Vinícius Fernandes Moretti

O pensamento computacional no ensino básico: potencialidades de desenvolvimento com o uso do *Scratch*

Porto Alegre

2019

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA

DEPARTAMENTO DE MATEMÁTICA PURA E APLICADA

O PENSAMENTO COMPUTACIONAL NO ENSINO BÁSICO: POTENCIALIDADES DE DESENVOLVIMENTO COM O USO DO *SCRATCH*

VINÍCIUS FERNANDES MORETTI

Porto Alegre
2019

Vinícius Fernandes Moretti

O pensamento computacional no ensino básico: potencialidades de desenvolvimento com o uso do *Scratch*

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Matemática Pura e Aplicada do Instituto de Matemática e Estatística da Universidade Federal do Rio Grande do Sul como requisito parcial para a obtenção do grau de Licenciado em Matemática.

Universidade Federal do Rio Grande do Sul

Instituto de Matemática e Estatística

Departamento de Matemática Pura e Aplicada

Orientador: Prof. Dr. Rodrigo Dalla Vecchia

Porto Alegre

2019

CIP - Catalogação na Publicação

Moretti, Vinícius Fernandes
O pensamento computacional no ensino básico: potencialidades de desenvolvimento com o uso do Scratch / Vinícius Fernandes Moretti. -- 2019.
105 f.
Orientador: Rodrigo Dalla Vecchia.

Trabalho de conclusão de curso (Graduação) -- Universidade Federal do Rio Grande do Sul, Instituto de Matemática e Estatística, Licenciatura em Matemática, Porto Alegre, BR-RS, 2019.

1. Pensamento computacional. 2. Scratch. 3. Tecnologias Digitais. 4. Construcionismo. 5. Algoritmos. I. Dalla Vecchia, Rodrigo, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

Vinícius Fernandes Moretti

O pensamento computacional no ensino básico: potencialidades de desenvolvimento com o uso do *Scratch*

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Matemática Pura e Aplicada do Instituto de Matemática e Estatística da Universidade Federal do Rio Grande do Sul como requisito parcial para a obtenção do grau de Licenciado em Matemática.

Banca examinadora. Porto Alegre, 05 de julho de 2019:

Prof. Dr. Rodrigo Dalla Vecchia
Orientador

Profa. Dra. Márcia Rodrigues Notare Meneghetti

Prof. Dr. Rodrigo Sychocki da Silva

Porto Alegre
2019

Ao amor da minha vida, Yasmin.

Agradecimentos

Recentemente, escutei uma entrevista do astrônomo Cliff Stoll, na qual ele relatou ter procurado por sua professora de Matemática do oitavo ano durante décadas. Ele queria encontrá-la para agradecê-la por um momento quando ela lhe mostrou o que seria uma matriz. Ele contou que aquela explicação foi crucial, que ela lhe inspirou a seguir a carreira científica e a buscar coisas novas e interessantes na sua vida. Semelhante ao professor Cliff, eu gostaria de agradecer aqui a todos os professores que me inspiraram durante a minha vida. Infelizmente não vou conseguir agradecer a todos da mesma maneira que ele o fez, alguns eu apenas lembro o primeiro nome, mas as suas marcas são sentidas independentemente disso. Queria deixar registrado toda a minha admiração e profundo agradecimento por terem feito parte da minha trajetória, de uma maneira ou de outra. Muito obrigado aos professores Siloê, Flávio, Dirce, João Maria, Ester, Luis Augusto, Cláudia Padão Rovani, Marisa, Luiz, Inédia, Maria Helena Pires, Marli Novelo, Leoni Cavagnoli, Patrícia, Gilcéu, Altemir, Liana Beatriz Costi Nácul, Marnes Augusto Hoff, Edson Prestes, Luis Fernando Carvalho da Rocha, Carlos Heuser, Raul Weber, Fábio Souto Azevedo, Vilmar Trevisan, Luis Gustavo Mendes, Cydara Ripoll, Carlos Hoppen, Luiz Emilio Allem, Fábio Souto de Azevedo, Luiz Carlos Bombassaro, Elisabete Zardo Búrigo, Débora da Silva Soares, Rodrigo Dalla Vecchia, Lisete Bampi, Paulo Zíngano e Eduardo Britzke.

Agradeço também à minha família e aos colegas de curso, em especial à Victória e à Rosana pelas tantas conversas e cafés.

Agradeço à professora e amiga Naira pela oportunidade de estagiar em suas turmas e realizar as atividades da pesquisa.

Agradeço também aos professores Rodrigo e Márcia por terem aceitado participar da minha banca, por terem lido o trabalho e por suas contribuições valiosas à versão final do texto.

Um muito obrigado especial ao professor Rodrigo pela confiança e paciência.

Agradeço imensamente à Janaína e ao Fábio por sempre estarem presentes e torcerem por mim.

E, sobretudo, agradeço à Yasmin por ser meu porto seguro, minha maior incentivadora e fonte de inspiração, a pessoa sem a qual este trabalho não existiria; obrigado por acreditar sempre em mim, teu apoio incondicional foi o que me manteve e me mantém sempre em frente, aconteça o que acontecer. Muito obrigado por ser minha companheira em todas as horas e por fazer eu querer me tornar uma pessoa melhor todos os dias.

So, carry on
There's a meaning to life
Which someday we may find
Carry on, it's time to forget
The remains from the past
To carry on

Carry On (Angra) – Andre Matos (1971–2019)

Resumo

O presente trabalho tem por finalidade analisar as potencialidades de desenvolvimento das habilidades fundamentais relacionadas ao pensamento computacional — decomposição, generalização, abstração, algoritmos e avaliação — por meio de atividades com a linguagem de programação *Scratch*. As atividades práticas foram realizadas em um laboratório de informática em uma escola estadual da cidade Porto Alegre, Rio Grande do Sul, com alunos do nono ano do Ensino Fundamental. Sob um viés qualitativo, e apoiados no Construcionismo de Seymour Papert (1988) e nas ideias levantadas por Jeannette Wing (2006), que trouxe à tona o movimento atual de difusão do pensamento computacional, analisamos as construções e caminhos dos estudantes, a procura de indícios das dimensões referentes ao pensamento computacional e de possíveis relações de construção de conhecimentos matemáticos em suas falas, discursos e produções de artefatos tecnológicos. Observamos que ao serem confrontados com atividades de caráter aberto em ambientes computacionais, os alunos foram capazes de demonstrar domínio de várias dimensões relativas ao pensamento computacional, o que converge com os ideais de Papert (1988).

Palavras-chave:
Pensamento Computacional. Scratch. Tecnologias Digitais. Construcionismo. Algoritmos.

Abstract

The present work aims to analyze the potentialities of development of fundamental skills related to computational thinking — decomposition, generalization, abstraction, algorithms and evaluation — through activities with the programming language *Scratch*. The practical activities were carried out in a computer lab at a public state school in Porto Alegre, Rio Grande do Sul, with students from the ninth grade of Elementary School level. Under a qualitative perspective, and supported by Seymour Papert's Constructionism (1988) and the ideas raised by Jeannette Wing (2006), who brought to light the current movement of diffusion of the computational thinking, we analyzed the students' constructions and paths, towards to looking for indications of the dimensions related to the computational thinking and possible relations of construction of mathematical knowledge in their speeches, discourses and productions of technological artifacts. We observed that when they were confronted with open-ended activities in computational environments, the students were able to demonstrate domain of various dimensions related to the computational thinking, which converges with Papert's ideals (1988).

Keywords:
Computational Thinking. Scratch. Digital Technologies. Constructionism. Algorithms.

Lista de ilustrações

Sumário

Introdução

Clouds I left below one day
Now are the ones who light the way

No Need To Have An Answer (Virgo) – Andre Matos

Os caminhos trilhados nunca são lineares. Planejamos um roteiro, preparamo-nos para ele, pensamos no futuro e em como atravessaremos as pontes a fim de irmos do lugar em que estamos para o lugar em que almejamos estar. Mas, silenciosamente, e quase sem notarmos, esse futuro chega e nos pega de sobreaviso, desnudos. E logo esse futuro já passou, já há outros planos, outros roteiros, outros anseios, outras memórias, outras vidas. Buscamos constantemente capturar o *kairós* em meio ao *cronos*, aquele tempo, aquele momento, aquele lugar, aquela posição, aquela oportunidade única. Mas a vida é como o *ichi-go ichi-e*[1] japonês: mesmo que haja a possibilidade de repetição de uma situação específica, os acontecimentos, os desdobramentos, os encontros nunca serão os mesmos.

Minha trajetória até aqui foi como todas as trajetórias são, repleta de desvios, descontinuidades (talvez não-enumeráveis), desencontros e reencontros: "[...] considerar o mundo como coisa a ser decifrada é, sem dúvida, um dom. Mas esse dom correria o risco de permanecer oculto em nós mesmos se não tivéssemos os encontros necessários [...]" (DELEUZE, 2003, p. 25). O caminho que me trouxe ao que sou hoje permeia todo esse devir instantâneo que insiste em acontecer, esse instante fugidio, que teima em sempre existir, mas em nunca se mostrar — ele é infinitesimal.

Nunca planejei me tornar e muito menos ser um professor. Apesar de ter crescido em um ambiente cheio de professores (minha mãe, minha tia, minha outra tia e minha outra tia, todas professoras), a observação das dificuldades e frustrações diárias da profissão me impeliram a ter como uma das únicas certezas na vida não seguir essa carreira. Meu primeiro curso universitário foi Ciência da Computação, na UFRGS, no qual ingressei logo após terminar o Ensino Médio. Depois de um tempo (um *cronos* de três anos e meio), tomei a difícil decisão de abandoná-lo e seguir outro rumo. A Matemática sempre esteve presente nos meus anseios e gostos pessoais, e assim acabei por escolher cursar bacharelado, já que a ideia de não ser professor ainda estava arraigada em mim. Pareceu-me uma escolha

[1] *Ichi-go ichi-e* é um lexema japonês (*yojijukugo*) que se refere a um conceito cultural de valorização dos encontros com as pessoas e da natureza não-replicativa de cada momento. Traduções comuns são "somente por esta vez", "nunca mais" ou então "uma chance em toda a vida" (ICHI-GO-ICHI-E, 2019).

natural na época, mesmo não sabendo exatamente o que um bacharel em Matemática fazia; eu não queria dar aulas, não fazia sentido fazer licenciatura.

Mas o caminho planejado nunca é o que trilhamos. Num desses (des)encontros, vi-me com oportunidade e vontade de encarar uma experiência nova: lecionar aulas em um cursinho popular. No ano de 2011, em meio à graduação do bacharelado, assumi duas turmas como professor de Matemática no CEUE Pré-Vestibular, cursinho popular que é oferecido pelo Centro dos Estudantes Universitários de Engenharia da UFRGS. Lembro-me muito bem da primeira vez em que eu entrei em sala de aula: estava muito nervoso e provavelmente passei o conteúdo de um mês inteiro em 30 minutos. Minhas mãos suavam, eu falei e escrevi rápido demais, não consegui olhar direito para o rosto dos alunos; entretanto, saí dali com a certeza de que era aquilo que eu queria fazer na minha vida.

Formei-me bacharel em Matemática, após isso obtive o grau de mestre em Matemática Aplicada, lecionei em inúmeros cursinhos e tive a oportunidade de estar professor substituto por duas ocasiões na UFRGS. A pesquisa ainda continua sendo um dos meus objetivos, mas motivado pelo desejo de me tornar docente, acabei ingressando no curso de Licenciatura em Matemática.

Na licenciatura, algumas disciplinas me deram uma visão muito mais ampla sobre o que é ser professor — que eu encarava anteriormente como um mero instrumento de transmissão de conhecimento —, e vi minha docência ser influenciada sobremaneira por leituras diversas realizadas durantes algumas disciplinas e também por conversas, conselhos e práticas de alguns professores, que com certeza carregarei comigo nas minhas novas trajetórias. Durante o curso, entrei em contato com tendências variadas em Educação Matemática, tais como a Modelagem Matemática, a Resolução de Problemas e as Tecnologias Digitais. Sendo oriundo da Matemática Aplicada, logo vi nessas concepções um grande potencial para se trabalhar em sala de aula. A Matemática, infelizmente, goza de certa reputação perante a sociedade e muito provavelmente o seu ensino nas escolas é um dos grandes responsáveis por isso. Vejo a pesquisa e o estudo dessas tendências como uma tentativa de romper com o estigma transmitido através de gerações e de dotar a Matemática de um caráter mais amistoso.

Segundo estatísticas divulgadas pela organização *Burning Glass*[2], metade dos empregos com os melhores salários nos Estados Unidos necessitam de conhecimentos de programação. Em 2015, outra organização americana, a *Change the Equation*[3], apurou que o número de americanos que se utilizam de computação em seus cotidianos de trabalho era mais do que o dobro do que o número de trabalhadores no setor divulgados pelo *BLS* (*U. S. Bureau of Labor and Statistics*, departamento de estatísticas dos Estados Unidos). Isso significa que as habilidades referentes à computação e a capacidade de

[2] Disponível em <https://www.burning-glass.com/research-project/coding-skills/>.
[3] Disponível em <http://ecs.force.com/studies/rstempg?id=a0r0g000009TLfB>.

usá-las não são necessárias somente àqueles que possuem carreiras ligadas à tecnologia, elas são valorizadas e requeridas em uma gama muito maior de carreiras. A importância do pensamento computacional é gradativamente mais evidente, ao passo que as tecnologias digitais são praticamente onipresentes em todos os âmbitos da nossa vida contemporânea.

Presenciamos uma mudança que ocorre de forma cada vez mais rápida. Quem imaginaria alguns anos atrás escolher qualquer filme ou série para assistir na televisão sem sair de casa? Essa mesma televisão estar conectada ao celular e ao computador? E solicitar comida, compras do supermercado ou transporte direto desse aparelho? E esse mesmo aparelho sendo cada vez menos um telefone e cada vez mais quase tudo? O papel que a tecnologia digital ocupa em nossas vidas só cresce, e estamos nos tornando progressivamente mais dependentes dela. Essa nova realidade vai nos exigir poder de adaptação e aprendizado contínuos.

Apesar dessa inegável presença da informática no mundo atual, movimentos na direção da inserção da tecnologia no âmbito educacional se mostram isolados. Há de se destacar que as pesquisas na área estão cada vez mais numerosas, no entanto o impacto dentro da sala de aula ainda é tímido, o que justifica a busca por meios e ambientes favoráveis à sua utilização no horizonte escolar. Diversos resultados, como os encontrados em Araújo e Soares (2002), Jenkins *et al.* (2009), Carneiro e Passos (2014), Fraiha-Martins e Gonçalves (2012), Fioreze *et al.* (2013), Jacinto e Carreira (2017), entre outros, dão conta de seus benefícios para a aprendizagem, em especial, a aprendizagem de Matemática.

Hoje podemos pensar e construir coisas distintas das quais a simples utilização de lápis e papel tornava possível, ou seja, os computadores influenciam e sensibilizam as pessoas e as suas formas de pensar (PEA, 1987). Mais do que isso: as tecnologias informáticas se apresentam como meios de transformação, como maneiras de transformar e de ser transformado. A fusão entre seres humanos e tecnologias são apreendidas pelos conceitos de humano-mídia (BORBA, 2002), seres-humanos-com-mídias (BORBA; VILLARREAL, 2005) e ferramentaparapensamentos (*toolforthougths*) (SHAFFER; CLINTON, 2006). Ademais, considera-se o pensamento computacional como uma capacidade fundamental e imprescindível a todos, o qual deveria ser trabalhado em todos os níveis educacionais (WING, 2006; RIESCO ALBIZU *et al.*, 2014).

Com essas ideias em mente nos propomos o presente trabalho, que visa a investigar as capacidades inerentes aos conceitos de pensamento computacional passíveis de serem manifestadas e/ou desenvolvidas na esfera da educação básica. Pretendemos obter possíveis respostas à pergunta ***"Quais as potencialidades do uso do Scratch para o desenvolvimento das habilidades relacionadas ao pensamento computacional?"***, seguindo um viés construcionista, baseado na teoria de Seymour Papert (1988), e nos valendo das ideias defendidas por Jeannette Wing (2006), que trouxe à tona o movimento atual de difusão do pensamento computacional. Para tal, foi realizada uma

prática em laboratório de informática, que ocorreu no período de estágio obrigatório do curso de Licenciatura em Matemática, no primeiro semestre do ano de 2019, junto a uma turma de nono ano da Escola Estadual de Ensino Fundamental Olegário Mariano, localizada na cidade Porto Alegre. Tivemos a intenção de investigar os processos de solução e de construção de soluções de problemas e como elas podem contribuir para o desenvolvimento do pensamento computacional, bem como observar indícios de construção de conhecimentos matemáticos surgidos.

A estruturação do presente trabalho se faz em quatro partes, divididas em tópicos que julgamos pertinentes à nossa pretensão. Na primeira delas, o Capítulo 1, fazemos um apanhado extensivo do que diz respeito aos múltiplos conceitos e definições do pensamento computacional, passando por diferentes autores e princípios, e possíveis conexões com a educação e Tecnologias Digitais. Apresentamos também uma revisão do que a nova BNCC sugere em seu texto no que tange às habilidades tomadas como fundamentais à caracterização do pensamento computacional: decomposição, abstração, generalização, algoritmos e avaliação.

Dando prosseguimento, no Capítulo 2, expomos a metodologia de pesquisa utilizada, destacando pontos relevantes às nossas ações durantes as práticas, que são também descritas neste capítulo. No Capítulo 3, apresentamos os dados produzidos, organizados na forma de cinco episódios. As suas descrições são feitas por meio de imagens, relatos e recortes de relatórios escritos e de falas transcritas dos sujeitos de pesquisa.

No Capítulo 4, temos como objetivo uma análise reflexiva frente aos dados obtidos e aos referenciais teóricos apresentados nos Capítulo 1 e Capítulo 2. Nele, procuramos encontrar indicações que evidenciam as habilidades referentes ao pensamento computacional permeadas nos discursos e ações dos alunos durante as práticas, procurando uma intersecção entre os dados produzidos e a teoria.

Por fim, tomamos espaço para tecermos possíveis caminhos de convergência para as ideias apresentadas ao longo de trabalho. Com o intuito de externarmos nossas perspectivas adquiridas sobre o explanado a respeito do pensamento computacional e de alguns de seus desdobramentos referentes à Matemática, colocamo-nos em posição de ação e de espera para novos desafios.

Além disso, apresentamos quatro apêndices: o primeiro deles (Apêndice A) abarca uma apresentação de um breve histórico das ideias e acontecimentos que culminaram na criação dos primeiros computadores, bem como algumas noções preliminares relativas à Computação em geral; o Apêndice B contém uma coleção de endereços eletrônicos de organizações e iniciativas em prol da disseminação das ideias do pensamento computacional; o Apêndice C comporta os planos de aula utilizados nas práticas em laboratório de informática durante o período de estágio; e o Apêndice D inclui os termos de consentimento esclarecido e de consentimento informado, que foram assinados pelos alunos e por seus

responsáveis, a fim de conceder permissão de uso de suas produções na presente pesquisa.

1 O pensamento computacional

And now it's clear
One day leads on to another
I dry my tears
There's so much else to discover
Somewhere

Innocence (Shaman) – Andre Matos

Neste capítulo são abordados aspectos referentes às definições do que seria o pensamento computacional (PC) — o qual não possui consenso na comunidade acadêmica desde as suas origens de uso na literatura — e ideias afins às Tecnologias Digitais (TD). São exploradas com mais detalhes as conceituações modernas do pensamento computacional, explicitando as principais habilidades relacionadas e apontadas como básicas à sua significação, de acordo com o indicado em BBC Learning (2015) e Csizmadia (2015), bem como possíveis conexões com a Matemática e a Educação Básica. Também apresentamos um panorama geral do que há no Brasil em resposta aos movimentos internacionais de inclusão do ensino de Computação (em especial, do pensamento computacional) em nível básico, por meio de recortes das declarações contidas na BNCC.

1.1 "Pensar como um computador": por quê?

A noção de que o pensamento estruturado é crucial para o desenvolvimento e para o bom relacionamento do ser humano com o mundo não é nova. Os primeiros hominídeos se valeram de técnicas e instrumentos criados para conquistar e dominar o espaço a sua volta e evoluir tanto como espécie quanto como sociedade: existem registros do uso de ferramentas e de comportamentos complexos datando de mais de 50.000 anos, provavelmente coincidindo com o surgimento da linguagem (WADE, 2003). Durante toda a história, o homem precisou construir artefatos que o auxiliasse nas mais diversas tarefas, desde o ato de lascar uma pedra, arar um terreno, localizar-se durante uma viagem, até colocar um satélite em órbita. Precisou também aprender a se relacionar com tais artefatos, de maneira que estes entes externos ao seu corpo fizessem exatamente o que lhes era proposto e previsto, obedecendo às intenções e finalidades da sua construção.

Quando pensamos na palavra tecnologia, comumente a associamos com equipa-

mentos eletrônicos e dispositivos modernos, esquecendo-nos que na verdade a sua definição abarca muito mais do que isso. Segundo o Dicionário Eletrônico Houaiss, tecnologia é a "teoria geral e/ou estudo sistemático sobre técnicas, processos, métodos, meios e instrumentos de um ou mais ofícios ou domínios da atividade humana" (TECNOLOGIA, 2009). Portanto, ela é muito mais do que meramente uma coleção de aparelhos digitais contemporâneos, diz respeito a um conjunto muito maior de objetos (físicos ou não) os quais são utilizados para conseguirmos nos relacionar com o mundo a nossa volta. Borba (2002) reflete sobre isso dizendo que o conhecimento não é só influenciado pelas mídias (como sinônimos de meios e tecnologias), mas também moldado por elas. Tal vínculo é inerente ao processo de transmissão de conhecimento ao longo da história da humanidade e não é exclusividade da nossa época informatizada atual. Ele propõe, então, uma unidade de conhecimento, batizada de humano-mídia, na qual o conhecimento passa a ser visto como um produto tanto do ser humano como ser coletivo quanto das mídias e tecnologias do seu tempo. Isso tem o poder de fomentar debates, em que alterações nos currículos se fazem urgentes. Transformações nos saberes produzidos vão exigir mudanças pedagógicas, dado o anacronismo do que é visto em sala de aula e as novas perspectivas de cidadania e possibilidades advindas desses novos conhecimentos dos humanos-mídia.

Na medida que as tecnologias interferem e se entrelaçam junto aos seres humanos na produção de conhecimento, elas não só estabelecem novas maneiras de se fazer coisas que já eram feitas antigamente, mas também

> [...] introduzem novas situações de resolução de problemas nas quais a Matemática é útil; elas introduzem novas normas e procedimentos para construção, argumentação e justificação; e expandem radicalmente os tipos de compreensões e habilidades matemáticas que contribuem para o sucesso nessas situações. (LESH, 2000, p. 178, tradução nossa)

Assim, o domínio do uso das tecnologias é condição essencial para que o ser humano exerça seu lugar no mundo, tanto como agente produtor de conhecimento quanto como paciente ou mero usuário dessa informação. No entanto, esse domínio não deve se restringir ao simples fato de se saber utilizar a tecnologia de maneira prática e procedural, ele diz respeito mormente à capacidade de se manipular e interpretar as mídias digitais, de modo a desenvolver aptidões e habilidades que serão úteis aos futuros cidadãos, em especial aos estudantes em sala de aula. Assume, portanto, dimensões maiores e designa virtudes de áreas diversas junto com as tecnológicas.

Tendo em vista a crescente demanda do uso de tecnologias digitais em diversas áreas, não só nas Ciências Exatas, o pensamento estruturado passou a ser uma habilidade desejável no âmbito da formação de praticamente todos os profissionais. Wing (2006) difundiu o termo pensamento computacional (PC), afirmando que "Ele representa uma atitude universalmente aplicável e um conjunto de habilidades que todos, não somente

cientistas da computação, deveriam almejar por aprender e usar." (WING, 2006, p. 33, tradução nossa). Ela advoga que o pensamento computacional será uma aptidão fundamental para todos até a metade do século XXI, tão necessária quanto ler, escrever e calcular, devendo ser adicionada a essas quando se pensa em educação. Diz ainda, em forma de previsão, que a Computação (e os computadores) auxiliarão na disseminação do pensamento computacional.

Entretanto, não foi a primeira vez que expressão foi empregada: deve-se a Seymour Papert o seu primeiro uso na literatura, no livro "*Mindstorms: children, computers, and powerful ideas*", publicado originalmente em 1980 e traduzido para o português com o título "*Logo: computadores e educação*". Na obra, ele cita e comenta o que chamou de ambientes computacionais: lugares ou espaços onde há uma cultura computacional com natureza matética (a que auxilia a aprender sobre a aprendizagem e não unicamente a aprender). Fala de tentativas de construção de tais ambientes e cultura — os quais contribuiriam para um aprendizado um tanto mais humanizado, por meio de relações essencialmente mais pessoais e menos distantes com o objeto de conhecimento — antevendo o seu surgimento inevitável.

> Em muitos casos, embora os experimentos [de construir os ambientes computacionais] tenham sido interessantes e excitantes, não vingaram porque eram muito primitivos. Seus computadores simplesmente não tinham a potência necessária para os tipos de atividades mais envolventes e compartilháveis. Sua visão de como integrar o **pensamento computacional** na vida diária estava insuficientemente desenvolvida. Mas haverá mais tentativas, e mais e mais. E, eventualmente, em algum lugar, todas as peças se juntarão e ele "pegará". Pode-se ter confiança nisso porque tais tentativas não serão experimentos isolados conduzidos por pesquisadores que podem perder os financiamentos ou simplesmente se tornarem desiludidos e desistir. Haverá manifestações de um movimento social de pessoas interessadas em computação pessoal, em seus próprios filhos e em educação.[1] (PAPERT, 1988, p. 217, complementos e grifos nossos)

Papert descreve um exemplo de "objeto-de-pensar-com" computacional, a Tartaruga, símbolo do seu ambiente (linguagem) LOGO. Ela tem a função de desempenhar papel de modelo para outros objetos(-de-pensar-com) a serem inventados, "[...] objetos em que há uma interseção de presença cultural, conhecimento implícito, e a possibilidade de

[1] "*In most cases, although the experiments have been interesting and exciting, they have failed to make it because they were too primitive. Their computers simply did not have the power needed for the most engaging and shareable kinds of activities. Their visions of how to integrate* ***computational thinking*** *into everyday life was insufficiently developed. But there will be more tries, and more and more. And eventually, somewhere, all the pieces will come together and it will 'catch'. One can be confident of this because such attempts will not be isolated experiments operated by researchers who may run out of funds or simply become disillusioned and quit. They will be manifestations of a social movement of people interested in personal computation, interested in their own children, and interested in education.*" (PAPERT, 1980, p. 182, grifo nosso)

identificação pessoal. [...] Essa Tartaruga serve ao único propósito de ser fácil de programar e boa para se pensar." (PAPERT, 1988, p. 26).

Aos cientistas como um todo, acostumados à análise de dados e à Matemática numérica em geral, o pensamento computacional nunca foi algo estranho. O termo foi cunhado por Papert em 1980, mas a ideia de pensamento computacional é mais antiga, aparecendo inclusive anteriormente na sua obra. Ele e outros pares descrevem o pensamento procedural como um recurso mental destacado desde pelo menos uma década antes (FEURZEIG *et al.*, 1970; PAPERT; SOLOMON, 1971). No início da década de 1960, houve uma iniciativa do cientista da computação Alan Perlis de disseminar entre diversos âmbitos a noção de que a programação constitui um valioso mecanismo para a resolução de problemas (FORSYTHE, 1959; PERLIS, 1962). Perlis defende que a ação de análise computacional de como as coisas são feitas — a qual batizou de "algoritmização" (*algorithmizing*) — deveria ser o foco, ao mesmo tempo que diz que esta ação está tão arraigada nos afazeres humanos que fatalmente todos deveriam aprendê-la de alguma maneira ou outra; a programação seria uma exploração de um processo, algo que mudaria a perspectiva das coisas, um passo em direção a um entendimento mais profundo do que seria a própria Computação, o que levaria os estudantes a reformularem suas visões de tópicos diversos em termos da Computação (GUZDIAL, 2008; PERLIS, 1962).

Nesta época, acontecia um movimento em prol da legitimação da Ciência da Computação como campo próprio, tentando desvinculá-la da Matemática, em especial da Análise Numérica. Acreditava-se que a ideia de algoritmo seria o pilar central da Ciência da Computação e que os atos de projetá-los e programá-los deveriam ser suas práticas fundamentais (TEDRE; DENNING, 2016). Dijkstra (1974) definiu o que seria para ele o pensamento algorítmico, dando-o características de transformação de linguagens (da materna para o formalismo computacional), invenção de soluções próprias para os problemas e agilidade no trato de diferentes níveis semânticos. Knuth (1985) se preocupou em definir tipos de pensamento algorítmico mediante a análise de alguns problemas matemáticos selecionados. Ele concluiu que as atitudes de representação da realidade, redução para problemas mais simples, raciocínio abstrato, estruturas de informação e atenção aos algoritmos (categorias destacadas na sua análise) apareciam fortemente no pensamento algorítmico, em contraste ao fazer matemático, onde eram menos visíveis. Divergindo desses pontos de vista, havia também movimentos de profissionais da área da Computação que advogavam no sentido de que ela deveria ter um caráter mais prático e experimental, o que ocasionou na época uma "fuga de cérebros" das universidades para as indústrias (TEDRE; DENNING, 2016).

