



Codificando saberes: oficinas de Scratch como estratégia para fomentar o pensamento computacional no ensino de ciências

Magdalena Dal Ponte Ceratti (PPGECI/UFRGS)¹

Márcia Finimundi Nóbile (PPGECI/UFRGS)²

Resumo

O presente artigo descreve um relato de experiência de aplicação de oficinas de Scratch nas aulas de Ciências, proporcionada a uma turma de 15 estudantes do 6º ano do Ensino Fundamental de uma escola pública do município de Farroupilha/RS, visando estimular o Pensamento Computacional (PC). Utilizou-se uma metodologia qualitativa, observações, criação de repositório de projetos e aplicação de questionário aos participantes. As atividades práticas foram projetadas para proporcionar uma abordagem dinâmica e interativa, promovendo a conexão entre conceitos científicos e habilidades de programação. Observou-se um impacto positivo, não apenas no desenvolvimento do Pensamento Computacional, mas também na compreensão mais profunda dos temas científicos abordados. Ao fim do estudo, conclui-se que a utilização de Scratch enriqueceu a experiência do processo de ensino e aprendizagem em Ciências, revelando-se uma ferramenta pedagógica eficaz ao mesmo tempo que promoveu o PC através da programação.

Palavras-chave: Pensamento Computacional (PC); Scratch; Ensino de Ciências.

Abstract

This article describes an experience report on the application of Scratch workshops in Science classes, provided to a class of 15 students in the 6th year of Elementary School at a public school in the city of Farroupilha/RS, aiming to stimulate Computational Thinking (PC). A qualitative methodology was used, observations, creation of a project repository and application of a questionnaire to participants. The practical activities were designed to provide a dynamic and interactive approach, promoting the connection between scientific concepts and programming skills. A positive impact was observed, not only on the development of Computational Thinking, but also on a deeper understanding of the scientific topics covered. At the end of the study, it was concluded that the use of Scratch enriched the experience of the teaching and learning process in Science, proving to be an effective tool at the same time that it promoted PC through programming.

Keywords: Computational Thinking (CT); Scratch; Science teaching.

¹ Contato: maguidalponte@gmail.com

² Contato: marciafinimundi@gmail.com



1. Introdução

O avanço da era digital tem redefinido o papel da escola diante da sociedade. Nesse contexto, a programação emerge como uma ferramenta poderosa para promover o desenvolvimento do Pensamento Computacional (PC). Nunes (2011) define o Pensamento Computacional como “habilidades comumente utilizadas na criação de programas computacionais, como uma metodologia para resolver problemas específicos nas mais diversas áreas”.

Lisbôa e Coutinho (2012, p. 5), apontam que

“o desafio imposto à escola por esta nova sociedade é imenso; o que se lhe pede é que seja capaz de desenvolver nos estudantes competências para participar e interagir num mundo global, altamente competitivo que valoriza o ser flexível, criativo, capaz de encontrar soluções inovadoras para os problemas de amanhã, ou seja, a capacidade de compreendermos que a aprendizagem não é um processo estático, mas algo que deve acontecer ao longo de toda a vida.”

Papert reconheceu que “[...] os computadores podem ser usados não apenas para fornecer informações e instruções, mas também para capacitar as crianças a experimentar, explorar e se expressar” (MIT NEWS, 2016).

No contexto educacional, o Scratch³, plataforma de programação visual desenvolvida pelo MIT, tem se destacado como uma ferramenta poderosa para o desenvolvimento do Pensamento Computacional. A programação, muitas vezes considerada uma habilidade restrita a especialistas em informática, emerge no Scratch como uma linguagem acessível e lúdica, capaz de proporcionar uma entrada amigável ao mundo da lógica computacional.

Para Maloney *et al.* (2004, p. 43), “o Scratch proporciona uma maneira fácil e acessível para que estudantes possam começar a explorar as possibilidades criativas da programação”.

Apesar da crescente importância atribuída ao Pensamento Computacional, muitos educadores ainda associam predominantemente o desenvolvimento desta habilidade ao âmbito das disciplinas exatas, negligenciando sua integração a outras áreas do conhecimento, como a de Ciências. Tal percepção limitada representa um desafio para abordagens mais abrangentes e interdisciplinares.

Diante deste cenário, este artigo descreve a execução de oficinas de programação utilizando o Scratch, aplicadas em uma turma de estudantes do 6º ano de uma escola pública municipal de Farroupilha, no estado do Rio Grande do Sul. A proposta visou

³ <https://scratch.mit.edu/>

proporcionar uma aprendizagem prática e envolvente durante as aulas de Ciências, promovendo de maneira simultânea o desenvolvimento do Pensamento Computacional. Por meio dessa abordagem pedagógica, buscou-se explorar como a utilização do Scratch pode enriquecer a experiência educacional, capacitando os estudantes para os desafios contemporâneos, potencializando o aprendizado.

2. Metodologia

A metodologia de pesquisa utilizada neste estudo baseou-se em uma abordagem qualitativa. Para Gil (1999), a pesquisa qualitativa é subjetiva ao objeto de estudo, ergue-se sobre a dinâmica e abordagem do problema pesquisado e visa descrever e decodificar de forma interpretativa os componentes de um sistema complexo.

O estudo visou aprofundar a compreensão dos impactos da implementação de oficinas de programação Scratch no ensino de Ciências, a fim de proporcionar o desenvolvimento de habilidades relacionadas ao Pensamento Computacional aos estudantes do 6º ano do Ensino Fundamental de uma escola pública municipal de Farroupilha, RS.

Durante as oficinas, foram coletados dados através de observações e registros da execução das atividades realizadas pelos estudantes por meio da criação de um repositório (estúdio) que a ferramenta disponibiliza para contas de educadores. As observações detalhadas durante as interações dos estudantes com o Scratch, as dinâmicas da sala de aula e a verificação dos códigos criados possibilitaram uma análise dos elementos envolvidos. No término da oficina, foi aplicado aos participantes um questionário misto, via *Google Forms*, contendo perguntas fechadas e abertas, projetado para coletar informações e avaliar a percepção dos participantes em relação à experiência.

A pesquisa seguiu princípios éticos, garantindo a confidencialidade e anonimato dos participantes, tendo sido obtido consentimento dos responsáveis dos estudantes previamente ao início da intervenção.

3. Pensamento Computacional: desenvolvendo competências e habilidades

O Pensamento Computacional é uma competência que transcende o universo da programação, abrangendo a capacidade de abordar problemas complexos de maneira lógica e sistemática.

