

Laterabilidade e Pensamento Computacional: caminhos para uma aprendizagem significativa usando o Scratch

David Machado¹, Vera Regina da Silva dos Santos¹

¹Colégio Farroupilha – Porto Alegre – RS – Brasil

david.machado.mv@hotmail.com, vrsklimach@yahoo.com.br

Abstract. *Computational Thinking is an approach that allows exploring the foundational principles of Computing, as well as systematizing a solution for a specific problem. Thus, this article presents a didactic sequence applied in 4th-grade classes of Elementary Education, in the Technology and Computing subject, at a private school in Rio Grande do Sul. The objective was to explore concepts of laterality through programming strategies using Scratch, which were addressed in the Mathematics classes that took place concurrently. The results demonstrate that the concepts of laterality can be effectively explored through the pillars of Computational Thinking.*

Resumo. *O Pensamento Computacional é uma abordagem que permite explorar os fundamentos basilares da Computação, bem como sistematizar uma solução para um determinado problema. Deste modo, este artigo apresenta uma sequência didática aplicada em turmas do 4º ano do Ensino Fundamental, na disciplina de Tecnologia e Computação, em uma escola privada do RS. O objetivo foi explorar conceitos de laterabilidade por meio de estratégias de programação usando Scratch, os quais foram abordados nas aulas de Matemática que ocorreram paralelamente. Os resultados mostram que os conceitos de laterabilidade podem ser explorados de maneira efetiva por meio dos pilares do Pensamento Computacional.*

1. Introdução

O ensino da Computação nas escolas já é uma realidade em muitos países, como Alemanha, Argentina, Austrália, Coreia do Sul, Escócia, França, Inglaterra, Estados Unidos da América, Finlândia, Grécia, Índia, Israel, Japão e Nova Zelândia, entre outros (SBC, 2017). Sendo assim, a Computação, diferentemente da Informática Educativa, na perspectiva de Ribeiro et al. (2022) tem sido vista como um campo urgente a ser abordado, no qual preconizam-se o desenvolvimento de habilidades relacionadas à resolução de problemas complexos, à compreensão e à fluência no mundo digital.

O Pensamento Computacional (PC) é uma abordagem essencial na era digital, pois permite lidar com situações complexas com base nos fundamentos da Computação, possibilitando a organização e sistematização da resolução de um determinado problema. De acordo com Brackmann (2017), esses fundamentos incluem a identificação dos elementos essenciais (abstração), o reconhecimento de padrões em soluções semelhantes (reconhecimento de padrões), a decomposição das etapas de um problema em etapas menores (decomposição) e a criação da sequência de passos necessária para resolver um problema (algoritmos).

Nesse cenário, o PC é evidenciado na Base Nacional Comum Curricular (BNCC, 2018), no currículo de Referência em Tecnologia e Computação (RAABE et al. 2018), do Centro de Inovação para a Educação Brasileira (CIEB), e nas diretrizes para o ensino de computação da Sociedade Brasileira de Computação (SBC, 2019). Além disso, com a homologação do parecer CNE/CEB nº 2/2022 (BRASIL, 2022), que estabelece a obrigatoriedade do ensino de Computação nas escolas, o PC passa a ser destacado como um eixo a ser desenvolvido como complemento à BNCC.

Este trabalho é um relato de experiência de uma proposta aplicada em uma disciplina intitulada “Tecnologia e Computação”, na qual são seguidas as diretrizes recomendadas pelo CIEB, e o PC é um dos eixos que é articulado junto às práticas e ao planejamento das professoras regentes de cada turma. A proposta foi realizada com turmas do 4º ano do Ensino Fundamental, tendo como objetivo desenvolver os conceitos de lateralidade, a partir da resolução de desafios que consistiam em criar algoritmos (sequências de instruções) dentro do ambiente de programação do Scratch. Pretendemos compartilhar os resultados destas práxis e fomentar o debate acerca do quanto a intersecção Tecnologia e Pedagogia pode ser desenvolvida de forma qualitativa, com ênfase numa intencionalidade pedagógica voltada para a criação autoral, cooperativa e lúdica.