A expressão "ferramentas de propósito geral" aparece inúmeras vezes na literatura quando há a intenção de se referir aos conceitos de pensamento computacional durante as décadas finais do século XX. Percebeu-se a importância que o aprendizado de programação

teria nas mais diversas áreas, e que o fato de se precisar ensinar algo a alguém — no caso, um computador — é o que realmente demonstra que se aprendeu esse algo de fato (KNUTH, 1974; MENDELSOHN; GREEN; BRNA, 1990). Na necessidade de ser preciso e de formular algoritmos, de modo que não haja espaço para ambiguidades, reside o cerne do aprender: o pensamento necessário a este tipo de entendimento é muito mais profundo do que qualquer outro tipo de pensamento (KNUTH, 1974).

Surgiu também na década de 1980 o termo máquina nocional, que seria basicamente um modelo mental do que um computador poderia fazer e de como controlá-lo (DU BOULAY; O'SHEA; MONK, 1981). Para se utilizar um computador de maneira a extrair o máximo de suas funcionalidades, há de se saber o que de fato é possível ser feito com ele, quais suas potencialidades e quais suas limitações. Este é o tema central da Teoria da Computação e é precisamente o que o conceito de máquina nocional captura. Tal aparato teórico serve de aporte para o desenvolvimento do aprendizado em Computação, que se baseia na construção de algoritmos, e a maneira como as pessoas pensam e aprendem sobre Computação é crucial na pesquisa em educação informática.

> Aprender uma máquina nocional sem qualquer tipo de notação é como ensinar poesia apenas por meio de palavras memorizadas e faladas. Pode ser possível, pelo menos para pequenos trechos de poesia. No entanto, é muito mais fácil e você pode explorar ideias mais poderosas se puder lê-las. (GUZDIAL, 2015, p. 3, tradução nossa)

O movimento atual de discussão acerca do pensamento computacional, iniciado por (WING, 2006), reacendeu o debate em torno de uma definição precisa para o termo e de suas contribuições para um conjunto muito maior de profissionais além daqueles ligados diretamente com a informática. Ainda hoje não há consenso e muitos debates e pesquisas existem a fim de definir o que seria e quais os propósitos básicos do pensamento computacional. A própria Wing dá definições diferentes para o termo ao longo de seus trabalhos.

> O pensamento computacional envolve a solução de problemas, o projeto de sistemas e a compreensão do comportamento humano, com base nos conceitos fundamentais da Ciência da Computação. O pensamento computacional inclui uma gama de ferramentas mentais que refletem a amplitude do campo da Ciência da Computação. (WING, 2006, p. 33, tradução nossa)

Ao longo do texto, ela distingue algumas características que seriam intrínsecas ao pensamento computacional: (1) conceituação, não somente programação; (2) habilidade fundamental, não rotineira; (3) a maneira como os humanos, não os computadores, pensam; (4) complementa e combina ideias da Matemática e da Engenharia; (5) baseado em ideias, não em artefatos tecnológicos; e (6) para todos, em todos os lugares (WING, 2006). Posteriormente, ela apresenta as seguintes definições:

> **Pensamento computacional** é o processo de pensamento envolvido na formulação de problemas e de suas soluções para que as soluções sejam representadas de uma forma que possam ser efetivamente realizadas por um agente de processamento de informações. (WING, 2010, p. 1, grifo do autor, tradução nossa)

> Pensamento computacional é o processo de pensamento envolvido na formulação de um problema e na expressão de sua(s) solução(ões) de tal forma que um computador — humano ou máquina — possa efetivamente executá-la(s). (WING, 2014, tradução nossa)

Aho (2011, p. 1, tradução nossa) define o termo de maneira semelhante: "[...] processo de pensamento envolvido na formulação de problemas, para que suas soluções possam ser representadas como etapas computacionais e algoritmos." Similarmente, encontramos uma definição dada por Liukas (2015, p. 110, tradução nossa):

> Pensar nos problemas de uma forma que permita aos computadores resolvê-los. O pensamento computacional é algo que as pessoas fazem, não os computadores. Inclui o pensamento lógico e a capacidade de reconhecer padrões, pensar com algoritmos, decompor um problema e abstrair um problema.

Podemos destacar também a definição em documento publicado pela *The Royal Society*, de Londres, que aponta o papel crescente que a Computação e a tecnologia desempenham no mundo atual nas mais diversas áreas, defendendo a inclusão de disciplinas afins à informática nos mais variados âmbitos educacionais:

> Pensamento computacional é o processo de reconhecer aspectos da computação no mundo que nos rodeia, e aplicar ferramentas e técnicas da Ciência da Computação para entender e raciocinar sobre sistemas e processos tanto naturais quanto artificiais. (FURBER, 2012, p. 29, tradução nossa)

Após sistemática pesquisa e levando em consideração correntes distintas de perspectivas epistemológicas acerca de uma elucidação clara do tema, Brackmann (2017, p. 29) propõe a seguinte definição, esquematizada na Figura 1.

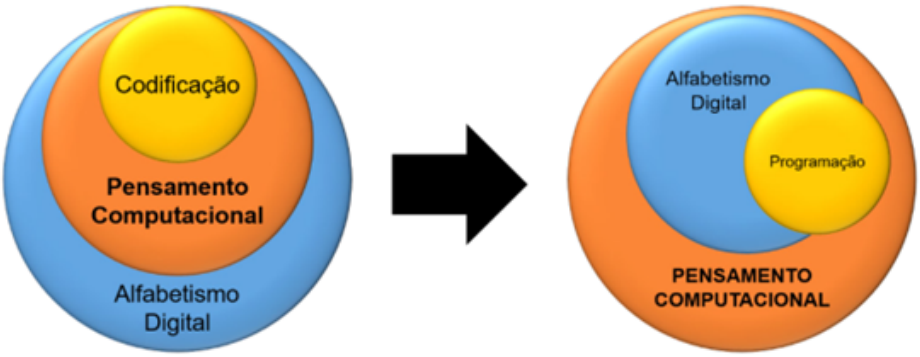
> O Pensamento Computacional é uma distinta capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da Computação, nas mais diversas áreas do conhecimento, com a finalidade de identificar e resolver problemas, de maneira individual ou colaborativa, através de passos claros, de tal forma que uma pessoa ou uma máquina possam executá-los eficazmente.

Figura 1 – Esquematização do PC e outras áreas.

Fonte: retirada de Brackmann (2017, p. 30).

Levando em consideração outros aspectos de formação do pensamento computacional e baseando-se em Johnson (2014) o mesmo autor propôs também uma segunda esquematização para as competências inerentes ao que tange as relações entre áreas distintas movimentadas no processo, apresentada na figura Figura 2. Segundo o autor, essa construção foi motivada pela compreensão de que a alfabetização digital não é condição necessária para a compreensão e o domínio do pensamento computacional: há inúmeras tarefas comuns, tais como utilizar um celular, regular a temperatura de um forno, selecionar um programa de televisão para ser gravado, entre outras, que demonstram que habilidades do pensamento computacional não estão diretamente ligadas ao ato de ser digitalmente alfabetizado (BRACKMANN, 2017).

Figura 2 – Ressignificação de competências ligadas ao PC.



Fonte: retirada de Brackmann (2017, p. 31).

Seguindo abundantes discussões entre inúmeros profissionais da Ciência da Computação, entre eles professores, pesquisadores e outros especialistas, publicou-se, em 2011, uma série de documentos nos quais foi divulgada o que seria a definição operacional de pensamento computacional. Lideraram esta empreitada a Sociedade Internacional para Tecnologia em Educação (*International Society for Technology in Education* — ISTE) e a Associação de Professores de Ciência da Computação (*Computer Science Teachers Association* — CSTA). Os documentos resumem e dão suporte para ações que visam à difusão dos conceitos, e elencam práticas e estratégias em direção à integração e ao desenvolvimento do pensamento computacional desde as idades iniciais nos currículos escolares (BARR; HARRISON; CONERY, 2011). Tal definição é a seguinte (CSTA; ISTE, 2011, p. 1, tradução nossa):

> O pensamento computacional (PC) é um processo de solução de problemas que inclui (mas não está limitado a) as seguintes características:
>
> - Formulação de problemas de uma forma que nos permita usar um computador e outras ferramentas para ajudar a resolvê-los
> - Organização e análise logica dos dados
> - Representação de dados através de abstrações como modelos e simulações
> - Automatização de soluções por meio de pensamento algorítmico (uma série de etapas ordenadas)
> - Identificação, análise e implementação de possíveis soluções com o objetivo de alcançar a combinação mais eficiente e eficaz de etapas e recursos
> - Generalização e transferência deste processo de resolução de problemas para uma ampla variedade de problemas
>
> Essas habilidades são apoiadas e realçadas por uma série de qualidades ou atitudes que são dimensões essenciais do PC. Essas qualidades ou atitudes incluem:
>
> - Confiança em lidar com a complexidade
> - Persistência no trabalho com problemas difíceis
> - Tolerância com ambiguidades
> - A capacidade de lidar com problemas em aberto
> - A capacidade de se comunicar e trabalhar com outras pessoas para alcançar um objetivo ou solução comum

Bers *et al.* (2014) adicionam dimensões de abstração, generalização e, principalmente, tentativa e erro às definições de pensamento computacional, dando atenção especial ao ato de depuração (*debugging*) das soluções: procur o porquê delas, nem sempre, funcionarem como o esperado. Grover e Pea (GROVER; PEA, 2013) chamam atenção para o papel que a abstração toma nas diversas definições de pensamento computacional encontradas na literatura, e concordam com (WING, 2010) quando dizem que ela é peça chave para o trato com a complexidade dos problemas. Particularizam uma série de

elementos que devem fazer parte das bases curriculares a fim de apoiar a aprendizagem e avaliar o desenvolvimento do PC (GROVER; PEA, 2013, pp. 39–40, tradução nossa):

- Abstrações e generalização de padrões (incluindo modelos e simulações)
- Processamento sistemático de informações
- Sistemas de símbolos e representações
- Noções algorítmicas de fluxo de controle
- Decomposição estruturada de problemas (modularização)
- Pensamento iterativo, recursivo e paralelo
- Lógica condicional
- Restrições de eficiência e performance
- Depuração e detecção sistemática de erros

Também há uma série de trabalhos que focam na diferenciação entre o pensamento computacional e outros tipos de pensamento (BARR; HARRISON; CONERY, 2011; DENNING, 2009; GROVER; PEA, 2013; EASTERBROOK, 2014; JONES, 2016). Em especial, podemos frisar algumas diferenças apontadas por Wing (2008) e Shute, Sun e Asbell-Clarke (2017) em relação ao pensamento matemático. Esse seria algo que envolve a habilidade de resolver problemas matemáticos em geral, como equações, e englobaria ao menos três aspectos principais: crenças e visões sobre a Matemática, processos de resolução de problemas e esquemas de provas (justificativas para soluções de problemas) (HAREL; SOWDER, 2005; SNEIDER *et al.*, 2014).

Fundindo as numerosas concepções e ideias do movimento contemporâneo em prol da popularização e inclusão do pensamento computacional nos currículos escolares, BBC Learning (2015) reuniu o que chamou de quatro pilares (técnicas chaves) do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmos. Cada um deles é independente do outro, todos tendo papel importante durante o processo. Csizmadia *et al.* (2015) acrescentam uma quinta dimensão à lista: a avaliação da solução. Nessas cinco habilidades foram baseadas nossas análises acerca das práticas realizadas e dos dados produzidos. A seguir, descrevemos brevemente o que seriam essas técnicas fundamentais.

A técnica de decomposição diz respeito ao pensar nos objetos em termos de suas partes, dividindo o todo em problemas menores, de forma a torná-las mais manejáveis (dividir para conquistar) (LIUKAS, 2015). "As partes podem ser entendidas, solucionadas, desenvolvidas e avaliadas separadamente." (CSIZMADIA *et al.*, 2015, p. 8, tradução nossa). Quebrar as tarefas em fragmentos menores torna o processo de solução mais claro e erros podem ser identificados com maior facilidade (processo de depuração). Programadores profissionais pensam em termos de decomposição o tempo todo, já que utilizam módulos para escreverem seus códigos (funções, procedimentos, objetos, classes, bibliotecas, etc.),

tornando possível a colaboração de outras pessoas no seu trabalho e descomplicando a manutenção.

O reconhecimento de padrões refere-se ao poder de generalização, de encontrar características, propriedades, qualidades ou similaridades que são comuns a diferentes instâncias de um problema. Explorar a natureza replicativa de certas construções pode fazer com que a solução de um problema se torne mais robusta, eficaz e elegante. Nesse movimento de se questionar se a estratégia considerada é válida para outros problemas, ou ainda, se soluções para problemas anteriores já resolvidos podem ser utilizadas em uma nova situação, fica implícito um certo poder de adaptação e apropriação (JENKINS *et al.*, 2009; SÁPIRAS, 2017), ou seja, aprender com algo já criado. Deste modo, algoritmos pensados e desenvolvidos anteriormente podem ser reutilizados ou ajustados para a uma nova classe de cenários: no momento que se identifica que um dado problema pertence a essa classe pode-se aplicar a solução genérica.

A habilidade identificada com a abstração tem relação com identificar variáveis ou elementos que não são relevantes a uma situação, reduzindo o número de detalhes desnecessários. Assim, é possível se concentrar no que de fato é substancial para se atingir o objetivo de resolver um problema. Wing (2014) defende que a abstração é o processo chave no pensamento computacional: ela é utilizada no momento de generalização, quando se detecta propriedades comuns a um conjunto de objetos e se descarta as distinções não importantes entre eles. Reconhecer tipos de estruturas e não deixar nenhum detalhe fundamental de fora faz com que os problemas se tornem mais simples, dando aparatos para se lidar com a complexidade.

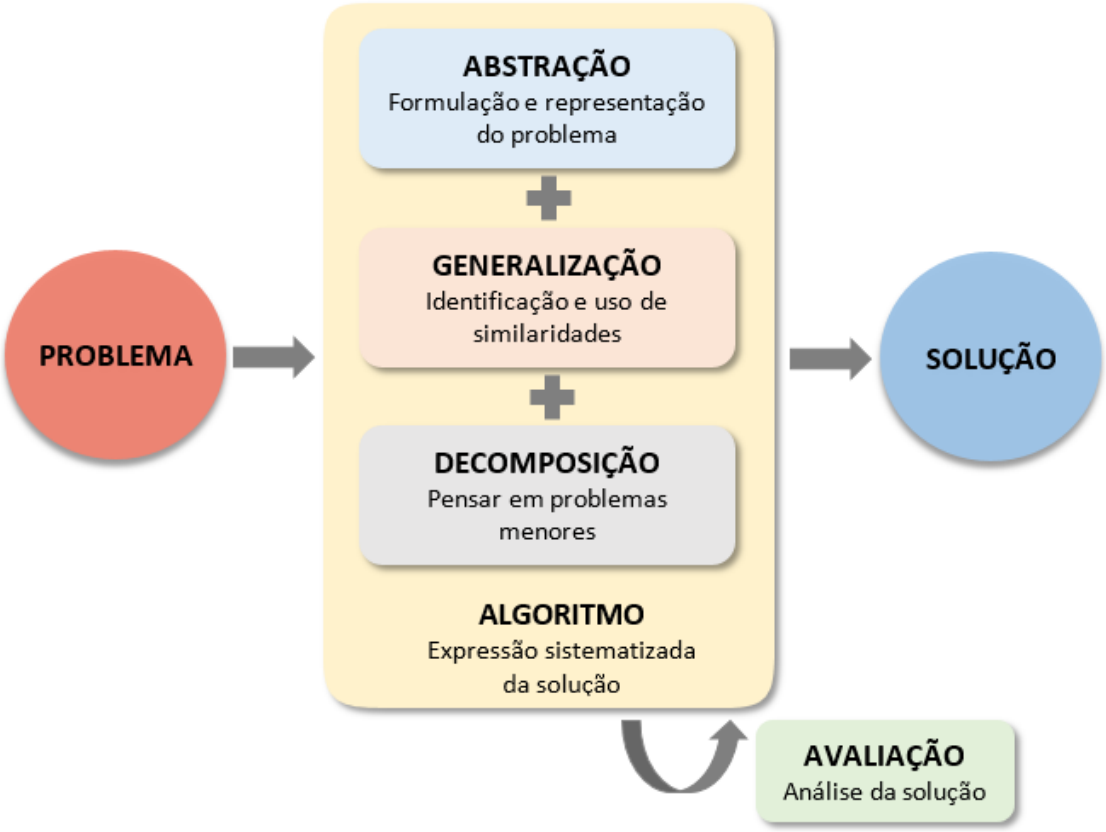
O algoritmo é a representação *per se* de uma solução para um problema. Pensar em termos de passos claramente especificados, de maneira estruturada e sem ambigüidades, define a essência do que é resolver um problema. A considerar aí estão os outros elementos descritos anteriormente, já que se procura um método que se encaixe em situações similares (generalização e abstração), sem que haja necessidade de se pensar novamente nas mesmas problemáticas toda vez que elas aparecem. A decomposição também cumpre papel importante no ato de se pensar em algoritmos, que podem ser arquitetados para problemas pequenos e depois compostos em soluções mais completas, construindo-se, assim, algoritmos maiores. Não à toa, Wing (2014) defende que o algoritmo é o componente de integralização no pensamento computacional.

Concomitante a essas quatro habilidades, deve ocorrer o processo de avaliação. A todo momento — desde quando se pensa em um problema, reconhece-se as suas características e elementos relevantes, divide-se o contexto em problemas menores, procura-se soluções ou situações semelhantes, e constitui-se um conjunto sequencial de regras e passos lógicos para a resolução — é necessário considerar se características como eficácia, consumo de recursos, clareza, rapidez, facilidade e outras métricas que a solução determinada possa

apresentar são as desejáveis. Considerar se as necessidades e propósitos das estratégias adotadas estão sendo atendidas é uma característica importante para que erros pequenos não se tornem complicações maiores e mais difíceis de serem tratadas posteriormente. Portanto, a avaliação deve ser um processo contínuo durante a solução de um problema, e por isso deve ser tratada como elemento fundamental ao se considerar o processo de pensamento computacional.

Esquematizamos as cinco habilidades consideradas como fundamentais ao pensamento computacional na Figura 3.

Figura 3 – Esquematização das habilidades fundamentais do PC.



Fonte: própria.

Esse esquema proposto compreende todo o movimento do pensamento computacional, desde o momento que se indentifica e se pensa no problema a ser resolvido, arquiteta-se as estratégias de resolução e considera-se a solução construída e testada. Nesse decurso, identificamos discretizadas as habilidades fundamentais de abstração, generalização, decomposição, algoritmos e avaliação, no entanto elas não são independentes e nem se mostram presentes em todas as possíveis soluções construídas, como poderemos evidenciar nas análises presentes no Capítulo 4.

1.2 O pensamento computacional e as Tecnologias Digitais na educação básica

Como vemos, as capacidades essenciais mobilizadas ao se considerar os conceitos de pensamento computacional estão relacionadas com diversas outras habilidades de pensamento e de resolução de problemas que podem permear inúmeras áreas. Pensar computacionalmente nos permite acessar uma parte do fazer computacional, sendo possível transpô-lo para uma variedade de sistemas, objetos, conjunto de dados e processos. Ele fornece uma série de estruturas mentais ligadas à Computação, mas que extrapolam esse campo. No processo de reconhecer aspectos computacionais no mundo que nos cerca e de utilizar técnicas e recursos originalmente pertencentes a um domínio restrito de conhecimento, e de raciocinar e pensar de acordo com essas técnicas, podemos expandir a nossa compreensão sobre diversos processos naturais, artificiais e sociais. Ao basearmos as ações no processo de criação e não no produto final, ideia defendida por Papert (1988, 1994), podemos criar outras bases para o entendimento e aprendizado sobre os problemas.

Segundo Borba (2002), houve duas correntes distintas no Brasil quanto à utilização de tecnologias, em especial o computador, na sala de aula. Uma delas afirmava que o uso dessas tecnologias faria com que os estudantes deixassem de adquirir ou perdessem certas competências anteriores à chegada das mídias, corrente essa que perdeu força ao passo que o computador e a tecnologia em geral são onipresentes no mundo atual. De outro lado, existiram aqueles que se tornaram deslumbrados pela inovação e acreditaram que o computador podia resolver (quase) tudo.

Inicialmente, tivemos a utilização do computador como um obstáculo a ser contornado. Não se sabia como usar as máquinas e não se sabia como ensinar a mesma coisa que se ensinava anteriormente se utilizando deles. Assim, antes de ser o pote de ouro no fim do arco-íris da educação, o computador também foi um inconveniente. Cessado o momento do estranhamento, passou-se a enxergar o computador como solução, e o primeiro ambiente em que isso se deu foi no âmbito da educação de gestão privada. Criaram-se slogans e propagandas, onde a utilização de tecnologia na sala de aula era mostrada como diferencial, algo que deveria ser exaltado como moderno e avançado. A medida que esse marketing se tornou lugar-comum na sociedade, aconteceram iniciativas do poder estatal para a inserção de computadores e de laboratórios de informática das escolas públicas, já que a informatização da sociedade parecia um caminho sem volta.

Introduzir as ideias de pensamento computacional em sala de aula foi uma busca constante e se mostrou um desafio desde que os primeiros conceitos acerca do tema começaram a aparecer. Pesquisadores em informática na educação baseiam suas concepções e conclusões tanto na Computação quanto na Educação — nenhuma das duas áreas isolada é suficiente.

> Ensinar pensamento computacional para todos requer abordagens distintas daquelas que usamos quando assumimos que nossos estudantes querem se tornar profissionais da Computação. Desenvolver abordagens que funcionarão para todos os estudantes exigirá responder perguntas difíceis, como o que estudantes comuns [aqueles que não seguirão na área da Computação] entendem sobre Computação, o que eles acharão desafiador, que tipos de ferramentas podem fazer o pensamento computacional mais facilmente acessível para eles, e como deveríamos organizar e estruturar nossas aulas de modo a fazer o pensamento computacional acessível para todos os estudantes. (GUZDIAL, 2008, p. 26, tradução e complemento nossos)

Considerou-se por muito tempo que antes de se incluir noções de Computação na educação era necessário que se soubesse utilizar os computadores de maneira adequada. Teve-se, então, como imagem, que isso significava ter literacia digital: saber usar e ter domínio operacional das tecnologias — o computador e seus programas, no caso. No entanto, essa concepção simplista da utilização do computador em sala de aula é insuficiente para que o seu uso seja justificado. Há de se considerar que a presença das tecnologias adiciona dimensões muito maiores e movimenta um conjunto de capacidades que vai muito além das meras habilidades mecânicas.

> Ter literacia digital requer mais do que simplesmente a capacidade de usar um software ou operar um dispositivo digital; ela inclui uma ampla variedade de habilidades complexas, tais como cognitivas, motoras, sociais e emocionais, que os usuários precisam ter de modo a utilizarem os ambientes digitais de maneira eficiente. (ESHET-ALKALI; AMICHAI-HAMBURGER, 2004, p. 421, tradução nossa)

Iniciativas com o intuito de incorporar os conceitos de Computação e desenvolver as capacidades inerentes ao pensamento computacional desde a educação básica têm acontecido em vários lugares distintos do mundo. Reconhece-se, cada vez com mais clareza, que a Computação é uma disciplina de caráter próprio, diferente de somente dominar o uso da máquina. Em seus trabalhos, Wing argumenta que, junto às habilidades de ler, escrever e contar, devemos acrescentar a habilidade básica de pensamento computacional. É comum o jargão "*the three R's*" (***R**eading, (w)**R**iting and (a)**R**ithmetic*), ou "os três erres", em inglês — para se referir à trinca fundamental de habilidades essenciais da educação básica. Há estudiosos que defendem que o pensamento computacional deveria ser incluído nesta lista como o "quarto erre" (*algo**R**ithms*) (WING, 2006; PECKHAM, 2011; STRAUSS, 2012).

No Reino Unido, os trabalhos realizados por Celia Hoyles e Richard Noss encabeçam o projeto chamado de *ScratchMaths*[2] (SM), que tem por objetivo principal entender o proveito que a introdução da programação nas escolas inglesas trouxe para o aprendizado e para o raciocínio matemático (BENTON *et al.*, 2016; BENTON *et al.*, 2017;

[2] Disponível em <https://www.ucl.ac.uk/ioe/research/projects/scratchmaths>.

CLARK-WILSON; HOYLES, 2017). Desde 2014, o ensino de informática é obrigatório em todo o território britânico, tanto no ensino primário quanto no secundário (entre 5 e 16 anos) (INGLATERRA, 2013). Tal iniciativa teve respaldo na constatação de que os estudantes estavam deixando a escola com pouco conhecimento da área de informática e do lado criativo da Computação (FURBER, 2012; BENTON *et al.*, 2016). O ensino de Computação não se foca somente no uso da tecnologia digital, mas também na ação de construção dos próprios programas pelos alunos, de maneira que o protagonismo do estudante surja e seja parte constituinte da metodologia de ensino.

Na Estônia, existe um projeto pioneiro de inclusão de conteúdos de Ciência da Computação nos currículos da educação básica. O país é um dos mais dependentes da internet e das tecnologias digitais do mundo, e faz esforços para informatizar o maior número de áreas do seu território, a ponto de somente três serviços estatais dependerem da presença física de um ser humano: transferência de imóveis, casamento e divórcio (BIGARELLI, 2018). Numa parceria público-privada, foi criada a plataforma digital *ProgeTiiger*[3], que integra recursos e documentos de apoio a professores e pais a fim de incentivar o estudo de programação nas escolas (COUTO; SILVA, 2016).

Nos Estados Unidos, há um movimento bastante forte de inclusão de aulas de programação na educação básica. Organizações como a *Computer Science Teacher Association*[4] (CSTA) e a *Code.org*[5] mantêm discussões e publicações constantes a respeito do tema, ajudando a elaborar currículos que se adaptem às novas exigências legais da educação, que colocam a Ciência da Computação no mesmo patamar que as demais disciplinas tradicionais da escola (BRACKMANN, 2017). No ano de 2013, foi lançada uma campanha nacional chamada *Hour of Code*[6], encabeçada pela organização *Code.org*, que teve, inclusive, como um dos garotos-propagandas o presidente do país à época, Barack Obama. Ele se tornou o primeiro presidente a escrever um programa, desenvolvendo um código que desenhava um quadrado na tela (FINLEY, 2014). Em vídeo para a campanha, Obama falou:

> Não apenas compre um novo videogame, faça um; não apenas faça download do aplicativo mais recente, ajude a projetá-lo; não apenas jogue no seu telefone, programe-o! Ninguém nasceu um cientista da computação, mas com um pouco de trabalho duro, e um pouco de Matemática e Ciência, praticamente qualquer um pode se tornar um. (YOUTUBE, 2013, tradução nossa)

Países como Finlândia, Grécia, Israel, Espanha, França, Austrália e Coréia do Sul, entre outros, também estão engajados em esforços na direção de inserir conteúdos de

[3] Disponível em <http://www.progetiiger.ee>.
[4] Disponível em <https://www.csteachers.org/>.
[5] Disponível em <https://code.org/>.
[6] Disponível em <https://hourofcode.com/>.

programação em seus currículos escolares (FRANÇA; SILVA; AMARAL, 2013; BRACKMANN, 2017). É global o entendimento da importância que as habilidades desenvolvidas ao se considerar os conceitos de pensamento computacional desempenham no âmbito escolar e fora dele. Não há mais como se pensar no mundo atual e suas idiossincrasias sem se considerar que a tecnologia digital é quase que onipresente e que os benefícios conseguidos por meio do desenvolvimento do pensamento computacional tendem a se propagar em diversas áreas do saber e do fazer. Entre os diversos argumentos levantados em defesa do pensamento computacional, podemos destacar (GUZDIAL, 2015; BRACKMANN, 2017):

- **Empregabilidade:** demanda por profissionais qualificados;
- **Aprender sobre o mundo:** a realidade contemporânea é informatizada;
- **Transdisciplinaridade:** aplicar conhecimentos computacionais em outras áreas;
- **Literacia digital:** aprender a usar e se beneficiar das mídias;
- **Produtividade**: pensar e aumentar os seus domínios de tempo, espaço e ação;
- **Ampliação da participação:** oportunidades para mais pessoas;
- **Diminuição de fronteiras:** acessível em diversas partes, a todo tempo;
- **Trabalho em equipe:** possibilidade de construções conjuntas e compartilhamento.

1.3 O pensamento computacional na educação básica brasileira

No Brasil, iniciativas governamentais que tenham paralelo com as de outros países em direção à inclusão de conteúdos afins à Computação na educação básica ainda são incipientes. A SBC (Sociedade Brasileira de Computação) possui a inicitiva Computação na Escola[7], a qual tem por objetivo estimular trabalhos interdisciplinares, sobretudo se utilizando de software livre e hardware aberto, que visam ao ensino de Computação nas escolas. Ela também organiza, junto com a UNICAMP, a OBI (Olimpíada Brasileira de Informática), que no ano de 2019 terá a sua vigésima primeira edição nacional.

Tanto os Parâmetros Curriculares Nacionais (PCN), bem como a novíssima Base Nacional Comum Curricular (BNCC), não trazem nada específico sobre o ensino do pensamento computacional ou Computação em nível básico. Contudo, podemos destacar a escrita do que a BNCC chama de "Competências gerais da Educação Básica", uma lista de dez comportamentos que devem ser desenvolvidos ao longo do ciclo básico na escola. Entre eles, é possível salientar alguns que fazem menção às ideias de pensamento computacional, mesmo que de maneira generalista (BRASIL, 2018, p. 9):

[7] Disponível em: <http://sbc.org.br/2-uncategorised/1925-computacao-na-escola>.

