima nos conceitos paralelismo e sincronização. Lógica obteve 1 ponto e os demais conceitos obtiveram 2 pontos cada.

Ao analisarmos o programa levando em consideração a proposta do estudante, podemos concluir que ele conseguiu criar, em seu projeto, praticamente tudo o que lhe fora proposto, indicando ter avançado em aspectos como a utilização de variáveis e a criação de blocos. A inserção, no início do jogo, de telas com as instruções de como jogar e suas regras, tornaram o jogo mais amigável e mostraram uma preocupação com o usuário que irá utilizá-lo. Todavia, identificamos também algumas falhas, como por exemplo, em relação à utilização de laço de repetição e com a criação de blocos.

Na Figura 39 podemos observar que o estudante inseriu um laço de repetição para repetir uma única vez um comando, sendo que esse é o procedimento usual ao inserir um comando, ou seja, ele é executado uma única vez. Portanto, retirar o laço

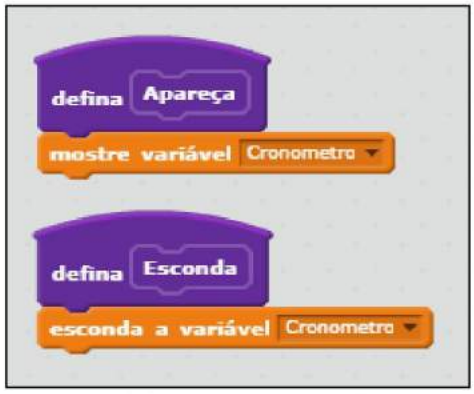
de repetição não causaria nenhuma alteração na execução do programa. Esse erro não foi identificado pelo *Dr. Scratch*.

Figura 39- Exemplo de uso desnecessário de laço de repetição.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Outro problema detectado foi com relação à criação e utilização de novos blocos. Na Figura 40 vemos que o estudante criou os blocos "Apareça" e "Esconda", sendo que já existem comandos com esses mesmos nomes dentro do conjunto de blocos "Aparência" do *Scratch*. Apesar de não ocasionar nenhum erro, de ser permitido criar blocos com o mesmo nome de comandos já existentes, isso pode confundir o programador além de ser nebuloso para quem olhar o código ou quiser modificá-lo.

Figura 40- Exemplo de criação de blocos com nomes de comandos já existentes.

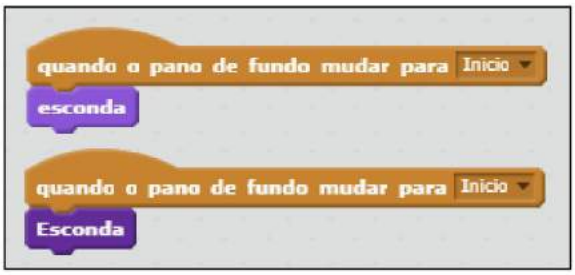


Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Os blocos definidos na Figura 40 só executam um único comando e, por isso, acabam não reduzindo o código, tornando-se descartáveis. Na Figura 41 observamos que o evento de mudança de plano de fundo para "Inicio" ocasiona a execução dos

comandos “esconda” e “Esconda”, com funções distintas. O primeiro comando oculta o ator e o segundo oculta a variável “Cronometro”.

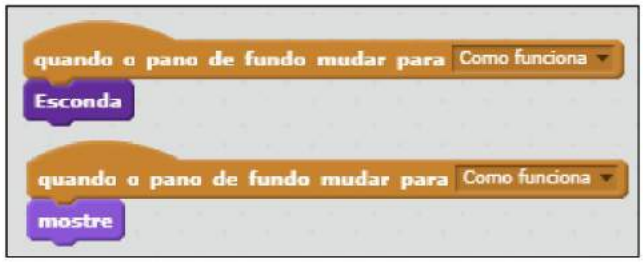
Figura 41- Exemplo de comandos com o mesmo nome, porém com funções diferentes.



Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Tanto na Figura 41, como na Figura 42, podemos observar que o evento mudança de plano de fundo é utilizado duplicadamente, sem necessidade. Bastava utilizar uma só vez e adicionar os dois comandos dentro dele. Essa questão também não é detectada pelo *Dr. Scratch.*

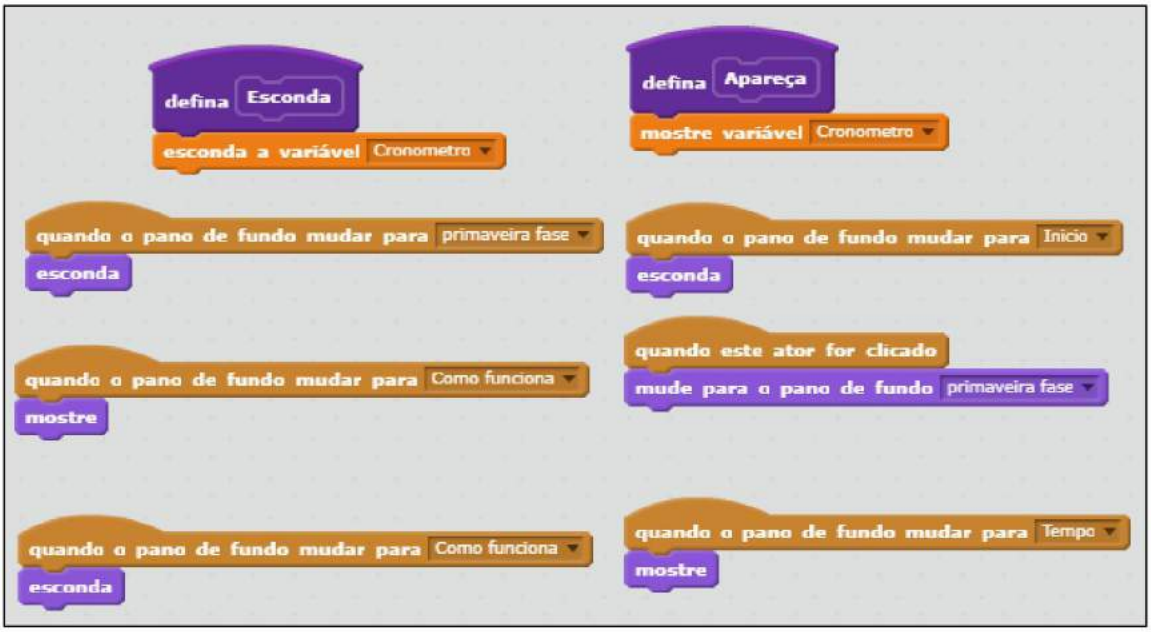
Figura 42- Exemplo de utilização duplicada de evento.



Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

Identificamos também a criação de blocos de comandos que não são utilizados. Na Figura 43 estão representados todos os comandos de determinado ator, dentre os quais estão a criação dos blocos “Esconda” e “Apareça”, porém não são utilizados em nenhum outro local. O *Dr. Scratch* também não detectou essa questão.

Figura 43- Exemplo de definição de blocos não utilizados.



Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise desse programa nos permitiu concluir que o estudante explorou outros comandos e elementos do *Scratch,* porém alguns conceitos não ficaram claros, evidenciando a necessidade de um maior aprofundamento nos estudos. O curto espaço de tempo da oficina pode ter contribuído para isso. O *Scratch* poderia contar com mecanismos que avisassem ou impedissem que alguns desses casos viessem a acontecer. O *Dr. Scratch* também deixa de avaliar alguns aspectos que poderiam contribuir para uma análise mais aprimorada do emprego do pensamento computacional. Todavia, levando em consideração o tempo que o participante teve para desenvolvê-lo, bem como a curta experiência com programação, podemos considerar que esse projeto contemplou o que foi proposto pelo estudante.

Articulando a análise dos projetos do participante E3

O participante E3 só faltou a uma aula, portanto esteve presente em seis encontros da oficina. A avaliação dos projetos feita pelo *Dr. Scratch* mostra uma evolução crescente do estudante, com aumento da pontuação e consequente troca de nível dos projetos, sendo que no último atingiu 15 pontos e foi considerado no nível "proficiente".

No questionário aplicado após a conclusão da oficina, esse estudante afirmou ter utilizado por diversas vezes o *Scratch* fora do horário das aulas, e também ressaltou que *"geralmente o que me motivou era fazer eu me superar a cada aula"* (**E3**). Tais afirmações evidenciam seu interesse pela ferramenta, o desejo de aprofundar seus conhecimentos e resolver as atividades propostas. O resultado desse empenho e motivação ficou registrado nos projetos desenvolvidos, com riqueza de detalhes e com interface pensada no usuário.

Em seu caderno de reflexões, quando questionado sobre suas estratégias para resolver as tarefas e esclarecer as dúvidas, o participante afirmou que *"em relação as dúvidas eu geralmente pesquiso na internet para sanar, em geral elas são de como estruturar a formatação dos códigos e quando não consigo resolver algo, peço dicas pro professor"* (**E3**). Com base nessa colocação, percebemos que o participante procurou pesquisar suas dúvidas, tentou resolver por conta própria e, quando não conseguiu, recorreu ao professor para que este indicasse caminhos. Essa atuação do professor está relacionada ao que postula Vigotski (2007) quando afirma que a interação, a mediação do professor e a atuação na zona de desenvolvimento proximal do aluno são aspectos importantes para a aprendizagem, uma vez que são as relações sociais que despertam processos internos de desenvolvimento.

Ao ser questionado sobre como está sendo trabalhar com a ferramenta, o participante E3 relatou que *"o que faço muito é ir testando direto no Scratch, coloco os comandos e vou vendo como o programa funciona. Algumas vezes é só trocar a ordem dos comandos ou fazer alguma mudança pra dar certo. Acho o Scratch uma ótima ferramenta"* (**E3**). Percebemos, nesse relato, indícios da mediação do *Scratch*, ao contribuir para o participante avançar no conhecimento do ambiente e seus comandos, passando a utilizá-lo na construção de seus projetos. Essa facilidade de utilização e dinamicidade propiciada pelo ambiente pode ter contribuído para o participante reconstruir internamente suas experiências externas, internalizando seu aprendizado de programação, assim como postula a teoria vigotaskiana.

Para Vigotski (2007) a internalização é um processo que causa uma reconstrução interna tendo por base uma operação externa. Já a aprendizagem e o desenvolvimento, segundo o autor, ocorrem por meio das relações entre as pessoas e com o mundo, sendo que para isso "a transmissão racional e intencional da

experiência e do pensamento a outros requer um sistema mediador, cujo protótipo é a fala humana, oriunda da necessidade de intercâmbio durante o trabalho" (VIGOTSKI, 1987, p. 5).

Nesse sentido, observamos na oficina momentos de interação e relacionamento do participante E3 com os demais membros, destacando seu caráter espontâneo e solícito. Auxiliou por diversas vezes os colegas, especialmente o estudante E5, com a inserção e edição de falas e sons nos projetos. Além disso, contribuiu com os participantes E1 e E2, dando dicas e sugestões sobre comandos e estratégias para a construção dos projetos e resolução das tarefas.

Outra característica desse participante que chamou a atenção foi seu entusiasmo em mostrar para os colegas o que estava desenvolvendo, expondo suas ideias e conquistas. Demonstrou também atenção e interesse com o trabalho dos demais membros da oficina, sempre disposto a ajudar.

Ao analisarmos os projetos sob a perspectiva do *framework* de Brennan e Resnick podemos destacar o crescimento do participante na utilização dos conceitos computacionais e no emprego das práticas computacionais. Ele demonstrou preocupação em criar projetos pensando no usuário e ficou muito claro seu desejo de aperfeiçoar cada vez mais seus trabalhos. Sua maneira de se expressar, de se questionar, sua empolgação e determinação indicam o uso das perspectivas computacionais.

4.3.2.4 Análise dos projetos do participante E4

O estudante E4 optou por enviar apenas dois projetos, mesmo tendo realizado outros projetos exploratórios durante a oficina. Como as tarefas não eram imposições e sim sugestões, o participante escolheu explorar o *Scratch* da maneira que achou mais conveniente, sem se preocupar em ter que cumprir determinada tarefa.

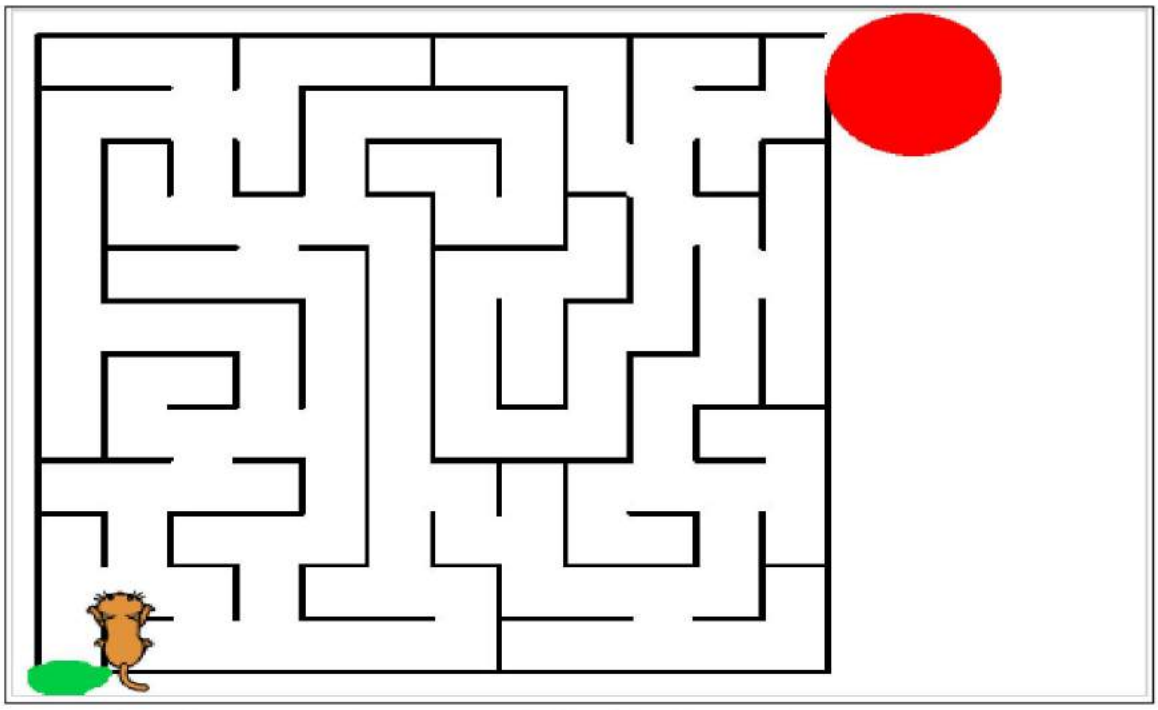
a) **Projeto 1**

O primeiro projeto foi desenvolvido tendo por base a proposta de criação de um labirinto, no qual um ator deveria ter um ponto de partida e outro de chegada. O

usuário deveria controlar o deslocamento desse ator e o programa não poderia permitir que ele ultrapassasse as linhas divisórias do labirinto. Foi proposta uma segunda fase, na qual os estudantes deveriam aperfeiçoar seu projeto, tornando-o mais atrativo e criando um controle de pontuação.

Para desenvolver essa tarefa o participante utilizou apenas um ator, representando um gato, conforme ilustrado na Figura 44. No labirinto o estudante pintou uma área em verde, determinando o ponto de partida e desenhou um círculo vermelho em outra extremidade, representando o ponto de chegada.

Figura 44- Projeto 1 do estudante E4.



Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 11 pontos e considerou que o programa atingiu o segundo nível, ou seja, "em desenvolvimento". Os conceitos de lógica, paralelismo, representação de dados e abstração obtiveram 1 ponto. Interatividade com o usuário e controle de fluxo ficaram com 2 pontos e a sincronização atingiu a pontuação máxima, com 3 pontos.

Ao analisar o programa levando em consideração o problema proposto e como a solução foi implementada podemos concluir que o projeto não contempla o

que foi proposto. Há indicativos de que o estudante teve dificuldades no emprego de alguns comandos e não conseguiu criar uma sequência lógica de passos para desenvolver o projeto. Alguns aspectos que merecem uma atenção maior são: o ator percorre o labirinto sem respeitar as paredes divisórias e, ao atingir o ponto de chegada, nada acontece.

Inicialmente, o ator escolhido pelo participante é muito grande (poderia ter sido reduzido de tamanho facilmente utilizando o *Scratch*). Por não ter ajustado o tamanho do ator à largura dos corredores do labirinto, não há como movimentar o ator pelo labirinto sem que ele ultrapasse as paredes divisórias. O estudante não implementou nenhuma mensagem ou outro mecanismo para indicar que o jogador atingiu o final do jogo. Além disso, não criou nenhum controle de pontuação ou outra melhoria.

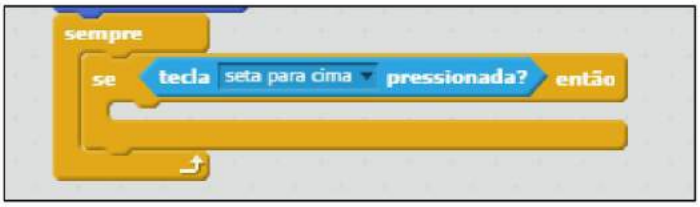
A análise do programa também revelou alguns erros no emprego dos controles do *Scratch,* entre eles o uso de comandos de maneira incompleta, como ilustrado na Figura 45. Esses blocos de comando podem ser removidos sem ocasionar nenhuma alteração na execução do programa.