O marco inicial do PC remonta ao trabalho visionário de Alan Turing na década de 1930. Sua concepção da Máquina de Turing não apenas estabeleceu as bases da Ciência da Computação, mas também lançou as bases para uma abordagem sistemática e lógica na resolução de problemas, destacando que, assim como suas máquinas, nossos pensamentos podem ser algorítmicamente estruturados para enfrentar os desafios (Hodges, 1983).

Com a criação dos primeiros computadores nas décadas de 1950, o desenvolvimento de linguagens de programação, como Fortran e COBOL, refletiu a necessidade crescente de expressar algoritmos de maneira compreensível (Ceruzzi, 2003).

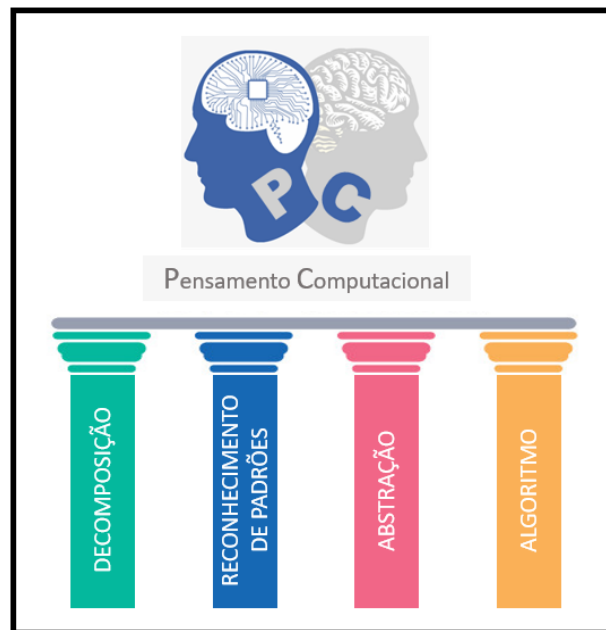
Seymour Papert, na década de 1980, desempenhou um papel crucial ao integrar o Pensamento Computacional na educação. Seu trabalho no desenvolvimento da linguagem de programação Logo e na teoria do Construcionismo destacou como a programação poderia ser uma ferramenta poderosa para promover o aprendizado. A Tartaruga Logo permitia que os alunos desenhassem padrões na tela, combinando a criatividade com a lógica de programação (Papert, 1980).

No início do século XXI, Jeanette Wing popularizou o termo Pensamento Computacional, definindo-o como uma habilidade fundamental para todos. Esse pensamento oferece às pessoas a possibilidade de desenvolverem o pensamento abstrato, pensamento algorítmico, pensamento lógico e pensamento dimensionável (Wing, 2006).

Blikstein (2008) destaca que o Pensamento Computacional amplia o poder cognitivo e operacional humano de modo a favorecer a produção de conhecimento e a criatividade.

Desde que Wing utilizou o termo Pensamento Computacional, buscou-se sistematizar e organizar as habilidades desse modo de pensar para unificar as diversas definições encontradas na literatura. Code.Org (2016), Liukas (2015) e BBC Learning (2015), com embasamento em outros autores e guias, como o guia da *Computing at School* (Sentance; Csizmadia, 2017), delimitaram os pilares do Pensamento Computacional em Decomposição, Reconhecimento de Padrões, Abstração e Algoritmos.

Figura 1: Pilares do Pensamento Computacional



Fonte: <https://comunidadesdesplugada.com.br/os-pilares-do-pensamento-computacional/>

A decomposição é a capacidade de particionar problemas complexos em tarefas menores. Esse pilar não apenas facilita a abordagem sistemática de desafios, mas também promove a eficiência na resolução de problemas multidimensionais, uma competência valiosa em um mundo caracterizado pela complexidade e interconexão. A decomposição pode ser entendida como a primeira atitude a ser tomada ao aplicar o Pensamento Computacional. Para Wing (2006), a decomposição é usada “ao atacar uma tarefa complexa grande. É a separação de inquietações. É escolher uma representação apropriada para um problema ou modelar os aspectos relevantes de um problema para torná-lo tratável.” Assim, o primeiro passo é analisar o problema e dividi-lo em partes menores e mais fáceis de gerenciar (Brackmann, 2017).

A identificação de padrões é a habilidade ligada à interpretação de dados. No século XXI, em que a informação é abundante, a capacidade de reconhecer padrões é crucial para a análise crítica, a previsão de tendências e a solução de problemas em diversas áreas. O reconhecimento de padrões é a busca por características similares, por tendências e regularidades entre os problemas que, depois de serem identificadas, tornarão mais fácil a resolução de outros futuros problemas, por já se ter a base de como solucioná-los, bastando reutilizar ou adaptar a solução a nova situação (Tabesh, 2017). Para Brackmann (2017), esta identificação de problemas parecidos que já foram solucionados anteriormente gera uma economia de tempo e esforços. Wing (2010) cita que o reconhecimento de padrões é

usado para deixar um objeto representar muitos, capturando propriedades essenciais comuns a um conjunto de objetos enquanto oculta distinções irrelevantes entre eles.

A abstração no Pensamento Computacional permite a simplificação de conceitos complexos, tornando-os mais acessíveis. Essa capacidade não apenas facilita a comunicação eficaz, mas também promove uma compreensão mais profunda e generalizável dos princípios subjacentes. No mundo moderno, no qual a informação é vasta e variada, a habilidade de abstração é inestimável. Para Wing (2010), a abstração pode ser entendida como o processo de pensamento mais importante e de mais alto nível, pois oferece o poder de escalar e lidar com a complexidade. Para Brackmann (2017), por meio dela tem-se o foco apenas no que é importante, enquanto informações irrelevantes são ignoradas, tornando mais clara a visualização do problema ao eliminar qualquer aspecto que não seja necessário para a sua resolução.

A criação de algoritmos, a habilidade de desenvolver sequências lógicas de passos para resolver problemas, é um pilar central do Pensamento Computacional. Essa competência não apenas prepara os indivíduos para a programação, mas também fortalece suas habilidades de resolução de problemas e pensamento lógico, fundamentais em um ambiente dinâmico e tecnologicamente orientado. Com o pensamento algorítmico, passos ou regras simples podem ser criados para resolver cada um dos subproblemas encontrados (Brackmann, 2017). Para Wing (2010), um algoritmo é uma abstração de um processo que recebe entradas, executa uma sequência de etapas e produz saídas para satisfazer um objetivo desejado.