> **1.** Valorizar e utilizar os conhecimentos historicamente construídos sobre o mundo físico, social, cultural e digital para entender e explicar a realidade, continuar aprendendo e colaborar para a construção de uma sociedade justa, democrática e inclusiva.
>
> **2.** Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.
>
> **4.** Utilizar diferentes linguagens — verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital —, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.
>
> **5.** Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

Encontramos explicitado nas diretrizes do documento uma definição de pensamento computacional, entremeada a um seção que fala sobre Tecnologias Digitais e Computação. A BNCC demonstra preocupação com as mudanças que a tecnologia impõe à sociedade e com as transformações que essas mudanças ocasionam no âmbito da educação básica. Ela coloca em destaque três características que devem ser articuladas e postas como objetivo da aprendizagem: o pensamento computacional, o mundo digital e a cultura digital. Ao definir o que seria o pensamento computacional, diz

> **pensamento computacional**: envolve as capacidades de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio do desenvolvimento de **algoritmos**; (BRASIL, 2018, p. 474, grifos nossos)

Há de se chamar a atenção também para o que é dito na parte específica que trata da Matemática do Ensino Fundamental:

> O desenvolvimento dessas habilidades [raciocinar, representar, comunicar e argumentar matematicamente] está intrinsecamente relacionado a algumas formas de organização da aprendizagem matemática, com base na análise de situações da vida cotidiana, de outras áreas do conhecimento e da própria Matemática. Os processos matemáticos de resolução de problemas, de investigação, de desenvolvimento de projetos e da modelagem podem ser citados como formas privilegiadas da atividade matemática, motivo pelo qual são, ao mesmo tempo, objeto e estratégia para a aprendizagem ao longo de todo o Ensino Fundamental. Esses processos de aprendizagem são potencialmente ricos para o desenvolvimento de competências fundamentais para o letramento matemático (raciocínio, representação, comunicação e argumentação) e para o desenvolvimento do **pensamento computacional**. (BRASIL, 2018, p. 266, complemento e grifo nossos)

> No Ensino Fundamental – Anos Iniciais, a expectativa em relação a essa temática é que os alunos resolvam problemas com números naturais e números racionais cuja representação decimal é finita, envolvendo diferentes significados das operações, argumentem e justifiquem os procedimentos utilizados para a resolução e avaliem a plausibilidade dos resultados encontrados. No tocante aos cálculos, espera-se que os alunos desenvolvam diferentes estratégias para a obtenção dos resultados, sobretudo por estimativa e cálculo mental, além de **algoritmos** e uso de calculadoras. (BRASIL, 2018, p. 268, grifo nosso)

> Outro aspecto a ser considerado é que a aprendizagem de Álgebra, como também aquelas relacionadas a Números, Geometria e Probabilidade e Estatística, podem contribuir para o desenvolvimento do **pensamento computacional** dos alunos, tendo em vista que eles precisam ser capazes de traduzir uma situação dada em outras linguagens, como transformar situações-problema, apresentadas em língua materna, em fórmulas, tabelas e gráficos e vice-versa.
>
> Associado ao **pensamento computacional**, cumpre salientar a importância dos **algoritmos** e de seus fluxogramas, que podem ser objetos de estudo nas aulas de Matemática. Um **algoritmo** é uma sequência finita de procedimentos que permite resolver um determinado problema. Assim, o **algoritmo** é a **decomposição** de um procedimento complexo em suas partes mais simples, relacionando-as e ordenando-as, e pode ser representado graficamente por um fluxograma. A **linguagem algorítmica** tem pontos em comum com a linguagem algébrica, sobretudo em relação ao conceito de variável. Outra habilidade relativa à Álgebra que mantém estreita relação com o **pensamento computacional** é a **identificação de padrões** para se estabelecer **generalizações**, propriedades e **algoritmos**. (BRASIL, 2018, p. 271, grifos nossos)

Há uma seção do documento intitulada "A progressão das aprendizagens essenciais do Ensino Fundamental para o Ensino Médio", na qual existe também menção explícita da preocupação com o desenvolvimento do pensamento computacional:

> A área de Matemática, no **Ensino Fundamental**, centra-se na compreensão de conceitos e procedimentos em seus diferentes campos e no desenvolvimento do **pensamento computacional**, visando à resolução e formulação de problemas em contextos diversos. No **Ensino Médio**, na área de **Matemática e suas Tecnologias**, os estudantes devem consolidar os conhecimentos desenvolvidos na etapa anterior e agregar novos, ampliando o leque de recursos para resolver problemas mais complexos, que exijam maior reflexão e abstração. Também devem construir uma visão mais integrada da Matemática, da Matemática com outras áreas do conhecimento e da aplicação da Matemática à realidade. (BRASIL, 2018, p. 471, grifo nosso (sublinhado) e da obra (negritos))

Tendo consciência de que o jovem está envolto em um mundo digitalizado, a BNCC distingue ações as quais as competências inerentes às Tecnologias Digitais teriam potencialidades para serem exploradas. Tais competências permitiriam aos estudantes:

> - buscar dados e informações de forma crítica nas diferentes mídias, inclusive as sociais, analisando as vantagens do uso e da evolução da tecnologia na sociedade atual, como também seus riscos potenciais;

> - apropriar-se das linguagens da cultura digital, dos novos letramentos e dos multiletramentos para explorar e produzir conteúdos em diversas mídias, ampliando as possibilidades de acesso à ciência, à tecnologia, à cultura e ao trabalho;
> - usar diversas ferramentas de software e aplicativos para compreender e produzir conteúdos em diversas mídias, simular fenômenos e processos das diferentes áreas do conhecimento, e elaborar e explorar diversos registros de representação matemática; e
> - utilizar, propor e/ou implementar soluções (processos e produtos) envolvendo diferentes tecnologias, para identificar, analisar, modelar e solucionar problemas complexos em diversas áreas da vida cotidiana, explorando de forma efetiva o raciocínio lógico, o **pensamento computacional**, o espírito de investigação e a criatividade.
>
> (BRASIL, 2018, pp. 474–475, grifo nosso)

Da seção em que a BNCC versa sobre o Ensino Médio, destacamos os seguintes trechos que adentram no tema do pensamento computacional:

> Além disso, a BNCC propõe que os estudantes utilizem tecnologias, como calculadoras e planilhas eletrônicas, desde os anos iniciais do Ensino Fundamental. Tal valorização possibilita que, ao chegarem aos anos finais, eles possam ser estimulados a desenvolver o **pensamento computacional**, por meio da interpretação e da elaboração de **algoritmos**, incluindo aqueles que podem ser representados por fluxogramas.
>
> Em continuidade a essas aprendizagens, no Ensino Médio o foco é a construção de uma visão integrada da Matemática, aplicada à realidade, em diferentes contextos. Consequentemente, quando a realidade é a referência, é preciso levar em conta as vivências cotidianas dos estudantes do Ensino Médio — impactados de diferentes maneiras pelos avanços tecnológicos, pelas exigências do mercado de trabalho, pelos projetos de bem viver dos seus povos, pela potencialidade das mídias sociais, entre outros. Nesse contexto, destaca-se ainda a importância do recurso a tecnologias digitais e aplicativos tanto para a investigação matemática como para dar continuidade ao desenvolvimento do **pensamento computacional**, iniciado na etapa anterior. (BRASIL, 2018, p. 528, grifos nossos)

Em consonância com as competências básicas listadas, deverá haver, segundo o documento, uma articulação de saberes para garantir o desenvolvimento de outras valências durante o ensino, denominadas de "Competências específicas de Matemática e suas Tecnologias para o Ensino Médio". Três delas fazem alusão direta ou indiretamente a habilidades relacionadas ao pensamento computacional (BRASIL, 2018, p. 531):

> **3.** Utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente.
>
> **4.** Compreender e utilizar, com flexibilidade e precisão, diferentes registros de representação matemáticos (algébrico, geométrico, estatístico, computacional etc.), na busca de solução e comunicação de resultados de problemas.

> **5.** Investigar e estabelecer conjecturas a respeito de diferentes conceitos e propriedades matemáticas, empregando estratégias e recursos, como observação de padrões, experimentações e diferentes tecnologias, identificando a necessidade, ou não, de uma demonstração cada vez mais formal na validação das referidas conjecturas.

Os documentos listam uma série de habilidades específicas, identificando-as com um código, em cada uma das categorias pormenorizadas. Podemos identificar algumas delas que têm referência ao pensamento computacional no Quadro 1.

Quadro 1 – Habilidades específicas do EM com alguma relação com o PC.

CÓDIGO	HABILIDADE
EM13MAT315	Investigar e registrar, por meio de um fluxograma, quando possível, um algoritmo que resolve um problema.
EM13MAT405	Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.
EM13MAT501	Investigar relações entre números expressos em tabelas para representá-los no plano cartesiano, identificando padrões e criando conjecturas para generalizar e expressar algebricamente essa generalização, reconhecendo quando essa representação é de função polinomial de 1° grau.
EM13MAT502	Investigar relações entre números expressos em tabelas para representá-los no plano cartesiano, identificando padrões e criando conjecturas para generalizar e expressar algebricamente essa generalização, reconhecendo quando essa representação é de função polinomial de 2° grau do tipo $y = ax^2$.
EM13MAT505	Resolver problemas sobre ladrilhamento do plano, com ou sem apoio de aplicativos de geometria dinâmica, para conjecturar a respeito dos tipos ou composição de polígonos que podem ser utilizados em ladrilhamento, generalizando padrões observados.

Fonte: retirado de Brasil (2018, pp. 537, 539, 541).

Como podemos observar, nos documentos oficiais existe uma tendência à inclusão de habilidades e competências transversais aos conceitos de pensamento computacional no cotidiano escolar brasileiro do ensino básico. As dimensões básicas do PC — decomposição, generalização, abstração, algoritmos e avaliação — aparecem diversas vezes nas descrições das competências ao longo do texto. Isso mostra uma certa preocupação com a transformação do conhecimento e uma atenção aos movimentos que vêm ocorrendo em outros lugares do planeta. Contudo, mesmo havendo diretrizes nacionais que indicam um caminho, as políticas educacionais nacionais ficam restritas basicamente ao letramento e à inclusão digital.

Frisamos também que a pesquisa em torno do tema do uso de Tecnologias Digitais na educação tem sido geradora de discussões crescentes no país (BASSO; NOTARE, 2015). A partir desses estudos é que ocorrem iniciativas em prol da popularização e introdução do pensamento computacional nas escolas de educação básica. Podemos destacar os trabalhos realizados por Andrade *et al.* (2013), Kologeski *et al.* (2016), Koscianski e Glizt (2017),

Brackmann (2017) e Geraldes (2017), todos com projetos voltados à Computação na escola, e em especial, os trabalhos de Barcelos e Silveira (2012), França e Amaral (2013) e Ventorini (2015), que utilizam o software *Scratch* nas suas atividades.

2 Método e Prática

Like a teenager discovery
What's more delightful than this?
Try to remember how good it was
Feeling the life as it is
To believe!

Z.I.T.O. (Angra) – Andre Matos

Este capítulo é destinado à descrição da metodologia de pesquisa utilizada durante o trabalho, que foi a qualitativa, junto com os aportes teóricos adotados para a elaboração dos planos de aula. Descrevemos também aspectos básicos do *Scratch*, que foi a linguagem de programação elencada para ser utilizada durante as práticas em laboratório de informática. Optamos por apresentar o conjunto de ideias construcionistas nesta seção pois, para o escopo da presente pesquisa, elas serviram de apoio e orientação metodológicas. Também descrevemos aqui as atividades realizadas e o contexto do ambiente e dos sujeitos de pesquisa.

2.1 Metodologia de pesquisa

Tendo como guia a pergunta "***Quais as potencialidades do uso do Scratch para o desenvolvimento das habilidades relacionadas ao pensamento computacional?***", caracterizamos a abordagem metodológica da presente pesquisa como sendo a de uma pesquisa qualitativa. Segundo Goldenberg (1997), o viés qualitativo envolve descrições detalhadas de situações as quais objetivam compreender os sujeitos e suas ações. Bogdan e Biklen (1994) especificam características que permeiam uma pesquisa do tipo qualitativa: (1) a fonte primordial dos dados é o próprio ambiente natural; (2) a investigação é descritiva; (3) o interesse principal é no processo e não nos resultados ou produtos; (4) a análise de dados é, em geral, feita de forma indutiva; e (5) o significado das ações e das análises constitui o seu cerne de abordagem.

Segundo Bicudo (2017), o adjetivo qualitativo incorpora conceitos suscetíveis de exporem emoções e opiniões, algo da ordem do subjetivo. Engloba, deste modo, aspectos humanos que se revelam no olhar, na reflexão e na experiência do pesquisador sobre e com os sujeitos/objetos da pesquisa, já que o pesquisador se torna parte da própria investigação

e os resultados ficam indissociáveis do viés adotado na análise dos acontecimentos. Foi com este anseio que os micromundos foram propostos aos alunos durante as práticas, na tentativa de entrelaçar o desenvolvimento e a busca por resultados apoiados nas inquietações iniciais da pesquisa, e norteando a produção dos dados, tendo bases no conceito de ressonância[1] proposto por Lincoln e Guba (1985).

Com relação ao acesso às tecnologias e à sua utilização como metodologia em sala de aula, Borba e Penteado (2015) apontam que este ato pode ter o efeito de tirar o professor da sua zona de conforto, sobretudo ao se considerar os professores em formação. Maltempi (2008) indica que há uma transformação no paradigma da prática docente, no qual a aula sofre uma mudança, passando de uma atividade centrada no professor a uma outra que é centrada no aluno.

> [...] um grande mérito das tecnologias é o de colocar diversos pesquisadores e educadores em um movimento de reflexão sobre a educação frente às modificações pelas quais a sociedade passa em decorrência da crescente inserção das tecnologias no dia-a-dia das pessoas. (MALTEMPI, 2008, p. 60)

Papert (1988, p. 18) argumenta que "aprender a comunicar-se com um computador pode mudar a maneira como outras aprendizagens acontecem". Na mesma linha vai o entendimento de Basso e Notare (2015, p. 3): "Os recursos tecnológicos e a possibilidade de representação e manipulação de objetos matemáticos abrem novas possibilidades para o pensamento matemático.". A programação pode entrar, então, como uma forma de fazer o aluno encarar os conceitos matemáticos sob novas vistas, exigindo deles um esforço adicional na direção da compreensão das ideias e definições (PONTE, 1991). Papert (1994) também menciona que na maioria dos espaços escolares o computador entrou para auxiliar um sistema de ensino já pré-estabelecido, mas que o desejável seria criar um ambiente diferente de aprendizado, e não reforçar um método que lima as habilidades e as expectativas das pessoas. Encorajar os alunos a estabelecerem projetos próprios, principalmente utilizando a interação e a colaboração mútua, seria um passo importante em direção a esta mudança. A tecnologia deve entrar nesta direção, facilitando e ampliando os horizontes da compreensão dos conteúdos escolares, fazendo com que as pessoas consigam alcançar e desenvolver competências talvez nunca antes imaginadas.

O advento do computador nos dá a oportunidade única de realmente revolucionarmos a educação e a forma como aprendemos. E o pensamento computacional se torna

[1] Segundo Lincoln e Guba (1985), um dos critérios que devem direcionar uma pesquisa qualitativa é a ressonância, entendida aqui como uma articulação entre o conhecimento prévio e a metodologia escolhida, ou seja, coerência entre a visão de mundo e os procedimentos adotados para a produção de dados. Sendo assim, é necessário levar em conta a sobreposição entre o sistema de crenças preliminar que orienta o paradigma assumido e os dados produzidos durante o processo de pesquisa. Ademais, o relatório deve conter as múltiplas realidades do tema pesquisado, esquivar-se de expectativas a priori, procurar explicações para as inconclusões que se revelam e ponderar sobre as influências que o pesquisador impinge ao projeto. Araújo e Borba (2017) chamam esse conceito de consonância.

aí algo imprescindível ao lidarmos com os problemas que surgem, dando subsídios para aprendermos estratégias na direção da resolução desses problemas e na comunicação clara das nossas ideias. Conrad Wolfram proclama:

> Abrimos muito mais possibilidades. Você pode resolver muito mais problemas. O que eu realmente penso que ganhamos disso é estudantes adquirindo intuição e experiência em quantidades muito maiores do que eles tiveram antes. Uma experiência em problemas mais difíceis, sendo capazes de brincar com a Matemática, interagir com ela, senti-la. Queremos pessoas que consigam sentir a Matemática instintivamente. É isso que o os computadores nos permitem fazer. (YOUTUBE, 2010, tradução nossa)

Esse é o cerne da teoria educacional desenvolvida por Seymour Papert, chamada de Construcionismo. Inspirado nas concepções de aprendizagem propostas por Jean Piaget, que sustenta a ideia de que o conhecimento é construído pelo indivíduo na medida em que ele age e sofre ações dos objetos (noções, pessoas ou coisas), ele estabeleceu as bases de uma nova metodologia de ensino. Papert (1986) defende que a aprendizagem seria um processo de (re)construção muito mais do que uma simples transmissão de conhecimento, no qual devem acontecer situações que fomentem o engajamento por parte dos aprendizes a fim de alimentar esse processo construtivo. O ato de se engajar pode vir da criação de uma pintura, um poema, uma peça de teatro, um programa, etc., enfim, de um projeto pessoal que possa ser compartilhado com outra(s) pessoa(s). Assim, ao mote "aprender fazendo" o Construcionismo acrescenta que o aprendizado é facilitado (e, em muitas vezes, propiciado) quando há motivação, quando se pensa e conversa sobre o que se faz.

> Dessa forma, ao trabalhar segundo as idéias construcionistas dois tipos de construção ocorrem, e mutuamente se reforçam, pois o aprendiz ao construir um produto no mundo está, simultaneamente, construindo conhecimento em sua cabeça. Este novo conhecimento o possibilita a construir produtos mais sofisticados, que o levam a novos conhecimentos, e assim por diante. (MALTEMPI, 2005, p. 3)

Ao analisarmos as concepções das ideias do Construcionismo, podemos perceber que ela não se baseia essencialmente no uso do computador ou de tecnologias digitais para viabilizar os seus ambientes de aprendizagem. Desde que as atividades realizadas coloquem o aluno como centro do processo de ensino, como agente principal do seu aprendizado, elas serão compatíveis com os propósitos imaginados por Papert. Entretanto, ao considerarmos o uso do computador, é possível termos a visão do grande potencial que esta mídia possui para a implementação dos ambientes de aprendizagem vislumbrados pelo Construcionismo.

E Papert soube explorar essa faceta da tecnologia. Seu trabalho na idealização da sua teoria se deu em cima da concepção do ambiente LOGO, no início da década de 1960, nos laboratórios do MIT. Mais que uma linguagem de programação, o LOGO é

um meio no qual o contato com a Matemática e o aprender em geral é experienciado de maneira singular. Cria-se nele miniculturas cognitivas, ou micromundos: ambientes onde é possível construir e reconstruir elementos do mundo real mediante simulações de operações concretas com o uso de uma linguagem (visual) de programação. Esses micromundos se mostram excelentes espaços de engajamento, troca de ideias, cooperação, enfim, de aprendizagem, e foi neles que se pensou as práticas da presente pesquisa.

> [...] o Construcionismo, visto como um processo de construção, pode se mostrar em harmonia com as ideias de processo educacional quando é considerado como um caminho, no qual a aprendizagem se dá na própria medida dos desdobramentos que as situações sofrem por meio da imersão dada pelos participantes. (DALLA VECCHIA, 2012, p. 68)

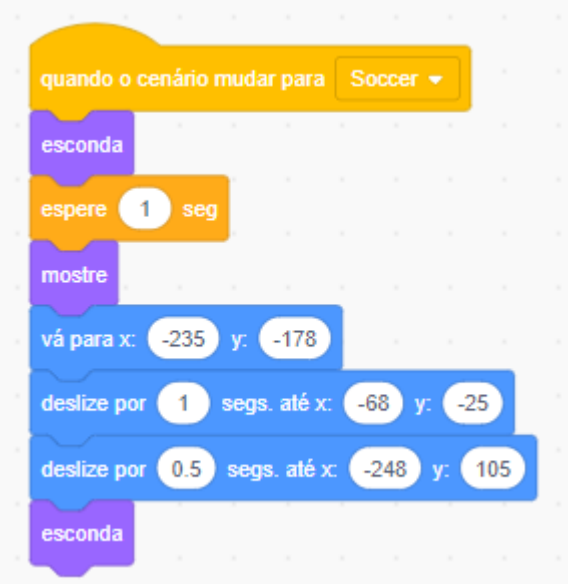
2.2 O *Scratch*

A escolha da linguagem (ou ambiente) de programação para a criação dos micromundos durante a prática foi um momento importante no delineamento da nossa pesquisa. Pretendíamos apresentar nas atividades uma vivência plena do que seria a construção de um programa de computador e a utilização de uma linguagem de programação. No entanto, também não tínhamos a intenção de inserir novos elementos que trouxessem novas dificuldades aos alunos, não havia tempo e nem preocupação de se ensinar a sintaxe própria de uma linguagem de propósitos comerciais. Isto tornaria o trabalho desinteressante, já que muito do foco estaria em aprender a utilizar a linguagem em si e não no processo de se beneficiar das oportunidades que o ato da programação tem a oferecer. Deste modo, uma escolha natural foi o *Scratch*, uma linguagem baseada em eventos, que possui uma interface extremamente intuitiva e visual e uma simplicidade na manipulação das suas estruturas e construção dos comandos.

O *Scratch*[2] é uma linguagem de programação desenvolvida em 2007 pelo MIT, que teve como foco principal inicial o ensino de programação para crianças e jovens (LIFELONG KINDERGARTEN GROUP, 2007). Por apresentar uma interface visual e orientada por meio de blocos pré-estabelecidos, passíveis de serem encaixados uns com os outros de acordo com a lógica desejada (Figura 4), e onde os resultados podem ser imediatamente testados e vistos na tela, ela se constitui numa maneira de se desenvolver as habilidades características do pensamento computacional e à computação em geral. O ambiente permite a criação de programas, jogos, vídeos, animações e histórias de maneira interativa, explicitando comandos para diferentes objetos em um cenário. Há a possibilidade de tradução das instruções para o português e tudo pode ser feito de maneira online.

[2] Disponível em <https://scratch.mit.edu/>.

Figura 4 – Exemplo de programação realizada no *Scratch*.



Fonte: própria.

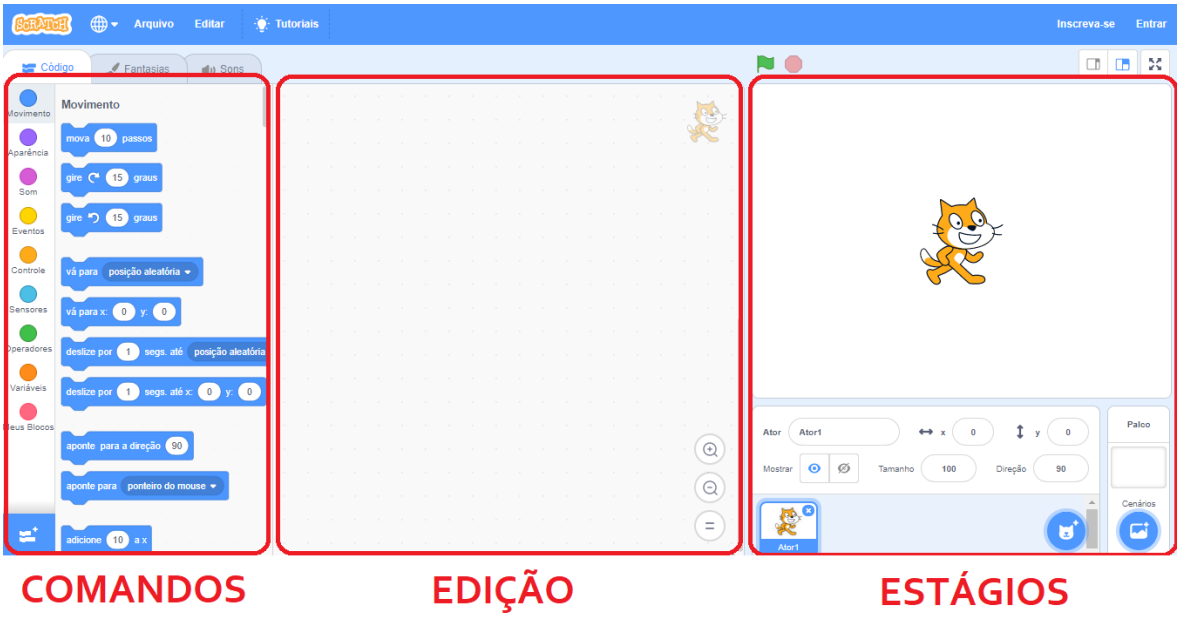
A interface do *Scratch* possui três áreas principais: armazém de comandos, área de edição de recursos e área de estágios (Figura 5). No armazém de comandos estão as estruturas pré-programadas, em forma de blocos, passíveis de serem utilizadas como *scripts* nas ações dos programas. Para isso, basta clicar em um deles com o mouse e arrastá-lo para a área de edição de recursos. Esses comandos estão separados de acordo com a sua natureza, como instruções de movimento, aparência, som, controle, entre outros. Cada tipo é identificado com uma cor e formato, sendo um indicativo de como eles podem ser combinados para comporem a programação. Esse é um dos aspectos pelo qual o *Scratch* se constitui numa ótima maneira de se iniciar no mundo da programação, já que esse modo de organizar as instruções facilita a visualização e o entendimento da lógica por trás do que acontece.

Na área de edição de recursos há espaço para que os blocos seja arrastados e combinados, de modo a formarem blocos maiores, que são as linhas de comando do programa. O funcionamento do *Scratch* se baseia em atribuir instruções a cada objeto utilizado na construção. Esses objetos podem ser os cenários — panos de fundo para as histórias — ou então os atores — personagens que são mostrados na tela. Ou seja, cada um desses elementos tem uma espécie de "vida própria" dentro de um programa, sendo possível designar ações diferentes para cada um deles. Então, ao selecionar uma dessas peças, é mostrada a programação atribuída a ele nessa área de edição. Convém destacar que este modelo de funcionamento incorpora a noção de paralelismo em programação, já que as ações de cada ator são executadas simultaneamente.

A parte de estágios é dividida em duas outras, uma que é o cenário de simulação

e outra na qual são mostrados os atores e o palco (cenário atual). Nesse espaço é onde o que foi programado na área de recursos acontece: as instruções dadas aos objetos dentro do programa são executadas no cenário e então podemos apreciar os resultados obtidos do que foi planejado, visualizando o que ocorre com os objetos. Há a opção de aumentar o espaço ocupado por essa tela, clicando-se nas setas do canto direito superior, o que faz com que o programa seja mostrado em tela cheia, escondendo o armazém de comandos e a área de edição. O palco e a área onde são mostrados os atores fazem alusão a um espetáculo, no qual os artistas esperam até chegar a sua vez de se mostrarem em cena ao público.

Figura 5 – Interface do *Scratch*.



Fonte: própria.

Toda a interação com o *Scratch* se dá de forma online, mas há também a opção de se fazer download de um arquivo executável do software para trabalhos offline. No site do ambiente é possível encontrar uma variedade enorme de exemplos, construídos por usuários do mundo todo. Qualquer um pode enviar suas criações para o site e fazer parte desta comunidade, bastando realizar cadastro na plataforma, o que é gratuito. Os arquivos disponibilizados ficam à disposição de quem quiser utilizá-los, editá-los ou simplesmente visualizá-los, podendo ser feito download deles para o computador pessoal.

Ainda que tenha sido inicialmente pensado para o ensino de programação a crianças e jovens, hoje em dia o *Scratch* possui muitas funcionalidade que extrapolam esse intuito inicial. É possível trabalhar nele com inúmeras extensões, como detecção de vídeo, recursos de detecção de fala e tradução, utilização de instrumentos musicais, conexão com placas Arduino e Raspberry Pi e projetos com LEGO.

2.3 O contexto, as circunstâncias e o que deles surgiu

A fim de investigar as habilidades consideradas fundamentais ao pensamento computacional passíveis de serem manifestadas por meio da aprendizagem e da prática de atividades de programação com o *Scratch*, e obter possíveis respostas à pergunta diretriz da pesquisa, pensou-se na realização de aulas em laboratório de informática com os sujeitos participantes. Durante a disciplina de estágio obrigatório do curso de Licenciatura em Matemática, da UFRGS (Universidade Federal do Rio Grande do Sul), houve a oportunidade de atuar como docente junto a uma turma do nono ano do Ensino Fundamental, na Escola Estadual de Ensino Fundamental Olegário Mariano, localizada na cidade de Porto Alegre, no estado do Rio Grande do Sul. As atividades na escola que serviram de apoio à presente pesquisa ocorreram no primeiro semestre do ano de 2019, entre os dias 26 de abril e 20 de maio. Foram organizados um total de 10 encontros — que possuíam duração variada devido ao número de períodos de 50 minutos destinados à disciplina de Matemática no dia em questão —, divididos em três etapas. As aulas aconteceram no turno da manhã no laboratório de informática da escola e contaram com o auxílio da professora regente da disciplina. Ao todo, havia 25 estudantes na turma participante das aulas. Os planos de aula são postos no Apêndice C a fim de registro e consulta.

A primeira das etapas, realizada nos dias 26 e 30 de abril e 03 de maio, foi destinada à apresentação do ambiente *Scratch* aos alunos. Houve uma breve conversa sobre aspectos rudimentares de programação em geral, algoritmos e suas possíveis relações com a Matemática. Nestes encontros, que totalizaram quatro períodos de 50 minutos cada, o objetivo era a introdução de alguns conceitos referentes ao funcionamento do ambiente e da linguagem do *Scratch*, em forma de estudo dirigido, seguindo como guia o tutorial[3] desenvolvido por Martinelli *et al.* (2019).