Figura 45- Exemplo de comandos utilizados de maneira incorreta.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

A Figura 46 ilustra outro tipo de emprego de comandos desnecessários, no qual o bloco "se – então" realiza um teste e, sendo o resultado desse verdadeiro, nada é executado. A remoção do bloco "se – então" bem como do comando "sempre" não causam nenhuma modificação na execução do programa.

Figura 46- Exemplo de comandos desnecessários.



Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

Já na Figura 47, podemos observar o emprego de um comando duplicado, sem necessidade. O estudante inseriu dentro do evento "quando a seta para cima for pressionada" outro comando que visa aguardar a seta para cima ser pressionada para então executar os comandos seguintes. Portanto, o comando "espere até tecla seta para cima pressionada?" pode ser removido sem alterar a execução do programa.

Figura 47- Exemplo de comandos duplicados, sem necessidade.

Fonte: projeto elaborado no *software Scratch* (2017) - https://scratch.mit.edu

Podemos concluir que a análise feita pelo *Dr. Scratch,* a qual atribuiu uma pontuação de 11 pontos e considerou que o programa ficou no nível "em desenvolvimento", não condiz com o que foi desenvolvido e com os erros encontrados. O fato do estudante ter participado de apenas quatro aulas da oficina e não possuir computador em casa pode ter contribuído para o projeto apresentado estar incompleto e distante do que foi proposto.

O participante E4 relatou em seu caderno de reflexões que para desenvolver a tarefa do labirinto utilizou *"algumas imagens da internet e outros criados no Scratch mesmo"* (**E4**). Isso evidencia os recursos que o ambiente disponibiliza para criação e edição de som e imagens, além da facilidade de incorporação de outros elementos multimídia.

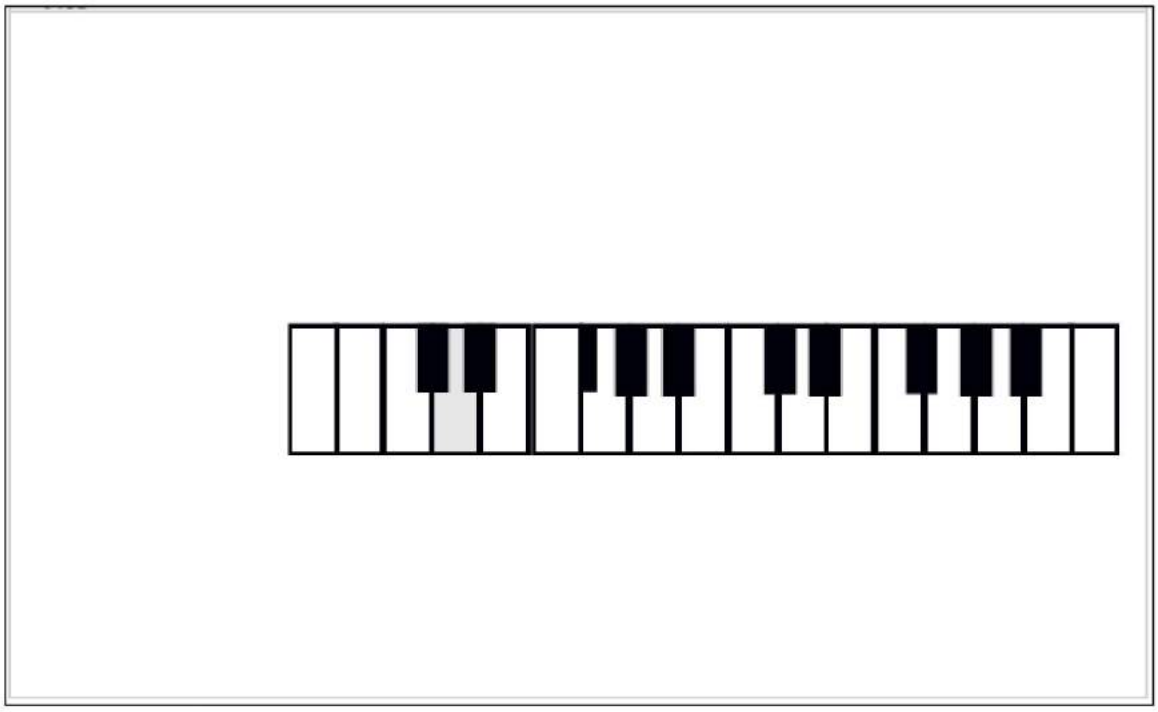
Com relação à sua percepção sobre o ambiente, o estudante destacou que a ferramenta propicia "d*esenvolvimento psicológico, criação e etc. Legal, até por você*

criar seu próprio jogo ou desafio ... é uma coisa diferente" (**E4**). Esse "desenvolvimento psicológico", citado pelo participante, está relacionado ao pensar e planejar o que se deseja criar. Observamos, com isso, que o estudante interagiu com o ambiente, explorando seus recursos e, assim, identificou possibilidades de emprego e potencialidades do *Scratch*. Todavia, no projeto 1, percebemos que essas descobertas ainda não foram postas em prática.

b) **Projeto 2**

Para o projeto final, de tema livre, esse participante explorou os projetos compartilhados na comunidade do *Scratch* e buscou inspiração no seu gosto pessoal por música. O estudante optou por criar um projeto simulando uma espécie de piano, conforme ilustrado na Figura 48.

Figura 48- Projeto 2 do estudante E4.



Fonte: autoria própria (2017) no *software Scratch* - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 6 pontos e com isso considerou que o programa ficou no nível "básico". Somente a interatividade com o usuário atingiu 2 pontos. Lógica e paralelismo não pontuaram e os demais conceitos obtiveram 1 ponto.

Ao analisarmos o projeto levando em consideração a proposta do estudante, concluímos que o projeto ficou incompleto, entretanto o estudante mostrou ter avançado na exploração de outros blocos, especialmente os que controlam sons e instrumentos musicais. O estudante explorou o conceito de criação e utilização de variáveis, porém ficou evidente a necessidade de auxílio para uma melhor compreensão e, com isso, um emprego melhor e mais eficiente desse recurso. Na Figura 49 podemos observar a utilização da variável "*which instrument*", porém ela nunca é inicializada ou alterada.

Figura 49- Utilização de variável.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Durante a análise do programa observamos também que, em algumas teclas, o estudante controlou adequadamente a troca de fantasia a fim de gerar a impressão que a tecla do piano foi pressionada, conforme código ilustrado na Figura 51. Já, a Figura 50 mostra duas trocas de fantasia sequenciais, sem nenhuma pausa entre elas, tornando-se assim imperceptíveis ao usuário.

Figura 50- Alteração de fantasia que o usuário não visualiza.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Figura 51- Alteração de fantasia visível ao usuário.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Na Figura 52 podemos observar um exemplo de código desnecessário ("use o instrumento 1"), pois no local onde ele foi colocado não faz nada e, portanto, sua remoção não causa nenhuma alteração na execução do programa.

Figura 52- Exemplo de comando que não tem efeito.

Fonte: projeto elaborado no *software Scratch* - https://scratch.mit.edu

Outro aspecto que ficou incompleto foi a forma escolhida pelo estudante para que o usuário interagisse com o piano. Algumas teclas são ativadas com o clique do mouse, porém a grande maioria só funciona com o pressionar de uma tecla específica do teclado. Somente analisando o código é que se descobre qual letra, ou número ou clique do mouse aciona cada tecla do piano. Faltou essa informação para o usuário.

Podemos concluir que mesmo esse projeto tendo atingido uma pontuação inferior na análise feita pelo *Dr. Scratch*, ao analisarmos o programa construído, bem como a proposta do estudante, foi possível observar uma evolução no seu conhecimento sobre programação e um melhor emprego dos recursos do *Scratch*. O programa apresenta diversos problemas, conforme já relatados, podendo ser melhor trabalhado, explorando e utilizando de forma mais adequada os recursos oferecidos.

Esse participante E4 faltou a três aulas da oficina, o que pode ter contribuído para o seu desempenho na realização das tarefas. O tipo de projeto escolhido, muito mais focado na geração de determinado som para cada tecla, não envolve muitos conceitos relacionados ao pensamento computacional e que, portanto, não são alvo da avaliação do *Dr. Scratch*. Porém, o estudante mostrou ter explorado e executado diversos testes a fim de conseguir reproduzir os sons de forma coerente em cada tecla do piano.

Em suas anotações o estudante afirmou que o *Scratch é "bom tanto para desafios, jogos, animações... Eu achei legal o desenvolvimento de um programa sem ter muita regra e funções..."* (**E4**). Com essa colocação podemos pressupor que, ao explorar a ferramenta, o participante conseguiu identificar diversas possibilidades de

aplicações, sendo motivado pela viabilidade de criar seus programas de forma livre, sem ter que se preocupar com sintaxe e formalismos. Características essas que fazem parte dos principais atributos do *Scratch,* que, por ser uma linguagem intuitiva, propicia aos estudantes a liberdade de criarem e regularem como as atividades serão elaboradas (RESNICK, 1998).

Apesar de ter gostado do *Scratch*, o estudante também destacou que ele *"poderia ser mais profundo, mais coisas deveriam ser feitas, mais funções o programa deveria ter... deveria ter mais opções"* (**E4**). Ou seja, alguns comandos a mais na ferramenta facilitariam o desenvolvimento dos projetos, principalmente para quem está iniciando na programação. Tal colocação também nos fez repensar a metodologia empregada na oficina, especialmente com os alunos mais introspectivos, que preferiram explorar o ambiente de maneira mais isolada, sem muita interação com os colegas e com o professor. Quando este realizava intervenções, geralmente obtinha como resposta do aluno que estava tudo certo, que não precisava de auxílio, que estava conseguindo desenvolver as tarefas.

Articulando a análise dos projetos do participante E4

O participante E4 precisou faltar a três aulas da oficina (por motivos de saúde e para poder frequentar os estudos orientados de matemática). Com isso, só conseguiu participar de quatro encontros e, em seu questionário pós-oficina[42], afirmou que não utilizou o *Scratch* fora do horário das aulas. Esses aspectos podem ter contribuído com o fato do estudante ter enviado somente dois projetos com problemas e incompletos, conforme já relatado.

A avaliação dos projetos feita pelo *Dr. Scratch* mostrou uma queda acentuada de pontuação, implicando na queda de nível, do primeiro para o segundo projeto. No primeiro todos os conceitos pontuaram e a sincronização obteve pontuação máxima. Já, no segundo, os conceitos de lógica e paralelismo não pontuaram e, com exceção da interatividade com o usuário que atingiu 2 pontos, os demais conceitos atingiram apenas 1 ponto cada.

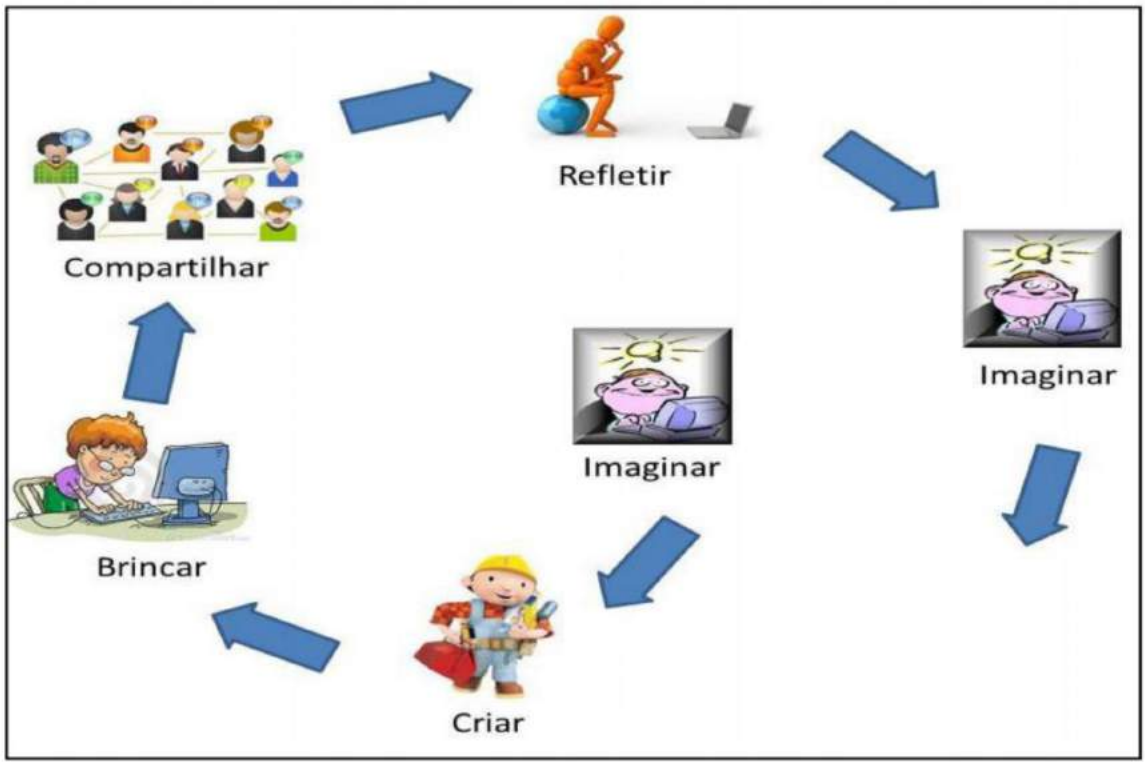
[42] Conforme resposta da questão número 8 do questionário: https://goo.gl/forms/Yt6YXQisImPzYJQJ3

Quando avaliamos os projetos levando em consideração a tarefa proposta observamos uma evolução do participante do primeiro para o segundo projeto. No primeiro, percebemos um projeto incompleto, que não possui uma sequência lógica ordenada de passos, ficando distante do que foi proposto. Já, no segundo trabalho, percebemos um avanço, com a utilização de novos comandos e, mesmo estando incompleto e com alguns erros, aproximou-se da ideia proposta pelo estudante. Nesse projeto observamos uma estruturação mais lógica e sequencial dos comandos.

As anotações do pesquisador revelam que esse estudante apresentou um perfil mais isolado, sem muitas interações com os colegas, apenas com os participantes E3 e E5. E, somente no segundo projeto, solicitou intervenções do professor para auxiliar na compreensão de alguns comandos e buscando sugestões para desenvolver determinadas ações. Esse participante fez pesquisas na internet e interagiu com o *Scratch* para buscar ideias e encontrar soluções para as dificuldades encontradas.

Quando o participante E4 relata em seu caderno de reflexões que o *Scratch* propicia "d*esenvolvimento psicológico, criação e etc. Legal, até por você criar seu próprio jogo ou desafio"* (**E4**), pode-se concluir que ele teve que pensar e lidar com situações problema e, a partir de suas interações com o ambiente de programação, conseguiu criar seus projetos. Nesse sentido, o *Scratch* foi criado para dar suporte ao que Resnick chamou de "Espiral do Pensamento Criativo", ilustrada na Figura 53. Essa espiral consiste em um processo no qual as pessoas imaginam o que desejam fazer, criam um projeto com base em suas ideias, brincam com suas criações, compartilham suas ideias e criações com os outros, refletem sobre as suas experiências e tudo isso os levam a desenvolver novas ideias e novos projetos. Passar por esse processo repetidas vezes, faz com que os alunos aprendam a desenvolver suas próprias ideias, as testem, busquem seus limites, experimentem alternativas e gerem novas ideias com base em suas experiências (PASSOS, 2014).

Figura 53- Espiral do Pensamento Criativo de Resnick.



Fonte: Passos (2014)

Com relação ao *framework* de Brennan e Resnick, podemos destacar que os conceitos computacionais foram pouco trabalhados por esse participante. Dentro das práticas computacionais percebemos que testes e depuração foram utilizados em diversos momentos no projeto final, a fim do estudante conseguir encontrar os sons específicos para associar com as respectivas teclas de seu piano. Todavia, não foi possível identificar o emprego de perspectivas computacionais.