4. Scratch: desenvolvendo o pensamento computacional de forma lúdica e criativa através da programação

A teoria do Construcionismo de Papert (2008), que fundamentou seu trabalho com o Logo, postulava que os estudantes aprendiam melhor quando envolvidos na construção ativa de artefatos significativos. Ele defendia que a programação não deveria ser apenas uma disciplina isolada, mas uma ferramenta integrada em diversas áreas do currículo, incentivando a interdisciplinaridade e o desenvolvimento integral dos estudantes.

Uma linguagem de programação, assemelha-se a uma língua natural, humana, na medida em que favorece certas metáforas, imagens e maneiras de pensar. [...] o computador é uma ferramenta que propicia à criança as condições de entrar em contato com algumas das mais profundas ideias em ciência, matemática e criação de modelos. Segundo a filosofia Logo, o aprendizado acontece através do processo de a criança inteligente “ensinar” o computador burro, ao invés de o computador inteligente ensinar a criança burra. (Papert, 1980, p. 25)

Nos últimos anos, a visão de Papert ganhou nova vida com iniciativas como o Scratch. Desenvolvido pelo grupo *Lifelong Kindergarten* no *Media Lab* do *Massachusetts Institute of Technology* (MIT), liderado por Mitchel Resnick. Sob o slogan “imaginar, programar, compartilhar”, torna a programação acessível e divertida para todas as idades, herdando a tradição do Logo ao fornecer uma plataforma visual onde os usuários podem criar e compartilhar projetos interativos, jogos e animações.

[...] a utilização de computadores na educação é muito mais diversificada, interessante e desafiadora, do que simplesmente a de transmitir informação ao aprendiz. O computador pode ser também utilizado para enriquecer ambientes de aprendizagem e auxiliar o aprendiz no processo de construção do seu conhecimento (Valente, 1999, p.10).

Resnick (2009, p. 35), inspirado nas propostas de Papert, formatou o conceito de Aprendizagem Criativa:

As crianças aprendem sobre o processo criativo: como começar uma ideia, criar algo baseado nela, compartilhá-la com os outros, testá-la para ver como funciona e revisá-la com base nos experimentos e nas interações com os colegas. Elas continuamente refinam suas ideias. Isso está no centro do processo criativo, que é tão importante no mundo de hoje.

A programação é identificada como uma extensão da escrita, e aprender a programar fornece aos estudantes as habilidades para escrever novos tipos de coisas, como histórias interativas, jogos, animações e simulações (Resnick, 2009). Ao aprender a programar, os jovens aprendizes não estão “...apenas aprendendo a codificar, eles estão codificando para aprender” (Resnick, 2009, p. 43). Por meio da programação, eles estão aprendendo a resolver problemas de maneira sistemática e têm o poder de expressar ideias e utilizar uma variedade de ferramentas. Construir e programar artefatos computacionais permite que as crianças se envolvam em processos cognitivos de alto nível envolvendo resolução de problemas, pensamento divergente e reflexão (Kafai; Burke, 2014).

O Scratch é uma ferramenta utilizada no Brasil para a disseminação do Pensamento Computacional cujas habilidades propostas por Wing (2006) influenciam o modo de se entender a computação e são parte da demanda educacional deste século.

Uma das vantagens da utilização do Scratch no processo do ensino de programação é o fato da ferramenta ser baseada em uma linguagem de programação por blocos, como um quebra cabeça, sendo um software educativo que desenvolve a criatividade e o Pensamento Computacional, não exigindo conhecimentos prévios de outras linguagens de programação.

A ferramenta Scratch é uma ferramenta educacional ligada a programação, oferecendo ao estudante um ambiente desafiador que o estimule a pensar, envolvendo habilidades como: capacidade de resolver problemas, criatividade, trabalho em grupo e raciocínio lógico e desenvolvimento do Pensamento Computacional. O raciocínio lógico é uma das habilidades fundamentais na aprendizagem de programação e se faz presente em qualquer área do conhecimento, sendo desenvolvido quando o indivíduo se depara com situações problemas e necessita organizar o pensamento detalhadamente, buscando uma solução para resolvê-los. Para desenvolver o Pensamento Computacional em sala de aula, é necessário incorporar estratégias no processo de ensino, no qual o aluno participa da construção do processo de forma flexível e criativa, podendo desfrutar dos benefícios da utilização das ferramentas educacionais como o Scratch (Brackmann, 2017).

O Scratch apresenta algumas potencialidades como liberdade de criação, comunicação, criatividade e compartilhamento, aprendizagens de conceitos escolares, manipulação de mídias, permitindo que os usuários compartilhem projetos entre si, reutilização e adaptação de projetos já existentes, e integração de objetos do mundo físico (Scratch, 2023).

5. Oficinas de Scratch integradas ao ensino de ciências

O planejamento das oficinas de Scratch para execução durante as aulas de Ciências foi estruturado para proporcionar uma experiência envolvente, alinhado às habilidades e competências da Base Nacional Comum Curricular (Brasil, 2018), considerando a Unidade Temática “Terra e Universo”, com ênfase nos movimentos da Terra e simulação do Sistema Solar.

A pesquisa envolveu uma turma de 15 estudantes do 6º ano de uma escola pública da Rede Municipal de Ensino de Farroupilha, RS. À medida que os participantes projetaram soluções com Scratch, eles se envolviam com um conjunto de conceitos computacionais, comuns a outras linguagens de programação. A oficina permitiu a exploração de vários conceitos: sequência, evento, paralelismo, loop, condicionais, operadores, sons e dados exigindo que os estudantes seguissem uma ordem lógica, utilizando o PC para solucionar os problemas propostos.

Foram criadas as contas dos estudantes, por meio dos recursos disponíveis na plataforma para educadores. Uma conta de “Professor Scratch” disponibiliza aos educadores funcionalidades adicionais de gestão, incluindo a possibilidade de criar contas dos estudantes, de organizar projetos em estúdios (repositório de projetos) e de monitorar os comentários.

Posteriormente foram aplicadas as oficinas, no total de 4 períodos (de 1 hora cada), incorporando habilidades relativas a Ciências da Natureza, alinhadas ao referencial curricular e a Base Nacional Comum Curricular (Brasil, 2018).

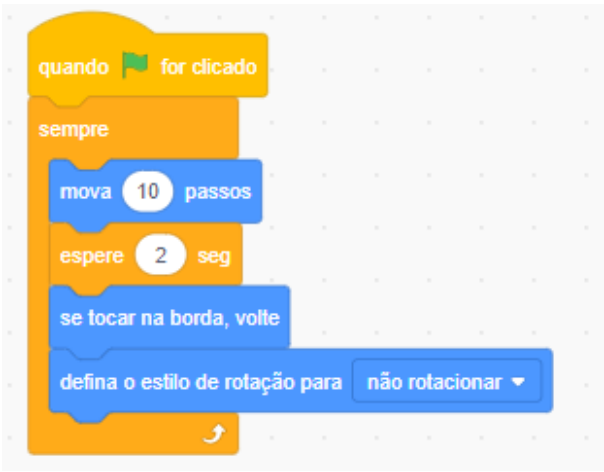
Como introdução à oficina, os estudantes assistiram ao tutorial de iniciação disponível na plataforma e, posteriormente, puderam explorar o ambiente do Scratch.