Na segunda parte das atividades, ocorrida nos dias 06, 07 e 10 de maio, houve espaço para os estudantes pensarem e realizarem seus projetos próprios. A proposta teve um caráter mais aberto: com a utilização do *Scratch*, os alunos deveriam construir um jogo, vídeo, animação ou apresentação. A natureza, a complexidade e a finalidade dos projetos ficou a cargo dos estudantes, com liberdade para pesquisarem na internet, em manuais, em tutoriais, ou consultarem os professores ou os colegas para coletarem ideias e soluções para o que desejavam construir. O trabalho poderia ser feito de forma individual ou em duplas e a sua realização fazia parte da avaliação do trimestre, sendo uns dos critérios de composição de pontos para a nota da disciplina de Matemática. Ao total, cinco períodos foram utilizados para esta fase das atividades.

A terceira e última das etapas consistiu na socialização dos projetos construídos

[3] Disponível em <http://scratchbrasil.net.br/index.php/materiais/tutoriais.html>.

na etapa anterior, e aconteceu nos dias 13, 14, 17 e 20 de maio, tomando um total de sete períodos de aula. Durante os três primeiros dias de apresentação, os estudantes expuseram suas construções ao grande grupo com o auxílio de um projetor e foram indagados pelos professores e pelos colegas sobre aspectos relevantes dos seus trabalhos: elaboração, estratégias utilizadas, dificuldades encontradas no processo e possíveis relações das atividades com conteúdos tradicionais de Matemática. No último dos encontros foi realizada uma mostra para um público externo ao da turma. Este momento não estava planejado no início, a ideia surgiu no decorrer das aulas ao se notar o bom rendimento e o engajamento geral dos estudantes nas atividades. No momento das apresentações aos colegas, a maioria dos estudantes mostrou alguma satisfação com os seus projetos, sentimento que foi reforçado pelas avaliações dos seus pares ao prestigiarem seus trabalhos. Deste modo, o último dos encontros, que teve duração de dois períodos, foi destinado para que aqueles que se sentissem à vontade mostrassem seus trabalhos a duas turmas selecionadas de sexto ano da escola. Este momento foi importante para o fechamento do ciclo dos trabalhos, em especial para mim, que estava encerrando as práticas do estágio. Foi uma oportunidade de mostrar para outros jovens e professores algumas potencialidades de uso do *Scratch* como instrumento e recurso educacional, focando o desenvolvimento do pensamento computacional como uma metodologia válida e poderosa tanto de construção de conhecimento quanto de imersão por parte dos alunos, não somente em aulas de Matemática, mas também em outros campos do conhecimento.

O Quadro 2 traz a esquematização das três etapas da pesquisa realizada na escola, nas quais os dados foram produzidos por meio de captura de vídeo da tela de alguns alunos, gravação de áudio das apresentações dos trabalhos, relatórios escritos e entregues pelos estudanes e anotações realizadas em diário de campo.

Quadro 2 – Organização das aulas práticas com o *Scratch*.

DATA	**PERÍODOS (50 minutos)**	**ATIVIDADE REALIZADA**
26/04	1	Ambientação e introdução do *Scratch*
30/04	2	
03/05	1	
06/05	2	Construção dos projetos
07/05	2	
10/05	1	
13/05	2	Socialização dos projetos com a turma
14/05	2	
17/05	1	
20/05	2	Mostra dos projetos para turmas do sexto ano

Fonte: própria.

Durante as etapas de ambientação e construção dos projetos, alguns participantes

tiveram as telas dos computadores gravadas por meio de um programa de captura de vídeo. Neste momento não houve gravação de áudio, somente as telas que mostravam suas construções foram salvas. As suas discussões foram acompanhadas e registradas por meio de observações em diário de campo. Ao longo dos momentos de apresentação dos projetos para a turma, foi registrado o áudio e partes destes registros foram selecionadas e transcri-tas, a fim de perceber nos seus discursos as estratégias executadas e tentar identificar os pontos relevantes acerca das dimensões do pensamento computacional demonstradas no decorrer do percurso.

Junto a esses registros, também foi solicitado que os alunos elaborassem um pequeno relatório das atividades, documentando de forma escrita as partes do trabalho desenvolvido. Alguns tópicos foram indicados como critérios bases para a composição desse relatório: (1) deveria conter o título do projeto; (2) a descrição do funcionamento do programa, pormenorizando a sua construção; (3) indicações da utilização de Matemática durante o processo; e (4) informações sobre possíveis dúvidas, dificuldades e pesquisas realizadas. Não foi nossa intenção que os alunos encarassem esses tópicos como perguntas a serem respondidas, mas sim que os utilizassem como guias para escreverem os seus pareceres. No entanto, grande parte deles esquematizou os textos de acordo com esses itens e escreveu o seu relatório na forma de questionário, o que pode ser notado em alguns trechos mostrados no próximo capítulo.

Apesar da construção de um produto final ter sido a intenção da prática proposta, não foi este o principal ponto considerado para análise da pesquisa. Seguindo os princípios do Construcionismo, a investigação e o entendimento do processo de construção, do andar da caminhada, junto com seus sucessos e insucessos, suas facilidades e dificuldades, foram os principais guias para apresentarmos nossas conclusões. Estas nuances foram as procuradas durante a realização da pesquisa, de modo a identificar possíveis pontos de convergência/divergência com a teoria estudada.

3 Dados produzidos: os episódios

Unshod feet traces on fresh sand
A map unfold
Spreading out knowledge
Magic and love
And then
Oh, and then

Holy Land (Angra) – Andre Matos

A apresentação dos dados produzidos será organizada na forma de episódios, cada um deles consistindo na trajetória trilhada pelos estudantes durante a prática. Episódios são histórias que se relacionam com os eventos sucedidos, tendo como característica o entrecruzamento de falas, relatos e análise que dizem respeito às condutas e circunstâncias dos envolvidos no processo, observadas com o intuito de obter possíveis esclarecimentos às dúvidas intrínsecas à pesquisa (DALLA VECCHIA, 2012). A partir de vídeos capturados das telas de alguns alunos, das suas produções parciais e finais, das suas falas durante as apresentações, do relatório escrito e das anotações e observações realizadas durante as atividades, foram selecionados cinco episódios. Frisamos que também fazem parte da descrição, e sobretudo da análise, os progressos parciais dos alunos no decorrer das tarefas. Ao final de cada dia era solicitado que fossem salvos os trabalhos em pastas destinadas no sistema para aquele encontro. Infelizmente nem todos cumpriam as atividades ou então as salvavam em arquivos como o requerido, mas na medida do possível tentamos apresentar as descrições e as análises tendo por base as evoluções dos projetos, não somente nos focando no resultado final.

Mesmo havendo a possibilidade da realização dos trabalhos em dupla, a maioria optou por fazê-lo de forma individual, de modo que ao total foram apresentados 16 trabalhos. No entanto, não foi imposta condição de não haver comunicações entre os pares: os estudantes poderiam trocar informações com seus colegas, com os professores e realizar pesquisas na internet para sanarem possíveis dúvidas. Desse conjunto de trabalhos, foram selecionados um total de cinco para comporem os episódios que serão descritos a seguir.

Os projetos escolhidos foram divididos em três grupos distintos de episódios: um que reúne construções interativas (jogos), outro que reúne construções do tipo animação (sem uma história elaborada) e outro que também agrupa animações, mas essas contendo

uma história ou desfecho por trás da construção. A escolha se deu pelo fato de que muitos dos estudantes realizaram construções parecidas e alguns deles somente modificaram algo do que foi apresentado no estudo dirigido, sem muito acréscimo ou desenvoltura perante o que foi proposto. Deste modo, o conjunto elegido para compor o corpo de dados e ser descrito aqui foi entendido ser o mais rico para as análises a serem explicitadas no próximo capítulo. Os nomes dos estudantes utilizados durante este texto são fictícios a fim de preservar suas identidades. Os episódios foram batizados pelos próprios alunos, que deram nome aos seus trabalhos no título dos seus relatórios. Os textos das imagens dos relatórios serão transcitos para a melhor compreensão e adaptados na medida que for necessário, corrigindo alguns erros de escrita e deixando de lado rasuras ou detalhes que julgamos não serem totalmente relevantes. Ao apresentarmos os excertos das falas dos estudantes durante as suas apresentações, mostramos juntos o tempo exato em que o diálogo é travado nos arquivos dos áudios e o nome do interlocutor.

3.1 Grupo 1: Animações

Neste grupo foram reunidos os episódios classificados como animações. No nosso entendimento, esses projetos se encaixam nesta divisão por serem constituídos de movimentos e interações entre os atores no *Scratch*, mas sem haver uma história ou um pano de fundo por trás das construções. Também, em alguns deles não há um encerramento propriamente dito, ou por estarem em estágio incompleto de elaboração ou por opção de quem o esquematizou, acontecendo às vezes de haver uma repetição da animação ou um encerramento abrupto da mesma. Foram elencados para esta categoria um total de três episódios.

3.1.1 Episódio I. "Vem pro Fut."

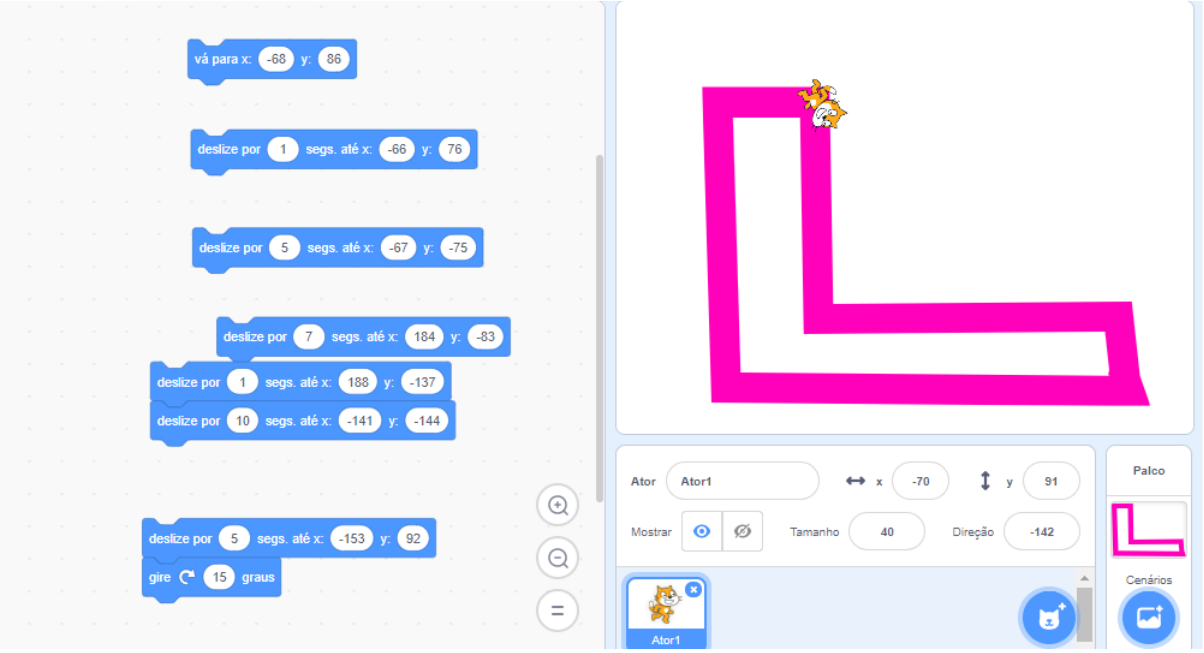
Este episódio foi construído pelos alunos Bruce e Steve e se constitui em uma animação que mostra a imagem de uma goleira de futebol, com alguns jogadores, a bola e um juiz (Figura 6), no que seria a esquematização de uma jogada que resulta em gol.

Ambos os alunos tiveram uma certa dificuldade no início do projeto. Nas aulas de ambientação, Steve não conseguia entender de maneira clara os movimentos dos personagens e as atividades do estudo dirigido não foram totalmente completadas pela dupla. O fato de ser possível (e imprescindível!) encaixar os comandos de forma a dar estruturação às instruções não foi compreendido de antemão pelo aluno, como pode ser evidenciado no primeiro arquivo entregue pelo estudante (que estava sozinho antes de formar a dupla, já que Bruce não se fazia presente nesse dia) logo após a última aula de apresentação do *Scratch* (Figura 7).

Figura 6 – Tela final do Episódio I.

Fonte: própria.

Figura 7 – Primeiro arquivo da dupla Bruce e Steve.



Fonte: própria.

Os alunos programaram cada ator com comandos próprios: dois jogadores de linha, um goleiro, o juiz e a bola. Entretanto, não se utilizaram de instruções muito diferentes daquelas que foram apresentadas durante a etapa de ambientação, que foram basicamente instruções de movimento ("*mover*", "*ir para*", "*deslizar*", "*apontar para direção*" e "*girar*"), trocas de fantasia, cenário, comandos de som e de eventos disparadores de ações. Mas houve progresso nas suas construções, ao menos no que diz respeito ao número de movimentos

que os atores realizaram na versão final, que foi aumentado conforme se analisa os arquivos preliminares disponibilizados pela dupla.

Durante a apresentação do projeto aos colegas, o aluno Steve não estava presente e somente Bruce falou sobre o trabalho. A seguir estão alguns excertos da sua fala.

Excerto 1: Tentativas

[17:29] **Bruce:** Só não consegui arrumar o goleiro ali que ficou parado deitado. [**risos**]

[17:39] **Bruce:** Mas consegui fazer a bola quicar também.

[17:41] **Vinícius:** Como é que tu conseguiu? Como é que tu fez a bola quicar?

[17:52] **Bruce:** Eu só fui acrescentando... eu só tive que acrescentar essas duas direções que a bola tinha que tomar, [por] que depois que ela entrava aqui ela só tinha que descer e fazer assim. Aí eu só acrescentei essas duas direções que é só pra subir e descer.

[18:11] **Vinícius:** E como é que tu descobriu as posições ali, quando tinha que ser na trave, como é que tu fez para descobrir a posição da trave?

[18:18] **Bruce:** Ah, eu arrastei a bola, aí apareceu a direção aqui assim. Mas na verdade quando eu arrastava a bola e parava ela aqui já mudava a direção ali, né?

Percebemos nas falas do aluno as suas estratégias para conseguir posicionar os personagens da sua animação na tela. Ele tenta, por meio de experimentações e obervações, descobrir as coordenadas a serem inseridas nas indicações de movimento dos atores no cenário.

Excerto 2: Tentativas novamente

[18:33] **Vinícius:** O que tu usou de Matemática aí?

[18:35] **Bruce:** De Matemática? Acho que bastante o tempo da bola. E dessa jogadora aqui que ela não fica parada, ela vinha até aqui para chutar a bola... ahn... O tempo do goleiro, porque se perceber aqui ele segue a bola, conforme a bola tá aqui ele segue a bola.

[18:57] **Vinícius:** Ok, ok.

[18:58] [**coloca mais uma vez a animação**]

[19:03] **Vinícius:** Como é que vocês conseguiram acertar esse tempo do goleiro?

[19:06] **Bruce:** Ah, a gente ficou fazendo uns testes, daí a gente acrescentava... No caso, se aumentasse o número, tipo, dos segundos ali pra 2 ele ia demorar mais pra chegar, se dimi[nuir]... A gente tentou botar menos 1, mas aí não deu muito certo...

[19:22] **Vinícius:** Não deu muito certo? O que aconteceu?

[19:25] **Bruce:** Ele ia muito rápido. No caso, ao invés de ele se afastar ele se teletransportava...

[19:32] **Vinícius:** Vocês colocaram tempo menos 1?

[19:33] **Bruce:** A gente tentou fazer, só que não deu certo. [**risos**]

[19:34] **Vinícius:** Mas faz sentido um tempo negativo?

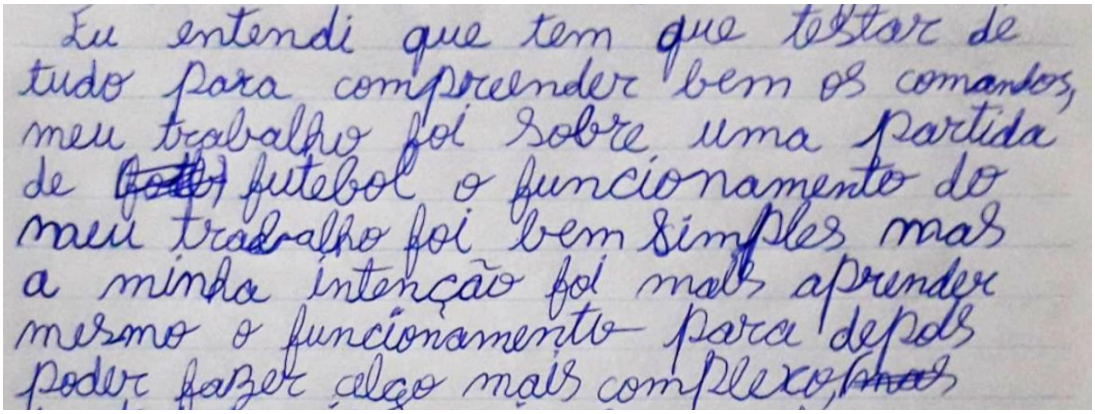
[19:36] **Bruce:** Não faz. [**risos**]

Vemos aí mais tentativas e experimentações. Essas experimentações assumem dimensões um tanto radicais ou apelativas, ao passo que o aluno comentou que a dupla

tentou atribuir um valor negativo a uma instância de tempo. Ele não explicou claramente quais as suas intenções ao fazerem isso, mas entendemos que o mundo virtual é um espaço para se tentar extrapolar conceitos, muitos deles impossíveis de serem testados de fato na realidade vivida, mostrando-se assim como uma oportunidade de realizar simulações.

A seguir, na Figura 8, está um trecho do relatório entregue pela dupla e logo abaixo a sua transcrição, com algumas modificações na estrutura do texto para melhor compreensão.

Figura 8 – Trecho do relatório para o Episódio I.



Fonte: própria.

Transcrição do trecho da Figura 8: "*Eu entendi que tem que testar de tudo para compreender bem os comandos. Meu trabalho foi sobre uma partida de futebol. O funcionamento do meu trabalho foi bem simples, mas a minha intenção foi mais aprender mesmo o funcionamento para depois poder fazer algo mais complexo*".

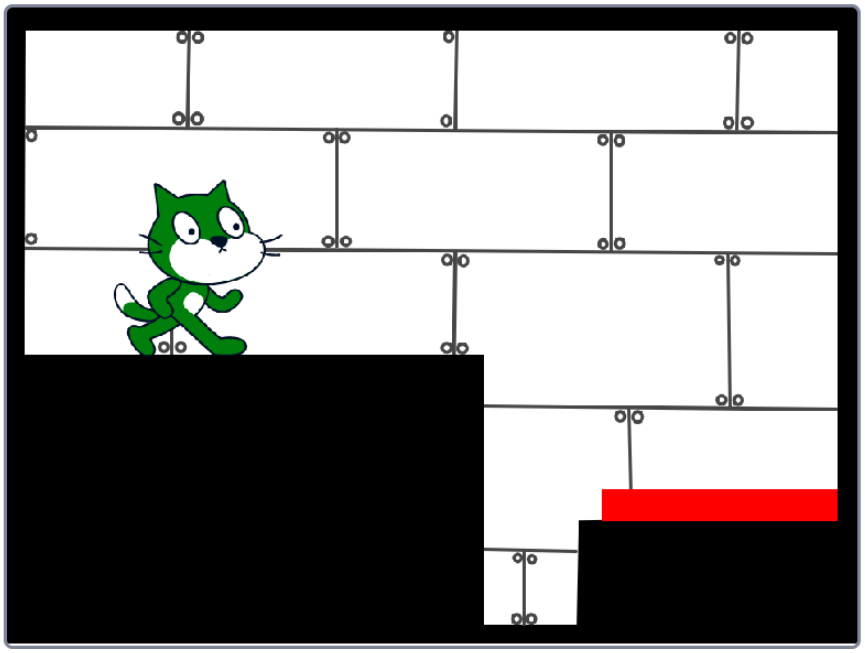
3.1.2 Episódio II. "Lemniscata do Gato Verde"

Este projeto foi construído pelo estudante Dave e é uma animação de uma espécie de viagem de um gato verde por diversos cenários. Não foi classificada como uma história porque não há um final propriamente dito e também não há indicação de que haja algum sentido na construção dos cenários (nem na fala do aluno nem no seu relatório). Ademais, a animação se repete indefinidamente, já que foi usado o comando de controle "*sempre*", que engloba todo o código. Na Figura 9 podemos ver a tela inicial da animação produzida pelo aluno.

Dave construiu um código de tamanho considerável comparado com os outros trabalhos da turma. Logo na aula de introdução ao ambiente do *Scratch*, foi possível notar que ele possuía alguma facilidade no trato com os comandos. Foi perguntado se já havia trabalhado com alguma linguagem de programação anteriormente, ao passo que respondeu

que já assistira alguns tutoriais no Youtube sobre jogos e programação. Nota-se que este aluno foi um dos poucos a usar os condicionais "*se/então*" no seu trabalho (Figura 10).

Figura 9 – Tela inicial do Episódio II.



Fonte: própria.

Figura 10 – Trecho do código da programação do Episódio II.

Fonte: própria.

A seguir, destacamos algumas passagens da fala do aluno em sua apresentação para os colegas.

Excerto 1: Experimentando sozinho

[13:43] **Vinícius:** Como é que você descobriu esse comando?

[13:43] **Dave:** Eu descobri tudo, na verdade eu deixei uma aba aberta como se fosse um bagulho de testes. Eu pegava o mesmo cenário só que fazia testes ali, então eu pegava todas as... todos os códigos e fazia de teste. E os que dessem certo eu passava pro principal.

[14:01] **Vinícius:** Então tu fazia teste pra ver o que o comando fazia?

[14:02] **Dave:** Sim. Aí eu abria outra aba aqui, botava no *Scratch*, botava o mesmo cenário, os mesmos personagens, e só ia testando, tanto que no outro deu quase duzentos e tantos comandos, praticamente porque eu usei quase todos os comandos só pra ver o que acontecia, como que cada um funcionava. Porque eu acho que pesquisar não dá muita graça, entendeu? Eu não vi tutorial, não vi nada. Foi só por descobrir, daí fica mais empolgante fazer por descobrir. Mais fascinante ficar descobrindo sozinho.

A experimentação também toma espaço na fala do aluno Dave, que orgulha-se do fato de ter descoberto as funcionalidades da linguagem de maneira individual. Observamos que essa exploração foi feita por meio de testes e simulações, semelhantes ao visto no Episódio I (subseção 3.1.1) e nos trabalhos em geral realizados pelos estudantes.

Excerto 2: Descobertas

[18:07] **Dave:** E outro erro que eu tinha encontrado era que a bola não tava indo certo, parecia que tava dando algum problema, mas é porque eu tava fazendo errado mesmo. Daí depois eu consegui consertar já na parte de teste, na aba de teste.

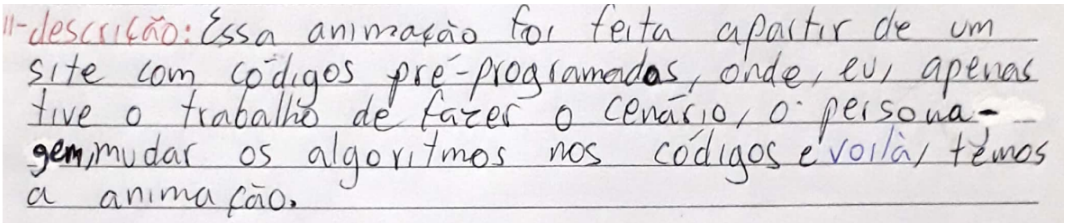
[18:15] **Vinícius:** O que tu estava fazendo errado?

[18:17] **Dave:** Tava dando errado que a bola não tava indo pro lugar certo e tipo, ela tava ficando parada no início da fase. Foi daí que eu descobri os comandos do *mostre* e do *esconda*. Daí quando eu boto no início ele esconde até a parte que eu quero que ele mostre.

Vemos também mais falas na direção da descoberta e do ato de experimentar e simular situações. Ao realizar testes em um cenário diferente, o aluno conseguia visualizar a ação dos comandos de maneira mais clara, o que possivelmente facilitou a construção do seu projeto no programa principal.

A seguir — Figura 11 e Figura 12 — estão trechos destacados do relatório entregue pelo aluno Dave, junto com suas transcrições.

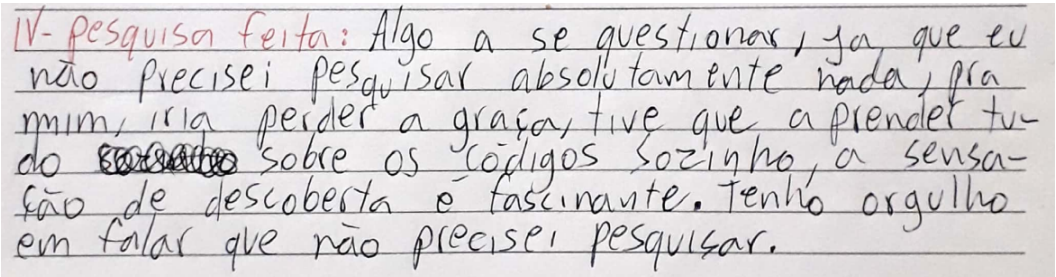
Figura 11 – Trecho do relatório do Episódio II.



Fonte: própria.

Transcrição do escrito da Figura 11: "*Essa animação foi feita a partir de um site com códigos pré-programados, onde eu apenas tive o trabalho de fazer o cenário, o personagem, mudar os algoritmos nos códigos e 'voilà', temos a animação*".

Figura 12 – Outro trecho do relatório do Episódio II.



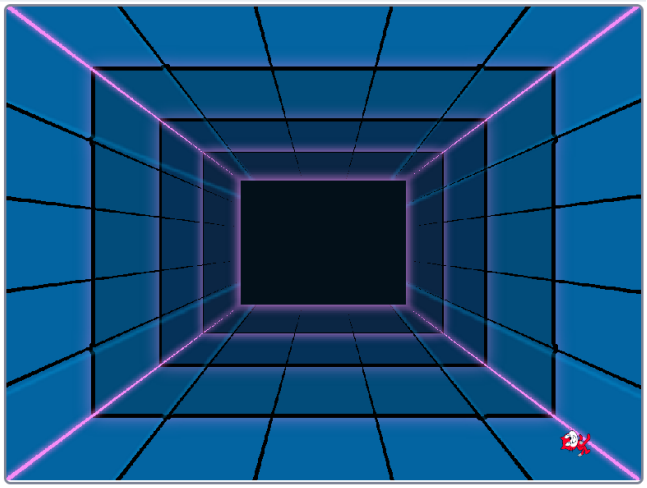
Fonte: própria.

Transcrição do escrito da Figura 12: "*Algo a se questionar, já que eu não precisei pesquisar absolutamente nada. Pra mim iria perder a graça. Tive que aprender tudo sobre os códigos sozinho. A sensação de descoberta é fascinante. Tenho orgulho em falar que não precisei pesquisar*".

3.1.3 Episódio III. "Corrida 3D Dimensional"

Este episódio consiste em uma animação construída pela aluna Sandy, que tem por personagem um gato que se "teletransporta" por cenários diversos. Também não foi classificada como uma história porque não há uma sequência de acontecimentos durante a sua execução, apenas movimentos aleatórios, que se repetem por três vezes, sem um final explícito. A Figura 13 mostra a tela inicial do projeto.

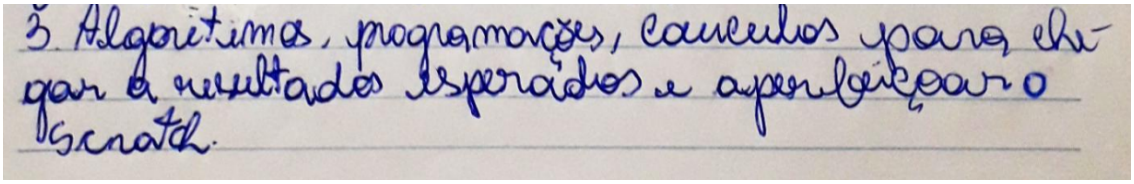
Figura 13 – Tela inicial do Episódio III.



Fonte: própria.

Não há nada de muito diferente no código além de movimentações e trocas de fantasias: apenas um comando "*repita*", que engloba todos os outros, escolhido para realizar três repetições. O relatório entregue pela aluna aponta as suas considerações acerca da Matemática que ela acreditou estar envolvida no processo de criação do seu trabalho (Figura 14).

Figura 14 – Trecho do relatório do Episódio III.



Fonte: própria.

Transcrição do texto da Figura 14: "*Algoritmos, programações, cálculos para chegar a resultados esperados e aperfeiçoar o Scratch*".

Em seguida, há fragmentos da fala da aluna durante a sua apresentação.

Excerto 1: Dividir...

[26:53] **Vinícius:** Sentiu alguma dificuldade para fazer o código?

[26:56] **Sandy:** Bom, eu senti dificuldade quando eu tava fazendo... muitas vezes ficou pronto e ele começava a bugar em alguma parte, eu não entendia [o] porquê.

[27:06] **Vinícius:** E como é que tu fazia pra descobrir onde que estava o erro?

[27:08] **Sandy:** Bom, eu dividia em partes e ficava rodando várias e várias vezes até eu encontrar o problema.

Neste trecho, observamos que a aluna traçou a estratégia de dividir os comandos em pedaços menores a fim de serem testados. Esse ato é precisamente o que definimos como decomposição, uma das táticas chaves que estão relacionadas ao pensamento computacional.

Excerto 2: ...em partes

[27:35] **Sandy:** Algumas vezes ele [o ator] chegava em um ponto que ele ia pra um lugar superaleatório que eu não tinha mandado. Eu não entendia [o] porquê, então eu tinha que dividir e fazer tudo de novo até entender o problema.

Novamente, fica evidente o caráter de teste realizao pela aluna mediante a separação em partes do que ela estava pretendendo realizar. As dimensões da decomposição e da avaliação se mostram aparentes no discurso da estudante.