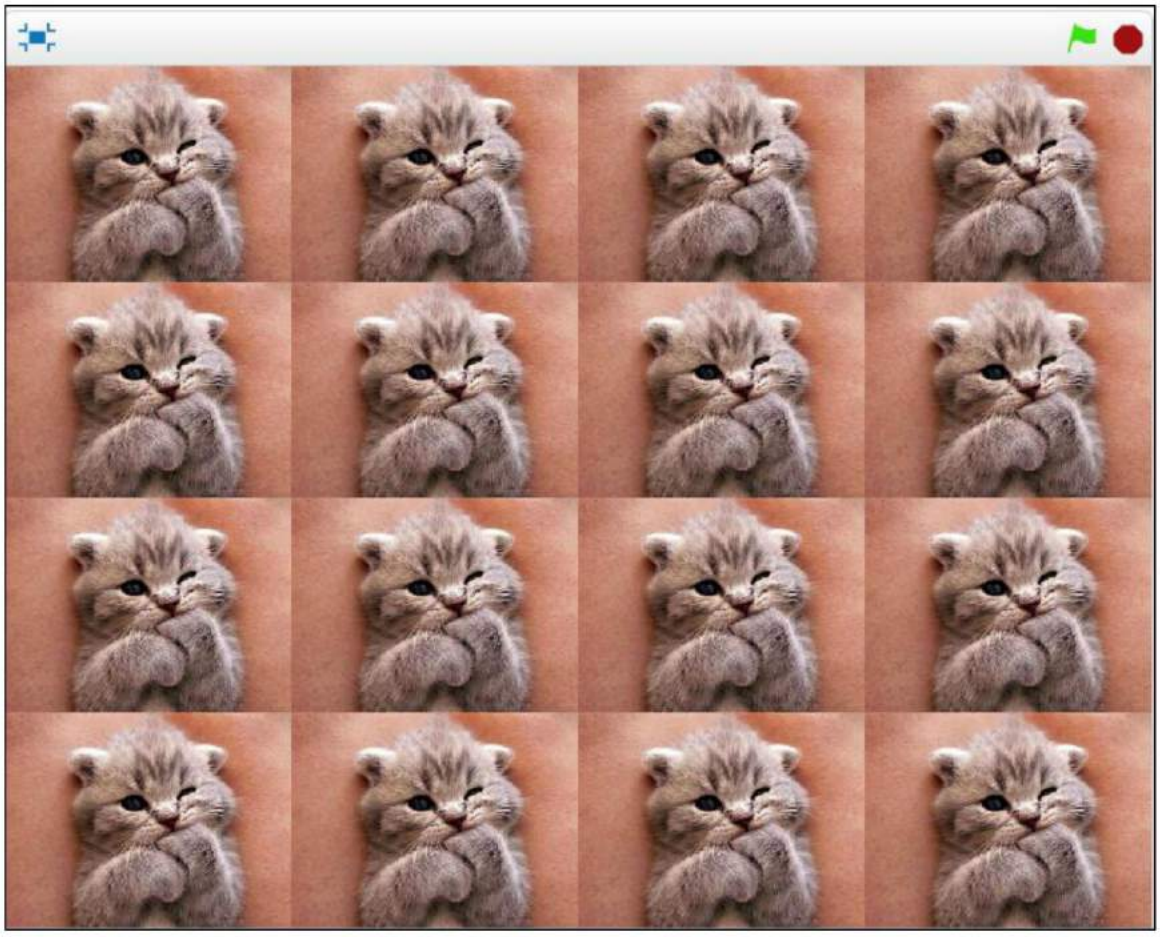
4.3.2.5 Análise dos projetos do participante E5

O estudante E5 optou por criar projetos de acordo com seus gostos e interesses. Como as tarefas não eram imposições e sim sugestões, o estudante escolheu explorar o *Scratch* da maneira que achou mais conveniente, sem se preocupar em ter que cumprir determinada tarefa. O foco principal de seus projetos foi a criação de animações envolvendo manipulação de imagens e sons.

a) **Projeto 1**

Esse foi o primeiro projeto desenvolvido, de caráter exploratório, no qual o estudante utilizou algumas imagens de plano de fundo, as quais eram alternadas e recebiam efeitos gráficos disponibilizados pelo *Scratch*. Um projeto que não empregou nenhum ator, apenas imagens de plano de fundo, conforme exemplificado na Figura 54. Entretanto o estudante demonstrou estar feliz e satisfeito por conseguir fazer sua animação.

Figura 54- Projeto 1 do estudante E5.



Fonte: autoria própria (2017). Projeto elaborado no *software Scratch* - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 4 pontos e com isso considerou que o programa ficou no nível "básico". Os conceitos de lógica, paralelismo e abstração não pontuaram, os demais conceitos obtiveram 1 ponto.

Ao analisarmos o programa levando em consideração a proposta do estudante, concluímos que ele criou um projeto sem se preocupar com os detalhes, indicando ter focado sua exploração nos recursos e efeitos de imagem. Concluímos que a pontuação e o nível atribuídos pelo *Dr. Scratch* foram coerentes com o projeto desenvolvido.

b) **Projeto 2**

O projeto 2 foi criado simplesmente para tentar apresentar algo tendo por base a proposta de criação de um labirinto. Entretanto, o que foi apresentado pelo participante E5 não faz absolutamente nada, ou seja, não possui nenhum comando. O estudante apenas desenhou um plano de fundo com um labirinto e adicionou dois atores, sem criar nenhum código, apenas deixou os atores estáticos, fixos no cenário.

A análise feita pelo *Dr. Scratch* atribuiu 0 (zero) pontos e com isso considerou que o projeto ficou no nível "básico". Como nenhum comando foi adicionado, todos os conceitos ficaram zerados.

Concluímos que o estudante apenas iniciou o projeto e logo em seguida o abandonou, quer seja por não ter se interessado pela tarefa proposta, quer seja por ter tido outra ideia que motivou-o a criar outro projeto. Por não ter nenhum comando, esse programa será desconsiderado para fins de análise.

c) **Projeto 3**

O estudante desenvolveu o projeto 3 com o objetivo de avançar na exploração do *Scratch*. Ele optou por criar uma animação na qual um mágico lança um raio com sua varinha mágica na personagem "*Peppa*", conforme ilustrado na Figura 55. Ao ser atingida por esse raio a personagem explode em uma bola de fogo e em seguida se transforma em um pedaço de bacon. Após alguns segundos a animação recomeça e é repetida dez vezes.

Figura 55- Projeto 3 do estudante E5.

Fonte: autoria própria (2017). Projeto elaborado no *software Scratch* - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 7 pontos e, com isso, considerou que o programa permaneceu no nível "básico", atribuindo 2 pontos ao controle de fluxo, não pontuando a lógica e os demais conceitos tiveram 1 ponto.

Ao compararmos com o primeiro projeto, a avaliação feita pelo *Dr. Scratch* mostra um grande aumento na pontuação, a qual praticamente dobrou. Ao analisarmos o programa levando em consideração a proposta do estudante para essa atividade, concluímos que o estudante conseguiu criar o que ele pretendia, mostrando avanço na utilização de alguns recursos, porém permanecendo no mesmo tipo de projeto, ou seja, a criação de uma animação.

Como resultado vemos um projeto sem interação com o usuário, sem a possibilidade de empregar recursos de lógica e outros comandos do *Scratch*. A animação ficou bem criativa e interessante, pois o estudante adicionou e editou sons e falas para sincronizar aos eventos, tornando a animação atrativa e engraçada. Alguns detalhes poderiam ser melhorados, como ocultar os ingressos do cinema que

o ator "*Peppa*" estava segurando nas mãos. Ao explodir, ele vira uma bola de fogo e posteriormente se transforma em bacon e os ingressos ficam soltos no ar.

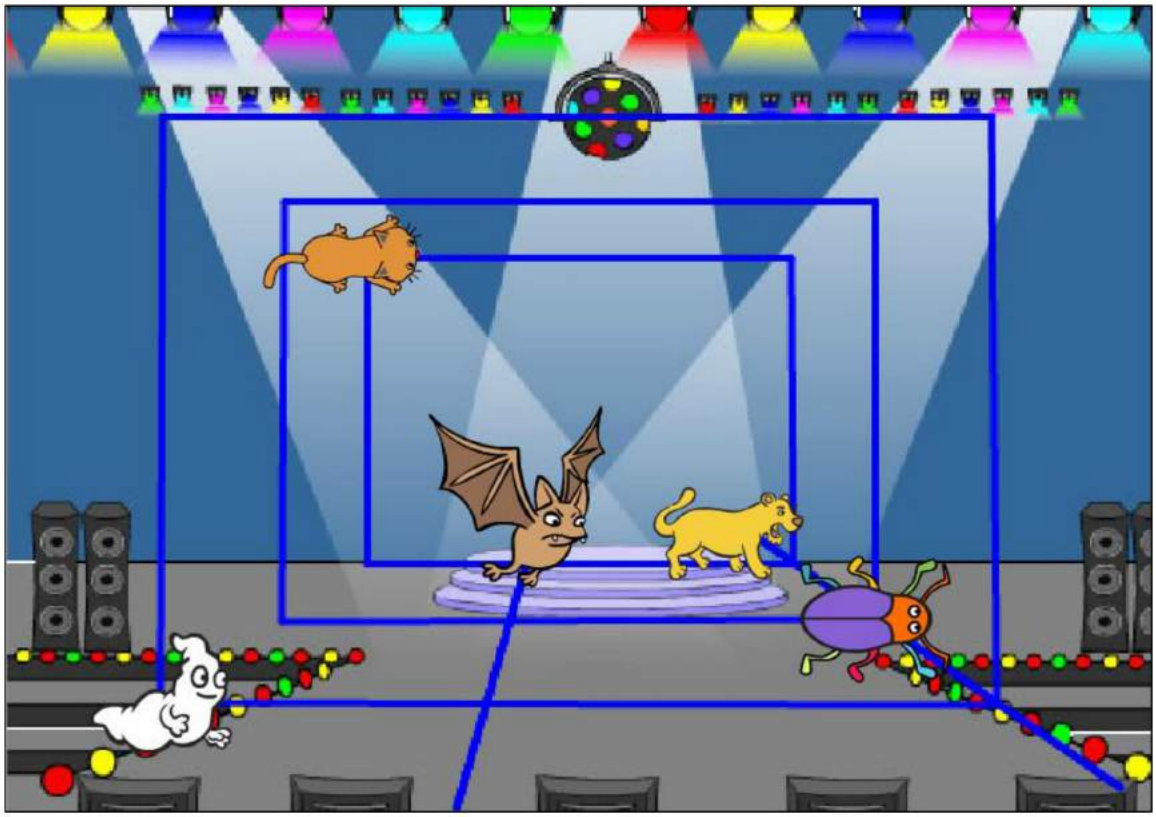
Para a construção do projeto 3 o estudante não se preocupou em utilizar outros tipos de comandos ou em criar interação com o usuário. Seu objetivo era desenvolver uma animação e por isso focou seus esforços na exploração dos elementos que poderiam contribuir para isso. Pela característica exploratória da oficina esse projeto ficou coerente com a proposta do estudante.

Ao programar no *Scratch* os estudantes deixam de ser apenas usuários passivos do computador e passam a criar, tornando a aprendizagem uma descoberta – uma das características mais valiosas a Papert (2008). Nesse sentido, o participante E5 teve uma ideia criativa, pesquisou e explorou o ambiente a fim de descobrir maneiras de implementar suas ideias, conseguindo concretizar, ao final, o que havia imaginado.

d) **Projeto 4**

No projeto 4, o estudante criou uma animação com cinco atores, sendo que três destes eram responsáveis por desenhar quadrados e os outros dois atores desenhavam linhas interligando os cantos inferiores dos quadrados, dando a ideia de profundidade, conforme ilustra a Figura 56. Novamente o estudante optou por montar um trabalho voltado à criação de uma animação. Para esse projeto o estudante buscou seguir a tarefa proposta na segunda aula da oficina, a qual propunha a criação de pelo menos duas formas geométricas. O estudante utilizou como inspiração uma imagem disponível nos planos de fundo do *Scratch* para tentar desenhar algo semelhante com seus atores.

Figura 56- Projeto 4 do estudante E5.



Fonte: autoria própria (2017). Projeto elaborado no *software Scratch* - https://scratch.mit.edu

A análise feita pelo *Dr. Scratch* atribuiu uma pontuação de 7 pontos e com isso considerou que o programa permaneceu no nível "básico", atribuindo 2 pontos ao controle de fluxo, não pontuando a lógica e os demais conceitos tiveram 1 ponto. A análise ficou exatamente igual ao projeto anterior, pontuando exatamente os mesmos conceitos. Mesmo utilizando comandos diferentes, a dinâmica das animações são bastante semelhantes.

Ao analisarmos o programa levando em consideração a proposta do estudante para essa atividade, concluímos que o projeto ficou coerente com a proposta do estudante. Ao utilizar recursos como a caneta e o posicionamento no plano cartesiano, verificou-se um avanço na utilização do *Scratch*, porém ele permaneceu no mesmo tipo de projeto, ou seja, a criação de uma animação. Novamente observamos um projeto sem interação com o usuário, sem a possibilidade de empregar recursos de lógica e outros comandos do *Scratch*.

Nessa animação o participante E5 utilizou recursos de som apenas em dois atores, com menos ênfase que na animação anterior. Os quadrados são desenhados corretamente, porém uma das linhas que deveria interconectar os cantos inferiores esquerdos dos quadrados não é desenhada na posição correta. Após cada ator desenhar seu respectivo objeto, todas as imagens desenhadas são apagadas e recomeça o desenho de todos os objetos. O estudante optou por executar essa ação infinitamente, até que o usuário pare o saia do programa.

A animação ficou interessante, porém poderia ter sido melhor trabalhada, especialmente ajustando a linha que não é desenha da posição correta e incrementando o projeto a fim de torná-lo mais atrativo. Poderia ter feito alguma interação com o usuário, possibilitando-lhe escolher alguns parâmetros.

Articulando a análise dos projetos do participante E5

O participante E5 esteve presente em cinco aulas da oficina, não comparecendo a dois encontros. Entretanto ele não fez nenhum registro em seu caderno de reflexões. Através das anotações do pesquisador, podemos destacar que o estudante ficou deslumbrado com a possibilidade de criar animações no *Scratch.* Todos os seus projetos versaram sobre esse tema, envolvendo inserção e edição de sons e imagens, aliados ao emprego de movimento. Com isso, deixou de explorar outras possibilidades do *Scratch* como a criação de jogos ou histórias interativas.

As animações criadas pelo participante, assim como a maioria das animações criadas na ferramenta, caracterizam-se pela utilização de apenas alguns comandos, não exercitando muitos dos conceitos de programação. Geralmente esse tipo de projeto é criado para ser executado sempre da mesma maneira, sem intervenção do usuário. Na maioria das vezes não são empregados conceitos de lógica, pois não é necessário criar condições para a execução. Pela avaliação do *Dr.* Scratch, nenhum dos projetos desse estudante pontuou o conceito de lógica. Somente o controle de fluxo conseguiu atingir 2 pontos, os demais atingiram no máximo 1 ponto.

Os projetos 3 e 4 foram os que mais pontuaram. Analisando-os, considerando a proposta do estudante, concluímos que ele conseguiu criar o que desejava, o que planejou. As anotações do pesquisador revelaram motivação e dedicação do participante para conseguir montar esses projetos. A empolgação foi tanta que em

alguns episódios ele deu gargalhadas, externando sua alegria por conseguir criar determinadas ações e com isso fazer seu programa funcionar. Essa mesma alegria e empolgação era demonstrada quando descobria algum recurso do *Scratch* que julgasse interessante ou que pudesse ser empregado na construção de seus projetos.

Ao conseguir progredir na criação dos trabalhos, o participante E5 fazia questão de mostrar o que havia desenvolvido para os demais participantes da oficina. Em muitos momentos os colegas eram contagiados pela sua empolgação e iam até ele para ver o que estava fazendo, o que havia descoberto e assistir suas criações. Nessas ocasiões ficava visível a sociointeração, especialmente quando algum colega, ao ver seus trabalhos, dava dicas para avançar ou melhorar os projetos. O participante E3 foi o que mais interagiu no sentido de contribuir para solucionar problemas ou melhorar os trabalhos.

O ambiente *Scratch* possui um *feedback* imediato: qualquer comando ou bloco de comandos pode ser testado imediatamente, diferente de linguagens tradicionais que exigem um processo de compilação. Dessa maneira, a ferramenta possibilitou que o participante fosse testando seus comandos e demais recursos que, conforme eram adicionados, iam sendo executados e com isso era possível identificar o que estava funcionando e o que precisava ser modificado.

Percebemos que o *Scratch* possibilitou ao estudante E5 transformar suas ideias e criatividade em programas computacionais, fornecendo os recursos necessários e mediando essas criações. Nesse sentido, Resnick, um dos criadores da ferramenta, vislumbrava um processo interativo e cíclico envolvendo aprendizado, reflexão e atividades lúdicas. Para tanto,

> O *Scratch* foi projetado para dar suporte ao processo criativo dos alunos, como descrito pela "Espiral do Pensamento Criativo" proposta por Resnik (2007) e faz parte de uma nova geração de tecnologias projetadas para ajudar os alunos a se preparar para uma Sociedade Criativa. (PASSOS, 2014, p.84)

O participante E5 não efetuou nenhum registro em seu caderno de reflexões e optou por montar apenas animações. Esse fato pode estar relacionado com o perfil da oficina, a qual não impôs um caminho a ser seguido, buscamos a exploração do *Scratch* de forma livre, apenas com indicações e sugestões de tarefas, sem avaliações. Esse estudante visualizou a oficina como um momento de explorar o *Scratch*, dando asas à sua imaginação e criando o que lhe interessava, com alegria e empolgação.

5 O QUE EMERGE DA ANÁLISE DO *CORPUS*

Apresentaremos a construção da resposta à pergunta de pesquisa[43], articulando os resultados da avaliação gerados pelo *Dr. Scratch* e os resultados da análise feita pelo pesquisador.

No Quadro 12, estão sintetizados os resultados das análises dos projetos de cada participante da oficina. Ao final foi calculada a pontuação média que cada participante obteve levando em consideração todos os trabalhos realizados. Somente o projeto 2 do participante E5 não foi considerado para análise por não possuir nenhum comando, somente uma imagem.

Quadro 12- Pontuação atribuída pelo *Dr. Scratch* na análise dos projetos.

ANÁLISE DOS PROJETOS PELO DR. SCRATCH – PONTUAÇÃO						
Participante	**Projeto 1**	**Projeto 2**	**Projeto 3**	**Projeto 4**	**Projeto 5**	**Média**
E1	13	7	11	16	-	11,75
E2	5	7	8	12	-	8,00
E3	6	11	11	13	15	11,20
E4	11	6	-	-	-	8,50
E5	4	0*	7	7	-	6,00
* Esse projeto foi desconsiderado pois nenhum código foi implementado						

Fonte: criado pelo autor (2018)

Com base nos dados apresentados no Quadro 12 é possível concluirmos que a maioria dos participantes apresentou um crescimento na pontuação de seus projetos e que, ao final, a maioria conseguiu atingir o nível "em desenvolvimento". Destaca-se o participante E1 pois conseguiu, em seu projeto final, atingir o nível "proficiente". Por outro lado, o estudante E5 permaneceu no nível "básico".