As atividades foram projetadas para desenvolver o Pensamento Computacional dos estudantes, integrando o aprendizado de Ciências. A resolução de problemas, a decomposição de tarefas complexas em etapas menores, o reconhecimento de padrões e a abstração foram habilidades enfatizadas durante as oficinas. Os estudantes foram encorajados a pensar algorítmicamente para alcançar os resultados desejados em suas criações, ao mesmo tempo que exploravam conceitos científicos de forma prática e interativa. A estruturação das atividades seguiu uma abordagem progressiva, começando pela utilização de conceitos mais simples e avançando para desafios mais complexos, garantindo a adaptação ao nível de aprendizado de cada estudante.

O intuito da primeira atividade (Quadro 1) foi proporcionar a ambientação dos estudantes ao Scratch utilizando funções básicas de Movimento, Controle e Aparência, necessárias para que a atividade fosse solucionada.

Quadro 1: Atividade 1

ATIVIDADE 1							
Desenvolva um algoritmo que faça o personagem mover 20 passos, esperar 2 segundos e se mover novamente. Quando o personagem tocar na borda ele deverá girar automaticamente e continuar se movendo. Utilizar uma estrutura de repetição para que o personagem não pare de se movimentar. Escolha um personagem que represente um astronauta e o cenário que representa a Lua.							
COMANDOS DO SCRATCH							
Movimento	Aparência	Som	Caneta	Controle	Sensores	Operadores	Variáveis
X	X			X			
PENSAMENTO COMPUTACIONAL							
<ul style="list-style-type: none"> Abstração <div style="text-align: center;"> <p>andar 20 passos esperar 2 segundos + andar novamente -----> borda ← não parar de se movimentar</p> </div> 							
<ul style="list-style-type: none"> Reconhecimento de Padrões <div style="text-align: center;"> <p>Estruturas de repetição Movimentos</p> </div> 							

- Decomposição
 - Mover 20 passos
 - Esperar 2 segundos
 - Tocar na borda e voltar
 - Alterar o estilo de rotação para não rotacionar
 - Não parar de se mover
- Algoritmo
 

Fonte: Elaborado pelas autoras (2023)

Na segunda atividade (Quadro 2), além de explorar a lógica de programação, buscou-se explorar a parte gráfica da ferramenta. Os estudantes desenvolveram um algoritmo para representar o Sol e utilizaram atores para representar os planetas, utilizando conceitos geométricos e espaciais.

Quadro 2: Atividade 2

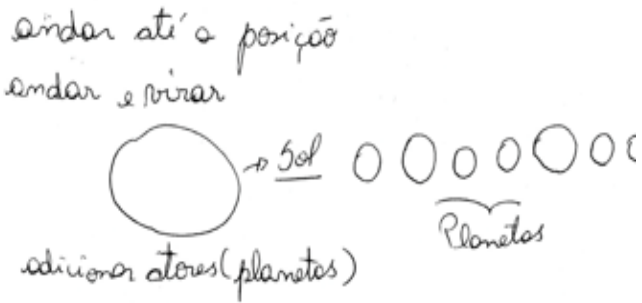
ATIVIDADE 2

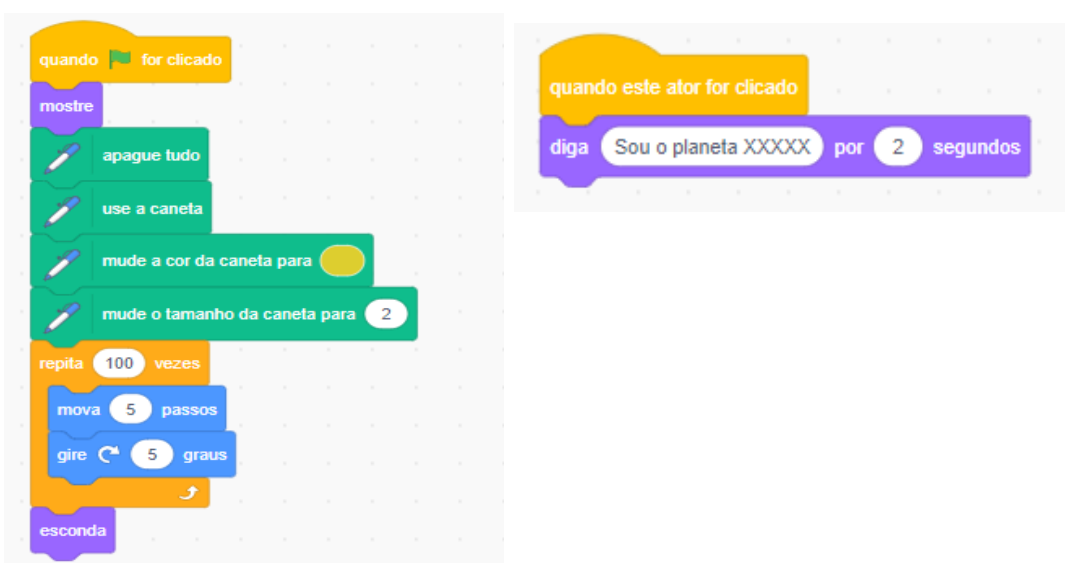
Desenvolva um algoritmo que mova o personagem e desenhe 1 círculo que irá representar o Sol. Posteriormente, adicione atores (imagens) que representarão os planetas que compõem o sistema solar. Ao clicar nestes atores o nome do respectivo planeta e deverá aparecer por 2 segundos. Utilize um cenário semelhante ao espaço e lembre-se que cada planeta possui um tamanho distinto.