3.2 Grupo 2: Histórias

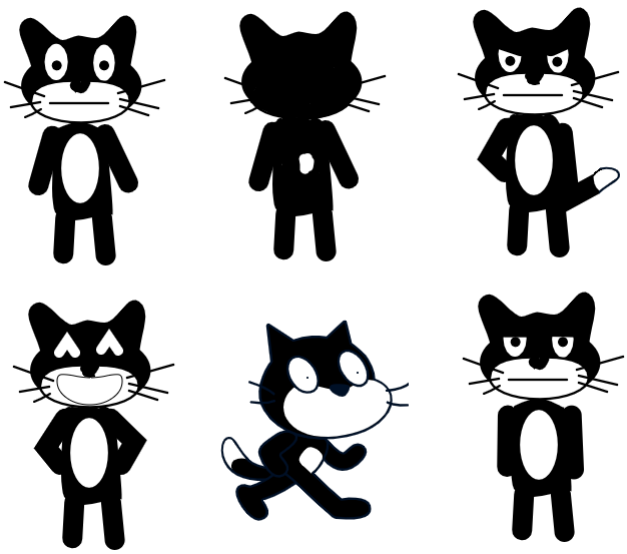
Neste grupo, caracterizamos os trabalhos que foram construídos através de uma animação, sem interatividade, mas que possuíam uma história com fio condutor. Há nesses episódios uma cronologia, que aparece como uma lógica dos acontecimentos, possuindo um começo, meio e fim. Apresentaremos aqui um dos trabalhos que teve essa classificação.

3.2.1 Episódio IV. "O Gato vs. O Programador"

Este projeto, construído individualmente pelo aluno Nicko, consiste em uma história animada, que conta o trajeto de um gato que passa por um cenário de uma praça e mantém um diálogo com o programador, pedindo para que ele troque os cenários ao fundo. O programador nem sempre faz as vontades do gato e isso o deixa zangado durante a história, o que fica aparente nas diferentes fantasias desenhadas pelo aluno, que podem ser conferidas na Figura 15.

Este aluno apresentava bastante dificuldade durantes as aulas, não só nas de Matemática, mas na maioria das disciplinas. Durante o período de ambientação com o funcionamento do *Scratch*, Nicko teve extrema dificuldade de entender o funcionamento dos comandos, mesmo os mais simples de movimento, solicitando ajuda constantemente. Passou tempo considerável tentando fazer desenhos para os cenários e personagens — o que podemos notar na Figura 15, que fez parte da sua versão final da história — e pouco tentando fazer a estruturação do seu código. Realizou a maioria do trabalho em casa, e apresentou um projeto bastante surpreendente nos quesitos criatividade e funcionamento. A Figura 16 mostra as telas iniciais da sua história.

Figura 15 – Fantasias do gato no Episódio IV.



Fonte: própria.

Figura 16 – Telas iniciais do Episódio IV.

Fonte: própria (adaptada para preservar identidade do aluno).

A seguir, estão alguns excertos da fala do aluno durante a sua apresentação e do relatório escrito entregue por ele ao final dos trabalhos.

Excerto 1: Quando não tem número não é Matemática

[11:56] **Vinícius:** O que tu usou de Matemática para construir o teu trabalho?
[11:59] **Nicko:** Bem, eu não sei Matemática, mas eu usei o que eu posso, né.
[12:03] **Vinícius:** E o que tu pôde usar?
[12:04] **Nicko:** **[suspiro]**
[12:06] **Vinícius:** Tu acha que não usou nada de Matemática?
[12:08] **Nicko:** Na verdade sim, né. Porque... 175. **[coordenada x do personagem na tela]**
[12:14] **Vinícius:** Ah, tem números ali. Quando tem número é Matemática?
[12:17] **Nicko:** É... **[risos]**
[12:18] **Vinícius:** E quando não tem número não é?
[12:21] **Nicko:** É óbvio!

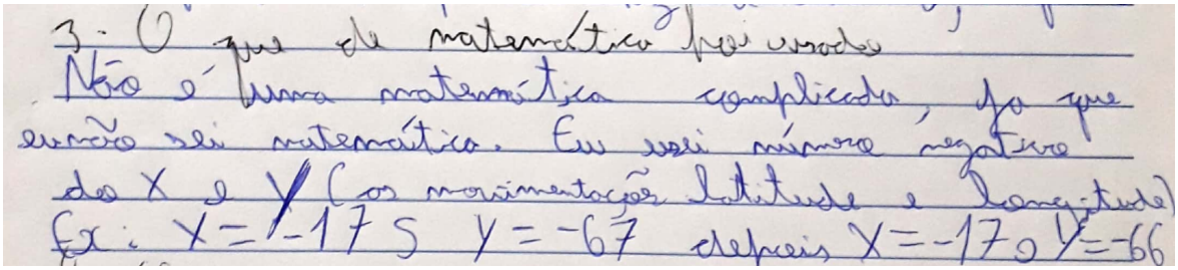
Percebemos na fala do aluno a sua autoimagem construída de quem não sabe Matemática. Mais ainda, parece-nos que essa identidade assumida pelo aluno é imutável, ou seja, alguém que não sabe Matemática nunca poderá aprendê-la. A ideia de que a Matemática trabalha somente com números também fica evidente no discurso do aluno, corroborando com o estigma social a disciplina carrega.

Excerto 2: Duplicar

[13:30] **[mostrando a animação novamente]**
[13:38] **Nicko:** Tipo, essa posição que ele tá na frente assim, ó. Foi a mesma, eu dupliquei e eu montei diferente, entendeu?

Neste trecho, o aluno tenta explicar como fez para construir as diversas fantasias do seu personagem. Como já mencionado, ele empregou bastante do seu tempo em aula nessa tarefa.

Figura 17 – Trecho do relatório do Episódio IV.



Fonte: própria.

Transcrição do texto da Figura 17: "*O que de Matemática foi usado? Não é uma matemática complicada, já que eu não sei Matemática. Eu usei número negativo do x e y (as movimentações latitude e longitude). Ex: $x = -175$ e $y = -67$, depois $x = -17$ e $y = -66$*".

3.3 Grupo 3: Jogos

Neste grupo, separamos os episódios que possuem algum tipo de interatividade com o usuário. Eles não se restringem a simplesmente mostrar algo acontecendo na tela, pré-programado e constante, há a possibilidade de agir e influenciar o que acontece, de maneira dinâmica, durante a execução. Dos trabalhos classificados como jogos, elencamos um para ser apresentado.

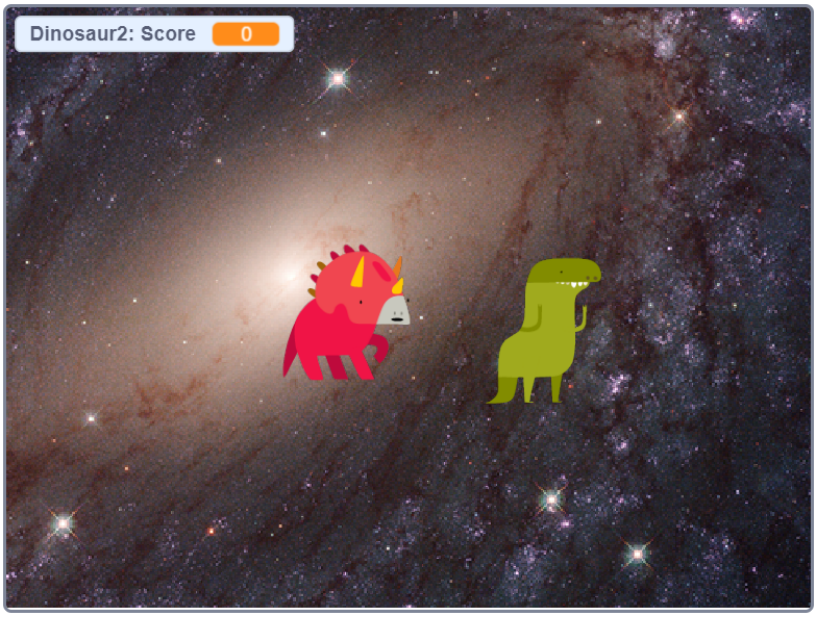
3.3.1 Episódio V. "Adventures of Dino"

Este episódio foi montado pelas alunas Shania e Simone e consiste em um jogo de "pega-pega" com personagens na tela que são dois dinossauros. O objetivo é fazer o dinossauro vermelho perseguir o dinossauro verde utilizando as setas do teclado, de modo a contar pontos cada vez que se consegue alcançá-lo: passados 40 segundos, se a pontuação for superior a 100 pontos o jogador é declarado vencedor, caso contrário, perdedor. A Figura 18 mostra a tela inicial do jogo.

Ambas as alunas se dedicaram durante todo o período das aulas, fazendo perguntas sobre a linguagem, questionando sobre dúvidas e funcionamento de alguns comandos e mostrando seus progressos. O relatório destas alunas mostra uma descrição do código

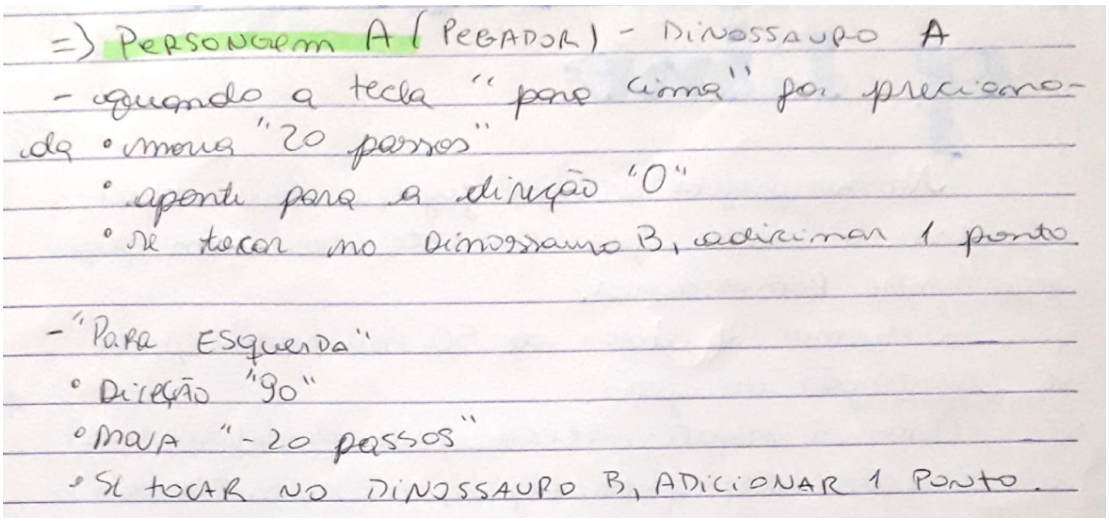
do programa feito por elas, praticamente em formato de pseudocódigo[1]. O escrito é basicamente as instruções dadas aos atores na programação do jogo, mas se utilizando um pouco mais de linguagem corrente, como podemos ver na Figura 19.

Figura 18 – Tela inicial do Episódio V.



Fonte: própria.

Figura 19 – Trecho do relatório do Episódio V: pseudocódigo.



Fonte: própria.

[1] Pseudocódigo é definido como um sistema genérico de alto nível de descrição de algoritmos ou programas, que se utiliza principalmente da língua materna. O intuito do seu uso é permitir que o desenvolvedor se preocupe mais com a lógica do algoritmo, sem haver a necessidade de se ater a detalhes da sintaxe de uma linguagem de programação específica, de forma que o escrito possa ser lido e compreendido por qualquer pessoa (RALSTON; REILLY, 1995).

Transcrição da Figura 19:

"*Personagem A (PEGADOR) – DINOSSAURO A*

– *quando a tecla 'para cima' for pressionada*

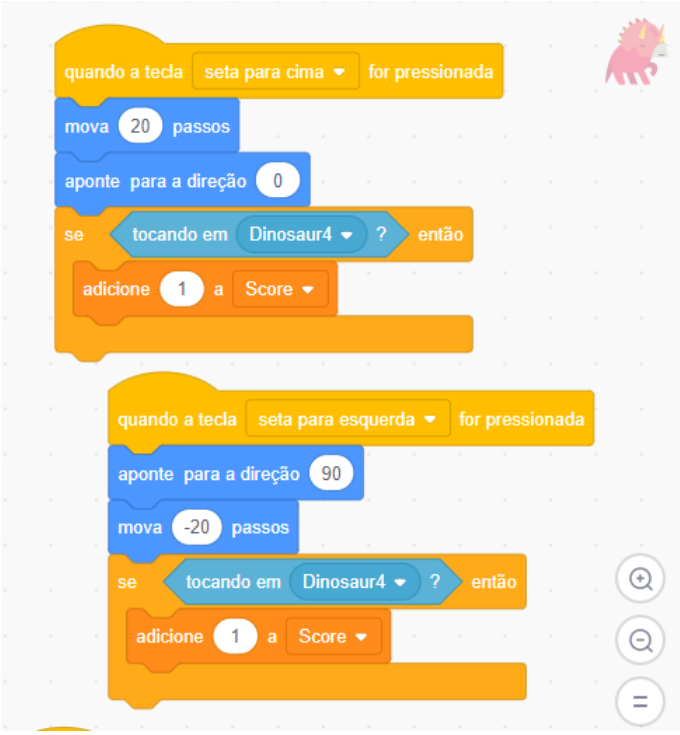
- *mova '20 passos'*
- *aponte para a direção '0'*
- *se tocar no DINOSSAURO B, adicionar 1 ponto.*

– *'Para esquerda'*

- *Direção '90'*
- *Mova '–20 passos'*
- *Se tocar no DINOSSAURO B, adicionar 1 ponto."*

Podemos notar que para a marcação dos pontos foi necessário utilizar o conceito de variável[2], o qual não foi utilizado por nenhum dos outros grupos durante as práticas. Junto a isso, as alunas precisaram empregar comandos do tipo "*se/então*" e dividiram os códigos em módulos, um para cada ação das teclas. Destacamos um trecho do código das estudantes (Figura 20) e das suas falas durante a apresentação a seguir.

Figura 20 – Trecho do código do Episódio V.



Fonte: própria.

[2] Uma variável é um identificador associado à uma posição de memória, utilizada pelo computador para o armazenamento de informação, podendo mudar ou não o seu conteúdo durante a execução do programa (RALSTON; REILLY, 1995).

Excerto 1: A variável

[24:09] **Vinícius:** Eu vi ali que vocês utilizaram um recurso que eu acho que ninguém tinha utilizado ainda, que é o recurso de variável.

[24:13] **Simone:** Sim, no placar.

[24:14] **Vinícius:** Isso! Variável é uma instância que vai modificando o seu valor de acordo com o programa, conforme o programa roda e o que acontece. Como é que vocês descobriram essa funcionalidade?

[24:46] **Simone:** Hum, a gente veio até aqui... [**mexendo o mouse e mostrando na tela do *Scratch***]

[24:47] **Shania:** Teste, no caso.

[24:47] **Vinícius:** Foram testando?

[24:50] **Simone:** Ah, foi um inferno. [**risos**]

Fica evidente também na fala das alunas que a experimentação cumpriu um papel importante nos seus trabalhos. Através de testes elas conseguiram descobrir uma funcionalidade pouco explorada pelos seus colegas, a variável.

Excerto 2: Andar para trás

[24:57] **Vinícius:** O que mais vocês utilizaram de Matemática?

[25:00] **Shania:** A gente usou bem da parte de deslizar, essas coisas. O tamanho deles a gente mudou também, e acho que foi isso pra falar a verdade. Acho que tem número negativo ali também que a gente teve que usar.

[25:15] **Simone:** Ah, no mova 20 passos, pra ele ir para trás a gente botou tipo mova menos 20. Pra andar pra trás tem que colocar negativo.

Aqui fica manifesta postura de descoberta e da capacidade de simulação (possível abstração), ao perceber que o movimento para a esquerda deveria ser feito diminuindo as coordenadas. Ou seja, as alunas compreenderam o funcionamento de um sistema de coordenadas cartesianas retangulares e se utilizaram dele para indicar as posições e movimentos dos seus personagens na tela.

4 Análise dos dados: consonâncias

I want you to know
For all that you do
Well it's not too late to rearrange
The pieces of your heart that made you go insane

I Want You To Know (Virgo) – Andre Matos

Este capítulo é destinado à análise e discussão acerca dos episódios apresentados no capítulo anterior, tendo como aporte teórico os referenciais desenvolvidos ao longo do texto nos capítulos 1 e 2. Utilizamo-nos dos fragmentos de fala e das imagens associadas aos episódios para destacarmos evidências de manifestações das dimensões básicas do pensamento computacional — decomposição, abstração, generalização, algoritmos e avaliação — e possíveis conexões com ideias matemáticas ao longo dos dados produzidos, sempre tendo em mente a questão orientadora "***Quais as potencialidades do uso do Scratch para o desenvolvimento das habilidades relacionadas ao pensamento computacional?***". Temos consciência de que o referencial elencado para a pesquisa não é único e que as pesquisas e iniciativas que visam a introduzir (ou ainda, analisar a presença de) rudimentos de computação nos currículos da educação básica estão se delineando, sobretudo quando falamos na realidade brasileira.

A análise dos dados pode ser compreendida como

> [...] o processo de busca e de organização sistemático de transcrições de entrevistas, de notas de campo e de outros materiais que foram sendo acumulados, com o objetivo de aumentar a sua própria compreensão desses mesmos materiais e de lhe permitir apresentar aos outros aquilo que encontrou. A análise envolve o trabalho com os dados, a procura de padrões, descoberta dos aspectos importantes e do que deve ser aprendido e a decisão sobre o que vai ser transmitido aos outros. (BOGDAN; BIKLEN, 1994, p. 205)

Como já exposto, baseamo-nos em anotações, em fragmentos das falas dos alunos, em seus escritos sobre seus projetos e em imagens capturadas de vídeo e dos próprios códigos disponibilizados pelos estudantes para proceder a produção dos dados e é nisso que nos firmamos para a presente análise. Frente a esses dados, procuramos dar um olhar reflexivo e cuidadoso, tentando capturar as nuances por eles reveladas.

De modo algum pretendemos colocar respostas definitivas à nossa questão ou a outras questões semelhantes referentes ao pensamento computacional no âmbito do ensino básico. Pretendemos aqui, de maneira elementar, analisar qualitativamente como alguns alunos manifestaram as habilidades de abstração, generalização (reconhecimento de padrões), decomposição, algoritmos e avaliação no encadeamento dos seus pensamentos ao longo da resolução de um problema. Essas instâncias não são manifestadas todas em conjunto, ao menos não nos casos a serem analisados, mostrando que o pensamento computacional pode aparecer de maneira incompleta. Isso revela um processo individual de pensamento, que se exprime de formas distintas e está em contínuo desenvolvimento.

No que diz respeito à habilidade de abstração, que é entendida por Brackmann (2017) como a capacidade de reconhecer elementos relevantes em detrimento a outros que são dispensáveis à situação, filtrando e classificando os dados e criando-se uma imagem ou representação do que se quer resolver, podemos dizer que ela não é uma característica que se manifesta de maneira tão explícita na fala dos alunos. A abstração é o componente fundamental quando lidamos com as dificuldades e complicações inerentes a um problema (WING, 2010), no entanto o ato de descartar detalhes não importantes pode passar despercebido quando tentamos elaborar uma estratégia para um problema. Em nosso entendimento, todos os alunos necessitaram dessa habilidade para construírem seus projetos durante as atividades propostas, já que o que se via na tela do computador era somente uma simulação que se utiliza de imagens para compor uma realidade virtual, ou seja, uma representação simbólica. Mas poucos se dão conta de que a usam para outras ações durante o desenvolvimento dos seus trabalhos.

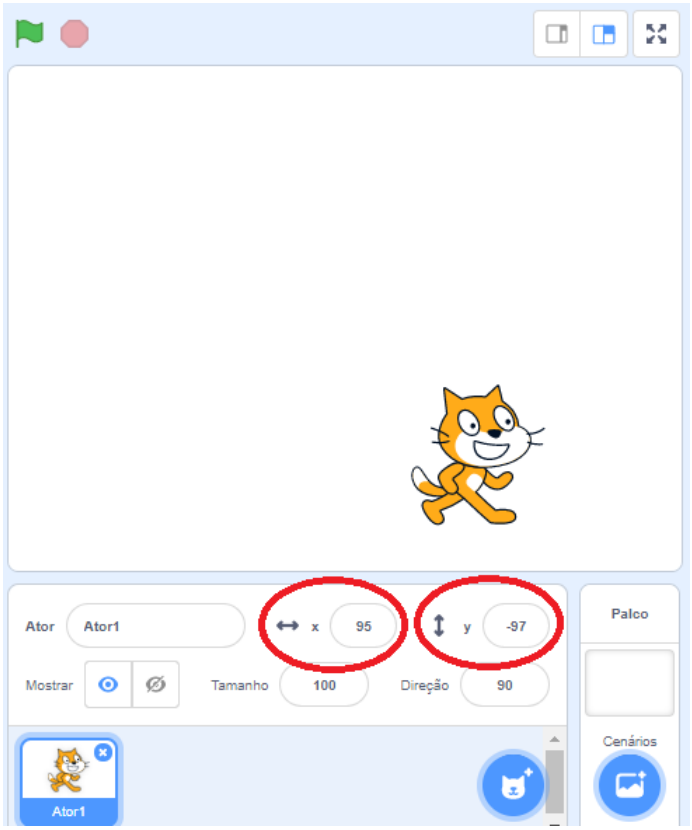
Ademais, o movimento de abstração parece estar amarrado com o movimento de generalização, ambos cumprindo papel crucial no delineamento das estratégias iniciais de solução, o que torna difícil a sua análise apenas se valendo dos códigos finais (ou até mesmo dos parciais) dos programas. Mesmo a presente pesquisa tendo observado o caminho do estudantes, devido ao grande número de sujeitos de pesquisa envolvidos nas práticas, não foi possível um acompanhamento mais direto dos alunos nos instantes em que eles estavam no princípio dos seus raciocínios. Deste modo, os dados que temos para analisar as manifestações dessa habilidade podem ser mais escassos.

A ideia da abstração, segundo a ótica aqui considerada, foi utilizada, segundo nosso entendimento, por todos os alunos ao terem que movimentar seus atores pelos cenários. Uma característica do *Scratch* é que o ponto central da tela tem posições marcadas em $x = 0$ e $y = 0$, ou seja, pode ser identificado com a origem de um sistema cartesiano de coordenadas retangulares. Salientamos que, no momento em que se encontravam nas práticas, nenhum dos alunos havia estudado sistema de coordenadas, assunto previsto para o final do ano letivo, segundo a professora titular. Deste modo, ao haver a necessidade de mover os personagens e as imagens pela tela, os alunos tiveram que se valer das

posições horizontal e vertical assinaladas segundo um sistema de coordenadas. Muitos alunos perceberam que, ao mover um ator na tela, a sua posição aparecia designada no canto direito, abaixo do cenário (Figura 21).

Valendo-se desse recurso, a maioria deles utilizou a estratégia de arrastar os atores e observar a posição indicada pelo programa para descobrirem quais valores deveriam colocar nos comandos de movimento ("*mover*", "*deslizar*", "*ir para posição*"). Porém, foi observado que alguns deles utilizaram a tática de "chutar" valores para as posições e verificar se o ator se colocava no lugar desejado. No Episódio I, o aluno Bruce se refere ao fato de arrastar a bola e observar as coordenadas, testando possíveis valores para as posições e para o tempo de descolamento dos personagens: "*Eu só fui acrescentando... eu só tive que acrescentar essas duas direções que a bola tinha que tomar, [por] que depois que ela entrava aqui ela só tinha que descer e fazer assim. Aí eu só acrescentei essas duas direções que é só pra subir e descer.*" (Episódio I, Excerto 1, página 49). Nessa fala (e nos excertos do primeiro episódio), podemos notar que quando Bruce diz "direção" ele na verdade quer se referir à "posição" do ator no cenário da animação. Outro trecho que evidencia a mesma tática está na fala "*Ah, a gente ficou fazendo uns testes, daí a gente acrescentava...*" (Episódio I, Excerto 2, página 49).

Figura 21 – Indicação de posição *XY* no *Scratch*.



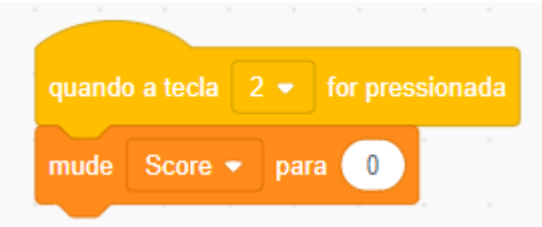
Fonte: própria.

Os alunos Nicko (Episódio IV) e Steve (Episódio I), assim como alguns outros, demonstraram dificuldades iniciais significativas no trato com os movimentos dos personagens na tela, possivelmente derivadas das suas adversidades com a disciplina de Matemática ou então do fato de ainda não conhecerem propriamente como funciona um sistema de coordenadas. Mesmo assim, ao longo das atividades, conseguiram se apropriar da linguagem visual utilizada pelo *Scratch* e realizar construções, manifestando resquícios de uso da abstração, no sentido de se utilizarem de uma representação simbólica de uma realidade.

Encontramos evidências de abstração também nas falas das alunas Simone e Shania (Episódio V), quando se referem ao fato de fazer o ator "andar para trás" no cenário. As estudantes perceberam que para movimentar o dinossauro do seu jogo para a esquerda (e também para baixo, apesar de não mencionarem isso no seu discurso) seria necessário indicar movimento negativo para o personagem. Destacamos um pedaço da fala da aluna Simone: "*Ah, no mova 20 passos, pra ele ir para trás a gente botou tipo mova menos 20. Pra andar pra trás tem que colocar negativo.*" (Episódio V, Excerto 1, página 60). Frisamos, novamente, que o Plano Cartesiano ainda não tinha sido estudado, mostrando aí um início de apropriação de um conceito que será utilizado mais adiante.

Ainda na construção do Episódio V, observamos a utilização de uma variável para a montagem do placar do jogo das alunas. O entendimento delas foi de que necessitavam de uma maneira de marcar a contagem da pontuação a fim de caracterizar a jogada como vencedora ou não. Houve a necessidade de se notar que o placar deveria ser zerado cada vez que reiniciava a partida, o que as alunas realizaram de maneira manual (Figura 22).

Figura 22 – Reinicialização manual da variável *Score*.



Fonte: própria.

Com relação à habilidade de generalização e reconhecimento de padrões, que diz respeito ao ato de identificar similaridades e propriedades comuns a objetos com intenção de resolver de forma eficiente problemas complexos (LIUKAS, 2015), podemos destacar que a maioria dos estudantes não foi tão bem-sucedida em distinguir correspondências comuns entre as várias ações que aconteciam nos seus projetos. Observamos que ao pensarem nos movimentos dos seus personagens ao longo dos cenários, longas linhas de códigos repetitivos apareceram. Estas linhas poderiam ser trocadas por comandos de controle, do

tipo "*repita*" ou "*repita até que*", mas quase nenhum dos alunos os usou ou pensou usá-los de modo a tornar seus códigos mais claros e enxutos. Na Figura 23 podemos examinar um exemplo de construção, mostrada pelo aluno Nicko (Episódio IV) no seu código, que poderia ter sido escrita de forma iterativa caso houvesse sido identificado um padrão nas ações do seu ator.

Apesar disso, esse aluno mostrou traços de ideias de generalização (e também de abstração) no seu discurso, quando explicou o modo como construiu as diversas fantasias do seu ator. Isso fica evidente na sua fala "*Tipo, essa posição que ele tá na frente assim, ó. Foi a mesma, eu dupliquei e eu montei diferente, entendeu?*" (Episódio IV, Excerto 2, página 56). De alguma maneira o aluno percebeu que as fantasias teriam algumas semelhanças entre elas, e, para construir novas, simplesmente se utilizou de uma já pronta para aproveitar essas equivalências e modificar somente os detalhes necessários.

Figura 23 – Linhas de comando repetitivas no Episódio IV.

Fonte: própria.

É possível observarmos certos sinais do uso de rudimentos de generalização no emprego das instruções "*sempre*" e "*repita*", que apareceram nos códigos feitos pelos alunos Dave e Sandy (Episódios II e III, respectivamente) (Figura 24). O uso desses comandos mostra que os alunos pensaram em uma repetição de coisas e notaram que não seria necessário escrever mais de uma vez essas instruções, bastava indicar para que se repetissem um certo número de vezes.

Figura 24 – Trechos dos códigos dos Episódios II e III, respectivamente.

Fonte: própria.

Conseguimos notar também delineamentos de generalização no relatório escrito pelo aluno Dave (Episódio II, Figura 11), que diz ter feito a sua animação "*a partir de um site com códigos pré-programados*" e que o trabalho foi apenas o de "*fazer o cenário, o personagem, mudar os algoritmos nos códigos*" e tudo apareceu praticamente em um passe de mágica ("*voilà*"). Parece aí que o aluno não valorizou o seu esforço ou então a atividade proposta na prática, talvez por tê-la achado um tanto simples, já que alegou possuir um conhecimento prévio acerca de programação. Mas o fato é que houve uma identificação do aluno de que os códigos podem ser reutilizados em outras construções, que eles podem servir para outras situações e não somente para aquela em questão, mostrando que a ideia de generalização foi movimentada por parte do estudante, ao passo que isso seria precisamente a ação de se utilizar de coisas já feitas anteriormente, aplicando uma solução genérica ao problema.