Os participantes E1 e E3 foram os que obtiveram maior pontuação média nos trabalhos, sendo que o estudante E1 não teve nenhuma falta e o E3 apenas uma ausência, de um total de sete encontros. Fato esse que pode ter contribuído para seus desempenhos nos projetos. Além disso, afirmaram ter utilizado o *Scratch* várias vezes fora do horário da oficina.

[43] Pergunta de pesquisa: como o *Scratch* pode mediar a aprendizagem de programação no Ensino Médio com vistas ao desenvolvimento do pensamento computacional a partir da teoria vigotskiana?

Os participantes E2 e E4 obtiveram médias semelhantes, porém, ao analisarmos os projetos individualmente, percebemos evoluções distintas. Enquanto o E2 realizou quatro projetos e demonstrou um crescimento linear da pontuação, o E4 só desenvolveu dois trabalhos e apresentou um declínio na pontuação. Esses dois participantes faltaram a três aulas da oficina, porém, quando questionados sobre a utilização do *Scratch* fora do horário das aulas, o estudante E2 relatou ter utilizado algumas vezes, enquanto o E4 (por não possuir acesso a computador fora da escola), afirmou não ter utilizado nenhuma vez.

O participante E5 obteve a menor média nos trabalhos apresentados e, desconsiderando o projeto 2, obteve uma pontuação crescente em seus trabalhos. Esse participante teve duas faltas e afirmou não ter utilizado o *Scratch* fora das aulas. Cabe destacar também que os participantes E4 e E5 não desenvolveram nenhuma das tarefas propostas, ou seja, criaram projetos de acordo com seus desejos e imaginação.

A metodologia empregada na oficina teve caráter exploratório, não obrigando os participantes a seguirem um roteiro rígido nem exigindo avaliações ou atribuindo notas aos trabalhos. O foco foi a interação do usuário com o ambiente, tendo o professor como mediador e com a possibilidade de diálogo e auxílio entre os colegas. Esse cenário possibilitou a cada participante seguir seu próprio caminho, atingindo com isso diferentes níveis e características de aprendizado. Isso porque, segundo os estudos de Vigotski (2007), a aprendizagem ocorre à medida que o indivíduo executa operações externas, com o uso de instrumentos e signos, e após reconstrói internamente essas ações, atribuindo significado a elas, em um processo de internalização.

Destacamos como pontos positivos dessa metodologia a possibilidade dos estudantes escolherem como iriam explorar o *Scratch*, seguindo seus desejos e movidos pela curiosidade de aprender como executar determinada ação ou como construir o que sua imaginação indicou. Com isso, os participantes demonstraram motivação e interesse, buscaram auxílio para suas dúvidas em diversos materiais de apoio (vídeos, tutoriais, ajuda do ambiente, comunidade do *Scratch*), além do professor, o qual buscou não dar respostas diretas às dúvidas e questionamentos,

mas sim indicar caminhos, fazendo questionamentos e instigando o estudante a pensar e testar suas hipóteses.

Com relação aos pontos negativos dessa metodologia podemos destacar que ela dificultou a avaliação do percurso dos estudantes, pois cada um criou um número diferente de projetos, cada qual com propostas distintas. Como as análises feitas pelo *Dr. Scratch* não levam em consideração o enunciado das tarefas ou os objetivos pretendidos com a realização de cada atividade, nem sempre a pontuação atribuída por ele ficou de acordo com o que foi desenvolvido. Já, a avaliação feita pelo pesquisador buscou analisar os projetos implementados levando em consideração a proposta do estudante, o tempo disponível para a criação, os conhecimentos prévios dos participantes e seu percurso para desenvolver cada uma das atividades. Além disso, a análise do *Dr. Scratch* por si só não foi suficiente para responder a pergunta de pesquisa.

Com base na análise dos projetos feita pelo pesquisador, podemos destacar que o participante E1 demonstrou um crescimento a cada projeto na apropriação dos comandos do *Scratch,* além de um aprofundamento na utilização da lógica e um aumento gradual na complexidade dos projetos. Ressaltamos também, a aplicação de práticas computacionais, como teste e depuração, modularização, reuso e reformulação. Seus questionamentos, sua forma de interação com o ambiente, com o professor e demais colegas apontam para o emprego de perspectivas computacionais.

A avaliação feita pelo *Dr. Scratch* no projeto 2 do participante E1 indicou um declínio frente o projeto 1. Entretanto, a avaliação do pesquisador revelou que o projeto 2 ficou coerente com a proposta do estudante, além de ter utilizado outros recursos do *Scratch,* como a inicialização de variáveis. No projeto 3, a análise do pesquisador destacou o uso de funções que permitiram uma redução no tamanho do código, além de o ter tornado mais claro. O uso de funções indicou um avanço significativo com relação ao pensamento computacional. Por fim, no projeto 4, o estudante demonstrou um avanço no emprego do pensamento lógico condicional através da utilização do comando "se–então–senão", combinado ao uso de expressões lógicas.

Sobre o participante E2, a análise do pesquisador nos permitiu destacar o emprego bem sucedido de alguns conceitos computacionais, porém necessitando aprofundar o estudo e a exploração de outros conceitos, como, por exemplo, dados e operadores. Percebemos que o participante buscou fazer teste e depuração, porém nem sempre conseguiu solucionar os problemas encontrados. Destacamos o emprego de reuso e reformulação nos dois últimos projetos, tanto na questão de imagens como na questão de ideias e comandos.

Quanto ao projeto 2 do estudante E2, a análise do pesquisador nos permitiu concluir que o estudante não conseguiu finalizar seu trabalho, sendo que algumas ações ficaram incorretas ou incompletas. O participante mostrou pouca evolução na exploração de outros recursos do *Scratch,* apenas avançou no paralelismo, com diversos atores se movimentando simultaneamente.

A análise do projeto 3 pelo pesquisador nos levou a concluir que esse trabalho também ficou incompleto, não tendo sido implementado um controle de pontuação. Com isso o estudante deixou de explorar e empregar as operações com variáveis e a criação de blocos – o que possibilitaria a redução de código. Entretanto, no projeto 4, a análise do pesquisador nos possibilitou indicar que o estudante E2 avançou em diversos aspectos, como a sincronização de vários objetos e a integração de recursos multimídia.

A análise do pesquisador com relação ao participante 3 ressaltou a preocupação deste em criar projetos pensando no usuário, bem como seu desejo de sempre aprimorar seus trabalhos. Além disso, permitiu-nos destacar seu crescimento na utilização dos conceitos computacionais e no emprego das práticas computacionais. Também percebemos o uso das perspectivas computacionais pela maneira do estudante se expressar, questionar-se, empolgar-se e mostrar determinação na construção dos projetos.

A avaliação feita pelo *Dr. Scratch* revelou um grande aumento na pontuação do projeto 1 para o projeto 2 do participante E3, porém, a análise feita pelo pesquisador nos permitiu concluir que o estudante atingiu de forma básica seus objetivos, mostrando avanço na utilização de alguns recursos. O projeto 2 ficou incompleto e apresentou objetos e códigos que não são utilizados para nada, também denominado de "código morto".

No projeto 3, a avaliação feita pelo *Dr. Scratch* mostrou que a pontuação de todos os conceitos foi exatamente igual ao projeto 2. Entretanto a análise do pesquisador revelou uma superioridade do projeto 3 em relação ao projeto 2, tanto pela programação desenvolvida como pelos recursos utilizados, resultando em um projeto mais aprofundado e detalhado. Percebemos que faltaram alguns ajustes no projeto 3, principalmente na inicialização de variáveis, mas o resultado final está coerente com problema proposto.

No projeto 4 do participante E3, a análise do pesquisador possibilitou destacarmos a criação de controles e melhorias que tornaram o jogo mais atrativo e amigável. Além disso, o estudante inseriu, no início do jogo, uma tela com as instruções de como jogar, a fim de esclarecer e orientar o usuário sobre as regras do jogo.

Por fim, a análise do projeto 5 por parte do pesquisador nos permitiu ressaltar um avanço em aspectos como a utilização de variáveis e a criação de blocos. Assim como ocorreu no projeto 4, o participante inseriu no início do jogo telas com as instruções de como jogar e suas regras, tornando o jogo mais amigável e demonstrando uma preocupação por parte do estudante com o usuário que irá utilizar seu jogo.

O participante E4, segundo análise do *Dr. Scratch,* apresentou uma queda acentuada de pontuação do projeto 1 para o projeto 2. Entretanto, a análise feita pelo pesquisador revelou uma evolução do primeiro para o segundo projeto. Conforme esta análise, o projeto 1 está incompleto e não possuí uma sequência lógica e ordenada de passos, ficando muito distante do que foi proposto. Já a análise do projeto 2 revelou um avanço, com a utilização de novos comandos e, mesmo estando incompleto e com alguns erros, está mais próximo do que foi pretendido pelo estudante. Nesse trabalho observamos uma estruturação mais lógica e sequencial dos comandos, com a criação e utilização de variáveis, as quais, entretanto, não foram eficientemente utilizadas, demonstrando a necessidade de auxílio para uma melhor compreensão.

No que se refere ao participante E5, esse ficou deslumbrado com a possibilidade de criar animações no *Scratch* sendo que todos os seus projetos focaram na criação de animações, envolvendo inserção e edição de imagens e sons, aliado ao emprego de movimento. O estudante deixou de explorar outras

possibilidades do *Scratch* como a criação de jogos ou histórias interativas. As animações criadas pelo participante, assim como a maioria das animações criadas na ferramenta, caracterizam-se pela utilização de apenas alguns comandos, não exercitando muitos conceitos de programação.

Conforme mencionado anteriormente, o projeto 2 do participante E5 não foi considerado para a análise por não conter nenhum comando. A análise do pesquisador revela que o projeto 1 foi criado sem nenhuma preocupação, sendo que o estudante buscou apenas explorar a inserção e edição de imagens disponibilizados pelo ambiente.

Já, os projetos 3 e 4 do participante E5 foram os que mais pontuaram. A análise do pesquisador revelou que nesse dois projetos o estudante conseguiu criar o que desejava, o que planejou. Entretanto, são projetos sem interação com o usuário, sem a possibilidade de empregar recursos de lógica e outros comandos do *Scratch*. As animações criadas em ambos os projetos ficaram criativas e interessantes. No projeto 3, foi realizada a inserção e edição de sons nos eventos, tornando a animação atrativa e engraçada. No projeto 4, a análise do pesquisador destacou o emprego de novos recursos, como a caneta e o posicionamento no plano cartesiano, demonstrando avanço na utilização do *Scratch*. A animação, apesar de interessante, poderia ter sido melhor trabalhada, cabendo também alguns incrementos no projeto a fim de torná-lo mais atrativo, como, por exemplo, a inserção de alguma interação com o usuário, a fim de possibilitar-lhe a escolha de alguns parâmetros.

Após essas considerações sobre as avaliações do *Dr. Scratch* e as análises feitas pelo pesquisador, podemos afirmar que os projetos criados pelos estudantes durante as atividades da oficina mostraram sua evolução no que diz respeito à aprendizagem dos conceitos relacionados à programação de computadores. Além disso, evidenciaram o emprego de princípios do pensamento computacional, independentemente do nível de ensino e dos conhecimentos prévios de cada participante. Cabe ressaltar que não houve intenção de comparar o desempenho entre os estudantes e sim a evolução individual de cada um.

De forma geral, foi possível observar que os participantes evoluíram, tanto no grau de complexidade dos elementos e blocos de comandos que usaram para compor os projetos, quanto nas relações lógicas para a construção do algoritmo por trás da

solução apresentada. Alguns participantes, em especial o E3, demonstrou preocupação e cuidado com o usuário final do projeto, buscando desenvolver seus trabalhos com informações bem detalhadas e com interface intuitiva e amigável.

Percebemos o interesse e a curiosidade dos participantes através do contato com as mídias que o ambiente *Scratch* oferece (imagens, atores, cenários, som, música etc). Observamos que os participantes conseguiram assumir o papel de autores, programaram, criaram e compartilharam suas criações, pensando criativamente. Ao descobrirem, na experiência lúdica, conceitos matemáticos e computacionais, treinaram o pensamento sistêmico, desenvolveram relacionamentos interpessoais e de cooperação, habilidades necessárias para o século XXI. (RESNICK, 2006).

Outro aspecto que aparece como resultado da análise do *corpus* é a importância do papel do professor, o qual necessita ser redimensionado.

Redimensionamento do Papel do Professor

Na efetivação da oficina, o pesquisador, atuando como professor, precisou redimensionar seu papel, a fim de sair do lugar do discurso e ir para o lugar da mediação. Para isso, foi necessário aprender a atuar como mediador, buscando avaliar o que os estudantes tinham compreendido, qual era sua zona de desenvolvimento proximal, para então encontrar táticas de atuação. Foi preciso autocontrole para não dar as respostas de imediato, para não apontar os erros e o que deveria ser feito para corrigi-los.

Conforme Vigotski (2007), a mediação é uma ação que ocorre no âmbito social por meio de instrumentos, de objetos, de palavras, de símbolos, dentre outros elementos. Assim, atuar como mediador implicou na busca por estratégias para que cada estudante construísse seu caminho até o conhecimento. Questionamentos, criação de situações problema, analogias e motivação foram algumas dessas estratégias adotadas.

No questionário pós-oficina[44], uma das perguntas se referia a percepção dos participantes sobre a forma de conduzir os encontros, a metodologia empregada e as

[44] Conforme resposta da questão 10 do questionário: https://goo.gl/forms/Yt6YXQisImPzYJQJ3

atividades propostas. Nessa questão, o estudante E1 relatou que "*as dúvidas que eu tinha na realização das atividades foram tiradas e muitas vezes o professor auxiliou na utilização da ferramenta. As aulas tiveram um método diferente, acho que dessa forma acabamos explorando mais o Scratch*" (**E1**). Já, o participante E2 afirmou que "*acho que não mudaria nada na forma como foi realizada a oficina. Apenas tive dúvidas em quais comandos dar ao ator para realizar a tarefa que eu desejasse, mas o professor me auxiliou muito nestas horas de dúvidas*" (**E2**). Os demais participantes deram respostas semelhantes, indicando que gostaram da forma como as aulas foram conduzidas, tendo o professor como mediador do processo.

Soares e Maschio (2017), realizando estudos sobre o uso de laptops educacionais no processo de ensino aprendizagem da Educação Básica e refletindo sobre a mediação pedagógica à luz da teoria vigotskiana, destacam que a mediação pode ser uma estratégia de interação entre sujeito e objeto de conhecimento. Essa mediação, segundo as autoras, pode ser representada pela tecnologia digital, a qual pode atuar como potencializadora da aprendizagem.

Para Vigotski (2007), os processos internos de desenvolvimento do sujeito estão diretamente relacionados ao ambiente em que este se desenvolve, à comunidade em que está inserido e aos indivíduos com os quais compartilha suas experiências. Dessa forma, as práticas vivenciadas pelos estudantes no ambiente Scratch podem ser ponto de partida para que representações mentais sejam desenvolvidas e, com isso, ocorram operações mentais relacionadas à internalização.

A internalização é um processo que origina uma reconstrução interna a partir de uma operação externa (VIGOTSKI, 2007). Desse modo, o professor precisa realizar intervenções para que as atividades realizadas no *Scratch* sejam significadas pelos estudantes. Ainda, segundo o autor, para uma maior eficácia, a mediação precisa ser realizada na ZDP[45] do aluno. Para isso, o professor necessita observar atentamente os estudantes: suas dúvidas, erros e procedimentos na realização das atividades e na exploração do ambiente, a fim de identificar a ZDP de cada um deles.

Quando o professor observa os participantes seguindo a teoria vigotskiana, ele consegue fazer intervenções que instiguem, motivem e auxiliem cada um dos

[45] ZDP – zona de desenvolvimento proximal.

estudantes, a fim de que possam transformar as ações externas realizadas no Scratch em movimentos internos de significação e compreensão. Esses movimentos internos são essenciais para que cada sujeito construa sua rede de conhecimentos, desenvolvendo sua lógica algorítmica, fazendo relação com os comandos estudados e com isso ocorra o aprendizado de programação.

Segundo Soares e Maschio (2017), o desafio das práticas pedagógicas, em especial no contexto das tecnologias digitais, é dar autonomia ao estudante para que ele seja capaz de colocar em movimento seus instrumentos internos. Nesse sentido, o professor necessita atuar como um agente transformador, assim como propõe a teoria sociointeracionista de Vigotski, ou seja, incentivando a participação ativa do estudante em seu processo de aprender, realizando intervenções mediadoras, explorando seu potencial criativo, conhecendo e atuando em sua ZDP.

Diante do exposto, percebemos a importância do professor nesse contexto, redimensionando seu papel a fim de deixar o lugar do discurso e atuar como mediador. Essa forma de atuação, em conjunto com o *Scratch,* possibilitou aos participantes da oficina vivenciarem momentos de reflexão e construção de novos conhecimentos na área de programação.