COMANDOS UTILIZADOS NO SCRATCH

Movimento	Aparência	Som	Caneta	Controle	Sensores	Operadores	Variáveis
X	X		X	X			

PENSAMENTO COMPUTACIONAL

- Abstração
 

<ul style="list-style-type: none"> Reconhecimento de Padrões Estruturas de repetição Movimentos
<ul style="list-style-type: none"> Decomposição Mover 5 passos girar 5° Remover imagem do personagem Adicionar atores Quando clicado, mostrar o nome do planeta por 2 segundos
<ul style="list-style-type: none"> Algoritmo 

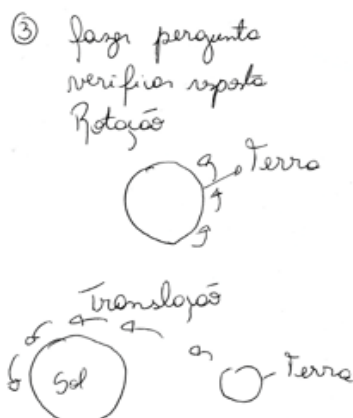
Fonte: Elaborado pelas autoras (2023)

Na terceira atividade (Quadro 3), os estudantes foram desafiados a utilizar estruturas de repetição, estruturas de controle e operadores. A atividade consistiu em criar cenários representando o Sol e a Terra e a criação de variáveis de interação com o usuário.

Quadro 3: Atividade 3

ATIVIDADE 3							
Desenvolva um algoritmo que ao ser executado, o personagem irá questionar o usuário sobre qual dos movimentos realizados pelo planeta Terra deseja visualizar. Se a resposta for Rotação, o ator (Terra) deverá realizar o movimento ao redor de si mesmo (seu eixo). Para isso, utilize uma estrutura de repetição que repita por 50 vezes: a cada intervalo de 1 segundo realize o movimento de giro de 45°. Se a resposta for Translação, o ator (Terra) deverá realizar o movimento em torno do sol. Para isso, repita 10 vezes para que a movimentação do ator, a cada intervalo de 1 segundo, simule o deslocamento para os pontos cardeais (Norte, Sul, Leste e Oeste). Lembre-se que você irá precisar de dois atores que representam o Sol e a Terra, além de um cenário relacionado ao assunto. Posicione o ator Sol no centro do cenário.							
COMANDOS UTILIZADOS NO SCRATCH							
Movimento	Aparência	Som	Caneta	Controle	Sensores	Operadores	Variáveis
X	X			X	X	X	X
PENSAMENTO COMPUTACIONAL							

• Abstração



• Reconhecimento de Padrões

- Estrutura condicional
- Sensores
- Operadores Lógicos
- Movimentos
- Variáveis

• Decomposição

- Realizar a pergunta
- Se resposta for Rotação (simular movimento em torno do ator Terra)
 - Repetir girar 45 °
 - Esperar 1 segundo
- Se a resposta for Translação
 - Repetir 10 vezes
- Simular movimento em torno do ator Sol (referência de deslocamento os Pontos Cardeais)
- Esperar 1 segundo após cada movimento

• Algoritmo

```

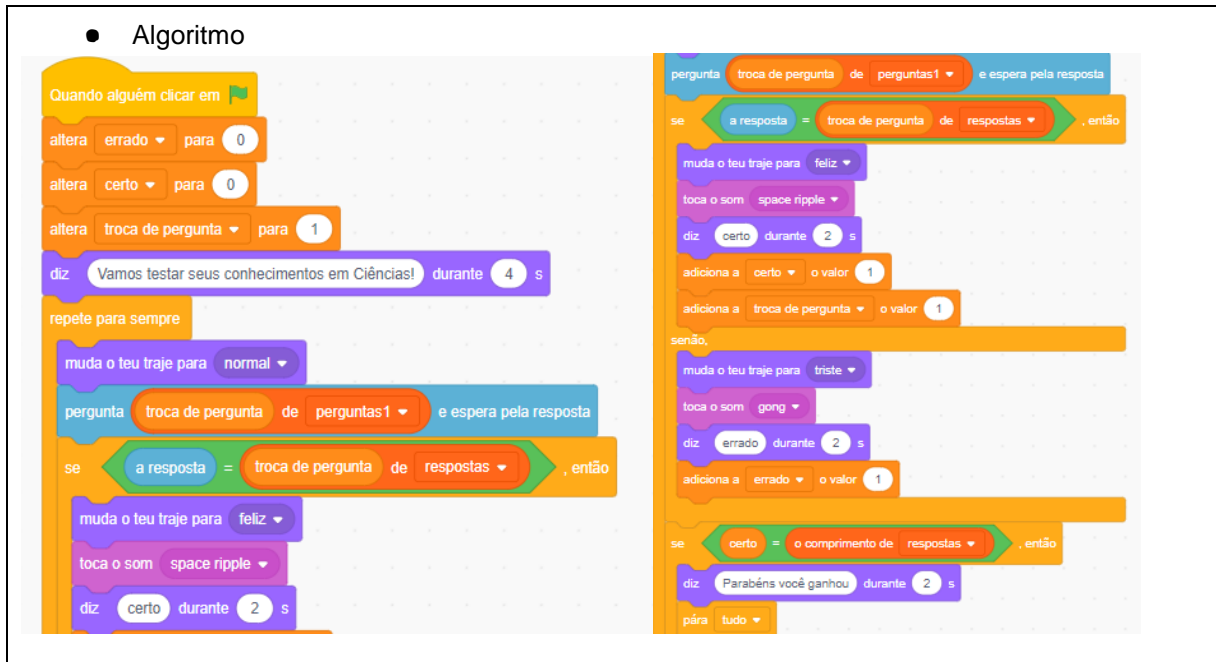
quando for clicado
pergunte Qual movimento você deseja visualizar? Rotação ou Translação? e espere
se resposta = Rotação então
  repita 30 vezes
    espere 1 seg
    gire 45 graus
  senão
    repita 10 vezes
      vá para x: 119 y: 28
      espere 1 seg
      vá para x: -21 y: 119
      espere 1 seg
      vá para x: -180 y: 30
      espere 1 seg
      vá para x: -14 y: -152
      espere 1 seg
    
```

Fonte: Elaborado pelas autoras (2023)

A quarta atividade (Quadro 4) consistiu em criar um quiz com conteúdo aprendidos nas aulas de Ciências. Para isso, foi necessária a utilização de estrutura de comandos que envolvessem a criação de perguntas, verificação das respostas e controle de pontuação. O projeto foi compartilhado com os demais colegas através do estúdio da turma, para que todos pudessem testar seus conhecimentos.