Este artigo está organizado da seguinte maneira: na seção 2 é apresentado o referencial teórico. Na seção 3 estão os passos metodológicos e os objetivos de aprendizagem de cada aula. Na seção 4 apresentamos os resultados e um feedback qualitativo dos alunos acerca do seu próprio processo de aprendizagem e, por fim, a seção 5 as considerações finais

2. Pensamento Computacional

O termo "Pensamento Computacional" ganhou destaque a partir da publicação do artigo intitulado "*Computational Thinking*" por Jeannette Wing em 2006 (Wing, 2006). Nesse artigo, Wing aborda a importância dos princípios e fundamentos da computação serem incorporados no currículo escolar desde os níveis iniciais da educação básica. A temática ganhou notoriedade, de fato, quando foi incluída na Base Nacional Comum Curricular (BNCC), em sua versão final homologada no ano de 2018.

A Base Nacional Comum Curricular (BNCC) é um documento normativo que estabelece de forma progressiva os conhecimentos considerados fundamentais para todos os alunos da educação básica. Em complemento à BNCC, um importante marco foi alcançado em fevereiro de 2022, quando o Conselho Nacional de Educação (CNE) aprovou um relatório que considera o PC como um dos eixos temáticos para a introdução do ensino de computação na Educação Básica. Esse relatório, atualmente homologado sob o parecer CNE/CEB nº 2/2022, destaca as contribuições da Sociedade Brasileira de Computação (SBC), do CNE e do Fórum Nacional de Licenciaturas em Computação (ForLic).

Diversos autores, como Brackmann (2017), Code.Org (2016), Liukas (2015) e BBC Learning (2015), convergem na concepção de que o Pensamento Computacional (PC) é composto por quatro pilares fundamentais: Decomposição, Reconhecimento de Padrões, Abstração e Algoritmos. De acordo com Vicari et al. (2018), essas quatro

dimensões ou "pilares" constituem a base do PC e são conceitos interdependentes durante o processo de formulação de soluções computacionalmente viáveis.

Em relação às abordagens para trabalhar com o PC, é possível identificar abordagens distintas: desplugada e plugada. As atividades de computação desplugada predominam na esfera concreta, permitindo que os estudantes aprendam de forma lúdica, dispensando o uso de dispositivos tecnológicos. De acordo com Rodrigues et al. (2022) a computação desplugada tem por objetivo ensinar o PC de forma lúdica, inovadora e criativa por meio de dinâmicas pitorescas que enfatizam o aprender fazendo, a lógica e a resolução de dilemas racionalmente. Em contrapartida, as atividades plugadas se baseiam no emprego de recursos tecnológicos, como é o caso da proposta apresentada.

Em suma, tanto as atividades desplugadas quanto as atividades plugadas desempenham um papel importante no ensino do PC. Enquanto as atividades desplugadas promovem a aprendizagem tangível e concreta, as atividades plugadas exploram as possibilidades das tecnologias digitais para aprofundar o conhecimento e as habilidades dos alunos no âmbito da computação. Nesse sentido, a proposta em questão se enquadra no contexto das atividades plugadas, aproveitando os recursos tecnológicos disponíveis para enriquecer o processo de aprendizagem do PC.

4. O Scratch e a Aprendizagem Criativa (AC)

O Scratch é um projeto desenvolvido pelo grupo de pesquisa *Lifelong Kindergarten*, liderado pelo professor e pesquisador Mitchel Resnick, no *Massachusetts Institute of Technology* (MIT). Lançado em 2007, o Scratch é uma plataforma online que permite aos usuários criar e compartilhar projetos por meio de uma linguagem de programação visual. Essa abordagem estimula a criatividade ao permitir que os usuários "imaginem, criem, brinquem, compartilhem, reflitam e imaginem novamente" (Resnick, 2020).

Ao contrário das linguagens de programação textuais, o Scratch utiliza blocos encaixáveis de cores variadas para atribuir comandos. Esses blocos são arrastados e encaixados para criar sequências de comandos (algoritmos). Essa abordagem visual e interativa permite que o usuário crie algoritmos e soluções de maneiras diversas. Essa característica flexível e adaptável da linguagem torna o Scratch uma ferramenta adequada para o ensino de programação e PC.