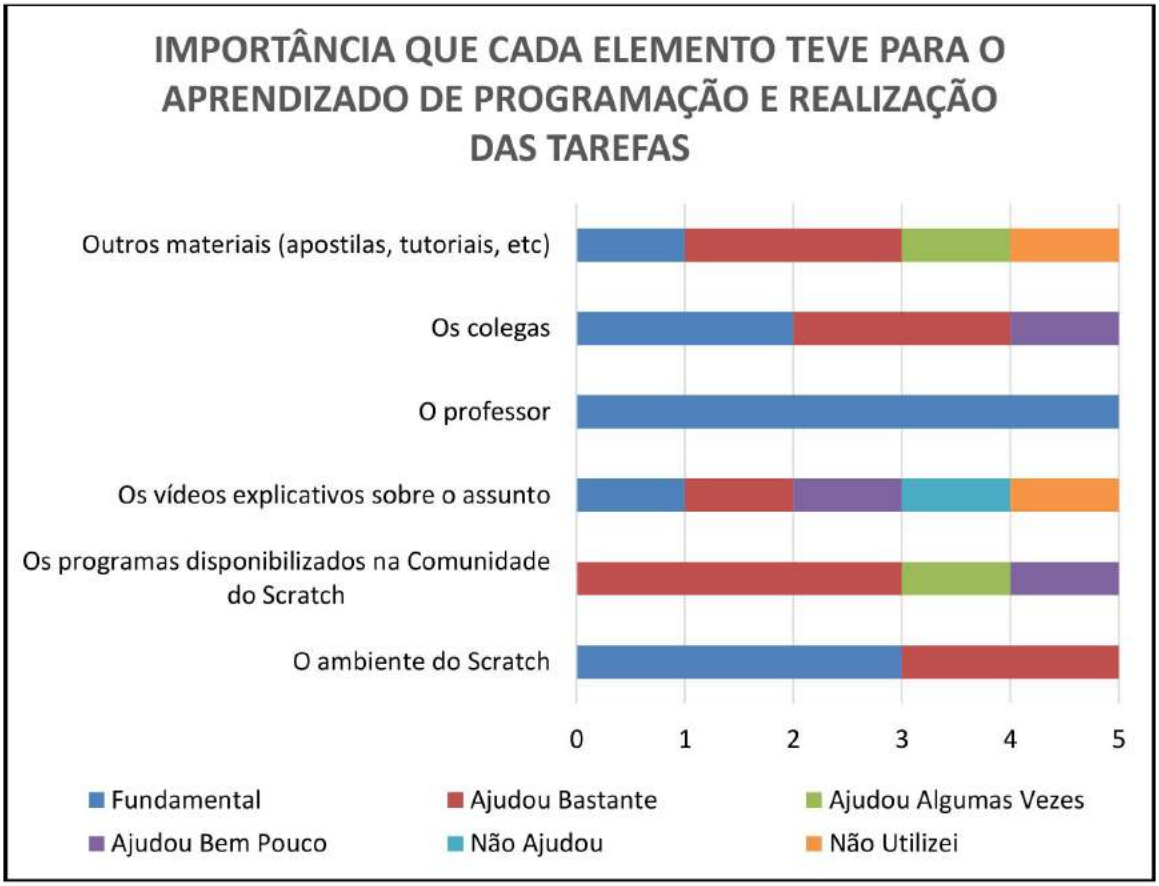
Vale citar o que o participante **E1**, em seu caderno de reflexões, afirmou: *"acho que para aprender a utilizar alguns recursos do Scratch e fazer pequenos projetos é suficiente só a ajuda do Scratch e sua comunidade, mas para realização de projetos um pouco mais elaborados, a ajuda do professor foi fundamental"* (**E1**). Assim, verifica-se sua percepção sobre o *Scratch* como elemento mediador e a importância do papel do professor na exploração do ambiente e realização das tarefas. Nessa mesma perspectiva, o participante E2 afirmou que o *Scratch* oferece "*uma boa base criativa para a realização dos jogos e acho que a ajuda e as dicas de programação do site também ajudam muito na hora de realizar o projeto, porém acho que tendo a ajuda de colegas e professores fica muito mais fácil e divertido*" (**E2**).

Os participantes da oficina demonstraram sua percepção sobre a importância do *Scratch* e do professor como mediadores do processo de aprendizagem de programação através, também, de suas respostas do questionário pós-oficina[46].

[46] Conforme resposta da questão número 6 do questionário: https://goo.gl/forms/Yt6YXQisImPzYJQJ3

Nesse questionário, uma questão abordava a importância que cada elemento teve para o aprendizado de programação e para a realização das tarefas propostas. Essas respostas estão expressas no Gráfico 2, onde podemos observar que todos os participantes consideraram fundamental o papel do professor e julgaram o *Scratch* fundamental ou que lhes ajudou bastante.

Gráfico 2- Importância dos elementos no aprendizado de programação e realização das tarefas.



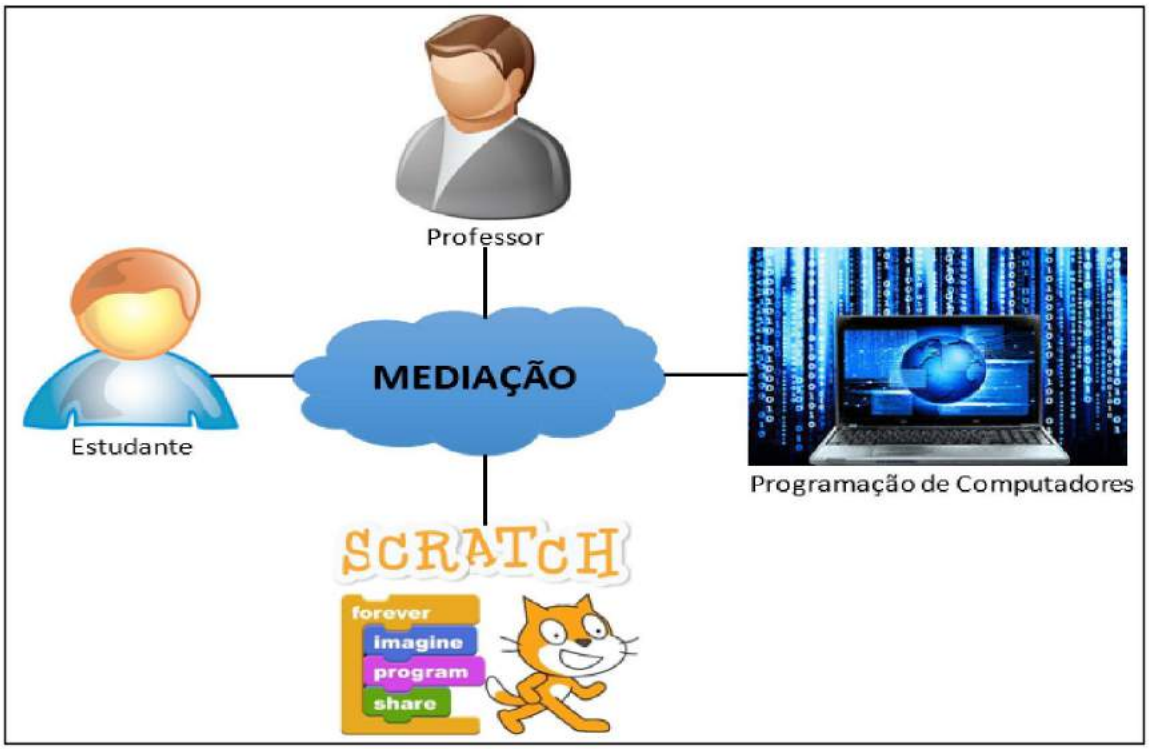
Fonte: criado pelo autor (2018)

Ao avaliarmos o *Scratch* como elemento mediador percebemos que, apesar dele possuir um ambiente amigável, estimulante, que motiva e propicia o trabalho autônomo, possibilitando uma iniciação fácil e intuitiva à programação de computadores, são necessários outros elementos nesse processo. Assim como Marques (2009) relatou em seus estudos, destacamos a importância da cooperação, do acompanhamento e da mediação por parte do professor, sem os quais "a produção

parece reduzir-se e a evolução não acontece a um ritmo satisfatório" (MARQUES, 2009, p. 32).

Nesse sentido, Matuí (1995, apud SOARES; MASCHIO, 2017) aponta que a estratégia mediadora é o elo entre o estudante e o objeto de conhecimento que necessita da ação do professor, ou ainda de outro colega, porém se fundamenta principalmente no papel do professor. No contexto desse estudo, entendemos que a estratégia mediadora reside nas relações e articulações estabelecidas entre as intervenções do professor, as possibilidades de comunicação do *Scratch* e o estudante. A Figura 57 ilustra esse cenário, onde as estratégias mediadoras podem conduzir o estudante ao aprendizado de programação.

Figura 57- Elementos de mediação na oficina de introdução à programação.



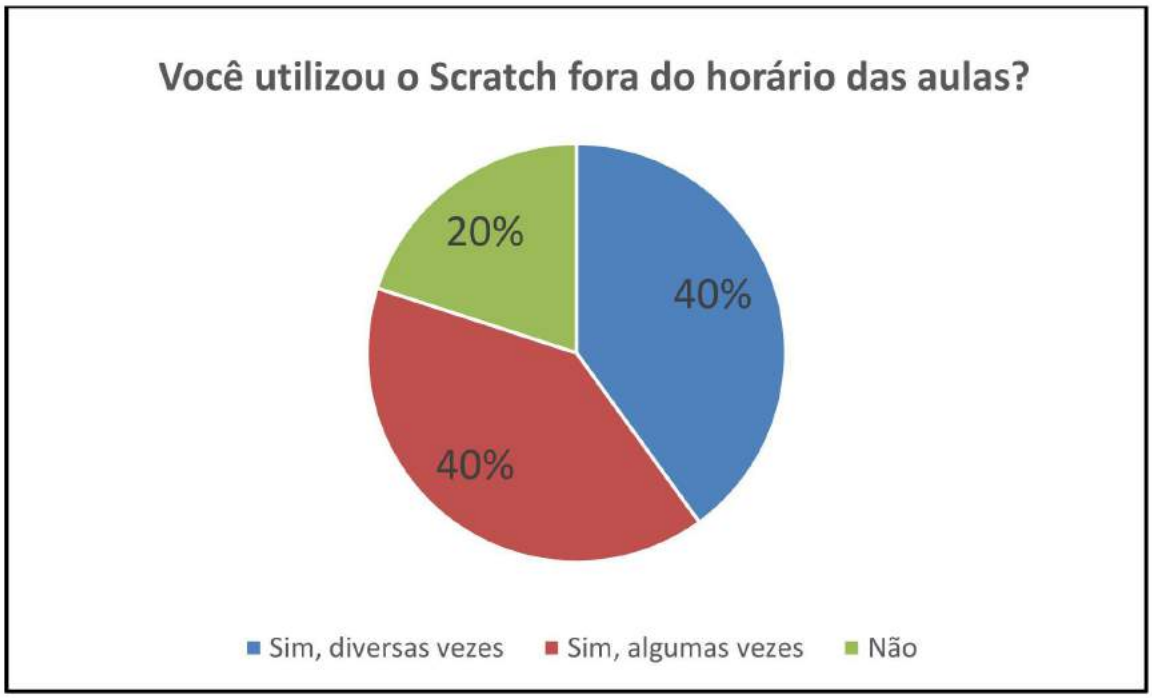
Fonte: criado pelo autor (2018).

As principais contribuições do *Scratch* para a realização dos projetos, conforme afirmou o participante E2 em seu caderno de reflexões, estão relacionadas ao fato de que "*ele facilita a programação, por ter um bom suporte criativo e por ser simples de utilizar. Eu me divirto tentando criar os projetos. É legal que a gente pode*

ir testando os comandos e se não dá certo tira fora e tenta de outra maneira" (**E2**). O estudante também destacou a importância da comunidade do *Scratch* e a grande quantidade de jogos disponíveis, segundo ele *"achei que ajudou bastante foi poder testar outros jogos no Scratch e ver os comandos que foram usados. [...] Aprendi muito olhando e tentando entender os comandos que foram usados para criar os jogos"* (**E2**).

A maioria dos participantes afirmou ter utilizado o *Scratch* fora do horário da oficina conforme resposta dada à respectiva pergunta no questionário pós-oficina[47]. Esses acessos tinham como objetivo conhecer melhor o ambiente e seus comandos, dar sequência aos projetos em andamento ou repensar alguma das tarefas. As referidas respostas estão representadas no Gráfico 3 e demonstram o interesse dos estudantes em conhecer mais o ambiente, avançando na sua exploração, além do desejo de desenvolver e aperfeiçoar seus projetos. Cabe ressaltar que um dos participantes não tinha acesso a computador fora da escola.

Gráfico 3- Utilização do Scratch pelos participantes fora do horário da oficina.



Fonte: criado pelo autor (2018)

[47] Conforme resposta da questão número 8 do questionário: https://goo.gl/forms/Yt6YXQisImPzYJQJ3

A análise da trajetória dos estudantes considerando a avaliação do *Dr. Scratch* e do pesquisador, com base no *framework* de Brennan e Resnick e à luz da teoria vigotskiana, além dos questionários e anotações do pesquisador e dos participantes, possibilitou-nos interpretar e dar sentido aos dados gerados. Revelou indicativos que contribuíram para construirmos a resposta à pergunta de pesquisa.

Permitiu-nos, também, considerar que o *Scratch* cumpre seu papel de mediador do processo de aprendizagem de programação e contribui para o desenvolvimento de princípios do pensamento computacional. Dessa maneira, possibilita a elaboração de situações-problema que favorecem a construção de conhecimentos pelo sujeito, fornecendo um sistema de objetos, relações e operadores para expressar e resolver problemas. Entretanto, nesse processo de aprendizagem, o professor tem papel fundamental, tanto na elaboração das situações-problema como na mediação baseada na ZDP de cada participante.

A metodologia empregada na oficina mostrou-se eficaz, pois os participantes se envolveram, demonstrando empenho na exploração do *Scratch* e na busca por realizar as tarefas. A forma como a oficina foi conduzida possibilitou que cada participante desse asas à sua imaginação, exercendo livremente sua criatividade e implementando no ambiente o fruto de sua imaginação.

6 CONSIDERAÇÕES FINAIS

A tecnologia pode ser considerada elemento central na sociedade contemporânea, influenciando todas as áreas do conhecimento e sendo cada vez mais necessária para proporcionar agilidade e eficiência nas ações e nas comunicações. Com isso, provoca transformações na forma de agir, de relacionar e de comunicar da sociedade como um todo.

A educação é uma das áreas do conhecimento que também sofreu essas mudanças, refletindo na maneira de atuar e de pensar dos educadores, os quais necessitaram rever suas práticas pedagógicas. Uma das técnicas que vem sendo estudada e que busca consolidação é a possibilidade de utilizar a tecnologia como mediadora do conhecimento.

Papert (1980) foi o precursor do uso de computadores na educação e seu legado está sendo perpetuado pelas pesquisas de Mitchel Resnick e do grupo do *Lifelong Kindergarten*[48], em conjunto com usuários e desenvolvedores de *softwares* para a educação em todo o mundo, buscando encontrar o equilíbrio entre educação e uso da tecnologia. Nesse sentido, esse trabalho buscou avaliar o ambiente *Scratch* como elemento mediador da aprendizagem de programação para alunos de primeiro ano do Ensino Médio.

Ao analisarmos o estudo piloto e a oficina dessa pesquisa podemos destacar que ambos tiveram pontos positivos e negativos. O piloto, realizado em uma turma de 31 estudantes, possibilitou que visualizássemos diversos momentos de sociointeração entre os participantes. O fato dos estudantes já se conhecerem, de existirem grupos dentro da turma pode ter contribuído para que houvesse mais diálogo, ajuda e troca de informações entre eles.

Na oficina, entretanto, percebemos uma menor interação entre os participantes, geralmente só ocorrendo com o colega sentado ao lado. Houve alguns momentos em que os participantes se deslocaram pelo laboratório para auxiliar outros colegas, porém no piloto isso foi mais intenso e frequente. A Figura 58 ilustra o

[48] O grupo *Lifelong Kindergarten* desenvolve novas tecnologias e atividades que, no espírito dos blocos e pintura de dedo do jardim de infância, envolvem pessoas em experiências criativas de aprendizagem. Mais informações em: https://www.media.mit.edu/groups/lifelong-kindergarten/overview/

ambiente criado para a oficina e apresenta um momento de sociointeração, no qual o participante E3 se deslocou até o computador onde estava o estudante E5 para auxiliá-lo na inserção de som em seu projeto.

Figura 58- Cenário da oficina com um exemplo de sociointeração.

Fonte: criado pelo autor (2017)

Muitos alunos afirmaram ter interesse em frequentar a oficina, porém não tinham condições de comparecer no contraturno, refletindo no pequeno número de inscritos. Além disso, a quantidade de participantes variou em cada encontro da oficina, conforme detalhado na seção 3.2.2, motivados por eventos climáticos, provas no contraturno, necessidade de frequentar atendimento de alguma disciplina, dificuldades com o deslocamento, entre outros.

O piloto, por sua vez, teve uma maior regularidade na presença dos alunos, com faltas esporádica. Pelo fato de ter sido realizado em horário regular de aula e com a possibilidade da livre exploração, pode ter contribuído para aguçar a vontade de conhecer e de tentar utilizar o *Scratch*. Assim, percebemos maior interesse dos estudantes em buscar descobrir como desenvolver as atividades e encontrar solução para os erros e dificuldades encontrados. Por outro lado, o elevado número de participantes dificultou a observação por parte do professor, que era muito requisitado para esclarecer dúvidas e prestar auxílios.