Quadro 4: Atividade 4

ATIVIDADE 4							
<p>Desenvolva um algoritmo para testar o conhecimento dos seus colegas sobre os assuntos aprendidos nas aulas de Ciências neste ano. Para isso, você deverá criar 5 perguntas sobre o conteúdo de Ciências. Ao iniciar, será mostrada a mensagem: “Vamos testar seus conhecimentos em Ciências”. Deverá ser criada uma variável de Pontos, que demonstre a quantidade de respostas corretas. Ao final das 5 questões, se o jogador tiver a pontuação maior ou igual a 4 pontos, deverá ser demonstrada uma mensagem de "Parabéns!" Utilize também um som no jogo. Use a sua imaginação e deixe seu jogo mais divertido.</p>							
COMANDOS UTILIZADOS NO SCRATCH							
Movimento	Aparência	Som	Caneta	Controle	Sensores	Operadores	Variáveis
	X	X		X	X	X	X
PENSAMENTO COMPUTACIONAL							
<ul style="list-style-type: none"> ● Abstração Criar 5 perguntas conteúdos de Ciências Mostrar mensagem de início Verificar pontuação Utilizar som 							
<ul style="list-style-type: none"> ● Reconhecimento de Padrões Estrutura de Condição Variáveis Estrutura de Repetição Controles Operadores Sensores Som 							
<ul style="list-style-type: none"> ● Decomposição Criar cenário Mostrar mensagem Criar variável Pontos Criar perguntas Verificar respostas Verificar pontuação Utilizar som 							

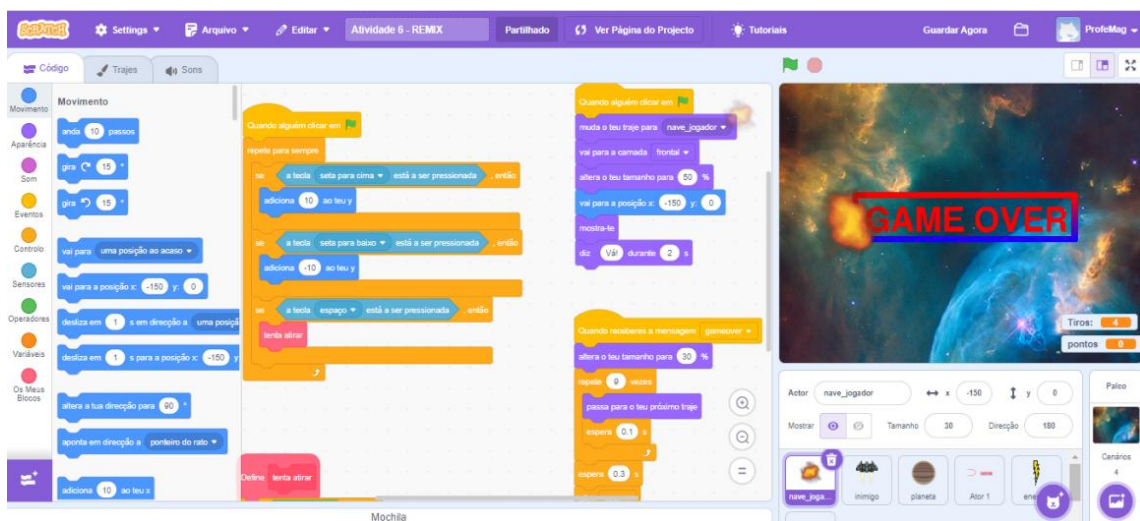


Fonte: Elaborado pelas autoras (2023)

Na quinta atividade foi proposto mixar um jogo. O jogo consistia em salvar o planeta Terra de uma chuva de meteoros. A funcionalidade "Remix" no Scratch é uma funcionalidade que permite aos usuários criar novos projetos a partir de projetos existentes.

Remixar um projeto permite editar um algoritmo, realizando modificações, adições ou adaptações. Esta atividade (Quadro 5), além de desenvolver o Pensamento Computacional dos estudantes, promoveu uma cultura de aprendizado colaborativo, criatividade e inclusão, aprofundando a compreensão do código por meio da análise dos algoritmos alterados.

Quadro 5: Atividade 5



Fonte: Estúdio da turma no Scratch

Como sexta atividade, foi oportunizado aos estudantes um momento de criação livre na ferramenta (jogo, história ou animação). O tema da atividade deveria estar relacionado ao conteúdo das aulas de Ciências. Esta atividade, permitiu que os estudantes utilizassem a imaginação e criatividade, para transformar livremente suas ideias em realidade digital.

Todas as atividades desenvolvidas pelos estudantes durante as oficinas foram compartilhadas no estúdio da turma no Scratch. Ao interagir com os projetos criativos de seus colegas, os estudantes foram incentivados a expandir sua própria imaginação agregando novas funcionalidades às suas criações.

A resolução de problemas no contexto das atividades no Scratch não se limitou apenas à programação, mas também se estendeu à resolução de desafios científicos. Os estudantes foram incentivados a abordar questões relacionadas à Ciências, utilizando os pilares do Pensamento Computacional para formular soluções e representar fenômenos de maneira visual.

A decomposição de tarefas complexas foi integrada à exploração de conceitos científicos específicos. Ao criar simulações interativas do Sistema Solar, os estudantes dividiram o projeto em etapas menores. Isso fortaleceu suas habilidades de programação e aprofundou a compreensão dos fenômenos relacionados.

O reconhecimento de padrões foi aplicado na reutilização dos códigos. Essa habilidade transcende a programação no Scratch, capacitando os alunos a aplicar uma mentalidade analítica na interpretação de dados e na identificação de correlações significativas.

A abstração foi incorporada ao entendimento das atividades e conceitos científicos mais amplos. Os estudantes simplificaram representações do Sistema Solar, concentrando-se nos aspectos mais relevantes. Essa prática contribuiu para a capacidade dos estudantes de compreender conceitos científicos abstratos

O pensamento algorítmico foi essencial na representação prática das atividades. Os estudantes desenvolveram sequências lógicas de passos para resolução das atividades.