No site do Scratch, é possível encontrar recursos pedagógicos que auxiliam no aprendizado e na exploração da linguagem. Um desses recursos são os cards de programação, que fornecem dicas e orientações sobre como empregar determinadas estratégias e combinações de códigos ao projeto. Desse modo, esse baralho é um recurso que pode envolver e motivar os estudantes a explorar diferentes combinações de blocos de acordo com as necessidades de cada projeto, ampliando as possibilidades de criação e personalização. Esse recurso reduz a necessidade de momentos expositivos sobre conteúdos específicos de codificação, uma vez que o baralho permite a realização de dinâmicas em que os estudantes podem testar diferentes cartas, compartilhar o que funcionou bem e trocar cartas entre si.

5. Metodologia

As aulas em que a sequência didática foi aplicada ocorreram na disciplina de Tecnologia e Computação (TechComp), compreendendo um total de quatro encontros semanais, cada um com duração de 50 minutos. Esses encontros contemplaram nove turmas do 4º ano dos Anos Iniciais. A disciplina de TechComp faz parte do currículo escolar e é oferecida para alunos do 1º ao 5º ano dos Anos Iniciais, sendo ministrada por dois professores que possuem vínculo tanto com os Anos Iniciais quanto com o setor de Tecnologia Educacional da escola. Esses professores desempenham um papel fundamental como interlocutores no desenvolvimento de práticas pedagógicas e projetos interdisciplinares.

É relevante destacar que os alunos do 4º ano já possuíam uma experiência prévia com o uso de computadores, especialmente por meio de atividades desplugadas, como jogos e outras atividades que visavam estimular a resolução de problemas e a criatividade, com o suporte das Tecnologias Digitais de Informação e Comunicação (TDIC), desde o 1º ano.

5.2 A proposta

A presente proposta surgiu a partir de uma reunião realizada com uma das professoras titulares do 4º ano, com o intuito de explorar questões relacionadas às noções de lateralidade, utilizando recursos disponíveis na ferramenta Scratch. Essa proposta foi fundamentada nas habilidades estabelecidas na matriz curricular da escola e do referencial do CIEB. As habilidades extraídas da matriz curricular foram as seguintes: (i) aplicar noções de lateralidade, utilizando diferentes suportes e deslocando-se em diferentes espaços nas situações apresentadas; (ii) explorar a noção de ângulo, associada às ideias de giro, mudança de direção e abertura nas situações apresentadas; e (iii) interpretar movimentação em diversas representações gráficas, como desenhos, malhas quadriculadas, croquis, itinerários, mapas e maquetes. Na Tabela 01 é apresentada uma síntese da proposta.

Tabela 1. Síntese da proposta

	Objetivo de cada aula	Metodologia	Habilidades
Aula 1	Criar um algoritmo para movimentar o personagem de tal forma que ele percorra todo o trajeto indicado evitando repetições no código.	Disponibilização do projeto contendo um cenário pelo qual o personagem será movimentado; Demonstrações expositivas acerca do funcionamento dos blocos de repetição e de rotação; Distribuição das cartas do baralho do Scratch para engajamento durante as soluções do problema.	PC05AL01- Conhecer e utilizar algoritmos com repetições. H3. Explorar a noção de ângulo associada às ideias de giro, mudança de direção e abertura nas situações apresentadas. HAV7. Aplica noções de lateralidade, utilizando diferentes suportes e deslocando-se em espaços diversos nas situações apresentadas.
Aula 2	Fazer com que o usuário digite uma das direções possíveis e o personagem ande de acordo com o que	Acessar o mesmo projeto para dar finalizar ou iniciar a resolução dos desafios. Na introdução do conceito de cada bloco (operadores, condicionais e entrada de	PC05AL01- Conhecer e utilizar algoritmos com repetições. H3. Explorar a noção de