Levando em consideração as observações sobre o piloto e a oficina, aliado as diretrizes dos PCNs e da nova BNCC (que enfatizam a necessidade de trabalhar

competências que desenvolvam a criatividade, a criticidade, o pensamento sistêmico e o trabalho em equipe), podemos concluir que a inserção da programação de computadores no currículo do Ensino Médio é um caminho para alcançar tais diretrizes.

Destacamos também o emprego bem sucedido do *Scratch* como elemento mediador no processo de aprendizagem de programação e desenvolvimento de princípios do pensamento computacional, atuando em parceria como o professor. Entretanto, os estudos indicaram a importância de um redimensionamento do papel do professor, saindo do lugar do discurso e passando a atuar como mediador.

Ao avaliarmos o *Scratch,* considerando aspectos pedagógicos e tecnológicos, encontramos nele características de micromundo, o qual é definido por Balacheff (1991) como sendo constituído por dois sistemas: um formal e outro fenomenal. O sistema formal, no sentido matemático, é composto por um conjunto de objetos primitivos, um conjunto de operações elementares e um conjunto de regras exprimindo como as operações podem ser executadas e associadas. Já, o sistema fenomenal é quem determina os comportamentos na interface em relação aos objetos do sistema formal e as operações sobre esses objetos. Esse sistema cria modelos de respostas e retornos para o usuário, de acordo com suas ações e decisões.

O autor também considera que os *softwares* precisam possibilitar a interação do sujeito com o objeto de conhecimento, pois é nesse processo que a construção do saber ocorre, e não simplesmente fazendo a leitura da tela do computador. Os *softwares* educativos denominados micromundos possuem essa característica, o que possibilita diferentes formas de atribuir significado aos conteúdos estudados (BALACHEFF, 2000).

Por meio da interação com o *Scratch,* cada sujeito, a sua maneira e a seu tempo, teve a oportunidade de experienciar nesse ambiente suas ideias e criatividade. Assim foram desenvolvendo outros processos cognitivos, desencadeando, por conseguinte, a construção de novas estruturas de raciocínio sobre programação de computadores. Nesse sentido, Webber et al. (2016) consideram que "o *Scratch* é extremamente válido pois permite a manipulação de seus objetos na tela e a simulação de múltiplas execuções" (*ibidem*, 2016, p. 8).

Um micromundo também pode ser definido como "uma linguagem de programação que permita a criação de objetos que poderão ser modificados pelo aluno. A interação deve possibilitar a construção de novos objetos e, a partir dessas construções concretas, acontecerão as mentais" (BARROS; STIVAN, 2012, p. 189). Para Balacheff (2000), conforme os conhecimentos do estudante se ampliam, o micromundo também progride. O *software* do tipo micromundo, como é o *Scratch,* possibilita a exploração e a investigação do estudante, que irá avançar conforme expande suas aprendizagens.

Para Bellemain (2002), os micromundos e simulações fazem parte da categoria de *software* que segue o modelo construtivista, os quais criam, no âmbito computacional, condições e ferramentas para expressão e resolução de problemas. Nesse modelo, a resolução de problemas e o desenvolvimento conceitual são fortemente interligados, sendo que a resolução de problemas tem um papel central na evolução dos saberes e na construção de conhecimentos pelo sujeito.

O micromundo, por sua vez, colabora para o desenvolvimento conceitual por meio de suas contribuições à resolução de problemas. Em resumo, "o micromundo contribui para a aprendizagem permitindo as manipulações pelo sujeito dos objetos e relações através dos operadores e favorecendo assim a construção dos conhecimentos sobre esses objetos e relações" (*ibidem*, 2002, p. 55).

Nesse sentido, Papert (1980) considera a linguagem de programação LOGO, desenvolvida por ele e sua equipe do MIT e que utiliza uma tartaruga para executar os comandos, um micromundo. "O mundo da tartaruga é um micromundo, um lugar, uma província da terra da matemática, onde certos tipos de pensamentos matemáticos poderiam nascer e crescer com específica facilidade" (*ibidem*, 1980, p. 125, tradução nossa[49]).

No desenvolvimento da linguagem LOGO (que depois serviu de base para a criação do *Scratch*), dois pontos merecem ser destacados: a integração dos comandos simbólicos a uma linguagem estruturada e o uso da tartaruga como uma integração concreta dos objetos matemáticos. A incorporação dos comandos simbólicos a uma linguagem estruturada (linguagem de programação) possibilita

[49] Do original: "The turtle world is a microworld, a place, a province of the mathland where certain kinds of mathematical thinking could hatch and grow with particular ease." (PAPERT, 1980, p.125).

explorar a comunicação com o computador, sendo essa uma forma de problematizar a maneira de especificar conhecimentos e organizar o pensamento.

Na visão de Papert (1980), o uso da linguagem estruturada pode favorecer a elaboração de situações de formulação. Assim, o objetivo do uso da tartaruga como uma integração concreta dos objetos matemáticos é facilitar a elaboração de significados das reações e das respostas às ações. Para o sujeito a significação das ações e das reações é um elemento crucial para a efetivação de aprendizagens.

Em estudo realizado por Webber et al. (2016), foi realizada uma avaliação do *Scratch* seguindo critérios pedagógicos e tecnológicos. Para isso, foi utilizado um instrumento de avaliação de *software* educativo, composto por questões relacionadas a aspectos pedagógicos e computacionais. Sob a perspectiva pedagógica, as avaliações foram unânimes em considerar que o *Scratch* favorece a utilização interdisciplinar e é adequado ao nível do aprendiz. Já, sobre a proposição, por parte do *software* analisado, de situações-problemas que envolvam a formulação de hipóteses, a investigação e/ou a comparação, metade dos professores pesquisados respondeu que o *Scratch* atende parcialmente a essa questão, enquanto a outra metade considerou que não se aplica a ele.

Outra questão pedagógica do estudo de Webber et al. (2016) visou verificar se o *software* motiva o usuário a buscar outras informações em diferentes fontes de pesquisa. Metade dos participantes julgou que o *Scratch* motiva essa busca enquanto a outra metade dos avaliadores considerou que instiga parcialmente. A última questão desse grupo buscou saber se o *software* oferece atividades variadas, diversificando os níveis de complexidade das tarefas. Os avaliadores divergiram, sendo que a maioria deles concluiu que essa questão não se aplica ao *Scratch*, enquanto outros consideraram que o *software* não apresenta essas características.

No aspecto tecnológico, o estudo de Webber et al. (2016) indica que os avaliadores foram unânimes na maioria das questões, afirmando que o *Scratch* cumpre os principais requisitos, possuindo uma interface amigável, de qualidade e adequada ao público-alvo, fazendo bom uso de imagens e animações. Também destacaram que o *Scratch* possui ajudas e dicas que facilitam a utilização e o esclarecimento de dúvidas, bem como opera de acordo com as instruções fornecidas na documentação. Além disso, ele possui bom tratamento de erros, ou seja, quando

um erro ocorre o sistema consegue contorná-lo e possibilita ao usuário cancelar a operação ou retornar a um ponto anterior.

Segundo Webber et al. (2016), nos aspectos tecnológicos os avaliadores discordaram apenas em alguns aspectos, quais sejam, com relação a adequação das imagens utilizadas pelo *software* aos conteúdos; clareza e precisão das instruções fornecidas pelo programa; adequação da linguagem e vocabulário ao público-alvo, ao tópico/conteúdo e características regionais; e, utilização do *software* por parte do aluno sem a interferência do professor. Os principais problemas destacados foram em relação a alguns termos que não são usuais para crianças e pelo fato de algumas palavras não terem sido traduzidas do inglês. Também ressaltaram a necessidade de levar em consideração a idade do aluno e seus conhecimentos prévios.

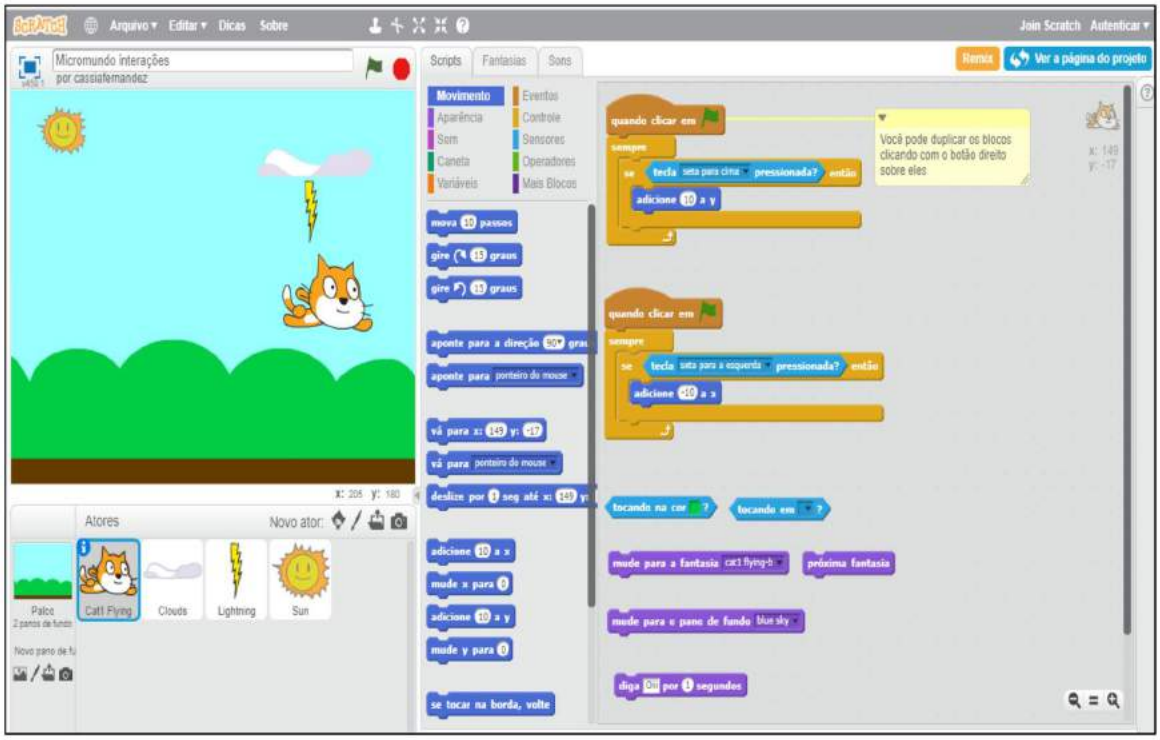
Na última etapa, os autores realizaram uma avaliação epistemológica do *Scratch*, ressaltando sua contribuição para a educação, possibilitando novas formas de representação de objetos e a exploração dinâmica de diferentes situações educativas. Enfatizaram a simulação como característica principal, pois compreende a experimentação, os ensaios e os erros, valorizando a operacionalidade e eficiência. "O *Scratch* foi considerado um *software* que possibilita a exploração e representação de diferentes objetos e lógicas expressas pelo estudante, por meio dos comandos dados pelo mesmo" (WEBBER et al., 2016, p. 8).

Com relação aos conhecimentos que o *software* possibilita ao aluno construir, Webber et al. (2016) trazem como aspecto relevante a ser considerado a possibilidade de interação e identificação dos caminhos percorridos pelo estudante, fornecendo respostas sobre as atividades realizadas e retorno sobre seu desempenho. Os autores consideraram que o *Scratch* atende ao requisito de interação e identificação dos caminhos percorridos, uma vez que "permanecem registrados e visíveis todos os comandos feitos pelo estudante no decorrer do uso do software, permitindo que se reavalie, a todo momento, suas ações e os resultados dela decorrentes" (*ibidem*, 2016, p. 8).

O ambiente *Scratch,* além de possuir as principais características de um micromundo, também possibilita a criação de micromundos, tanto para apresentar o próprio *Scratch* como para introduzir conceitos de programação. Além disso, possibilita a criação dos mais variados tipos de desafios, pois limita a quantidade de

recursos disponíveis, permitindo com isso direcionar o foco da atenção do usuário para um número restrito de funções. A Figura 59 ilustra o ambiente de programação padrão do *Scratch*. Nesse ambiente é possível acessar todos as categorias dos blocos de comando (faixa central da tela), as abas de fantasias e sons (faixa central da tela), além de poder incluir novos atores (região inferior esquerda da tela) e planos de fundo (região inferior esquerda da tela).

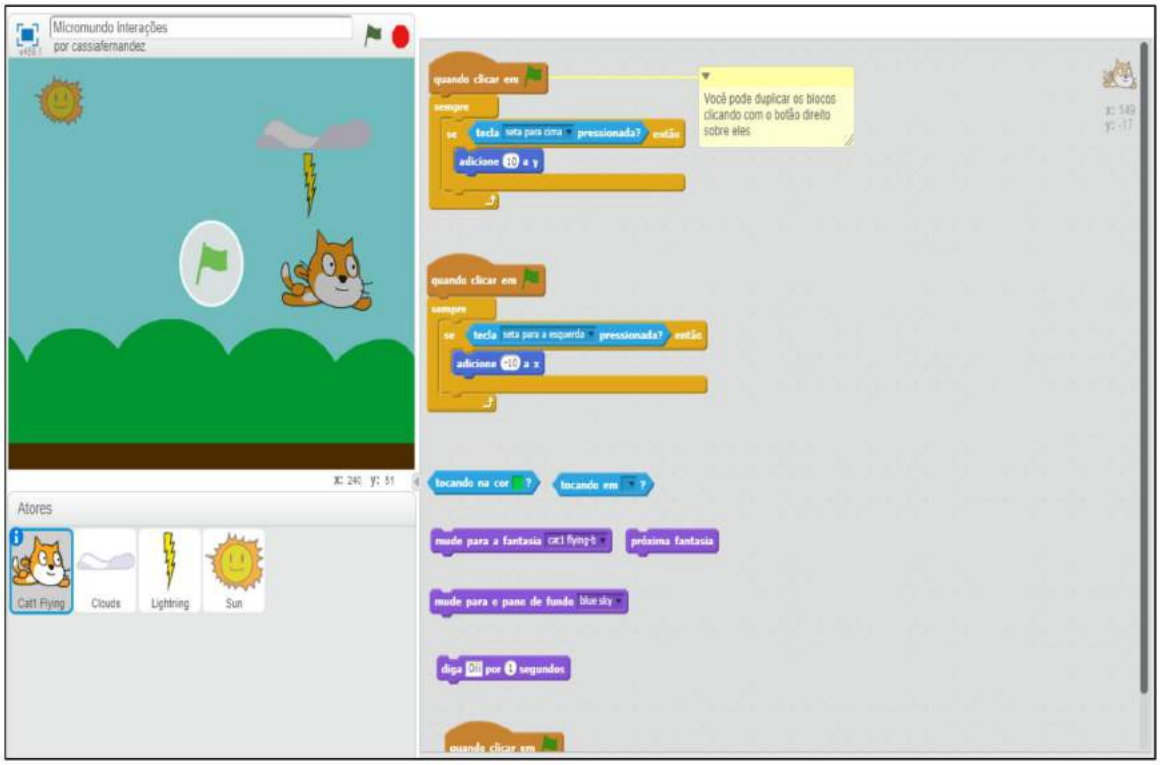
Figura 59- Ambiente de programação padrão do *Scratch*.



Fonte: criado pelo autor (2018). Projeto compartilhado na comunidade *Scratch* - https://scratch.mit.edu

A Figura 60 representa um exemplo de micromundo, na qual o ambiente *Scratch* está com suas funções restritas. Nessa imagem é possível identificar a ausência da barra de menu, das ferramentas de cursor, da paleta de blocos de comandos, de fantasias, de sons, de inclusão de personagens e planos de fundo. Com isso observamos o ambiente fica mais enxuto e simplificado.

Figura 60- Exemplo de emprego do *Scratch* como micromundo.



Fonte: criado pelo autor (2018). Projeto compartilhado na comunidade *Scratch* - https://scratch.mit.edu

Um exemplo de micromundo criado com o *Scratch,* denominado de "Plano Cartesiano", está ilustrado na Figura 61. Nessa imagem observamos no lado esquerdo o desenho de um plano cartesiano e, no lado direto, a proposta de um desafio, com suas regras e instruções. Ao acessar esse projeto o usuário visualiza, assim como representado na Figura 59, um conjunto específico de comandos para criar uma solução para o desafio proposto.

Figura 61- Exemplo de micromundo - Plano Cartesiano

Fonte: criado pelo autor (2018). Projeto compartilhado na comunidade *Scratch* - https://scratch.mit.edu

Na comunidade do *Scratch* é possível localizar diversos exemplos de micromundos, os quais estão disponíveis para teste e utilização. A possibilidade de utilização do *Scratch* como micromundo e para a criação de outros, potencializa e facilita a introdução de comandos e conceitos de programação. Esse recurso amplia as possibilidades de emprego do ambiente, podendo ser integrado aos mais diversos conteúdos e disciplinas. Além disso, aliado ao que foi elencado na análise do *corpus,* contribui para reforçar nossa convicção que o *Scratch* possibilita a mediação do aprendizado de programação.