6. Resultados Obtidos

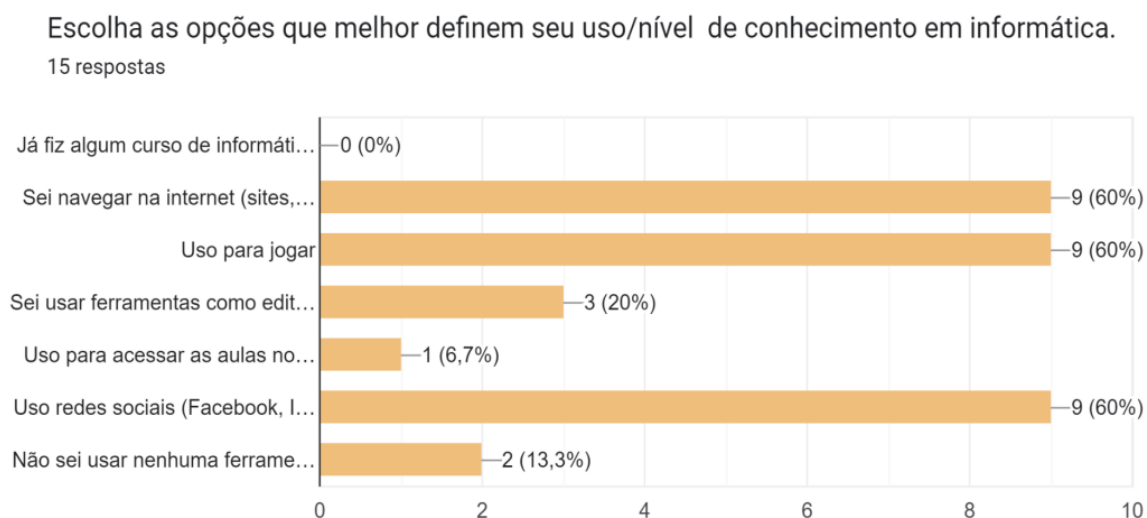
Por meio da análise dos códigos gerados para a solução das atividades e das observações realizadas durante a execução das oficinas e da análise dos resultados do questionário aplicado pós-oficina, buscou-se descrever os impactos observados,

evidenciando as implicações mais amplas desses resultados para o campo de estudo em questão.

Conforme dados obtidos no questionário pós-oficina, 9 (60%) estudantes se declararam do sexo masculino, 5 (33,3%) do sexo feminino e 1 (6,7%) preferiu não responder.

Todos relataram que possuíam acesso à internet fora do ambiente da escola, sendo que 40% afirmaram terem experiência prévia com o Scratch. Quando questionados sobre sua competência em informática, a maior parte mencionou utilizar principalmente para jogos, navegação em diferentes sites e redes sociais, conforme representado no Gráfico 1.

Gráfico 1: Nível de uso/conhecimento em informática



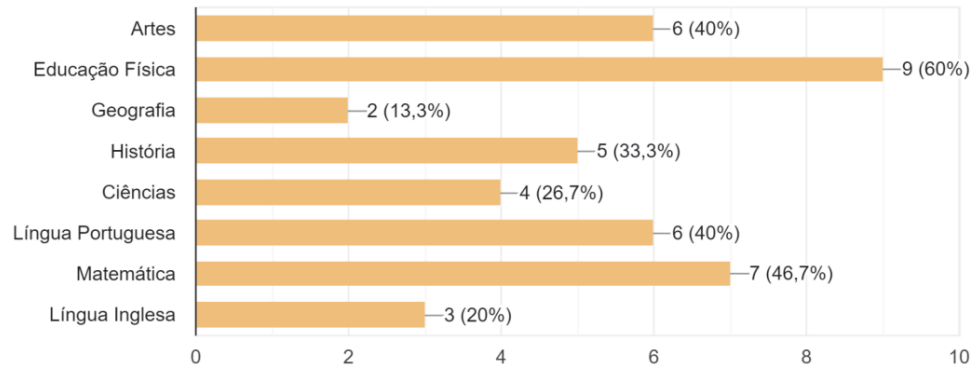
Fonte: Análise das repostas ao questionário pós-oficina

Ao serem indagados sobre a disciplina que mais se identificavam, a de Ciências, ficou à frente apenas de Geografia e Língua Inglesa, o que demonstra um nível de desinteresse dos estudantes nessa área (Gráfico 2). Essa constatação evidencia a necessidade de implementar estratégias que tornem o aprendizado de Ciências mais atrativo e relevante.

Gráfico 2: Disciplina que mais gosta/tem facilidade de aprender na escola

Qual (is) disciplina(s) você MAIS gosta/tem facilidade de aprender na escola?

15 respostas



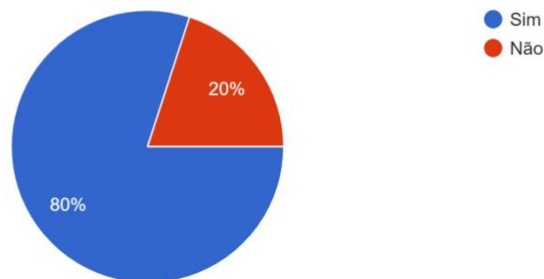
Fonte: Análise das repostas ao questionário pós-oficina

Nove dos participantes (60%) relataram dificuldades em utilizar o Scratch e destes, 80% conseguiram superá-las durante as oficinas (Gráfico 3).

Gráfico 3: Superação de dificuldades de uso do Scratch

Se você sentiu dificuldade em utilizar o Scratch ao longo das atividades você conseguiu superar as dificuldades?

15 respostas



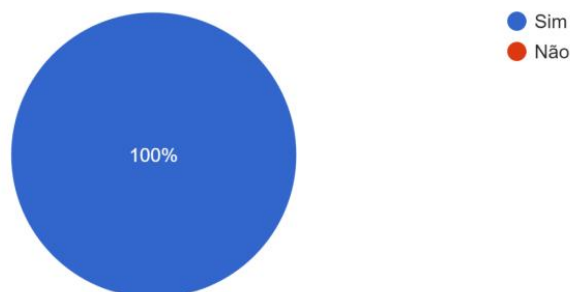
Fonte: Análise das repostas ao questionário pós-oficina

Para a totalidade dos estudantes participantes, as atividades desenvolvidas durante a oficina auxiliaram no entendimento dos conteúdos relacionados à Ciências, conforme demonstrado no gráfico 4. Esse mesmo número demonstrou o interesse em utilizar a ferramenta novamente.

Gráfico 4: Auxílio das aulas para entendimento do conteúdo de Ciências

As aulas te auxiliaram para entendimento dos conteúdos de Ciências envolvidos?

15 respostas



Fonte: Análise das repostas ao questionário pós-oficina

Por meio das observações realizadas durante a aplicação das oficinas e da análise dos códigos gerados pelos estudantes, pode-se avaliar que as oficinas de Scratch ofereceram uma jornada transformadora de aprendizado de habilidades e competências relacionadas à Ciências e ao desenvolvimento do PC, permitindo explorar os fundamentos da programação de uma forma acessível e divertida, ao mesmo tempo que solucionaram problemas, pensaram de forma lógica e expressaram sua criatividade. Os participantes puderam compartilhar ideias e projetos, aprimorando suas habilidades de trabalho em equipe, proporcionando diferentes perspectivas e abordagens para resolução de problemas. À medida que as oficinas progrediram, percebeu-se, como o PC estava se tornando uma parte integrante do repertório dos participantes. Eles passaram a enfrentar desafios de forma mais sistemática e organizada, identificando padrões, particionando problemas complexos e elaborando soluções por meio de algoritmos mais otimizados.