	<p>for digitado, seguindo os desafios propostos:</p> <ul style="list-style-type: none"> - Como podemos fazer com que o personagem percorra todas as direções usando os blocos de decisão? - Quantas situações de decisões (bloco SE e ENTÃO), ou quantas vezes o “Scratch deverá pensar” para conseguir percorrer todo o caminho? 	<p>dados), o professor fará alguns questionamentos: “Qual a utilidade de cada bloco?” Ou “O que aconteceria se alguém digitasse algo diferente do esperado?”</p>	<p>ângulo associada às ideias de giro, mudança de direção e abertura nas situações apresentadas.</p> <p>HAV7. Aplica noções de lateralidade, utilizando diferentes suportes e deslocando-se em espaços diversos nas situações apresentadas.</p>
Aula 3	<p>Utilização de entrada de dados para manipulação usando blocos de condicionais.</p>	<p>Exploração do significado e possibilidade de encaixe de cada um dos blocos que serão utilizados.</p> <p>Realizar diferentes exemplos e combinações acerca do uso de condicionais dentro do Scratch;</p>	<p>PC05AL01- Conhecer e utilizar algoritmos com repetições.</p> <p>H3. Explorar a noção de ângulo associada às ideias de giro, mudança de direção e abertura nas situações apresentadas.</p> <p>HAV7. Aplica noções de lateralidade, utilizando diferentes suportes e deslocando-se em espaços diversos nas situações apresentadas.</p>
Aula 4	<p>Compartilhamento individual de cada projeto no estúdio da turma, para apreciação coletiva.</p>	<p>Realização de rodadas de exploração de projetos, nas quais os alunos poderão comentar e interagir com as produções de seus colegas;</p> <p>Organização de rodas de conversa sobre o compartilhamento de estratégias de programação utilizadas.</p>	

A fim de apoiar a referida proposta, optou-se pelo uso do Scratch como ferramenta principal, na qual foram elaboradas situações de aprendizagem pelos docentes. Essa escolha permitiu que os estudantes desenvolvessem habilidades nas áreas da Matemática, especificamente em relação à lateralidade, e da tecnologia, particularmente no que se refere ao Pensamento Computacional.

Para viabilizar a realização da atividade, o professor realizou adaptações na proposta original, a fim de tornar os blocos utilizados mais adequados à faixa etária dos alunos. Para tanto, foi disponibilizado aos estudantes um projeto pré-elaborado, contendo implementações adicionais, conforme ilustrado na Figura X.

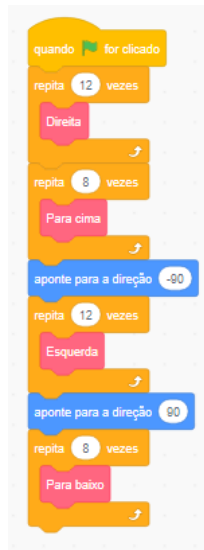


Figura 2. Algoritmo etapa 1

5.2.2 Aula 02 - Aprimorando

Na segunda etapa foram apresentadas mais duas situações de aprendizagem, nas quais os alunos tivessem de criar diferentes algoritmos para resolvê-las, sendo elas: (i) ir até a goleira da direita, passando pelo centro do campo; e (ii) dar duas voltas no campo e chegar na marca do pênalti da goleira esquerda. Novamente, foram instigados a utilizar as noções de laços de repetição (PC05AL01) para compor suas sequências.

5.2.3 Aula 03 – Introdução às condicionais e entrada de dados

No terceiro encontro, teve-se como objetivo introduzir os comandos de decisão do Scratch “se”, “então” e “senão”, entrada de dados, por meio do bloco “pergunte” e como manipular os dados de entrada utilizando os blocos de condicional mencionados. O desafio era criar um algoritmo que fizesse o personagem se movimentar, dependendo da direção digitada pelo usuário, como mostra a Figura 3.



Figura 3. Algoritmo acrescido de condicionais

Conforme a Figura 3, o algoritmo pergunta ao usuário a direção que o personagem deverá percorrer e, em seguida, o conjunto de caracteres digitado é analisado pela condicional, de tal forma que se o resultado (o que foi digitado e

armazenado) for exatamente igual à “direita” o personagem se movimentará para a direita. O professor introduziu o conceito de cada bloco e os alunos auxiliaram no encaixe, alguns questionamentos foram feitos. Em seguida os alunos tiveram de continuar a programação com o intuito de fazer com que o personagem desse a volta completa pelo campo.