A decomposição, que é a ação de dividir um problema em subproblemas, com o intuito de diminuir a sua complexidade (WING, 2014), se mostrou evidente na fala da aluna Sandy, quando ela explicou a maneira como tentava achar os erros no seu código (essa ação também faz parte da habilidade de avaliação, pontuada mais à frente): "*Bom, eu dividia em partes e ficava rodando várias e várias vezes até eu encontrar o problema.*" (Episódio III, Excerto 1, página 54). Logo em seguida ela repete e complementa seu raciocínio: "*Algumas vezes ele [o ator] chegava em um ponto que ele ia pra um lugar superaleatório que eu não tinha mandado. Eu não entendia porque, então eu tinha que dividir e fazer tudo de novo até entender o pro-blema.*" (Episódio III, Excerto 2, página 54). Essa é exatamente a estratégia que programadores profissionais executam ao procurarem por erros nos seus códigos: ao dividir o problema em partes é possível observar com maior cuidado e clareza o

que está realmente acontecendo — o que deveria e o que não deveria acontecer. Tal técnica é responsável por tornar tanto o trabalho em construir códigos quanto a manutenção deles muito menos custosa em termos de tempo e complexidade, sendo uma prática essencial a um bom profissional da área. Além disso, quando se olha para pedaços menores dos problemas tem-se uma nitidez maior do que se deve focar, desviando a atenção de aspectos irrelevantes (dimensão relacionada à abstração).

Outra observação importante que podemos fazer com relação à decomposição está no fato das alunas Shania e Simone (Episódio V) terem se utilizado de blocos separados para cada uma das ações que o seu personagem deveria seguir (Figura 20). Este tipo de construção abarca o pensamento de que dividir um problema em subproblemas (no caso, sub-rotinas ou módulos) torna a solução mais simplificada. Isso também faz parte de indicações em diversos manuais de programação: "Simples é melhor que complexo" (PETERS, 2010, p. 301, tradução nossa).

No tocante ao algoritmo, elemento integralizador das habilidades do pensamento computacional (WING, 2014), podemos apontar especialmente o relatório da dupla Shania e Simone (Episódio V, Figura 19). Nele as alunas descreveram o seu programa em uma espécie de pseudocódigo, indicando as instruções utilizadas para fazer seu personagem se movimentar no cenário. Não houve sinais e também não foi observado que este pseudocódigo ou algum tipo de fluxograma tenha sido construído a priori, o que seria o indicado para a programação. Entretanto, vemos aí uma certa preocupação e movimentos na direção de transformação das linguagens e uma procura por clareza de expressão, apontados por Dijkstra (1974) como características do que ele chamou de pensamento algorítmico.

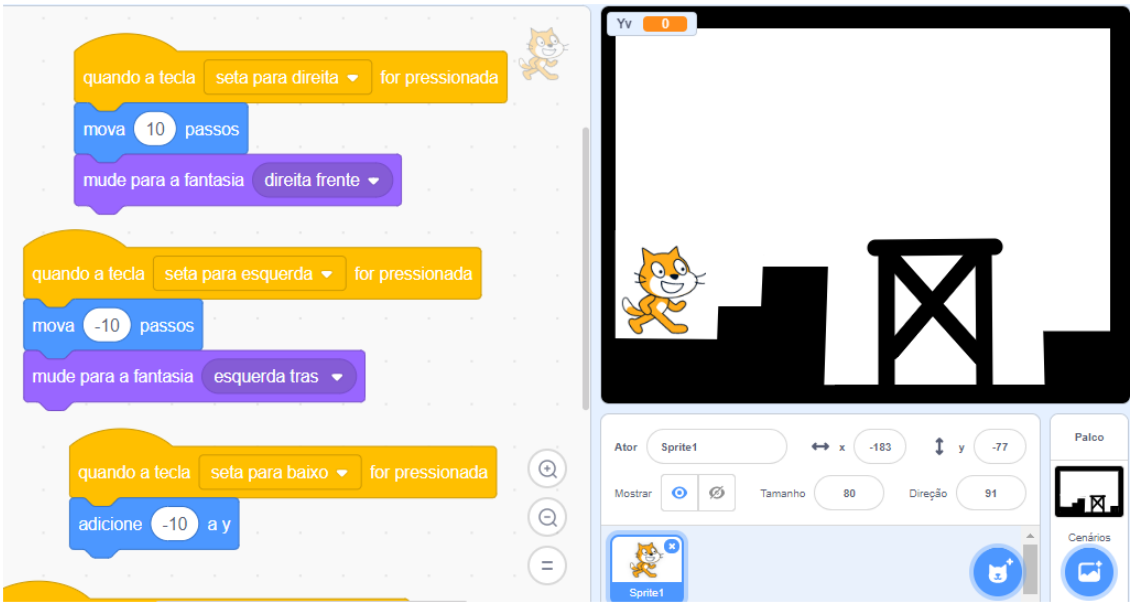
No trecho selecionado do relatório da aluna Sandy (Episódio III, Figura 14), vemos as palavras "*algoritmo*" e "*programação*" sendo citadas. Essa foi a única vez que essas palavras apareceram nas nossas observações, seja de forma escrita ou oral. Pensamos que a aluna porventura tenha sido influenciada pelas falas iniciais que antecederam as atividades no laboratório, nas quais foi conversado sobre o que seria um algoritmo, utilizando-se de exemplos corriqueiros como uma receita de bolo e ações quando se levanta da cama e se vai em direção à escola. No mesmo trecho, a aluna fala em "*cálculos para chegar a resultados esperados*". Esse seria o entendimento dela referente ao ato de pensar algoritmicamente: realizar cálculos — ela não especifica quais, mas em nossa interpretação entendemos isso como o caráter cultural da Matemática na sociedade, que é utilizada como sinônimo de fazer cálculos — a fim de atingir um objetivo desejado e organizar uma estratégia com a finalidade de resolver um problema.

Ainda no que tange a noção do algoritmo, podemos salientar as codificações dos Episódios II e V, nas quais foram utilizados os condicionais "*se/então*" (Figura 10 e Figura 19). O uso dessas instruções de controle não foi explorado por muitos estudantes, talvez por causa da natureza das suas construções, que não possuíam caráter interativo.

Na animação do aluno Dave (Episódio II) não seria necessária a sua utilização, já que não há uma lógica condicional envolvida. O programa sempre executará os comandos que estão internos à estrutura: uma linha sequencial de comandos bastaria, caracterizando aí um emprego supérfluo desse tipo de instrução. Acreditamos que o intuito do aluno ao usar tal tipo de instrução teria sido alguma reminiscência da sua ideia inicial de projeto, que seria um jogo no qual o personagem se movimentaria de acordo com as setas do teclado (Figura 25); assim, caso o personagem encostasse em determinada cor, algo aconteceria, caracterizando uma utilização adequada da instrução "*se/então*".

Já no código das alunas Simone e Shania (Episódio V), o uso do comando "*se/então*" se faz essencial, já que a marcação do placar é feita somente quando um personagem alcança o outro. Nesse ponto, temos uma caracterização do entendimento do funcionamento do comando: a instrução interna a ele será executada somente se o critério imposto for satisfeito (no caso, se tocar no ator). Ainda assim, foi observado um fenômeno assinalado por Miller (1981) e Pane, Ratanamahatana e Myers (2001): ao se especificar instruções com o uso de controle de fluxo (repetições e condicionais), as pessoas geralmente não utilizam cláusulas do tipo "*senão*", fazendo os seus condicionais serem exclusivamente unívocos. Nas suas pesquisas, eles encontraram evidências de que as pessoas considerariam essa como uma instrução inútil, já que, para elas, parece óbvio o que se deve fazer quando a condição de teste não é satisfeita, não sendo importante tal especificação.

Figura 25 – Primeira construção (abandonada posteriormente) do aluno Dave.



Fonte: própria.

Analisando o aspecto definido como avaliação, podemos notar que essa habilidade apareceu praticamente em todas as construções, nem que seja timidamente. A tática de tentativa e erro foi a mais usada, sendo observada explicitamente nos discursos dos alunos.

O aluno Bruce (Episódio I, Excerto 2, página 49) diz: "*Ah, a gente ficou fazendo uns testes, daí a gente acrescentava...*"; "*A gente tentou fazer, só que não deu certo.*". Já o aluno Dave (Episódio II, Excerto 2, página 52) fala de precisar consertar um erro no seu personagem: "*E outro erro que eu tinha encontrado era que a bola não tava indo certo, parecia que tava dando algum problema, mas é porque eu tava fazendo errado mesmo. Daí depois eu consegui consertar já na parte de teste, na aba de teste.*". Na fala da aluna Sandy (Episódio III) também podemos identificar o fato de ter havido um processo de avaliação. Ela menciona que dividia o código em partes para realizar testes e descobrir onde estava acontecendo o erro (Excerto 1, página 54), em um processo de depuração: "*Bom, eu senti dificuldade quando eu tava fazendo... muitas vezes ficou pronto e ele começava bugar em alguma parte, eu não entendia [o] porquê.*"; "*Bom, eu dividia em partes e ficava rodando várias e várias vezes até eu encontrar o problema.*". A aluna também escreveu no seu relatório o que seria uma tentativa de avaliação, ao citar que tinha que realizar procedimentos para "*aperfeiçoar o Scratch*".

Gostaríamos de tecer mais alguns comentários sobre dois aspectos referentes aos dados produzidos e às observações feitas durante a prática, que dizem respeito mais ao ambiente construcionista. O primeiro deles é concernente ao ato da descoberta. Ficou explícitado para nós na fala do aluno Dave (Episódio II) que esse aspecto foi deveras importante durante as suas atividades: "*Eu descobri tudo [...]. Eu pegava o mesmo cenário só que fazia testes ali, então eu pegava todas as... todos os códigos e fazia de teste. E os que dessem certo eu passava pro principal.*"; "*[...] eu acho que pesquisar não dá muita graça, entendeu? Eu não vi tutorial, não vi nada. Foi só por descobrir, daí fica mais empolgante fazer por descobrir. Mais fascinante ficar descobrindo sozinho.*" (Excerto 1, página 52). A redação do seu relatório vai na mesma direção (Figura 12): "*A sensação de descoberta é fascinante. Tenho orgulho em falar que não precisei pesquisar*". Apesar de vermos aí uma atitude um tanto arredia ao ato da pesquisa — que faz parte da descoberta —, o aluno se mostrou satisfeito com a possibilidade de conceber ideias e perceber coisas de maneira autônoma mediante a experimentação.

O descobrir também se manifestou no discurso das alunas Shania e Simone (Episódio V), no momento em que falaram do uso da variável para contar pontos no placar. Vemos também a experimentação aparecer quando o aluno Bruce (Episódio I) diz que a sua dupla tentou atribuir um tempo negativo para o movimento: "*No caso, se aumentasse o número, tipo, dos segundos ali pra 2 ele ia demorar mais pra chegar, se dimi[nuir]... A gente tentou botar menos 1, mas aí não deu muito certo...*" (Excerto 1, página 49). Não ficou claro o que os alunos pensaram com isso, já que ele mesmo admitiu na sua apresentação que tal ação não faria sentido, mas conseguimos perceber aí também a tentativa de se realizar algo e a descoberta sendo feita, nem que seja por um exemplo que falhou.

O segundo aspecto ao qual queremos chamar atenção diz respeito ao estigma social que a Matemática carrega e como ele pode afetar a autoestima e a visão do indivíduo sobre si mesmo. Identificamos uma identidade pré-concebida no discurso do aluno Nicko (Episódio IV). Segundo o que foi possível perceber a partir de conversas com outros professores e observações, esse aluno é bastante tímido e contido, tendo alguns problemas de se expressar, especialmente em público, demonstrando também dificuldades no aprendizado em diversas disciplinas. Ele fala: "*Bem, eu não sei Matemática, mas eu usei o que eu posso, né.*" (Episódio IV, Excerto 1, página 56). No seu relatório também há a ratificação da sua autoimagem: "*Não é uma Matemática complicada, já que eu não sei Matemática.*" (Figura 17). Papert (1988) fala sobre essa identidade construída ao citar o fluxo que a criança, originalmente ávida por conhecimento, percorre ao passar do matéfilo — amante da aprendizagem — ao matófobo — aquele que tem medo ou até aversão ao saber. A passagem de um estágio para o outro se dá por vários fatores conforme a criança atravessa a infância e se torna um adulto, e varia de indivíduo para indivíduo.

O ambiente construcionista vislumbrado por Papert tem como um dos objetivos reorganizar essas ideias no indivíduo. Os micromundos propostos seriam espaços onde a descoberta, o "[...] *vir a conhecer* [...], *explorar* [...], *adquirir sensibilidade* [...]" (PAPERT, 1988, p. 166, grifos do autor) teriam lugar cativo e se concretizariam de uma maneira mais consistente e afetiva. Mais ainda, as tecnologias digitais, em especial o computador, teriam um papel fundamental no rompimento dessas autoimagens de fracasso, seriam o *algo* (BAMPI *et al.*, 2013) notável a acontecer no ambiente escolar a fim de haver a quebra de paradigmas internos na autoimagem intelectual.

> Trabalhar nos micromundos da Tartaruga é um modelo de como a aprendizagem de uma ideia pode ser semelhante à maneira como conhecemos uma pessoa. Os alunos que trabalham nesses ambientes certamente descobrem fatos, fazem generalizações de proposições, e aprendem habilidades. Mas a experiência primordial de aprendizagem não é a memorização de fatos ou a prática de habilidades. (PAPERT, 1988, pp. 166–167)

A oportunidade de interagir com objetos-de-pensar-com típicos dos micromundos — no caso do *Scratch* temos os atores, no LOGO, a Tartaruga — tornaria a aprendizagem um momento de satisfação pessoal (como dito pelo aluno Dave, no Episódio II). Teria, então, o potencial de criar um movimento contrário ao observado tradicionalmente, fazendo o matófobo se transformar novamente em matéfilo, ou então bloquear o fluxo comum de transformação, como dito por Papert (1988).

5 Convergências

Here we go, carrying no longer sorrow
Standing up in the wind
Walk along, marching on for tomorrow
In this neverending way

For Tomorrow (Shaman) – Andre Matos

Aqui pretendemos atingir um fechamento, mesmo que incompleto e passageiro, frente ao que foi exposto durante o trabalho. Durante a pesquisa, tivemos a preocupação de estruturar as leituras, as práticas, os pensamentos e a escrita de modo a deixar as dissidências e os encontros dos múltiplos eus envolvidos no processo se confrontarem. Conversas entre o eu-aluno, o eu-pesquisador e o eu-professor convergiram para o presente texto, que se fez a partir dos múltiplos movimentos de dúvidas, procuras, inquietações, conversas, frustrações, contentamentos, respostas, meias-respostas e novas dúvidas.

Seguindo a interrogação primária "***Quais as potencialidades do uso do Scratch para o desenvolvimento das habilidades relacionadas ao pensamento computacional?***", procuramos encontrar algumas evidências de possíveis manifestações relacionadas às habilidades consideradas como pilares do pensamento computacional — decomposição, generalização, abstração, algoritmos e avaliação — em meio ao desenrolar das atividades realizadas. Mediante um olhar construcionista, analisamos os trabalhos, os percursos, os discursos e as expressões dos alunos quando confrontados com uma proposta de atividade de caráter aberto (ao menos bem mais aberto do que as tradicionalmente vivenciadas), que envolvia a produção de um artefato próprio e de interesse pessoal se utilizando da linguagem de programação *Scratch*.

Notamos, a princípio, que o aspecto mais livre da atividade causou um estranhamento nos alunos. Alguns não souberam como começar, perguntaram várias vezes "*o que é pra fazer mesmo?*", e demoraram para iniciar alguma construção. A proposta gerou um certo desconforto, já que o que vemos geralmente em sala de aula, sobretudo quando falamos em uma aula de Matemática, é um tipo de atividade mais impositiva: espera-se que as ordens sejam claras e objetivas, sem margem para discussão e inventividade. No entanto, ao adotarmos como viés epistemológico e metodológico o Construcionismo e a utilização de Tecnologias Digitais, temos uma oportunidade única de propiciar aos alunos um ambiente diferenciado, um momento de construção e produção de conhecimento, uma

tensão geradora de novas compreensões.

Pretendemos, acima de tudo, ao propormos uma atividade com o uso do recurso tecnológico, ir além da sua simples inserção e transposição de atitudes que não necessitariam da tecnologia para acontecerem. A nossa intenção foi nos aproveitarmos do humano-mídia, do objeto-de-pensar-com e do ambiente para avançar na direção de uma aprendizagem que, ao mesmo tempo que fosse profunda — pois geraria novos conhecimentos e os movimentaria com os já adquiridos — fosse também mais proveitosa e satisfatória — por se basear em interesses pessoais na construção de um produto próprio e passível de ser compartilhado. As Tecnologias Digitais entraram no contexto como integrantes do processo de produção dos saberes, como uma mídia capaz de manipular e ser manipulada em prol da aprendizagem.

Pegamos emprestadas as palavras de Rosa (2018, p. 257):

> [...] produzimos conhecimento com o mundo, com as Tecnologias Digitais que se encontram no mundo, e não sobre o mundo, sozinhos, de forma que essas tecnologias simplesmente nos auxiliam a pensar sobre algo.

Sendo mais específico em relação à nossa questão norteadora, observamos indícios de manifestação de todos os aspectos inerentes ao pensamento computacional durante as práticas. As perspectivas teóricas e os acontecimentos vivenciados durante as atividades se entrecruzaram, impelindo-nos a refletir sobre cada um dos pilares do pensamento computacional que iam emergindo. Suas conotações, confluências e possíveis relações com a Matemática também foram observadas, mostrando que de fato há um potencial a ser explorado no que diz respeito à inserção de noções de programação nas aulas, seja esse ensino concomitante à alguma disciplina — à Matemática ou às Ciências em geral — ou mesmo como um componente isolado, o que é defendido por alguns pesquisadores como Wolfram (2017).

As dimensões alusivas ao pensamento computacional apareceram no processo de construção e nas falas dos alunos. Ao compartilharem seus projetos com o grande grupo, eles puderam mostrar suas intenções e conquistas, ao passo que adentraram em mundo novo, já que a imensa maioria nunca havia programado antes. Percebemos que a familiarização com o *Scratch* e o início das estratégias se desenharam de formas diversas, mas os trabalhos selecionados, descritos e analisados aqui mostraram que as habilidades do pensamento computacional estão intrinsecamente ligadas ao ato de programar, sendo ele um potencial vetor para o desenvolvimento e aprimoramento delas.

Como destaque final, queremos mencionar que as reflexões e debates matemáticos e acerca do pensamento computacional em geral tomaram lugar de maneira natural no ambiente propiciado pelo *Scratch*. A colaboração mútua também se fez presente, sendo observada frequentemente em conversas entre alunos tentando auxiliar os colegas. Assim, esse ambiente se apresentou também como uma ótima ocasião para o estreitamento de

laços e reforço de conhecimentos por meio de troca, corroborando com as ideias de Piaget ao argumentar que a interação é parte importante do processo de aprendizagem.

Ademais, o uso da informática pode auxiliar na mudança do panorama de ensino da Matemática, mudança esta que se dá na direção de que a Matemática "pronta e acabada" tende a perder espaço. Há cada vez menos sentido em decorar tabuadas ou tabelas de integrais, visto que as respostas aos antigos exercícios estão na literatura, disponíveis na internet inclusive. A questão se torna identificar os problemas e realizar reflexões e simulações, buscando alternativas para eles, e aí entra a tecnologia como um fator potencializador e descomplicador. O uso dos softwares possibilita a visualização quase que imediata de soluções e estratégias, sejam elas intencionais ou não, e viabiliza a realização de testes que talvez fossem impossíveis ou imensamente trabalhosos se feitos a mão apenas com lápis e papel. Em especial à programação, ela "[...] é singular, pois a execução do computador oferece um feedback imediato e fiel, desprovido de qualquer interferência intelectual ou emocional." (MALTEMPI, 2004, p. 273).

Ao pensarmos a questão das Tecnologias Digitais na sala de aula, estamos ensinando e aprendendo ciência. Quando imaginamos uma maneira de solucionar um problema presente, investigamos recursos dentro do programa (da linguagem de programação) e essa pesquisa tem potencial de gerar novas perguntas que nos levarão a mais pesquisas. Este é o movimento que existe em nosso cotidiano e também é este viés que devemos apresentar dentro do ambiente escolar.

Por derradeiro, queremos dizer que a pesquisa não se finda aqui. Ela continua no dia-dia da profissão de professor e na vida cotidiana. Aprender a programar não é mais um privilégio e uma habilidade que está relegada aos profissionais da área da Computação. Como apontado por Wing (2006), ela se fará crucial na vida de todo cidadão em um mundo cada vez mais envolto em Tecnologias Digitais, no qual a aprendizagem deve ser constante. Pretende-se continuar nesse caminho, a fim de ampliar os horizontes do ensino de programação e torná-lo, de fato, parte integrante de uma educação livre.

Referências

AHO, A. V. Ubiquity symposium: Computation and computational thinking. *Ubiquity*, ACM, n. January, 2011. Citado na página 22.

ANDRADE, D. *et al.* Proposta de atividades para o desenvolvimento do pensamento computacional no ensino fundamental. In: *Anais do Workshop de Informática na Escola*. [S.l.: s.n.], 2013. v. 1, p. 169–178. Citado na página 35.

ARAÚJO, D. A. d.; SOARES, E. S. Calculadoras e outras geringonças na escola. *Presença Pedagógica*, v. 8, n. 47, p. 13–27, 2002. Citado na página 14.

ARAÚJO, J. d. L.; BORBA, M. d. C. Construindo pesquisas coletivamente em educação mateática. In: *Pesquisa qualitativa em Educação Mateática*. 5. ed. Belo Horizonte: Autêntica, 2017, (Coleção Tendências em Educação Mateática). p. 31–51. Citado na página 38.

BAMPI, L. R. *et al.* Numa brincadeira de aprendiz de feiticeira... surge algo. *Revista Sul Americana de Filosofia e Educação*, Brasília, v. 7, n. 10, p. 170–184, 2013. Citado na página 70.

BARCELOS, T. S.; SILVEIRA, I. F. Teaching computational thinking in initial series – an analysis of the confluence among mathematics and computer sciences in elementary education and its implications for higher education. *XXXVIII Conferencia Latinoamericana En Informática (CLEI)*, p. 1–8, 2012. Citado na página 36.

BARR, D.; HARRISON, J.; CONERY, L. Computational thinking: a digital age skill for everyone. *Learning & Leading with Technology*, v. 38, n. 6, p. 20–23, 2011. Citado 2 vezes nas páginas 24 e 25.

BASSO, M.; NOTARE, M. R. Pensar-com tecnologias digitais de Matemática Dinâmica. *RENOTE*, v. 13, n. 2, p. 1–10, 2015. Citado 2 vezes nas páginas 35 e 38.

BBC Learning. *Introduction to computational thinking*. 2015. Disponível em: <https://www.bbc.com/bitesize/guides/zp92mp3/revision/1>. Acesso em: 25 mai. 2019. Citado 2 vezes nas páginas 17 e 25.

BENTON, L. *et al.* Building mathematical knowledge with programming: insights from the scratchmaths project. Suksapattana Foundation, 2016. Citado 2 vezes nas páginas 29 e 30.

BENTON, L. *et al.* Bridging primary programming and mathematics: Some findings of design research in england. *Digital Experiences in Mathematics Education*, v. 3, n. 2, p. 115–138, 2017. Citado 2 vezes nas páginas 29 e 30.

BERS, M. U. *et al.* Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, v. 72, p. 145–157, 2014. Citado na página 24.

BICUDO, M. A. V. Pesquisa qualitativa e pesquisa qualitativa segundo a abordagem fenomenológica. In: *Pesquisa qualitativa em Educação Mateática*. 5. ed. Belo Horizonte: Autêntica, 2017, (Coleção Tendências em Educação Matemática). p. 111–124. Citado na página 37.

BIGARELLI, B. *Como a Estônia construiu uma sociedade digital*. 2018. Disponível em: <https://epocanegocios.globo.com/Tecnologia/noticia/2018/08/como-estonia-construiu-uma-sociedade-digital.html>. Acesso em: 26 mai. 2019. Citado na página 30.

BOGDAN, R. C.; BIKLEN, S. K. *Investigação qualitativa em educação: uma introdução à teoria e aos métodos*. Lisboa: Porto Editora, 1994. Citado 2 vezes nas páginas 37 e 61.

BORBA, M. d. C. O computador é a solução: mas qual é o problema? In: *Formação docente: rupturas e possibilidades*. 1. ed. Campinas: Papirus, 2002, (Cidade Educativa). p. 141–161. Citado 3 vezes nas páginas 14, 18 e 28.

BORBA, M. d. C.; PENTEADO, M. G. *Informática e Educação Matemática*. 5. ed. Belo Horizonte: Autêntica, 2015. (Coleção Tendências em Educação Matemática). Citado na página 38.

BORBA, M. d. C.; VILLARREAL, M. E. *Humans-with-media and the reorganization of mathematical thinking: Information and communication technologies, modeling, visualization and experimentation*. 1. ed. Nova Iorque: Springer US, 2005. (Mathematics Education Library, 39). Citado na página 14.

BRACKMANN, C. P. *Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica*. Tese (Doutorado) — UFRGS, Porto Alegre, 2017. Disponível em: <https://lume.ufrgs.br/handle/10183/172208>. Citado 6 vezes nas páginas 22, 23, 30, 31, 36 e 62.

BRASIL. *Base Nacional Comum Curricular – Educação é a Base*. 2018. Disponível em: <http://basenacionalcomum.mec.gov.br/abase>. Acesso em: 27 mai. 2019. Citado 5 vezes nas páginas 31, 32, 33, 34 e 35.

BROWDER, F. E. *Mathematical developments arising from Hilbert problems*. Dekalb, IL: American Mathematical Society, 1976. (Proceedings of Symposia in Pure Mathematics XXVIII, Part I). Citado na página 86.

CARNEIRO, R. F.; PASSOS, C. L. B. A utilização das tecnologias da informação e comunicação nas aulas de matemática: limites e possibilidades. *Revista Eletrônica de Educação*, v. 8, n. 2, p. 101–119, 2014. Citado na página 14.

CLARK-WILSON, A.; HOYLES, C. *Dynamic Digital Technologies for Dynamic Mathematics – Implications for teachers' knowledge and practice (Final Report)*. Londres: UCL Institute of Education Press, 2017. Citado 2 vezes nas páginas 29 e 30.

COMPUTADOR. *Houaiss Eletrônico*. Rio de Janeiro: Objetiva, 2009. Citado na página 82.

COMPUTER. *Online Etymology Dictionary*. 2019. Disponível em: <https://www.etymonline.com/search?q=computer>. Acesso em: 12 mai. 2019. Citado na página 82.

COUTO, G. M.; SILVA, M. d. G. M. da. *Educação e pensamento computacional: estudo de caso sobre a e-Estônia*. São Paulo: [s.n.], 2016. Disponível em: <http://www.abed.org.br/congresso2016/trabalhos/263.pdf>. Citado na página 30.

CSIZMADIA, A. *et al.* Computational thinking – a guide for teachers. 2015. Disponível em: <https://community.computingatschool.org.uk/resources/2324/single>. Citado 2 vezes nas páginas 17 e 25.

CSTA; ISTE. *Operational Definition of computational thinking for K–12 education*. NSF, 2011. Disponível em: <https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf>. Acesso em: 25 mai. 2019. Citado na página 24.

DALLA VECCHIA, R. *A modelagem matemática e a realidade do mundo cibernético*. Tese (Doutorado) — UNESP, Rio Claro, SP, 2012. Disponível em: <http://hdl.handle.net/11449/102151>. Citado 2 vezes nas páginas 40 e 46.

DELEUZE, G. *Proust e os signos*. 2. ed. Rio de Janeiro: Forense Universitária, 2003. Citado na página 12.

DENNING, P. J. Beyond computational thinking. *Communications of the ACM*, v. 52, n. 6, p. 28–30, 2009. Citado na página 25.

DIJKSTRA, E. W. Programming as a discipline of mathematical nature. *The American Mathematical Monthly*, v. 81, n. 6, p. 608–612, 1974. Citado 2 vezes nas páginas 20 e 67.

DIVERIO, T. A.; MENEZES, P. B. *Teoria da Computação: Máquinas Universais e Computabilidade*. 1. ed. Porto Alegre: Sagra Luzzatto, 1999. (Livros Didáticos, 5). Citado 4 vezes nas páginas 83, 86, 87 e 88.

DU BOULAY, B.; O'SHEA, T.; MONK, J. The black box inside the glass box: presenting computing concepts to novices. *International Journal of Man-Machine Studies*, v. 14, n. 3, p. 237–249, 1981. Citado na página 21.

EASTERBROOK, S. From computational thinking to systems thinking: A conceptual toolkit for sustainability computing. In: *ICT for Sustainability 2014 (ICT4S-14)*. [S.l.]: Atlantis Press, 2014. (Advances in Computer Science Research). Citado na página 25.

ENIAC. *McGraw-Hill Dictionary of Scientific & Technical Terms*. 2003. Disponível em: <http://encyclopedia2.thefreedictionary.com/ENIAC>. Acesso em: 2019-05-13. Citado na página 84.

ESHET-ALKALI, Y.; AMICHAI-HAMBURGER, Y. Experiments in digital literacy. *CyberPsychology & Behavior*, v. 7, n. 4, p. 421–429, 2004. Citado na página 29.

FEURZEIG, W. *et al.* Programming-languages as a conceptual framework for teaching mathematics. *SIGCUE Outlook*, v. 4, n. 2, p. 13–17, 1970. Citado na página 20.

FINLEY, K. *Obama Becomes First President to Write a Computer Program*. 2014. Disponível em: <https://www.wired.com/2014/12/obama-becomes-first-president-write-computer-program/>. Acesso em: 27 mai. 2019. Citado na página 30.

FIOREZE, L. A. *et al.* Análise da construção dos conceitos de proporcionalidade com a utilização do software geoplano virtual. *Ciência & Educação*, v. 19, n. 2, p. 267–278, 2013. Citado na página 14.