Por fim, cabe apresentar um dos frutos dessa pesquisa. No mês de janeiro de 2018, durante as férias escolares, um dos participantes da oficina entrou em contato com o pesquisador demonstrando interesse em desenvolver outros trabalhos com o uso do *Scratch*. Sua proposta é desenvolver alguns jogos e animações para facilitar a compreensão de determinados conteúdos de química. O estudante também deseja participar do "Prêmio Mercosul de Ciência e Tecnologia" e gostaria que o pesquisador fosse seu orientador na elaboração de um projeto para esse evento.

Como trabalhos futuros, pretendemos desenvolver outras oficinas de programação com *Scratch*, tanto para alunos do Ensino Médio como para professores. Na próxima oficina, pretendemos utilizar o recurso de micromundo para focar e simplificar a exploração do ambiente.

Outro trabalho que planejamos desenvolver é a inserção do *Scratch* na disciplina de Matemática Computacional (para alunos de Licenciatura em Matemática). Pretendemos com isso, habilitar os futuros professores a utilizar o ambiente para montarem jogos e atividades voltadas aos conteúdos de matemática, além de capacitá-los a difundirem os principais conceitos de programação.

Os resultados do estudo aqui apresentado indicaram que o *Scratch* tem potencial para mediar o aprendizado de programação no Ensino Médio, apoiando o desenvolvimento de habilidades relacionadas ao pensamento computacional. Com isso, esperamos contribuir para as reflexões sobre a inserção, no currículo da Educação Básica, de práticas que promovam o pensamento crítico, criativo e sistêmico, com vistas ao desenvolvimento do pensamento computacional.

REFERÊNCIAS

ACM. **Association for Computing Machinery.** Disponível em: <http://www.acm.org/>. Acesso em: 08 jun. 2016.

AMARAL, Érico Marcelo Hoff do. **Processo de ensino e aprendizagem de algoritmos integrando ambientes imersivos e o paradigma de blocos de programação visual.** 2015. 255 f. Tese (Doutorado) - Curso de Programa de Pós-graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2015. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/131053/000978112.pdf>. Acesso em: 29 mar. 2016.

ANDRADE, Gabriel. **O que são Linguagens de Programação.** InfoEscola. Disponível em: <http://www.infoescola.com/informatica/o-que-sao-linguagens-de-programacao/>. Acesso em: 08 nov. 2015.

BALACHEFF, Nicolas. **Entornos informaticos para la enseñanza de las matemáticas:** complejidad didactica y expectativas. 2000. Disponível em: http://www.cvrecursosdidacticos.com/download_file.php?file=1288115856_U2_Balacheff.pdf. Acesso em: 18 jan. 2018.

_____. **Contribution de la didactique et de l'épistémologie aux recherches em EIAO**, actes des 13ème Journées Francophones sur l'Informatique, Formation Intelligemment Assitée par Ordinateur, Genève, páginas 9-38. 1991. Disponível em: <https://hal.archives-ouvertes.fr/hal-01027408/document>. Acesso em: 19 jan. 2018.

BARROS, Ana Paula Rodrigues Magalhães de; STIVAM, Elen Priscila. O software GeoGebra na Concepção de Micromundo. **Revista do Instituto Geogebra Internacional de São Paulo**, São Paulo, v. 1, n. 1, p.184-194, jan. 2012. Semestral. Disponível em: <http://revistas.pucsp.br/index.php/IGISP/article/view/8388>. Acesso em: 27 jan. 2018.

BELLEMAIN, Franck. O Paradigma Micromundo. **História e Tecnologia no Ensino da Matemática**, Rio de Janeiro, v. 1, n. 1, p.51-63, 2002. Luiz M. Carvalho e Luiz. C. Guimarães (organizadores). Editora IME-UERJ. Disponível em: <https://www.academia.edu/3217386/O_paradigma_micromundo>. Acesso em: 18 jan. 2018.

BERRY, Miles. **CS For All: the English Experience.** 2016. Disponível em: <https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Voice/csta_voice_09_2016.pdf>. Acesso em: 01 set. 2016.

BOSSE, Yorah; GEROSA, Marco Aurélio. **Reprovações e Trancamentos nas Disciplinas de Introdução à Programação da Universidade de São Paulo:** Um Estudo Preliminar. Instituto de Matemática e Estatística – Departamento de Ciência da Computação - Universidade de São Paulo (USP). 2015. Disponível em:

<http://lapessc.ime.usp.br/public/papers/18929/Artigo.pdf>. Acesso em: 20 mar. 2017.

BRASSCOM. **Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação**. Disponível em: <https://brasscom.org.br/>. Acesso em: 18 maio 2016.

BRASSCOM. **Internet das Coisas.** 2013. Disponível em: <https://brasscom.org.br/wp-content/uploads/2017/08/internet_das_coisas_final.pdf>. Acesso em: 18 maio 2016.

BRASSCOM. **O Mercado de Profissionais de Tecnologia da Informação e Comunicação no Brasil:** uma análise do período de 2006 a 2013. 2015. São Paulo. Disponível em: <https://brasscom.org.br/wp-content/uploads/2017/08/estudo_profissionais_tic_brasil_2006_2013.pdf>. Acesso em: 18 maio 2016.

BRASIL. Leis etc. Lei n. 9394 de 20 de dezembro de 1996. **Estabelece as Diretrizes e Bases da Educação Nacional.** Diário Oficial da União. Brasília, 1996.

_____. Ministério da Educação. Secretaria de Educação Média e Tecnológica. **Parâmetros Curriculares Nacionais para o Ensino Médio** - Bases Legais. Brasília: MEC/SEMTEC, 2000.

_____. Ministério da Educação. **Base Nacional Comum Curricular – BNCC 2ª versão**. Brasília, DF, 2017. Disponível em: <http://basenacionalcomum.mec.gov.br/wp-content/uploads/2018/02/bncc-20dez-site.pdf>. Acesso em: 12 fev. 2018.

BRENNAN, Karen; RESNICK, Mitchel. **New frameworks for studying and assessing the development of computational thinking**. In: Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada, p 1-25.

COCCHIERI, Tiziana. Conceito de Abdução: Modalidades de Raciocínio Contidas no Sistema Lógico Peirceano. **Clareira**: Revista de filosofia da região amazônica, Rondônia, v. 2, n. 1, p.75-92, jul. 2015. Disponível em: <http://www.revistaclareira.com.br/index.php/clareira/article/view/41/37>. Acesso em: 27 fev. 2017.

CORMEN, Thomas H. et al. **Algoritmos:** Teoria e prática. 2. ed. Rio de Janeiro: Elsevier, 2002. 916 p. Tradução de: Vandernberg D. de Souza. Disponível em: <https://www.cin.ufpe.br/~ara/algoritmos-%20portugu%EAs-%20cormen.pdf>. Acesso em: 23 fev. 2018.

CSTA - Computer Science Teacher Association. **K–12 Computer Science Standards:** The CSTA Standards Task Force. 2011. ACM - Association for

Computing Machinery. Disponível em: <http://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/CSTA_K-12_CSS.pdf>. Acesso em: 07 maio 2016.

CSTA - Computer Science Teacher Association; ISTE – International Society for Technology in Education. **Operational definition of computational thinking:** for k-12 education. 2011. Disponível em: <https://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>. Acesso em: 07 maio 2016.

DIETRICH, Gustavo Luís Vieira. **Scratch:** tornando a programação acessível. 2015. Disponível em: <http://blog.render.com.br/programacao-2/scratch-tornando-a-programacao-acessivel/>. Acesso em: 03 out. 2017.

DINIZ, Ana Maria. **Programação é o novo inglês.** 2017. Estadão - Educação. Disponível em: <http://educacao.estadao.com.br/blogs/ana-maria-diniz/programacao-e-o-novo-ingles/>. Acesso em: 19 mar. 2017.

EDUCATION, U. S. Department Of. **Definitions.** Departamento de Educação dos Estados Unidos. Disponível em: <https://www.ed.gov/race-top/district-competition/definitions>. Acesso em: 25 fev. 2017.

EVANS, Dave. **A Internet das Coisas – Como a próxima evolução da internet está mudando tudo**. Cisco IBSG. 2011. Disponível em: <http://www.academia.edu/12106350/Internet_das_Coisas_-_Cisco>. Acesso em: 13 jan. 2017.

FRANÇA, Rozelma S.; AMARAL, Haroldo J. C.. **Proposta metodológica de ensino e avaliação para o desenvolvimento de pensamento computacional com o uso do Scratch. In:** XIX Workshop de Informática na Escola, 2013, Campinas.

GOMES, Márcio. **Técnica japonesa de gesticular e falar sozinho reduz erros no trabalho.** 2016. Jornal Nacional. Disponível em: <http://glo.bo/2etsimd>. Acesso em: 12 out. 2016.

HAZZAN, O.; GAL-EZER, J.; BLUM, L. "**A model for high school computer science education: the four key elements that make it!**". Proceedings of the 39th SIGCSE technical symposium on Computer science education, 2008, Portland, OR, USA. Pág. 281-285.

HEUVELDOP, Niklas. **Ericsson Mobility Report**. 2017. Disponível em: <www.ericsson.com/mobility-report>. Acesso em: 03 out. 2017.

ISTE. **International Society for Technology in Education.** Disponível em: <https://www.iste.org>. Acesso em: 20 jun. 2016.

KITCHENHAM, B. A. **Guidelines for performing Systematic Literature Reviews in Software Engineering.** Technical report, Computer Science Department, Keele University.

2007. Disponível em: <https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf>. Acesso em: 17 jul. 2016.

KITCHENHAM, Barbara et al. **Systematic literature reviews in software engineering:** a systematic literature review. **In:** Information and Software Technology, v.51, 2009.

LAMIM, Jonathan. **O que é Programação de Computadores.** 2008. Disponível em: <https://www.oficinadanet.com.br/artigo/1310/o_que_e_programacao_de_computadores>. Acesso em: 21 ago. 2016.

LEITÃO, Antônio. **Linguagem de Programação.** 1995. Disponível em: <http://www.dca.fee.unicamp.br/courses/EA072/lisp9596/node2.html>. Acesso em: 23 ago. 2016.

LEMOS, André. **A comunicação das coisas. Internet das coisas e a Teoria de Ator-Rede**. Artigo Seminário SimSocial, 2012. Salvador-BA. Disponível em: <http://www.seminariosmv.org.br/textos/Andre%20Lemos.pdf>. Acesso em: 13 jan. 2017.

LÉVY, Pierre. **A inteligência colectiva:** para uma antropologia do ciberespaço. Lisboa, Portugal: Instituto Piaget, 1997.

_____. **As tecnologias da inteligência.** Rio de Janeiro: Editora 34, 1995.

_____. **Cibercultura**. 2.ed. São Paulo: Ed. 34, 2000.

_____. **O que é virtual?** São Paulo: Ed. 34, 1996.

LIMA, Márcio Roberto. **Construcionismo de Papert e ensino-aprendizagem de programação de computadores no Ensino Superior.** 2009. 143 p. Dissertação (Mestrado) - Universidade Federal de São João Del-Rei, São João Del-Rei, MG.

LOPES, Marina. **Como a internet das coisas pode entrar na escola.** 2016. Disponível em: <http://porvir.org/como-internet-das-coisas-pode-entrar-na-escola/>. Acesso em: 15 fev. 2017.

LUDKE, Menga; ANDRÉ, Marli E.D.A. **Pesquisa em educação:** abordagens qualitativas. São Paulo, Editora Pedagógica e Universitária, 1986.

MARQUES, Maria T. P. M. **Recuperar o engenho a partir da necessidade, com recurso às tecnologias educativas**: contributo do ambiente gráfico de programação Scratch em contexto formal de aprendizagem. Universidade de Lisboa, 2009. Disponível em: <http://hdl.handle.net/10451/847>. Acesso em: 20 abr. 2017.

MARTINS, R. DE Q. **Usando o Scratch para Potencializar o Pensamento Criativo em Crianças do Ensino Fundamental.** Dissertação (Mestrado em Educação) Universidade de Passo Fundo, 2012. Orientador: Professor Dr. Adriano Canabarro

Teixeira. Disponível em: <https://secure.upf.br/pdf/2012AmiltonRodrigodeQuadrosMartins.pdf> Acesso em: 10 out. 2016.

MORAES, Ronny M. de. **Gênese da linguagem na teoria sócio-cultural de vygotsky.** 2010. Universidade Federal do Mato Grosso do Sul. Disponível em: <http://pt.slideshare.net/ronnymm/gnese-do-pensamento-e-da-linguagem-l-s-vygotsky>. Acesso em: 16 out. 2016.

MORENO-LEÓN, Jesús; ROBLES, Gregorio. Analyze your Scratch projects with Dr. Scratch and assess your computational thinking skills. In: **Scratch Conference**. 2015. p. 12-15.

MOREIRA, Marco Antonio. **Teorias de Aprendizagem.** São Paulo: E.P.U. 2ª.ed.ampl. 2011.

MOTA, Fernanda P. et all. Desenvolvendo o Raciocínio Lógico no Ensino Médio: uma proposta utilizando a ferramenta Scratch. In: **XXV Simpósio Brasileiro de Informática na Educação – SBIE**, 2014. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/2964/2698>. Acesso em: 23 abr. 2016.

OLIVEIRA, Emanuelle. **Estudo de Caso.** InfoEscola. Disponível em: <http://www.infoescola.com/sociedade/estudo-de-caso>. Acesso em: 18 set. 2016.

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

_____. **Mindstorms:** Children, Computers, and Powerful Ideas. BasicBooks, New York, 1980.

P21 - Partnership for 21st Century Learning. **Framework for 21st Century Learning.** 2015. Disponível em: <http://www.p21.org/our-work/p21-framework>. Acesso em: 19 maio 2016.

PAPERT, S.; HAREL, I. **Situating constructionism**. In: HAREL, I.; PAPERT, S. "Constructionism". Norwood: Ablex Publishing Corporation, 1991. Disponível em: <http://www.papert.org/articles/SituatingConstructionism.html>. Acesso em: 05 ago. 2016.

PASSOS, Marize Lyra Silva. Scratch: uma ferramenta construcionista no apoio a aprendizagem no século XXI. **Revista Eletrônica Debates**, v. 04, n. 02, p. 68- 85, 2014. Disponível em: <http://ojs.ifes.edu.br/index.php/dect/article/viewFile/123/205>. Acesso em: 25 nov. 2017.

POCRIFKA, Dagmar Heil; SANTOS, Taís Wojciechowski. **Linguagem logo e a construção do conhecimento.** 2009. IX Congresso Nacional de Educação - EDUCERE. Disponível em: <http://www.pucpr.br/eventos/educere/educere2009/anais/pdf/2980_1303.pdf>. Acesso em: 19 fev. 2007.

PRADO, Maria Elisabette Brisola Brito. **O Uso do Computador na Formação do Professor**: um enfoque reflexivo da prática pedagógica. Coleção Informática para a Mudança na Educação. Brasília: MEC/SED, 1999.

RAMOS, José L.; ESPADEIRO, Rui G. **Pensamento computacional na escola e práticas de avaliação das aprendizagens.** Uma revisão sistemática da literatura. 2015. Disponível em: <http://hdl.handle.net/10174/18147>. Acesso em: 23 mai. 2016.

RESNICK, Mitchel. **O computador como pincel**. In: VEJA. Limpeza de alto risco. Especial: um guia do mundo digital. São Paulo: Abril Cultural, n. 41, out. 2006.

_____. Technologies for Lifelong Kindergarten. **Educational Technology Research & Development**. Berkshire, v. 46, n. 4, p.21-33, abr. 1998. Disponível em: <https://llk.media.mit.edu/papers/llk/>. Acesso em: 13 jun. 2016.

RESNICK, Mitchel et al. Scratch: Programming for All. **Communications of the ACM**, v. 52, n. 11, p.60-67, 2009.

SANTOS, Victor. A internet das coisas e a TAR: onde está o homem? **Revista Temática, João Pessoa**, v. 9, n. 10, p.1-15, outubro 2013. Disponível em: <http://docplayer.com.br/5061389-A-internet-das-coisas-e-a-tar-onde-esta-o-homem.html>. Acesso em: 04 mar. 2017.

SAUVÉ, Jacques Philippe. **Frameworks.** Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm>. Acesso em: 20 set. 2017.

SCAICO, Pasqueline D. et all. Ensino de Programação no Ensino Médio: Uma Abordagem
Orientada ao Design com a linguagem Scratch. **Revista Brasileira de Informática na Educação**, Volume 21, Número 2, 2013.

SOARES, Eliana Maria do Sacramento; MASCHIO, Elaine Cátia Falcade. Práticas, representações e mediação: o uso dos laptops educacionais e as intervenções docentes no processo de aprendizagem da educação básica. **Revista Ibero-Americana de Estudos em Educação**, Araraquara, v. 12, n. esp. 2, p. 1372-1390, ago./2017.

SOUZA, Paulo Roberto de Azevedo; DIAS, Lucimeri Ricas. **Kodu Game Lab:** Estimulando o Raciocício Lógico através de Jogos. Rio de Janeiro, 2012. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/1733>. Acesso em: 12 ago. 2015.

_____. **Kodu Game Lab Brasil:** Apresentação e reflexão sobre os jogos criados e publicados na comunidade Kodu BR. São Paulo, 2013. Disponível em: <http://www.sbgames.org/sbgames2013/proceedings/cultura/Culture-9_full.pdf >. Acesso em: 27 ago. 2015.

VALENTE, José Armando. **A Espiral da Espiral de Aprendizagem:** o processo de compreensão do papel das tecnologias de informação e comunicação na educação. 2005. 232 f. Tese (Doutorado Livre Docência) - Curso de Instituto de Artes, Universidade Estadual de Campinas, Campinas, 2005. Disponível em: <http://www.bibliotecadigital.unicamp.br/document/?code=000857072&fd=y>. Acesso em: 12 set. 2016.