Aula 4 - Compartilhamento de resultados e testagem coletiva

Na última etapa foi realizada uma exposição dos trabalhos e os alunos puderam compartilhar a sua estratégia para resolver cada situação.

6. Resultados

Durante as aulas, os estudantes demonstraram um envolvimento positivo na realização das atividades, apresentando interesse e curiosidade em descobrir recursos para compor sua programação. No entanto, foi observada uma certa dificuldade em lidar com o código fornecido pelo professor, uma vez que estavam acostumados a criar projetos a partir do zero, sem comandos previamente dispostos na tela de programação. Isso demandou um tempo da aula para que o professor fizesse intervenções e os estudantes conseguissem desenvolver a compreensão dos comandos e criarem a partir das orientações e desafios apresentados.

A criação dos blocos "esquerda", "direita", "para cima" e "para baixo" facilitou a assimilação do conceito de direção na movimentação do personagem. No entanto, o professor também explicou o uso das coordenadas X e Y para a movimentação. Dessa forma, os alunos puderam utilizar tanto os blocos criados pelo professor como aqueles relacionados à movimentação através das coordenadas X e Y.

O cerne da proposta foi possibilitar a interseção entre os conceitos de lateralidade e os pilares do Pensamento Computacional (BRACKMANN, 2017; CODE.ORG, 2016; LIUKAS, 2015; e BBC LEARNING, 2015). No que se refere ao pilar da abstração, foi necessário utilizar um filtro e selecionar apenas os blocos essenciais para resolver a situação-problema em questão. Isso ocorreu devido ao fato de que os alunos frequentemente tentavam fazer modificações no código utilizando blocos que não estavam dentro do escopo da proposta, exigindo a intervenção do professor para enfatizar apenas os blocos relevantes para alcançar o objetivo desejado.

A decomposição também foi aplicada quando o professor fez perguntas orientadoras sobre o uso de blocos de repetição. Nesse momento, foi necessário dividir o código em partes menores, utilizando estruturas de repetição. Em relação ao reconhecimento de padrões, acredita-se que esse tenha sido um dos pilares menos enfatizados durante a proposta. No entanto, essa habilidade é fundamental quando os blocos necessários para executar uma determinada ação não são encontrados. Nesses casos, é preciso identificar os padrões existentes e relacioná-los a outros blocos disponíveis.

Por fim, o pilar do algoritmo foi enfatizado ao longo de toda a atividade, pois foi utilizado na criação da sequência de blocos que deu vida ao projeto desenvolvido. De acordo com Raabe et al. (2022), o algoritmo é um dos pilares mais importantes do Pensamento Computacional, uma vez que "assim como o produto do raciocínio lógico é

a prova, o produto do raciocínio computacional é a sequência de regras que define a transformação, comumente chamada de algoritmo".

Com relação ao uso das cartas de programação, pode-se perceber que houveram momentos de troca de informações sobre aspectos importantes para a realização de projetos no Scratch. Esses aspectos incluíam entrar na conta, salvar o projeto, escolher um ator, mover-se, escolher um fundo, utilizar condições, criar diálogos, utilizar teclas para movimento, sincronização, voz e webcam, diretrizes e compartilhamento. Dessa forma, seguindo a afirmação de Valente (2018), foi possível centralizar o ensino mais no aluno e menos no professor, envolvendo os estudantes na descoberta, investigação e resolução de problemas, proporcionando uma aprendizagem ativa.

7. Considerações Finais

A proposta buscou desenvolver as habilidades de lateralidade extraídas da matriz da escola com os pilares do PC, como a abstração, a decomposição, o reconhecimento de padrões e algoritmo. Os desafios elaborados pelos docentes possibilitaram a resolução de situações-problema dentro do ambiente de programação do Scratch, tendo que vista que por meio programação visual os alunos conseguiram sistematizar comandos como esquerda, direita, para cima e para baixo para criar o algoritmo.