A interação dos temas ligados à Ciências proporcionou uma aplicação prática dos conceitos científicos e a compreensão significativa dos conteúdos relacionados. Os estudantes não apenas programaram no Scratch; eles exploraram, questionaram e representaram conceitos científicos.

Com base nos resultados obtidos, este estudo demonstra que a utilização da ferramenta Scratch não se restringe a sessões de programação. Representa uma jornada de exploração, descobertas, aprendizagem e desenvolvimento de habilidades e competências, quando incorporado às atividades curriculares, podendo ser empregado como ferramenta pedagógica no ensino de Ciências ao mesmo tempo em que fomenta o PC.

7. Considerações Finais

À medida que avançamos em uma era cada vez mais digital, o PC emerge como uma habilidade fundamental e, a escola, para além de apenas transmitir conhecimentos, tem o compromisso de proporcionar o desenvolvimento integral aos estudantes. Este estudo demonstrou que a utilização do Scratch promoveu uma experiência enriquecedora aos estudantes. Ao integrar criatividade e programação, as oficinas foram além de atenderem aos requisitos curriculares, estimularam uma abordagem inovadora para o ensino de Ciências ao mesmo tempo que potencializaram o Pensamento Computacional.

Os trabalhos futuros podem abranger o desenvolvimento de projetos temáticos para explorar fenômenos naturais e teorias científicas; a integração de sensores e dispositivos externos como *micro:bit* e Arduíno, para coletar dados e incorporá-los aos projetos científicos; a criação de recursos educacionais complementares, como *e-book* para apoiar a integração da ferramenta Scratch ao ensino de Ciências.

A habilidade de pensar de forma algorítmica, decompor problemas complexos, reconhecer padrões e abstrair informações são habilidades essenciais para o êxito pessoal e profissional em uma sociedade cada vez mais orientada pela informação e tecnologia. Portanto, é fundamental que o Pensamento Computacional seja incorporado de forma integrada nas diversas disciplinas e áreas de estudo.

8. Referências

- BBC LEARNING. **Learning computing for kids**: beginner coding resources. 2015. Disponível em: <https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1>. Acesso em: 5 jan. 2024.
- BLIKSTEIN, P. **Travels in Troy with Freire**: Technology as an Agent for Emancipation. 2008. Doctoral dissertation - Northwestern University, School of Education and Social Policy, Evanston, IL, 2008. Disponível em: <https://tltlab.org/wp-content/uploads/2019/02/2008.Book-B.Travels.pdf>. Acesso em: 29 jan. 2024.
- BRACKMANN, C. P. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica**. Orientador: Dante Augusto Couto Barone. 2017. 226 f. Dissertação (Mestrado em Tecnologia educacional) - Universidade Federal do Rio Grande do Sul, Porto Alegre, Rio Grande do Sul, 2017. Disponível em: <https://lume.ufrgs.br/handle/10183/172208>. Acesso em: 30 jan. 2024.
- BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**: Ensino Fundamental. Brasília: MEC/Secretaria de Educação Básica, 2018. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 25 jan. 2024.
- CERUZZI, P. E. **A History of Modern Computing**. Cambridge, MA: MIT Press, 2003.

CODE.ORG. **About us**. 2016. Disponível em: <https://code.org/about>. Acesso em: 5 fev. 2024.

GIL, A. C. **Método e técnicas de pesquisa social**. São Paulo: Atlas, 1999.

HODGES, A. **Alan Turing**: The Enigma. New York: Simon and Schuster, 1983.

KAFAI, Y. B.; BURKE, Q. Constructionist gaming: understanding the benefits of making games for learning. **Educational Psychologist**, v. 50, n. 2, p. 313-334, 2015. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4784508/>. Acesso em: 10 out. 2023.

LISBÔA, E. S.; COUTINHO, C. P. O Processo da Comunicação na Sociedade da Informação em Busca de um Referencial Teórico. *In*: J. B. Bottentuit Junior, & C. P. Coutinho (Org.). **Educação On-line**, Curitiba, PR: CRV Editora, 2012, p. 41-61.

LIUKAS, L. **Hello Ruby**: adventures in coding. New York: Feiwei & Friends, 2015.

MALONEY, J. H.; RESNICK, M.; RUSK, N.; SILVERMAN, B.; EASTMOND, E. (2004). Scratch: a sneak preview. *In*: SECOND INTERNATIONAL CONFERENCE ON CREATING, CONNECTING, AND COLLABORATING THROUGH COMPUTING, 2004, Kyoto. **Proceedings...** Kyoto, Japan: 2004. p. 104-109. Disponível em: <https://ieeexplore.ieee.org/document/1314376>. Acesso em: 10 out. 2023.

MIT NEWS. **Professor Emeritus Seymour Papert, pioneer of constructionist learning, dies at 88**. Cambridge, Massachusetts, 2016. Disponível em: <https://news.mit.edu/2016/seymour-papert-pioneer-of-constructionist-learning-dies-0801>. Acesso em: 15 out. de 2023.

NUNES, D. J. Ciência da Computação na Educação Básica. **Jornal da Ciência**, n. 4340, 09 set. 2011. Disponível em: <http://jcnoticias.jornaldaciencia.org.br/12-ciencia-da-computacao-na-educacao-basica/>. Acesso em: 16 out. 2023.

PAPERT, S. **Mindstorms**: Children, Computers, and Powerful Ideas. New York: Basic Books, 1980.

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

RESNICK, M. **Scratch**: Programming for All. Cambridge, MA: Lifelong Kindergarten Group, MIT Media Lab, 2009.

SCRATCH. **Scratch for Parents**, 2023. Disponível em: <https://scratch.mit.edu/parents>. Acessado em: 23 dez. 2023.

SENTANCE, S., CSIZMADIA, A. Computing in the curriculum: challenges and strategies from a teacher's perspective. **Educ Inf Technol**, v. 22, p. 469-495, 2017. Disponível em: <https://doi.org/10.1007/s10639-016-9482-0>. Acesso em: 10 out. 2023.

TABESH, A. **Pattern Recognition and Problem Solving**. Boston: Academic Press, 2017.

VALENTE, J. A. **Tecnologias na Educação**: dos paradigmas às práticas. Campinas: UNICAMP/NIED, 1999.

WING, J. M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006. Disponível em: <https://doi.org/10.1145/1118178.1118215>. Acesso em: 24 dez. 2023.

WING, J. M. Computational thinking: What and why? **The Link Magazine**, v. 8, n. 1, p. 27-30, 2010. Disponível em: <http://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf>. Acesso em: 26 dez. 2023.