VIGOTSKI, Lev Semenovich. **Pensamento e Linguagem**. Tradução de Jefferson Luis Camargo. São Paulo: Martins Fontes, 1987.

_____. **A formação social da mente:** o desenvolvimento dos processos psicológicos superiores. 7. ed. São Paulo: Martins Fontes, 2007. Org. Michael Cole.

WAZLAWICK, Raul S., **Metodologia de Pesquisa para Ciência da Computação**, Editora Campus, Elsevier, 2009.

WEBBER, Carine G. et al. Reflexões sobre o software scratch no ensino de ciências e matemática. **Renote - Revista Novas Tecnologias na Educação,** Porto Alegre, v. 14, n. 2, p.1-10, dez. 2016. Semestral. Disponível em: <http://seer.ufrgs.br/index.php/renote/article/view/70668/40105>. Acesso em: 16 set. 2017.

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

YIN, Robert K. **Estudo de caso:** planejamento e métodos. 3 ed. Porto Alegre: Bookman, 2005.

ZORZO, Avelino Francisco. **Computação transformando a sociedade.** 2017. Diretor de educação da Sociedade Brasileira de Computação (SBC). Disponível em: <http://www.baguete.com.br/noticias/19/01/2017/computacao-transformando-a-sociedade>. Acesso em: 19 mar. 2017.

APÊNDICE A – FORMULÁRIO PESQUISA CURSO DE PROGRAMAÇÃO

Pesquisa - Curso de Programação

Levantamento para caracterização dos participantes

*Obrigatório

Endereço de e-mail *

Seu e-mail

1. Nome Completo *

Sua resposta

2. Data de nascimento *

Data

dd/mm/aaaa

3. Sexo *

- ◯ Masculino
- ◯ Feminino

4. Turno das aulas *

- ◯ Manhã
- ◯ Tarde

5. Qual série está cursando? *

- ◯ 1º Ano do Ensino Médio
- ◯ 2º Ano do Ensino Médio
- ◯ 3º Ano do Ensino Médio

PRÓXIMA

Pesquisa - Curso de Programação

*Obrigatório

Informações sobre o uso de tecnologia

6. Possui computador em casa? *

- ◯ Sim - computador de mesa (desktop)
- ◯ Sim - notebook/netbook/ultrabook
- ◯ Sim- computador de mesa (desktop) e também notebook/netbook/ultrabook
- ◯ Não possuo

7. Possui Celular/Smartphone/iPhone? *

- ◯ iPhone com acesso à internet
- ◯ iPhone sem acesso à Internet (somente WI-FI)
- ◯ Smartphone com acesso à Internet
- ◯ Smartphone sem acesso à Internet (somente WI-FI)
- ◯ Celular sem ser smartphone
- ◯ Não possuo celular

8. Possui Tablet/iPad? *

- ◯ Tablet
- ◯ iPad
- ◯ Não possuo

9. Quando utiliza a Internet, o que costuma fazer?

- [ ] Acessar as redes sociais (como o Facebook, por exemplo)
- [ ] Assisitir filmes
- [ ] Ouvir música
- [ ] Jogar (jogos individuais)
- [ ] Jogar (jogos em rede, dois ou mais jogadores simultaneamente)
- [ ] Ficar informado sobre as notícias, novidades (jornais, revistas, etc)
- [ ] Acessar e-mail
- [ ] Fazer pesquisas e trabalhos escolares
- [ ] Diversão e passatempo (piadas, memes, textos diversos, etc)
- [ ] Estudar ou conhecer sobre algum tema ou assunto
- [ ] Pesquisas de preços/compra de produtos
- [ ] Criar páginas Web ou blogs
- [ ] Outro: _____

10. Escolha a(s) categoria(s) de jogo(s) que mais lhe atrai:

- [ ] Ação
- [ ] Shooter (Atirador)
- [ ] Aventura
- [ ] Construção/Gerenciamento
- [ ] Simulação de Vida
- [ ] Música/ritmo
- [ ] Festa
- [ ] Quebra-cabeças
- [ ] Esportes
- [ ] Estratégia
- [ ] RPG
- [ ] Simulação de Veículo
- [ ] Outro: _____

11. Você já fez algum curso de programação de computadores? *

- ◯ Sim
- ◯ Não

12. Em caso afirmativo, como foi o curso? Gostou?

Sua resposta

13. Em caso negativo, gostaria de aprender a programar?

- ◯ Sim
- ◯ Não
- ◯ Talvez
- ◯ Outro:

14. Você considera importante para os alunos do ensino médio aprenderem a programar computadores? *

- ◯ Sim, todos os estudantes deveriam aprender
- ◯ Sim, como uma disciplina optativa (não obrigatória)
- ◯ Não no Ensino Médio, só na graduação
- ◯ Não considero importante

15. Descreva o que você sabe sobre Pensamento Computacional ou o que você acha que é. *

Sua resposta

VOLTAR ENVIAR

Segue o link para o formulário on-line: https://goo.gl/forms/BY9blNYIN17ICMG83

APÊNDICE B – FORMULÁRIO DE AVALIAÇÃO DA OFICINA

Avaliação da Oficina com Scratch

*Obrigatório

Endereço de e-mail *

Seu e-mail

1- Como você avalia a oficina utilizando o Scratch? *

- ◯ Excelente
- ◯ Boa
- ◯ Regular
- ◯ Não gostei

2- Nessas aulas não foi empregada uma metodologia tradicional (aulas expositivas, com o professor explicando passo a passo). Com relação à metodologia empregada: *

- ◯ Gostei e foi adequada para conhecer o básico de programação
- ◯ Gostei mas acho que a metodologia tradicional teria propiciado um aprofundamento maior
- ◯ Gostei, mas acho que com mais explicações e exemplos ficaria mais fácil
- ◯ Não gostei, prefiro aulas expositivas (metodologia mais tradicional)
- ◯ Outro:

3- O que você achou do Scratch com relação a: *

	Excelente	Muito Boa	Boa	Regular	Não Gostei
Sua interface gráfica	○	○	○	○	○
Disponibilidade de recursos multimídia	○	○	○	○	○
Compreensão dos comandos	○	○	○	○	○
Possibilidade de testar os programas a fim de localizar os erros	○	○	○	○	○
Compartilhamento dos programas	○	○	○	○	○

4- Você teria interesse em aprofundar os conhecimento em Programação utilizando o Scratch? *

- ○ Sim
- ○ Não
- ○ Talvez

5- Na sua opinião o aprendizado de programação deveria fazer parte do currículo do Ensino Médio? *

- ○ Sim, mas só no Ensino Médio Técnico
- ○ Sim, em todas as escolas
- ○ Deveria ser opcional
- ○ Não

6- Relacione a importância que cada elemento teve para o aprendizado de programação e realização das tarefas. *

	Fundamental	Ajudou Bastante	Ajudou Algumas Vezes	Ajudou Bem Pouco	Não Ajudou	Não Utilizei
O ambiente do Scratch	○	○	○	○	○	○
Os programas disponibilizados na Comunidade do Scratch	○	○	○	○	○	○
Os vídeos explicativos sobre o assunto	○	○	○	○	○	○
O professor	○	○	○	○	○	○
Os colegas	○	○	○	○	○	○
Outros materiais (apostilas, tutoriais, etc)	○	○	○	○	○	○

7- Qual foi o fator que mais motivou no desenvolvimento das tarefas propostas? *

Sua resposta

8- Você utilizou o Scratch fora do horário das aulas? *

- ○ Sim, diversas vezes
- ○ Sim, algumas vezes
- ○ Raramente
- ○ Não

9- Relacione os Conceitos abaixo com o nível de utilização em seus programas: *

	Muitas Vezes	Algumas Vezes	Poucas Vezes	Nunca Usado	Não Sei
Sequências: expressar uma tarefa em uma série de passos	○	○	○	○	○
Laços (loops): mecanismo utilizado para repetição de sequências	○	○	○	○	○
Eventos: situações que dão início a outras	○	○	○	○	○
Paralelismo: sequências que acontecem ao mesmo tempo	○	○	○	○	○
Condicionais: habilidade de fazer decisões baseado em certas condições	○	○	○	○	○
Operadores matemáticos	○	○	○	○	○
Operadores relacionais	○	○	○	○	○
Operadores lógicos	○	○	○	○	○
Dados: elementos que o computador armazena, recupera e atualiza	○	○	○	○	○

10- Que mudanças você acha importante serem feitas na condução da oficina e/ou nas atividades propostas? *

Sua resposta

ENVIAR

Segue o link para o formulário on-line: https://goo.gl/forms/yMTeFlbrxGTUXiXt2

APÊNDICE C – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

O termo de consentimento livre e esclarecido foi entregue ao responsável pelo participante da oficina, o qual assinou uma das vias, autorizando a participação de seu filho(a) na pesquisa, e a devolveu ao pesquisador.

Projeto de pesquisa: desenvolvimento do pensamento computacional no Ensino Médio: aprendizagem de programação mediada por uma linguagem visual.

Pesquisador responsável: Esp. Leonardo Poloni

Orientadora: Dra Eliana Maria do Sacramento Soares.

Instituição: Universidade de Caxias do Sul

Objetivo: Identificar e analisar as formas de mediação possibilitadas pelo *Scratch* no processo de ensino aprendizagem de programação no Ensino Médio.

Benefício: O benefício da pesquisa é produzir novos conhecimentos que poderão auxiliar nas estratégias de ensino e aprendizagem de programação, principalmente para alunos do Ensino Médio. A pesquisa busca também analisar uma ferramenta de aprendizagem de programação e sua relação com o desenvolvimento do pensamento computacional.

Risco: Mínimo, pois o estudo constitui-se de análise de programas desenvolvidos pelos estudantes, anotações dos alunos e observações do pesquisador. No entanto, as atividades e observações podem gerar desconforto nos participantes e estes podem desistir da participação na pesquisa a qualquer momento.

Procedimentos: Para analisar as interações de seu filho, serão realizadas observações no laboratório de informática da escola. Serão cinco momentos de três horas cada um. Os programas desenvolvidos por seu filho e suas anotações serão enviadas ao pesquisador para posterior análise.

Alternativas: A autorização é voluntária e a recusa em autorizar não acarretará qualquer penalidades. Seus dados pessoais não serão identificados em nenhuma publicação que resultar deste estudo. Aefetivação do envolvimento com esta pesquisa somente se dará a partir da assinatura deste termo, com o qual estará consentindo em participar do trabalho, sendo-lhe reservado o direito de recusar-se a participar ou de desistir de sua participação a qualquer momento. Sua desistência ou não participação não irá prejudicá-lo e os dados obtidos a partir das coletas realizadas com você até o momento de sua desistência serão descartados.

Confidencialidade: O pesquisador e sua orientadora certificaram-me que todos os dados desta pesquisa serão confidenciais e as informações publicadas somente em eventos ou periódicos científicos. De acordo com a Resolução 466/12, fica garantido o sigilo absoluto das informações obtidas nessa pesquisa, sendo a guarda dos materiais produzidos neste estudo de inteira responsabilidade do pesquisador. Após o período de cinco anos os programas e anotações serão descartados.

Problemas e perguntas: O pesquisador - Leonardo Poloni - é graduado em Ciência da Computação, é Especialista em Redes de Computadores e é mestrando do programa de Pós-graduação em Educação da Universidade de Caxias do Sul, sob orientação da professora Dr.ª Eliana Maria do Sacramento Soares, e se compromete a esclarecer devida e adequadamente qualquer dúvida ou necessidade de informações que o participante, venha a ter no momento da pesquisa ou posteriormente través do telefone: 54 99937-4323 ou por e-mail lpoloni@ucs.br, e/ou o Comitê de Ética em Pesquisa pelo telefone 54 3218-2829.

Custos: Sua participação no estudo não acarretará custos, assim como não será disponibilizada nenhuma compensação financeira.

Termo de Consentimento

Após ter sido devidamente informado/a de todos os aspectos da pesquisa e ter esclarecido todas as minhas dúvidas, concordo em autorizar a participação do meu filho(a) na referida pesquisa e participar das atividades propostas.

Nome legível do responsável pelo estudante: _____

Assinatura do responsável: _____

Atesto que expliquei a natureza e o objetivo de tal estudo, bem como os possíveis riscos e benefícios do mesmo, junto ao participante. Acredito que ele recebeu todas as informações necessárias que foram fornecidas em uma linguagem adequada e compreensível e que o (a) participante compreendeu tal explicação.

Endereços para contato:

E-mail: _____

Fone: _____

Pesquisador responsável:

Nome legível: Leonardo Poloni

Assinatura: _____

Este Projeto está vinculado ao Comitê de Ética em Pesquisa da Universidade de Caxias do Sul, sob o endereço Rua Francisco Getúlio Vargas, 1130, Bloco A, sala 302B, Caxias do Sul, RS, CEP: 95070-560, fone(54) 32182100 ramal, 2829.

APÊNDICE D – TERMO DE ASSENTIMENTO LIVRE E ESCLARECIDO

O termo de assentimento livre e esclarecido foi entregue aos participantes da oficina, os quais assinaram e devolveram ao pesquisador uma das vias, concordando assim em participar da pesquisa.

Projeto de pesquisa: desenvolvimento do pensamento computacional no Ensino Médio: aprendizagem de programação mediada por uma linguagem visual.

Pesquisador responsável: Esp. Leonardo Poloni

Orientadora: Dra. Eliana Maria do Sacramento Soares.

Instituição: Universidade de Caxias do Sul

Objetivo: Identificar e analisar as formas de mediação possibilitadas pelo *Scratch* no processo de ensino aprendizagem de programação no Ensino Médio.

Benefício: O benefício da pesquisa é produzir novos conhecimentos que poderão auxiliar nas estratégias de ensino e aprendizagem de programação, principalmente para alunos do Ensino Médio. A pesquisa busca também analisar uma ferramenta de aprendizagem de programação e sua relação com o desenvolvimento do pensamento computacional.

Risco: Mínimo, pois o estudo constitui-se de análise de programas desenvolvidos pelos estudantes, anotações dos alunos e observações do pesquisador. No entanto, as atividades e observações podem gerar desconforto nos participantes e estes podem desistir da participação na pesquisa a qualquer momento.

Procedimentos: Para analisar as interações de seu filho, serão realizadas observações no laboratório de informática da escola. Serão cinco momentos de três horas cada um. Os programas desenvolvidos por seu filho e suas anotações serão enviadas ao pesquisador para posterior análise.

Alternativas: A autorização é voluntária e a recusa em autorizar não acarretará qualquer penalidades. Seus dados pessoais não serão identificados em nenhuma publicação que resultar deste estudo. Aefetivação do envolvimento com esta pesquisa somente se dará a partir da assinatura deste termo, com o qual estará consentindo em participar do trabalho, sendo-lhe reservado o direito de recusar-se a participar ou de desistir de sua participação a qualquer momento. Sua desistência ou não participação não irá prejudicá-lo e os dados obtidos a partir das coletas realizadas com você até o momento de sua desistência serão descartados.

Confidencialidade: O pesquisador e sua orientadora certificaram-me que todos os dados desta pesquisa serão confidenciais e as informações publicadas somente em eventos ou periódicos científicos. De acordo com a Resolução 466/12, fica garantido o sigilo absoluto das informações obtidas nessa pesquisa, sendo a guarda dos materiais produzidos neste estudo de inteira responsabilidade do pesquisador. Após o período de cinco anos os programas e anotações serão descartados.

Problemas e perguntas: O pesquisador - Leonardo Poloni - é graduado em Ciência da Computação, é Especialista em Redes de Computadores e é mestrando do programa de Pós-graduação em Educação da Universidade de Caxias do Sul, sob orientação da professora Dr.ª Eliana Maria do Sacramento Soares, e se compromete a esclarecer devida e adequadamente qualquer dúvida ou necessidade de informações que o participante, venha a ter no momento da pesquisa ou posteriormente través do telefone: 54 99937-4323 ou por e-mail lpoloni@ucs.br, e/ou o Comitê de Ética em Pesquisa pelo telefone 54 3218-2829.

Custos: Sua participação no estudo não acarretará custos, assim como não será disponibilizada nenhuma compensação financeira.

Termo de Assentimento

Após ter sido devidamente informado/a de todos os aspectos da pesquisa e ter esclarecido todas as minhas dúvidas, concordo em participar da referida pesquisa e participar das atividades propostas.

Estarei assinando as duas vias, sendo uma retida por mim, e a outra via arquivada com o pesquisador.

Eu, _____
declaro meu consentimento em participar desta Pesquisa.

Assinatura do participante: _____

Atesto que expliquei a natureza e o objetivo de tal estudo, bem como os possíveis riscos e benefícios do mesmo, junto ao participante. Acredito que ele recebeu todas as informações necessárias que foram fornecidas em uma linguagem adequada e compreensível e que o (a) participante compreendeu tal explicação.

Endereços para contato:

E-mail: _____

Fone: _____

Pesquisador responsável:

Nome legível: Leonardo Poloni

Assinatura: _____

Este Projeto está vinculado ao Comitê de Ética em Pesquisa da Universidade de Caxias do Sul, sob o endereço Rua Francisco Getúlio Vargas, 1130, Bloco A, sala 302B, Caxias do Sul, RS, CEP: 95070-560, fone(54) 32182100 ramal, 2829.