No que diz respeito ao uso de condicionais, foi observada uma considerável dificuldade por parte dos alunos. Embora tenham demonstrado compreensão do emprego das condicionais durante as explicações do professor sobre o significado de cada bloco e as possibilidades de aplicação, quando se deparavam com a necessidade de realizar o processo de busca pelos blocos de forma independente ou aplicar em um novo conceito, percebia-se certa confusão. Portanto, nas próximas atividades que envolvam o uso de condicionais, será enfatizada a realização de mais exercícios que demandem o emprego dessas estruturas para solucionar situações-problema.

Acredita-se que seja a compreensão do conceito de condicional seja desafiadora para os alunos, especialmente quando estão aprendendo a programar. A ideia de tomar decisões com base em condições específicas pode ser abstrata e requer uma compreensão sólida dos conceitos lógicos subjacentes, portanto, a mediação do professor é um elemento constante que deve encorajar os alunos a experimentar diferentes combinações e testar suas soluções.

Além disso, o uso do baralho de programação foi enfatizado como uma ferramenta para o trabalho cooperativo. Os alunos foram encorajados a compartilhar ideias, colaborar uns com os outros e trocar cartas do baralho de programação. Essa dinâmica promoveu a interação entre os estudantes, o trabalho em equipe e o compartilhamento de conhecimentos.

Referências

BBC LEARNING, B. *What is computational thinking?* , 2015. Disponível em:<
<https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1>>. Acesso em:
12/05/2023.

- BRACKMANN, C. P. Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica. 2017. Tese (Doutorado em Informática na Educação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- Brasil. 2022. Normas sobre Computação na Educação Básica – Complemento à Base Nacional Comum Curricular (BNCC). Diário Oficial [da] República Federativa do Brasil. http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=235511-pceb002-22&category_slug=fevereiro-2022-pdf&Itemid=30192 Parecer Homolog
- BRASIL. Ministério da Educação. Resolução CNE/CP n. 02/2022. Brasília, 2022b. Disponível em: < http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=240741-rc_p002-22&category_slug=agosto-2022-pdf&Itemid=30192>. Acesso em: 20 jan. 2023.
- CODE.ORG. Where computer science counts. Disponível em: <https://advocacy.code.org/>. Acesso em: 12/05/2023.
- LIUKAS, L. Hello Ruby: adventures in coding. Feiwei & Friends, 2015.
- PISCHETOLA, Magda. Inclusão digital e educação: a nova cultura da sala de aula. Editora Vozes Limitada, 2019.
- RAABE, André L. A.; BRACKMANN, Christian P.; CAMPOS, Flávio R. Currículo de referência em tecnologia e computação: da educação infantil ao ensino fundamental. São Paulo: CIEB, 2018. E-book em pdf.
- RAABE, André; ZORZO, Avelino F.; BLIKSTEIN, Paulo. Computação na educação básica: fundamentos e experiências. Penso Editora, 2020.
- RESNICK, Mitchel. Jardim de Infância para a vida toda: por uma aprendizagem criativa, mão na massa e relevante para todos. Penso Editora, 2020.
- RIBEIRO, Leila; CAVALHEIRO, Simone André da Costa; FOSS, Luciana; CRUZ, Marcia Elena Jochims Kniphoff da; FRANÇA, Rozelma Soares de. Proposta para Implantação do Ensino de Computação na Educação Básica no Brasil. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 33. , 2022, Manaus. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2022 . p. 278-288. DOI: <https://doi.org/10.5753/sbie.2022.225231>.
- RODRIGUES, Amanda K. M.; GOMES, Kamily C. O.; CARNEIRO, Murillo G.. Scratchim: uma abordagem para o ensino do Pensamento Computacional para crianças de forma remota e desplugada. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 33. , 2022, Manaus. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2022 . p. 1444-1455. DOI: <https://doi.org/10.5753/sbie.2022.225797>.
- VALENTE, José Armando. A sala de aula invertida e a possibilidade do ensino personalizado: uma experiência com a graduação em midialogia. In: BACICH, Lilian; MORAN, José (org.). Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática. Porto Alegre: Penso Editora, 2018.p.26-44.
- WING, Jeanette Marie. Computational thinking. Commun. ACM 49, 3, 33–35, 2006.