FORSYTHE, G. E. The role of numerical analysis in an undergraduate program. *The American Mathematical Monthly*, v. 66, n. 8, p. 651–662, 1959. Citado na página 20.

FRAIHA-MARTINS, F.; GONÇALVES, T. V. O. Informática na educação matemática e científica dos anos iniciais de escolaridade: um estudo sobre as pesquisas da área ensino de ciências e matemática. *Revista Ensaio*, v. 14, n. 3, p. 313–331, 2012. Citado na página 14.

FRANÇA, R. S. d.; AMARAL, H. J. C. d. Ensino de computação na educação básica no brasil: um mapeamento sistemático. In: *XXI Workshop sobre Educação em Computação*. [S.l.: s.n.], 2013. p. 426–431. Citado na página 36.

FRANÇA, R. S. d.; SILVA, W. C. d.; AMARAL, H. J. C. d. Despertando o interesse pela ciência da computação: Práticas na educação básica. In: *Proceedings of International Conference on Engineering and Computer Education*. [S.l.: s.n.], 2013. v. 8, p. 282–286. Citado na página 31.

FURBER, S. *Shut down or restart? The way forward for computing in UK schools*. Londres: The Royal Society, 2012. Citado 2 vezes nas páginas 22 e 30.

GERALDES, W. B. *O pensamento computacional no ensino profissional e tecnológico*. Tese (Mestrado) — Universidade Católica de Brasília, Brasília, DF, 2017. Citado na página 36.

GOLDENBERG, M. *A arte de pesquisar: como fazer pesquisa qualitativa em Ciências Sociais*. 13. ed. Rio de Janeiro: Record, 1997. Citado na página 37.

GROVER, S.; PEA, R. Computational thinking in k–12: A review of the state of the field. *Educational researcher*, v. 42, n. 1, p. 38–43, 2013. Citado 2 vezes nas páginas 24 e 25.

GUZDIAL, M. Paving the way for computational thinking. *Communications of the ACM*, v. 51, n. 8, p. 25–27, 2008. Citado 2 vezes nas páginas 20 e 29.

GUZDIAL, M. *Learner-centered design of computing education: Research on computing for everyone*. 1. ed. State College, PA: Morgan & Claypool, 2015. v. 8. (Synthesis Lectures On Human-Centered Informatic, 33). Citado 2 vezes nas páginas 21 e 31.

HAREL, G.; SOWDER, L. Advanced mathematical-thinking at any age: Its nature and its development. *Mathematical thinking and learning*, v. 7, n. 1, p. 27–50, 2005. Citado na página 25.

ICHI-GO-ICHI-E. 2019. Wikipedia. Disponível em: <https://en.wikipedia.org/wiki/Ichi-go_ichi-e>. Acesso em: 15 jun. 2019. Citado na página 12.

IFRAH, G. *Os números – A história de uma grande invenção*. 11. ed. São Paulo: Globo, 2005. Citado na página 82.

INGLATERRA, U. D. f. E. National curriculum in england: computing programmes of study. 2013. Disponível em: <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>. Acesso em: 01 jul. 2018. Citado na página 30.

JACINTO, H.; CARREIRA, S. Diferentes modos de utilização do geogebra na resolução de problemas de matemática para além da sala de aula: evidências de fluência tecno-matemática. *Boletim de Educação Matemática*, v. 31, n. 57, p. 266–288, 2017. Citado na página 14.

JACQUARD LOOM. The Editors of Encyclopædia Britannica, 2017. Encyclopædia Britannica. Disponível em: <https://www.britannica.com/technology/Jacquard-loom>. Acesso em: 12 mai. 2019. Citado na página 82.

JENKINS, H. *et al. Confronting the challenges of participatory culture: Media education for the 21st century*. Cambridge, MA: MIT Press, 2009. (The John D. and Catherine T. MacArthur Foundation Reports on Digital Media and Learning). ISBN 978-0-262-51362-3. Citado 2 vezes nas páginas 14 e 26.

JOHNSON, P. *France to offer programming in elementary school*. 2014. Disponível em: <https://www.itworld.com/article/2696639/france-to-offer-programming-in-elementary-school.html>. Citado na página 23.

JONES, E. The trouble with computational thinking. *ACM*, 2016. Disponível em: <http://csta.acm.org/Curriculum/sub/CurrFiles/JonesCTOnePager.pdf>. Citado na página 25.

KNUTH, D. E. Computer science and its relation to mathematics. *The American Mathematical Monthly*, v. 81, n. 4, p. 323–343, 1974. Citado na página 21.

KNUTH, D. E. Algorithmic thinking and mathematical thinking. *The American Mathematical Monthly*, v. 92, n. 3, p. 170–181, 1985. Citado na página 20.

KOLOGESKI, A. L. *et al.* Desenvolvendo o raciocínio lógico e o pensamento computacional: Experiências no contexto do projeto logicando. *RENOTE*, v. 14, n. 2, 2016. Citado na página 35.

KOSCIANSKI, A.; GLIZT, F. R. d. O. O pensamento computacional nos anos iniciais do ensino fundamental. *RENOTE*, v. 15, n. 2, 2017. Citado na página 35.

LAVINGTON, S. H. *A history of Manchester computers*. 1. ed. Manchester: NCC, 1975. Disponível em: <http://archive.org/details/HistoryOfManchesterComputers>. Acesso em: 16 mai. 2019. Citado na página 85.

LESH, R. Beyond constructivism: identifying mathematical abilities that are most needed for success beyond school in an age of information. *Mathematics Education Research Journal*, v. 12, n. 3, p. 177–195, 2000. Citado na página 18.

LIFELONG KINDERGARTEN GROUP. *Programing Concepts and Skills Supported in Scratch*. MIT Media Lab, 2007. Disponível em: <http://web.media.mit.edu/~mres/scratch/documentation/Scratch-CS-concepts.pdf>. Acesso em: 05 jun. 2019. Citado na página 40.

LINCOLN, Y. S.; GUBA, E. G. *Naturalistic inquiry*. Newbury Park, CA: Sage Publications, 1985. Citado na página 38.

LIUKAS, L. *Hello Ruby: adventures in coding*. 1. ed. Crawfordsville, IN: R. R. Donnelley & Sons Company, 2015. Citado 3 vezes nas páginas 22, 25 e 64.

MALTEMPI, M. V. Construcionismo: pano de fundo para pesquisas em informática aplicada à educação matemática. In: BICUDO, M. A. V.; BORBA, M. C. (Ed.). *Educação Matemática: pesquisa em movimento*. 4. ed. São Paulo: Cortez, 2004. p. 264–282. Citado na página 73.

MALTEMPI, M. V. Novas tecnologias e construção de conhecimento: reflexões e perspectivas. In: *Congresso Ibero-Americano De Educação Matemática*. Porto, Portugal: [s.n.], 2005. Citado na página 39.

MALTEMPI, M. V. Educação matemática e tecnologias digitais: reflexões sobre prática e formação docente. *Acta Scientiae*, v. 10, n. 1, p. 59–67, 2008. Citado na página 38.

MARTINELLI, S. R. *et al. Tutorial de introdução ao Scratch (Parte 1)*. Scratch Brasil, 2019. Disponível em: <http://scratchbrasil.net.br/images/download-materiais/Introdu%C3%A7%C3%A3o%20ao%20Scratch%20(passo%201)%20-%20Scratch%20Brasil.pdf>. Acesso em: 24 abr. 2019. Citado 2 vezes nas páginas 43 e 90.

MENDELSOHN, P.; GREEN, T.; BRNA, P. Programming languages in education: The search for an easy start. In: *Psychology of Programming*. Londres: Academic Press, 1990. v. 12, p. 175–200. Citado na página 21.

MILLER, L. A. Natural language programming: styles, strategies, and contrasts. *IBM Systems Journal*, v. 20, n. 2, p. 184–215, 1981. Citado na página 68.

PAGE, D. *A practical introduction to computer architecture*. Ithaca, NY: Springer-Verlag, 2009. (Texts in Computer Science). Citado na página 85.

PANE, J. F.; RATANAMAHATANA, C. A.; MYERS, B. A. Studying the language and structure in non-programmers' solutions to programming problems. *International Journal of Human-Computer Studies*, v. 54, n. 2, p. 237–264, 2001. Citado na página 68.

PAPERT, S. *Mindstorms: children, computers, and powerful ideas*. 1. ed. Nova Iorque, NY: Basic Books, Inc., 1980. Citado na página 19.

PAPERT, S. Constructionism: A new opportunity for science education. 1986. Citado na página 39.

PAPERT, S. *Logo: computadores e educação*. 3. ed. São Paulo: Brasiliense, 1988. Citado 8 vezes nas páginas 7, 8, 14, 19, 20, 28, 38 e 70.

PAPERT, S. *A máquina das crianças: repensando a escola na era da informática*. Porto Alegre: Artes Médicas, 1994. Citado 2 vezes nas páginas 28 e 38.

PAPERT, S.; SOLOMON, C. Twenty things to do with a computer. In: *Studying the Novice Programmer*. Massachusetts: Lawrence Erlbaum Associates, Inc., 1971. Disponível em: <http://www.stager.org/articles/twentythings.pdf>. Citado na página 20.

PARKER, S. P. *McGraw-Hill Dictionary of Scientific and Technical Terms*. 6. ed. New York, NY: McGraw-Hill Education, 2002. Citado na página 86.

PEA, R. D. Cognitive technologies for mathematics education. *Cognitive Science and Mathematics education*, p. 89–122, 1987. Citado na página 14.

PECKHAM, J. Is computational thinking the fourth "r"? *CSTA Voice*, v. 7, n. 2, p. 1–2, 2011. Citado na página 29.

PERLIS, A. The computer in the university. *Computers and the World of the Future*, p. 180–219, 1962. Citado na página 20.

PETERS, T. The zen of python. In: *Pro Python*. Berkeley, CA: Apress, 2010. p. 301–302. Citado na página 67.

PONTE, J. P. O computador na educação matemática. *Cadernos de Educação Matemática*, v. 2, 1991. Citado na página 38.

PROOF OF CONCEPT. 2019. Wikipedia. Disponível em: <https://en.wikipedia.org/wiki/Proof_of_concept>. Acesso em: 30 mai. 2019. Citado na página 85.

RALSTON, A.; REILLY, E. D. *Encyclopedia of Computer Science*. 3. ed. Boston, MA: International Thomson Computer Press, 1995. Citado 4 vezes nas páginas 58, 59, 85 e 86.

RIESCO ALBIZU, M. *et al.* Informática: materia esencial en la educación obligatoria del siglo xxi. *ReVisión*, v. 7, n. 3, 2014. Citado na página 14.

ROJAS, R.; HASHAGEN, U. *The first computers: history and architectures*. Cambridge, MA: MIT press, 2000. Citado na página 85.

ROSA, M. Tessituras teórico-metodológicas em uma perspectiva investigativa na educação matemática: da construção da concepção de cyberformação com professores de matemática a futuros horizontes. In: *Abordagens teóricas e metodológicas nas pesquisas em educação matemática*. Brasília: SBEM, 2018, (Coleção SBEM, 13). p. 322. Citado na página 72.

SÁPIRAS, F. S. *Relações entre a literacia digital e o ambiente Scratch: um olhar por meio de perspectivas matemáticas com alunos do sétimo e oitavo anos do Ensino Fundamental*. Tese (Mestrado) — ULBRA, Canoas, 2017. Disponível em: <http://ppgecim.ulbra.br/teses/index.php/ppgecim/article/view/275>. Citado na página 26.

SCIENCE MUSEUM. *Lovelace, Turing and the invention of computers*. 2018. Science Museum. Disponível em: <https://www.sciencemuseum.org.uk/objects-and-stories/lovelace-turing-and-invention-computers>. Acesso em: 12 mai. 2019. Citado na página 82.

SHAFFER, D. W.; CLINTON, K. A. Toolforthoughts: Reexamining thinking in the digital age. *Mind, Culture, and Activity*, v. 13, n. 4, p. 283–300, 2006. Citado na página 14.

SHUTE, V. J.; SUN, C.; ASBELL-CLARKE, J. Demystifying computational thinking. *Educational Research Review*, v. 22, p. 142–158, 2017. Citado na página 25.

SIPSER, M. *Introduction to the Theory of Computation*. 3. ed. Boston, MA: Cengage Learning, 2013. Citado 2 vezes nas páginas 86 e 87.

SNEIDER, C. *et al.* Computational thinking in high school science classrooms. *The Science Teacher*, v. 81, n. 5, p. 53, 2014. Citado na página 25.

STRAUSS, V. *A fourth "r" for 21st century literacy*. 2012. Disponível em: <https://www.washingtonpost.com/blogs/answer-sheet/post/a-fourth-r-for-21st-century-literacy/2011/12/29/gIQAxx2BWP_blog.html>. Acesso em: 27 mai. 2019. Citado na página 29.

TECNOLOGIA. *Houaiss Eletrônico*. Rio de Janeiro: Objetiva, 2009. Citado na página 18.

TEDRE, M.; DENNING, P. J. The long quest for computational thinking. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, p. 120–129, 2016. Citado na página 20.

VENTORINI, A. E. *Construção de relações funcionais através do software Scratch*. Tese (Mestrado) — UFSM, Santa Maria, RS, 2015. Citado na página 36.

WADE, N. Early voices: the leap to language. *The New York Times*, v. 15, n. 7, p. D1–D3, 2003. Disponível em: <https://www.nytimes.com/2003/07/15/science/early-voices-the-leap-to-language.html>. Citado na página 17.

WILLIAMS, M. R. *A History of Computing Technology*. 2. ed. Los Alamitos, CA: Wiley-IEEE Computer Society Press, 1997. Citado 4 vezes nas páginas 82, 83, 84 e 85.

WING, J. M. Computational thinking. *Communications of the ACM*, v. 49, n. 3, p. 33–35, 2006. Citado 8 vezes nas páginas 7, 8, 14, 18, 19, 21, 29 e 73.

WING, J. M. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, v. 366, n. 1881, p. 3717–3725, 2008. Citado na página 25.

WING, J. M. *Computational thinking: what and why?* 2010. Disponível em: <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>. Acesso em: 26 mai. 2019. Citado 3 vezes nas páginas 22, 24 e 62.

WING, J. M. Blog, *Computational thinking benefits society*. 2014. Disponível em: <http://socialissues.cs.toronto.edu/2014/01/computational-thinking/>. Acesso em: 25 mai. 2019. Citado 4 vezes nas páginas 22, 26, 66 e 67.

WOLFRAM, C. Blog, *Computational thinking is the code to success*. 2017. Disponível em: <https://www.tes.com/magazine/article/computational-thinking-code-success>. Citado na página 72.

YANDELL, B. *The honors class: Hilbert's problems and their solvers*. Natick, MA: AK Peters/CRC Press, 2002. Citado 2 vezes nas páginas 86 e 87.

YOUTUBE. *Conrad Wolfram: Teaching kids real math with computers*. 2010. Disponível em: <https://www.youtube.com/watch?v=60OVlfAUPJg>. Acesso em: 29 abr. 2019. Citado na página 39.

YOUTUBE. *President Obama asks America to learn computer science*. 2013. Disponível em: <https://www.youtube.com/watch?v=6XvmhE1J9PY>. Acesso em: 27 mai. 2019. Citado na página 30.

APÊNDICE A – Computadores e Computação

Neste espaço são apresentadas algumas notas históricas sobre Computação em geral, máquinas de calcular, a concepção dos primeiros computadores e também apontamentos prelimares acerca de Teoria da Computação.

A.1 O homem precisa contar!

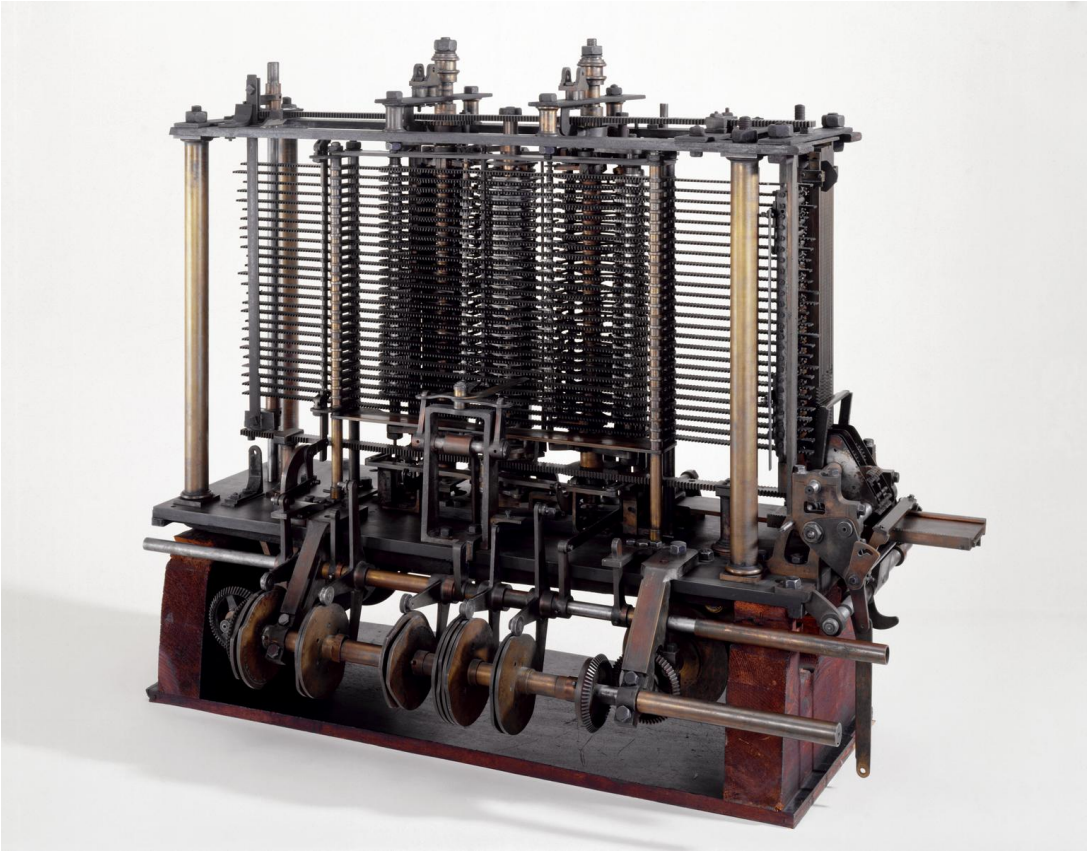
A Ciência da Computação é o conjunto de conhecimentos que sistematiza toda a Computação. Suas origens apontam às primeiras civilizações, como os mesopotâmios, que inventaram os primeiros ábacos, tendo desenvolvimentos substanciais feitos por povos da Grécia, Egito, Índia, China, entre outros (IFRAH, 2005). Muito tempo se passou até que surgissem as primeiras máquinas de computar (calcular) de maneira automatizada. O termo *computador* foi usado durante muito tempo para designar pessoas com habilidades extremamente desenvolvidas de cálculo (usado como sinônimo de "fazer contas") e que ajudavam em trabalhos tais como a contabilidade de impostos. Encontram-se registros da palavra sendo usada neste sentido que datam dos séculos XIV (do latim *computist* — aquele que é perito em reconhecimento calendárico ou cronológico) e XVI (do latim *computator* — aquele que calcula) (COMPUTADOR, 2009; COMPUTER, 2019).

Contribuíram para o surgimento das primeiras calculadoras os trabalhos de John Napier, Wilhelm Schickard e de Blaise Pascal, ao qual é atribuída a construção da primeira máquina calculadora (WILLIAMS, 1997). Durante séculos, diversos instrumentos mecânicos destinados a cálculos e medidas foram desenvolvidos para serem usados com fins de navegação e de estudos astronômicos. Entretanto, tais máquinas realizavam apenas cálculos com as entradas inseridas, não era possível agregar instruções. Somente no início do século XIX houve a primeira tentativa bem-sucedida de construção[1] de uma máquina programável, idealizada por Charles Babbage. Ele se baseou nas ideias de Joseph-Marie Jacquard que construíra anos antes o primeiro tear mecânico programável, que funcionava a partir de instruções em cartões perfurados (JACQUARD LOOM, 2017). O Engenho Analítico, como ficou conhecida a máquina de Babbage, possuía unidades lógico-aritméticas,

[1] Charles Babbage construiu, de fato, o que é chamado de Máquina Diferencial, um mecanismo destinado ao trato de polinômios para cálculos de navegação. Babbage deu-se conta de que uma máquina muito mais geral poderia ser construída, o que foi chamado de Engenho Analítico. Esta máquina, teorizada por Babbage, e posteriormente descrita e estudada minuciosamente por Ada Lovelace, teve sua construção iniciada, mas nunca concretizada, e é considerada o primeiro computador de propósito geral criado (SCIENCE MUSEUM, 2018; WILLIAMS, 1997)

memória integrada e a linguagem utilizada permitia o emprego de estruturas de controle, tais como repetições e condicionais. É considerado o primeiro projeto de um computador de propósito geral, tendo o poder computacional equivalente ao da Máquina de Turing[2].

Figura 26 – Modelo experimental do Engenho Analítico de Charles Babbage (1871).



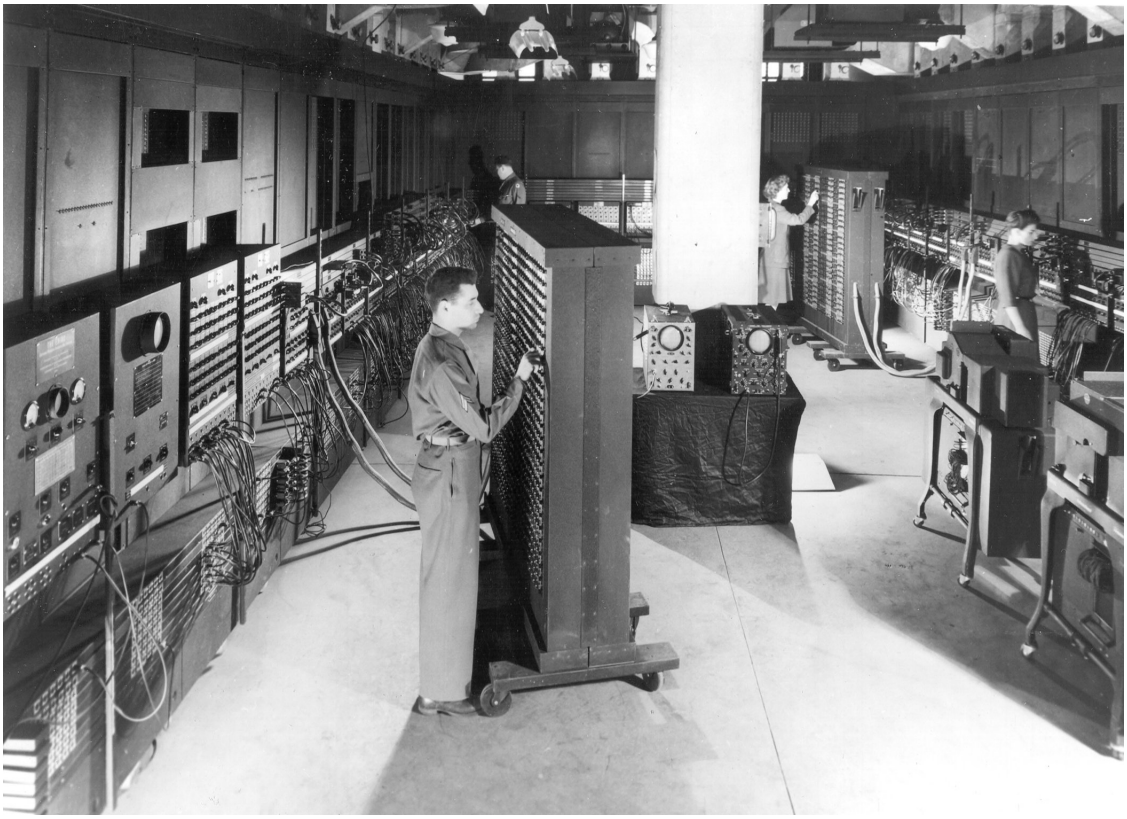
Fonte: *Science Museum*: <https://collection.sciencemuseum.org.uk/objects/co62245>.

No entanto, a Ciência da Computação só passou a se tornar um campo de conhecimento estruturalmente definido a partir da virada do século XIX para o XX, com David Hilbert e, mais fortemente, na década de 1930, com trabalhos de Kurt Gödel, Alonzo Church e Alan Turing (DIVERIO; MENEZES, 1999). A partir do advento do primeiro computador eletrônico com poder comparável ao da Máquina de Turing, o ENIAC (*Electronic Numerical Integrator and Computer*), completado em 1945, nos laboratórios da Escola de Engenharia Elétrica Moore School, na Universidade da Pensilvânia (WILLIAMS, 1997), o uso de máquinas programáveis passou a se disseminar e a se desenvolver mais fervorosamente. O funcionamento do ENIAC era baseado em uma estrutura que pesava

[2] A Máquina de Turing foi um modelo teórico proposto pelo matemático britânico Alan Turing em um artigo publicado em 1936, que descrevia um mecanismo abstrato capaz de realizar operações de memória, estado e de transição; em suma, um computador. Um sistema de regras (real ou teórico) é dito ser Turing-equivalente se, e somente se, ele for capaz de manipular (computar) a mesma classe de funções que a Máquina de Turing, isto é, se ele pode simular ou ser simulado por qualquer Máquina de Turing (DIVERIO; MENEZES, 1999)

30 toneladas, possuía mais de 18.000 válvulas termiônicas de 16 tipos, 1.500 relés, 10.000 capacitores, 70.000 resistores e milhares de indutores, utilizando 140 quilowatts de energia elétrica e ocupando uma área de mais de 30 metros quadrados. O orçamento inicial para a realização do projeto de desenvolvimento do megacomputador foi de US$ 150.000, mas no final o gasto total foi muito maior: US$ 486.804,22 (ENIAC, 2003; WILLIAMS, 1997).

Figura 27 – ENIAC (1945).



Fonte: *Wikipedia*: <https://en.wikipedia.org/wiki/ENIAC>.

Outras máquinas eletromecânicas foram desenvolvidas na mesma época — seja de forma teórica ou de forma prática — com propósitos semelhantes, tanto nos Estados Unidos quanto no Reino Unido e União Soviética. Após juntar-se ao time que trabalhava na construção do ENIAC, o matemático húngaro-americano John von Neumann escreveu, em junho de 1945, o documento "*First Draft of a Report on the EDVAC*" (abreviatura para *Electronic Discrete Variable Automatic Computer*), no qual foi descrito pela primeira vez, em detalhes, o conceito de máquina digital de programa armazenado (*stored-program digital computer*), ou seja, um computador que era capaz de armazenar instruções em memória e não somente executá-las a partir comandos mecânicos pré-estabelecidos mediante cabeamentos entre seus componentes. O fato de von Neumann estar listado como único autor do documento fez com que o conceito de programa armazenado estivesse associado ao seu trabalho. No entanto, ele nunca se creditou como criador das ideias,

apenas as organizou em forma de relatório. Tal situação impeliu diversos membros da equipe que desenvolveu o ENIAC e o EDVAC a ficarem extremamente desgostosos e muitos deles deixaram o laboratório por causa de conflitos internos. Junto com von Neumann trabalharam nos laboratórios Moore School o físico John William Mauchly, o engenheiro elétrico John Adam Presper Eckert — ambos líderes do projeto do ENIAC —, além do matemático Herman Heine Goldstine e outros cientistas, todos com contribuições significativas ao desenvolvimento do EDVAC e do conceito de máquina de programa armazenado (RALSTON; REILLY, 1995). Computadores baseados nestes conceitos (armazenamento de dados e programas na mesma memória) são conhecidos hoje como máquinas de von Neumann.

Mesmo tendo sido proposto em 1945 de forma teórica, o EDVAC só ficou totalmente operacional em 1951 (RALSTON; REILLY, 1995). O título de primeira máquina eletrônica a executar um programa armazenado em memória é tema de controvérsias e vários projetos clamam pela honraria. Entretanto, tende-se a considerar o computador SSEM (*Small-Scale Experimental Machine*), também conhecido como *Manchester Baby*, desenvolvido pela equipe do engenheiro Frederic Calland Williams, do matemático Tom Kilburn e do engenheiro eletrônico Geoff Tootill, na universidade de Manchester, no Reino Unido, como o pioneiro neste quesito (PAGE, 2009; RALSTON; REILLY, 1995; ROJAS; HASHAGEN, 2000; WILLIAMS, 1997). No dia 21 de junho de 1948, esta máquina foi utilizada para resolver alguns problemas simples a fim de checar sua viabilidade (prova de conceito[3] ou *PoC — proof of concept*). Uma rotina foi projetada para se dividir o número $2^{30} - 1$ por 31, a resposta sendo encontrada em aproximadamente um segundo e meio. Essa mesma rotina foi usada para mostrar que 314.159.256 e 271.828.183 são relativamente primos. O maior programa a ser executado nessa máquina consistia em determinar o maior fator de 2^{18} testando-se cada inteiro a partir de $2^{18} - 1$ de forma decrescente realizando as divisões por meio de subtrações sucessivas. Essa computação levou em torno de 52 minutos, na qual foram executadas mais de 3 milhões e 500 mil operações (LAVINGTON, 1975; WILLIAMS, 1997).

A.2 Algoritmos

No âmbito de resolução de problemas, devemos considerar que tanto um problema quanto um dispositivo ou instrumento devem (ou podem) ser utilizados no processo. Tal instrumento pode ser humano, uma máquina ou uma combinação de ambos. Também não há limites ao que diz respeito à natureza dos problemas: eles podem ser de origem

[3] Prova de conceito é um termo utilizado comumente em engenharia e computação para designar um modelo (prático) construído com o intuito de demonstrar a efetividade e a factibilidade de um certo método ou idéia (teórica). Em geral, não possui caráter de protótipo ou versão inicial de um produto, sendo implementado com fins de realizar testes comprobatórios para diminuir erros ou correções futuras nos projetos (PROOF OF CONCEPT, 2019).

matemática ou não, simples ou complexos, com solução única ou não. No entanto, há restrições para que um problema seja considerado bem-posto (PARKER, 2002; RALSTON; REILLY, 1995):

a) as informações referentes aos dados estão claramente especificadas;

b) pode-se decidir quando o problema foi resolvido;

c) o problema não muda durante a sua resolução.

Intuitivamente, um algoritmo pode ser descrito como a "solução de um problema, como uma forma de descrever se uma propriedade é verificada ou não para uma dada classe de entrada" (DIVERIO; MENEZES, 1999, p. 65). Ainda, segundo Sipser (2013, p. 164, tradução nossa), "um algoritmo é uma coleção de instruções simples para realizar alguma tarefa". Apesar da natureza intuitiva da definição de algoritmo, ele deve ser uma descrição precisa e não-ambígua de um método de solução para um problema, considerados ambos o problema e o dispositivo a ser usado para sua resolução (RALSTON; REILLY, 1995). Muitas vezes são chamados de procedimentos ou receitas na vida cotidiana, dada a constância que tais estruturas aparecem no dia a dia. Especialmente na Matemática, desempenham papel fundamental na construção das ideias e há evidências de que desde muito cedo a humanidade se utilizou de algoritmos para inúmeras tarefas: existem métodos milenares descritos para se encontrar números primos, para se construir polígonos regulares, para se encontrar raízes de certas equações, entre outros tantos.

Embora sua história seja antiga, foi somente no início do século XX que se deu uma definição precisa do que seria um algoritmo. Até então, a simples noção intuitiva era suficiente para se resolver problemas matemáticos e não havia a necessidade de formalização rigorosa. Entretanto, houve um acontecimento importante que fez com que a definição de algoritmo necessitasse ser dada de maneira formal e o seu entendimento fosse mais profundo. Esta definição precisa foi crucial para a solução de um problema importante da Matemática.

No ano de 1900, o matemático alemão David Hilbert proclamou uma fala que se tornou um marco na história da Matemática. Tal palestra aconteceu no II Congresso Internacional de Matemáticos (*International Congress of Mathematicians* — ICM), que teve lugar em Paris. No seu discurso, Hilbert enunciou uma lista de 10 problemas[4]indexICM que até então não possuíam solução conhecida e que deveriam servir de guia para o trabalho dos matemáticos durante os próximos anos, frisando "O profundo significado de certos

[4] Na palestra conferida no II Congresso Internacional de Matemáticos em Paris, no dia 8 de agosto de 1900, David Hilbert apresentou 10 problemas, os de número 1, 2, 6, 7, 8, 13, 16, 19, 21 e 22 da sua lista. No entanto, a seleção compilada por ele possuía um total de 23 problemas e foi publicada posteriormente no texto "*Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematike-Congreß zu Paris 1900*", Göttinger Nachrichten, 1900, pp. 253–297, e mais notavelmente em língua inglesa em "*Bulletin of the American Mathematical Society 8*", julho de 1902, pp. 437–479, tradução feita por Dr. May Winston Newson com permissão do autor (BROWDER, 1976; YANDELL, 2002)

problemas para o avanço da ciência matemática em geral [...][5]." (YANDELL, 2002, p. 389, tradução nossa).

O problema de número 10 apresentado por Hilbert propunha a concepção de um algoritmo capaz de testar se um polinômio possui raiz inteira. Ele não utilizou diretamente a expressão algoritmo, ao invés disso falou em "[...] um processo de acordo com o qual pode ser determinado por um número finito de operações [...][6]" (YANDELL, 2002, p. 406, tradução nossa). Implícito nas palavras de Hilbert estava o fato de que ele assumiu que tal algoritmo existia, o que foi provado ser impossível em 1970, pelo matemático russo Yuri Vladimirovich Matiyasevich, apoiado em trabalhos anteriores de Julia Hall Bowman Robinson, Martin David Davis e Hilary Whitehall Putnam. Para se chegar a uma conclusão sobre o problema foi necessário que a noção intuitiva de algoritmo fosse sistematizada de forma rigorosa, o que só aconteceu em 1936 com os trabalhos de Alonzo Church e Alan Turing. Church propôs um formalismo chamado Cálculo-λ (ou λ-Cálculo) para a definição de algoritmo e Turing introduziu o conceito de máquina universal, conhecido como Máquina de Turing. As duas definições foram mostradas serem equivalente entre si, o que deu origem à chamada hoje de Tese de Church-Turing, que afirma que "A capacidade de computação representada pela Máquina de Turing é o limite máximo que pode ser atingido por qualquer dispositivo de computação" (DIVERIO; MENEZES, 1999, p. 128). Isso quer dizer que a definição de algoritmo é correspondente a da Máquina de Turing, tomando-se, portanto, os dois como sinônimos (SIPSER, 2013).

Ademais, um algoritmo é caracterizado pelas seguintes propriedades:

a) a aplicação de um certo algoritmo para um determinado problema ou conjunto de problemas resulta em um número finito de passos;

b) a sequência de ações tem uma única ação inicial;

c) cada ação na sequência tem uma única ação sucessora;

d) a sequência de ações termina ou com a solução do problema ou com a decisão de que o problema é insolúvel (para o conjunto particular de dados iniciais).

A imposição de que um algoritmo deve conter um número finito de instruções pode estabelecer algumas restrições no que concerne à natureza dos dados e informações tratados. Como exemplo, tomemos um número irracional, digamos π: qualquer descrição que se possa dar a este número é somente uma aproximação, visto que sua expansão é infinita e não-periódica. Deste modo, restringe-se o escopo de abrangência a algoritmos

[5] "*The deep significance of certain problems for the advance of mathematical science in general and the important role which they play in the work of the individual investigator are not to be denied.*" (YANDELL, 2002, p. 389)

[6] "*To devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.*" (YANDELL, 2002, p. 406)

naturais, ou, melhor dizendo, a conjuntos contáveis (que possuem bijeção com o conjunto dos números naturais) (DIVERIO; MENEZES, 1999).

APÊNDICE B – Endereços eletrônicos

A título de registro, citamos alguns endereços eletrônicos de associações, pesquisadores e entusiastas ligados à disseminação da computação no âmbito educacional e do pensamento computacional em geral. Neles há uma variedade de informações acerca de assuntos afins, junto com inúmeros materiais disponíveis para serem usados em sala de aula, não somente no âmbito de aulas de Matemática, mas em todas as disciplinas.

The Computer Science Teachers Association (CSTA): <https://www.csteachers.org/>
Google for Education:
<https://edu.google.com/resources/programs/exploring-computational-thinking/>
Center for Computational Thinking (Universidade Carnegie Mellon, Pittsburgh, PA): <http://www.cs.cmu.edu/~CompThink/>
PhET Interactive (Universidade do Colorado, Boulder, CO):
<https://phet.colorado.edu/>
Projeto ScratchMaths (UCL, Londres):
<https://www.ucl.ac.uk/ioe/research/projects/scratchmaths>
Computer Science For Fun (CS4FN): <https://teachinglondoncomputing.org/>
Computing at School: <https://www.computingatschool.org.uk/>
Code.org: <https://code.org/>
Laboratório de Inovação Tecnológica na Educação (LITE, UNIVALI):
<http://lite.acad.univali.br/pt/pensamento-computacional/>
Pensamento Computacional Brasil: <http://www.computacional.com.br/>
Programaê: <http://programae.org.br/>
Computação na Escola (UFSC): <http://www.computacaonaescola.ufsc.br/>
Computação na Escola (SBC):
<http://sbc.org.br/2-uncategorised/1925-computacao-na-escola/>
Olimpíada Brasileira de Informática: <https://olimpiada.ic.unicamp.br/>

APÊNDICE C – Planos de aula

Neste apêndice, estão incluídos, a título de registro, os planos de aula elaborados para a disciplina obrigatória de Estágio em Educação Matemática II que serviram de guia para a realização das atividades na Escola Estadual de Ensino Fundamental Olegário Mariano. Destas atividades surgiram os dados produzidos descritos e analisados no presente trabalho.

As outras foram estruturadas em três momentos distintos: ambientação com o *Scratch* — em forma de estudo dirigido baseado nas atividades contidas em Martinelli *et al.* (2019) —, construção dos projetos e socialização dos trabalhos com a turma e com alunos de outras turmas. O quadro a seguir esquematiza as aulas.

Quadro 3 – Organização das aulas práticas com o *Scratch*.

DATA	**PERÍODOS (50 minutos)**	**ATIVIDADE REALIZADA**
26/04	1	Ambientação e introdução do *Scratch*
30/04	2	
03/05	1	
06/05	2	Construção dos projetos
07/05	2	
10/05	1	
13/05	2	Socialização dos projetos com a turma
14/05	2	
17/05	1	
20/05	2	Mostra dos projetos para turmas do sexto ano

Fonte: própria.

PLANO DE AULA

AULA 11

Professor responsável: Vinícius Fernandes Moretti

Disciplina: Matemática

Turma: 9A e 9B

Data: 26-30 de abril de 2019

Duração: um período de 50 minutos

Professora Fernanda Wanderer

Professora Naira Girotto

Resumo da(s) atividade(s) a ser(em) desenvolvida(s)

Atividade inicial de apresentação e ambientação com o *Scratch*.

Objetivo geral da(s) atividade(s)

Introduzir o software *Scratch* para os alunos, bem como algumas de suas funcionalidades e possibilidades.

Conceitos de matemática presentes na atividade

Raciocínio lógico e programação.

Público alvo

Alunos do nono ano do Ensino Fundamental.

Justificativa/Relevância

As tecnologias digitais fazem parte do mundo atual e são indissociáveis de praticamente todos os âmbitos da vida contemporânea, e a educação não é exceção. Apesar de ainda estarmos em estágios incipientes em níveis educacionais, já há diversas pesquisas e iniciativas em prol do seu uso em sala de aula. Em especial ao conhecimento de informática e ao pensamento computacional , podemos identificar habilidades relacionadas ao seu desenvolvimento: formular e resolver problemas, organizar e analisar dados, representar dados através de modelos, automatizar soluções através de pensamento algorítmico, identificar soluções mais eficientes (ou ótimas) e generalização (GROVER; PEA, 2013).

Descrição das atividades

O *Scratch* é uma linguagem de programação desenvolvida em 2007 pelo MIT, tendo como foco principal o ensino de programação para crianças e jovens. Por apresentar uma interface visual e orientada por meio de blocos pré-estabelecidos, passíveis de serem encaixados uns com os outros de acordo com a lógica desejada, e onde os resultados podem ser imediatamente testados e vistos na tela, ela se

constitui numa maneira de se introduzir conceitos relativos ao pensamento computacional.

Esta aula é pensada a partir de uma apresentação do software e da linguagem do *Scratch* aos alunos. Será realizada uma ambientação, em estilo de estudo dirigido, expondo características e comandos básicos da linguagem. Para tal, seguir-se-á o manual organizado por Martinelli et al. ([s.d.]), que introduz conceitos sobre o funcionamento do software através de exemplos ilustrativos.

Procedimentos e materiais

Aula expositiva dialogada no laboratório de informática.

Referências

GROVER, S.; PEA, R. **Computational thinking in K–12:** a review of the state of the field. Educational Researcher, v. 42, n. 1, pp. 38–43, 2013.

MARTINELLI, S. R et al. **Tutorial:** Introdução ao Scratch (passo 1). [s.d.] Disponível em: <http://scratchbrasil.net.br/images/download-materiais/Introdu%C3%A7%C3%A3o%20ao%20Scratch%20(passo%201)%20-%20Scratch%20Brasil.pdf>. Último acesso: 25 abr. 2019.

PLANO DE AULA

AULA 17-18-19-20-21

Professor responsável: Vinícius Fernandes Moretti

Disciplina: Matemática

Turma: 9A

Data: 06-07-10 de maio de 2019

Duração: cinco períodos de 50 minutos

Professora Fernanda Wanderer

Professora Naira Girotto

Resumo da(s) atividade(s) a ser(em) desenvolvida(s)

Construção de projetos utilizando o *Scratch*.

Objetivo geral da(s) atividade(s)

Desenvolvimento de projetos individuais ou em duplas utilizando o *Scratch*.

Conceitos de matemática presentes na atividade

Primordialmente, raciocínio lógico e programação. No entanto, vários outros conceitos matemáticos podem (e devem surgir) no decorrer das atividades.

Público alvo

Alunos do nono ano do Ensino Fundamental.

Justificativa/Relevância

As tecnologias digitais fazem parte do mundo atual e são indissociáveis de praticamente todos os âmbitos da vida contemporânea, e a educação não é exceção. Apesar de ainda estarmos em estágios incipientes em níveis educacionais, já há diversas pesquisas e iniciativas em prol do seu uso em sala de aula. Em especial ao conhecimento de informática e ao pensamento computacional, podemos identificar habilidades relacionadas ao seu desenvolvimento: formular e resolver problemas, organizar e analisar dados, representar dados através de modelos, automatizar soluções através de pensamento algorítmico, identificar soluções mais eficientes (ou ótimas) e generalização (GROVER; PEA, 2013).

Segundo Piaget, as pessoas produzem (constroem) conhecimento a partir da sua ação sobre os objetos — coisas, pessoas, idéias —, sendo esta construção feita na medida em que há interação, em um modelo de ação-reação entre pessoas e objetos de conhecimento. Baseando-se nessa idéia, Papert (1985) propôs a sua teoria do Construtivismo, na qual sugere que que o aprendizado ocorre sobretudo quando o aprendiz está engajado na construção de um *produto* que tenha significado pessoal, cujo resultado possa ser compartilhado com outrem.

Descrição das atividades

Após as últimas aulas, onde foram introduzidos conceitos básicos de programação e alguns comandos específicos do Scratch, estas aulas são reservadas para a construção de um projeto de interesse próprio dos alunos.

O trabalho consiste na elaboração de um jogo, vídeo, animação ou apresentação que se utilize do *Scratch* para ser concebido. A natureza, a complexidade e a finalidade do projeto ficará a cargo dos estudantes, estando eles livres para pesquisarem na internet, em manuais, em tutoriais, ou consultarem o professor ou os colegas para coletarem idéias e soluções para o que desejam realizar. O trabalho deverá ser feito de forma individual ou em duplas e contará como critério de avaliação para composição de nota no trimestre.

O tempo destinado para a realização deste trabalho inicialmente foi estipulado em cinco aulas (períodos de 50 minutos), mas, no entanto, pode ser que ele varie de acordo com o rendimento e o engajamento por parte dos estudantes. Também está planejado que seja feita uma apresentação dos trabalhos para o grande grupo, a qual também comporá parte da nota da atividade.

Procedimentos e materiais

Aula expositiva dialogada no laboratório de informática.

Referências

GROVER, S.; PEA, R. **Computational thinking in K–12:** a review of the state of the field. Educational Researcher, v. 42, n. 1, pp. 38–43, 2013.

MALTEMPI, M. V. **Novas tecnologias e construção de conhecimento:** reflexões e perspectivas. In: Congresso Ibero-Americano De Educação Matemática. 2005.

MARTINELLI, S. R et al. **Tutorial:** Introdução ao Scratch (passo 1). [s.d.] Disponível em: <http://scratchbrasil.net.br/images/download-materiais/Introdu%C3%A7%C3%A3o%20ao%20Scratch%20(passo%201)%20-%20Scratch%20Brasil.pdf>. Último acesso: 25 abr. 2019.

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

PLANO DE AULA

AULA 22-23-24-25-26

Professor responsável: Vinícius Fernandes Moretti

Disciplina: Matemática

Turma: 9A

Data: 13-14-17 de maio de 2019

Duração: cinco períodos de 50 minutos

Professora Fernanda Wanderer

Professora Naira Girotto

Resumo da(s) atividade(s) a ser(em) desenvolvida(s)

Apresentação dos projetos construídos com o *Scratch*.

Objetivo geral da(s) atividade(s)

Socialização dos projetos individuais ou em duplas construídos com o *Scratch* para avaliação do professor e dos seus pares.

Conceitos de matemática presentes na atividade

Raciocínio lógico e programação estruturada junto com outros vários conceitos matemáticos (sobretudo geométricos) que podem emergir no decorrer das apresentações e discussões.

Público alvo

Alunos do nono ano do Ensino Fundamental.

Justificativa/Relevância

As tecnologias digitais fazem parte do mundo atual e são indissociáveis de praticamente todos os âmbitos da vida contemporânea, e a educação não é exceção. Apesar de ainda estarmos em estágios incipientes em níveis educacionais, já há diversas pesquisas e iniciativas em prol do seu uso em sala de aula. Em especial ao conhecimento de informática e ao pensamento computacional, podemos identificar habilidades relacionadas ao seu desenvolvimento: formular e resolver problemas, organizar e analisar dados, representar dados através de modelos, automatizar soluções através de pensamento algorítmico, identificar soluções mais eficientes (ou ótimas) e generalização (GROVER; PEA, 2013).

Segundo Piaget, as pessoas produzem (constroem) conhecimento a partir da sua ação sobre os objetos — coisas, pessoas, idéias —, sendo esta construção feita na medida em que há interação, em um modelo de ação-reação entre pessoas e objetos de conhecimento. Baseando-se nessa idéia, Papert (1985) propôs a sua teoria do Construtivismo, na qual sugere que que o aprendizado ocorre sobretudo quando o aprendiz está engajado na construção de um *produto* que tenha significado pessoal, cujo

resultado possa ser compartilhado com outrem.

Descrição das atividades

Dando seguimento às atividades no laboratório de informática da escola, esta semana está reservada para a apresentação e socialização dos trabalhos realizados pelos alunos com o uso do *Scratch*.

Na última semana houve espaço e tempo para que os estudantes construíssem algum projeto de motivação própria: jogo, vídeo, apresentação, animação ou alguma história que se utilizasse do *Scratch* para ser desenvolvido. Será feita, portanto, uma mostra coletiva para que os alunos apresentem suas construções aos professores e aos colegas, a fim de serem avaliados e arguidos sobre o processo de desenvolvimento e seus resultados esperados e obtidos.

Junto ao projeto no computador foi solicitado que os alunos entregassem um relatório que deveria conter obrigatoriamente os seguintes itens:

1. Título do trabalho;
2. Descrição do funcionamento e da construção do projeto;
3. O que de matemática foi enxergado no processo de construção;
4. O que foi necessário pesquisar externamente caso se aplique (na internet ou com colegas/professor).

A apresentação deverá contemplar basicamente tais aspectos do processo e a arguição também se dará neste sentido. Também é esperada (e incentivada sob peso de composição de nota), críticas dos estudantes com relação ao trabalho dos colegas.

A avaliação do aluno no trabalho, que terá peso de prova, dar-se-á a partir dos seguintes critérios:

a) Engajamento nas atividades de laboratório: atividades de familiarização com o ambiente do *Scratch* e também durante a construção dos seus projetos próprios;
b) Entrega e apresentação do projeto;
c) Relatório descrevendo aspectos relevantes do projeto;
d) Participação na apresentação dos demais colegas, com críticas, sugestões e comentários.

Procedimentos e materiais

Aula expositiva dialogada no laboratório de informática.

Referências

GROVER, S.; PEA, R. **Computational thinking in K–12:** a review of the state of the field. Educational Researcher, v. 42, n. 1, pp. 38–43, 2013.

MALTEMPI, M. V. **Novas tecnologias e construção de conhecimento:** reflexões e perspectivas. In: Congresso Ibero-Americano De Educação Matemática. 2005.

MARTINELLI, S. R et al. **Tutorial:** Introdução ao Scratch (passo 1). [s.d.] Disponível em: <http://scratchbrasil.net.br/images/download-materiais/Introdu%C3%A7%C3%A3o%20ao%20Scratch%20(passo%201)%20-%20Scratch%20Brasil.pdf>. Último acesso: 25 abr. 2019.

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

PLANO DE AULA

AULA 27-28

Professor responsável: Vinícius Fernandes Moretti

Disciplina: Matemática

Turma: 9A

Data: 20 de maio de 2019

Duração: dois períodos de 50 minutos

Professora Fernanda Wanderer

Professora Naira Girotto

Resumo da(s) atividade(s) a ser(em) desenvolvida(s)

Mostra coletiva dos projetos construídos com o *Scratch*.

Objetivo geral da(s) atividade(s)

Socialização dos projetos construídos com o *Scratch* para outras turmas da escola, a fim de valorização das produções feitas e de incentivo ao trabalho com a plataforma pelos demais professores, mostrando possibilidades diferenciadas de se trabalhar com o computador.

Conceitos de matemática presentes na atividade

Raciocínio lógico e programação estruturada junto com outros vários conceitos matemáticos (sobretudo geométricos) que podem emergir no decorrer das apresentações e discussões.

Público alvo

Alunos do nono ano do Ensino Fundamental.

Justificativa/Relevância

As tecnologias digitais fazem parte do mundo atual e são indissociáveis de praticamente todos os âmbitos da vida contemporânea, e a educação não é exceção. Apesar de ainda estarmos em estágios incipientes em níveis educacionais, já há diversas pesquisas e iniciativas em prol do seu uso em sala de aula. Em especial ao conhecimento de informática e ao pensamento computacional, podemos identificar habilidades relacionadas ao seu desenvolvimento: formular e resolver problemas, organizar e analisar dados, representar dados através de modelos, automatizar soluções através de pensamento algorítmico, identificar soluções mais eficientes (ou ótimas) e generalização (GROVER; PEA, 2013).

Segundo Piaget, as pessoas produzem (constroem) conhecimento a partir da sua ação sobre os objetos — coisas, pessoas, idéias —, sendo esta construção feita na medida em que há interação, em um modelo de ação-reação entre pessoas e objetos de conhecimento. Baseando-se nessa idéia, Papert (1985) propôs a sua teoria do Construtivismo, na qual sugere que que o aprendizado ocorre sobretudo quando o aprendiz está engajado na construção de um *produto* que tenha significado pessoal, cujo

resultado possa ser compartilhado com outrem.

Descrição das atividades

Tendo em vista o bom rendimento dos estudantes nas atividades com o *Scratch* e o engajamento geral da turma, foi pensado que seria interessante mostrar a outros colegas e à comunidade do colégio em geral os resultados obtidos com as produções. Apareceram construções bastante interessantes e surpreendentes, mostrando o potencial que a plataforma tem e também o desenvolvimento das idéias e do pensamento computacional da turma ao criarem suas animações, histórias e jogos.

Na última aula (sexta-feira, dia 17 de maio) foi conversado com os estudantes sobre a possibilidade de se apresentar os trabalhos para outros colegas e houve concordância com a idéia. Sendo assim, estes períodos serão designados para uma pequena mostra de trabalhos realizados com o *Scratch*, a ser feita na sala de vídeo da escola. Também já foi conversado com alguns professores sob a viabilidade de levarem suas turmas para assistirem às apresentações, ao passo que diversos deles concordaram e se interessaram. Então, serão levadas algumas turmas, uma por vez, à sala de vídeo e os alunos apresentarão suas produções, mas sem muita discussão sobre o processo de criação como foi feito nas últimas aulas. Espera-se alcançar o maior número de turmas possíveis neste tempo, a fim de mostrar para outros jovens e professores algumas potencialidades do *Scratch* como instrumento e recurso educacional e o pensamento computacional como uma metodologia válida e poderosa tanto de construção de conhecimento quanto de engajamento por parte dos alunos não somente em aulas de matemática, mas em todos os campos do conhecimento.

Procedimentos e materiais

Aula expositiva dialogada na sala de vídeo.

Referências

GROVER, S.; PEA, R. **Computational thinking in K–12:** a review of the state of the field. Educational Researcher, v. 42, n. 1, pp. 38–43, 2013.

MALTEMPI, M. V. **Novas tecnologias e construção de conhecimento:** reflexões e perspectivas. In: Congresso Ibero-Americano De Educação Matemática. 2005.

MARTINELLI, S. R et al. **Tutorial:** Introdução ao Scratch (passo 1). [s.d.] Disponível em: <http://scratchbrasil.net.br/images/download-materiais/Introdu%C3%A7%C3%A3o%20ao%20Scratch%20(passo%201)%20-%20Scratch%20Brasil.pdf>. Último acesso: 25 abr. 2019.

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

APÊNDICE D – Termos de consentimento

Neste apêndice, estão os termos referentes às concessões de uso dos trabalhos realizados durante as práticas em laboratório para serem usados como fontes de dados para a presente pesquisa. São inclusos o ***Termo de Consentimento Livre e Esclarecido***, assinado pelos sujeitos de pesquisa, e o ***Termo de Consentimento Informado***, assinado pelos responsáveis dos alunos. Ambos estão de posse dos pesquisadores para serem conferidos a qualquer momento.

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Prezado(a) participante:

Sou estudante do curso de graduação de Licenciatura em Matemática da Universidade Federal do Rio Grande do Sul. Estou realizando a prática do Trabalho de Conclusão de Curso na Escola Estadual de Ensino Fundamental Olegário Mariano, sob supervisão do professor Doutor Rodrigo Dalla Vecchia.

Sua participação consiste em atuar em atividades que envolvem o software *Scratch*. A sua cooperação neste estudo é voluntária e se você decidir não participar ou quiser desistir de continuar em qualquer momento, tem absoluta liberdade de não o fazer.

Na publicação dos resultados desta pesquisa, sua identidade, bem como seus dados, será mantida em total sigilo. Serão omitidas todas as informações que permitem identificá-lo individualmente. Indiretamente, você estará contribuindo para a compreensão do fenômeno estudado e para produção de conhecimento científico.

Quaisquer dúvidas relativas à sua participação poderão ser esclarecidas pelo pesquisador e ou pelo supervisor.

Porto Alegre, _____ de _____ de 2019.

Declaro estar ciente dos objetivos da pesquisa e concordo em participar do referido estudo.

_____

Nome e assinatura do(a) participante

_____

Nome e assinatura do estudante pesquisador

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA

TERMO DE CONSENTIMENTO INFORMADO

Eu, _____, RG _____, responsável pelo(a) aluno(a) _____, da turma _____, declaro, por meio deste termo, que concordei em que o(a) aluno(a) participe da pesquisa intitulada ***O PENSAMENTO COMPUTACIONAL NO ENSINO BÁSICO: POTENCIALIDADES DE DESENVOLVIMENTO COM O USO DO SCRATCH***, desenvolvida pelo pesquisador ***VINÍCIUS FERNANDES MORETTI***. Fui informado(a), ainda, de que a pesquisa é coordenada/orientada pelo Professor Doutor ***RODRIGO DALLA VECCHIA***, a quem poderei contatar a qualquer momento que julgar necessário.

Tenho ciência de que a participação do(a) aluno(a) não envolve nenhuma forma de incentivo financeiro, sendo a única finalidade desta participação a contribuição para o sucesso da pesquisa. Fui informado(a) dos objetivos estritamente acadêmicos do estudo, que, em linhas gerais, são:

- ***REALIZAÇÃO DE ATIVIDADES EM LABORATÓRIO DE INFORMÁTICA QUE SE UTILIZEM DO AMBIENTE DE PROGRAMAÇÃO SCRATCH POR MEIO DE UM VIÉS CONSTRUCIONISTA;***
- ***INVESTIGAR PROCESSOS DE SOLUÇÃO E CONSTRUÇÃO DE SOLUÇÕES PARA PROBLEMAS QUE ENVOLVAM PENSAMENTO ESTRUTURADO;***
- ***ANALISAR O DESENVOLVIMENTO E/OU MANIFESTAÇÃO DE HABILIDADES RELACIONADAS AO PENSAMENTO COMPUTACIONAL.***

Fui também esclarecido(a) de que os usos das informações oferecidas pelo(a) aluno(a) será apenas em situações acadêmicas (artigos científicos, palestras, seminários etc.), identificadas apenas por pseudônimo.

A colaboração do(a) aluno(a) se fará por meio de entrevista/questionário escrito etc., bem como da participação em oficina/aula/encontro/palestra, em que ele(ela) será observado(a) e sua produção analisada. No caso de fotos ou filmagens, obtidas durante a participação do(a) aluno(a), autorizo que sejam utilizadas em atividades acadêmicas, tais como artigos científicos, palestras, seminários, etc., sem identificação. Esses dados ficarão armazenados por pelo menos 5 anos após o término da investigação.

Cabe ressaltar que a participação nesta pesquisa não infringe as normas legais e éticas. No entanto, poderá ocasionar algum constrangimento dos entrevistados ao precisarem responder a algumas perguntas sobre o desenvolvimento de seu trabalho na escola. A fim de amenizar este desconforto será mantido o anonimato das entrevistas. Além disso, asseguramos que o estudante poderá deixar de participar da investigação a qualquer momento, caso não se sinta confortável com alguma situação

Como benefícios, esperamos com este estudo, produzir informações importantes sobre ***O PENSAMENTO COMPUTACIONAL NO ÂMBITO DA EDUCAÇÃO BÁSICA***, a fim de que o conhecimento construído possa trazer contribuições relevantes para a área educacional.

Qualquer dúvida quanto a procedimentos éticos também pode ser sanada com o Comitê de Ética em Pesquisa (CEP) da Universidade Federal do Rio Grande do Sul (UFRGS), situado na Av. Paulo Gama, 110, sala 317, Prédio Anexo 1 da Reitoria — Campus Centro, Porto Alegre/RS — CEP 90040-060, e que tem como fone (51) 3308-3738 e e-mail ***etica@propesq.ufrgs.br***.

Fui ainda informado(a) de que o(a) aluno(a) pode se retirar dessa pesquisa a qualquer momento, sem sofrer quaisquer sanções ou constrangimentos.

Porto Alegre, _____ de _____ de _____.

Assinatura do Responsável:

Assinatura do pesquisador:

Índice