ISSN: 2358-8829

INTRODUÇÃO AO PENSAMENTOCOMPUTACIONAL COM SCRATCH NOENSINO FUNDAMENTAL II: UM RELATO DE EXPERIÊNCIA

Franciele Cavalcante Bezerra de Sá [1]
Gislene da Silva Laurenço [2]
Antônio Marcos da Silva Souto [3]

INTRODUÇÃO

A expansão da cultura digital transformou as formas de comunicar, produzir e acessar informações e passou a exigir da escola a formação de sujeitos capazes de compreender, interpretar e agir criticamente sobre problemas de natureza lógica e computacional. Nesse cenário, o *pensamento computacional* configura-se como competência transversal, que ultrapassa os limites da Ciência da Computação e articula abstração, decomposição de problemas, reconhecimento de padrões e elaboração de algoritmos (WING, 2006). A Base Nacional Comum Curricular (BNCC) reforça essa perspectiva ao reconhecer a cultura digital e o *pensamento computacional* como dimensões formativas da Educação Básica (BRASIL, 2018), o que legitima a inserção de práticas de programação já nos anos finais do Ensino Fundamental.

Entre as ferramentas de iniciação à programação, o *Scratch* ocupa posição de destaque. Desenvolvido pelo Lifelong Kindergarten Group, do Massachusetts Institute of Technology (MIT), apoia-se na tradição construcionista de Papert (1980) e permite que crianças e adolescentes criem animações, jogos e histórias interativas pelo encaixe de blocos visuais, sem exigir domínio prévio de sintaxe textual (RESNICK et al., 2009). Ao reduzir essa barreira de entrada, a plataforma amplia o acesso de estudantes com perfis diversos a experiências significativas de criação digital.

A disponibilidade de uma ferramenta acessível, porém, não assegura aprendizagem por si só. Nas turmas reais convivem repertórios tecnológicos desiguais, ritmos distintos de apropriação, estudantes neurodivergentes e laboratórios com poucos equipamentos, e é nesse ponto que a mediação pedagógica se torna decisiva. Este artigo apresenta uma intervenção desenvolvida no âmbito do Programa Institucional de Bolsa de Iniciação à Docência (PIBID), subprograma de Computação do Instituto Federal de Pernambuco (IFPE), Campus Afogados da Ingazeira, em duas turmas de 7º e 8º anos de

[1] Graduando do Curso de Licenciatura em Computação do Instituto Federal de Pernambuco- IFPE, fcbs@discente.ifpe.edu.br ;
[2] Graduando do Curso de Licenciatura em Computação do Instituto Federal de Pernambuco- IFPE, gsl16@discente.ifpe.edu.br ;
[3] Mestre em Matemática pela UFPB, Licenciado em Ciências com habilitação em Matemática. Professor EBTTdo IFPE/Campus Afogados da Ingazeira, antonio.souto@afogados.ifpe.edu.br ;

ISSN: 2358-8829

uma escola da rede municipal de Afogados da Ingazeira (PE), e examina como o *Scratch* se comporta em um contexto marcado por heterogeneidade e por restrições de infraestrutura.

O objetivo geral consistiu em introduzir os fundamentos do *pensamento computacional* por meio do *Scratch,* favorecendo o protagonismo discente na criação de projetos autorais. Especificamente, buscou-se: (i) desenvolver a compreensão do conceito de algoritmo a partir de atividades desplugadas e em linguagem natural; (ii) apresentar a interface e os blocos fundamentais da plataforma; (iii) mediar a construção de projetos que articulassem sequência, repetição e variáveis; (iv) identificar as principais dificuldades conceituais e operacionais dos estudantes; e (v) propor estratégias diferenciadas capazes de atender à diversidade de perfis e ritmos, incluindo o caso de um estudante neurodivergente acompanhado durante a ação. Metodologicamente, trata-se de um relato de experiência qualitativo, organizado em três etapas (atividades desplugadas, apropriação da plataforma e projetos autorais), com dados produzidos por observação participante e diário de campo.

Os resultados indicam que a maioria dos estudantes produziu projetos que combinavam sequência, eventos e repetição; que as maiores dificuldades recaíram sobre as variáveis e sobre estruturas de comando aninhadas; e que o suporte diferenciado permitiu a um estudante neurodivergente concluir uma animação simples. Conclui-se que o *Scratch* é adequado à iniciação ao pensamento computacional nessa etapa de ensino quando associado a mediação intencional, ao passo que a limitação de recursos tecnológicos permanece como desafio a ser enfrentado institucionalmente.

2 REFERENCIAL TEÓRICO

2.1 Pensamento computacional e atividades desplugadas

O *pensamento computacional* pode ser entendido como o conjunto de processos mentais envolvidos na formulação de problemas e de suas soluções, de modo que estas possam ser executadas por um agente, humano ou máquina (WING, 2006). A literatura costuma apoiá-lo em quatro pilares: a decomposição, que fragmenta um problema complexo em partes menores; o reconhecimento de padrões; a abstração, que identifica o essencial e descarta detalhes irrelevantes; e a construção de algoritmos, isto é, sequências finitas e ordenadas de instruções que conduzem à resolução de uma tarefa.

As atividades desplugadas, realizadas sem computador e a partir de situações do cotidiano, são uma estratégia consolidada para trabalhar esses pilares na Educação Básica (BRACKMANN, 2017). Seu valor pedagógico está em permitir que a noção de sequência lógica se consolide antes de ser traduzida para a sintaxe de uma linguagem, evitando que conceito e ferramenta sejam introduzidos simultaneamente e imponham ao estudante uma sobrecarga cognitiva desnecessária.

ISSN: 2358-8829

2.2 O Scratch e a perspectiva construcionista

O *Scratch* foi concebido no MIT Media Lab sob coordenação de Mitchel Resnick e fundamenta-se na teoria construcionista de Seymour Papert, segundo a qual a aprendizagem é mais efetiva quando o sujeito se engaja na construção de artefatos significativos e compartilháveis (PAPERT, 1980; RESNICK et al., 2009). No ambiente, esses artefatos assumem a forma de jogos, animações, histórias interativas e simulações.

Do ponto de vista técnico, os comandos são organizados em categorias de blocos coloridos e encaixáveis (Movimento, Aparência, Som, Eventos, Controle, Sensores, Operadores, Variáveis e Blocos Personalizados), arrastados e conectados na área de programação (workspace). Como blocos incompatíveis simplesmente não se encaixam, eliminam-se os erros de sintaxe próprios das linguagens textuais, e o foco do estudante se desloca da memorização de comandos para a lógica de construção da solução. A ferramenta se estrutura em torno de personagens, os sprites, que atuam sobre um palco (stage) e têm seu comportamento definido por scripts.

Esse arranjo permite mobilizar, de modo progressivo, os principais conceitos computacionais: a sequência, que define a ordem de execução das instruções; os eventos, que disparam um script a partir de uma ação do usuário ou do ambiente, como o clique na bandeira verde; a repetição, por meio de laços como “repita”, “sempre” e “repita até”; as condicionais, que introduzem a tomada de decisão; as variáveis, espaços de memória nomeados pelo estudante para armazenar dados que mudam durante a execução, como pontuação ou número de vidas; os operadores, empregados em cálculos e comparações; e o paralelismo, no qual vários sprites executam scripts simultaneamente e se comunicam por mensagens (broadcast).

Para Resnick (2017), ambientes como o *Scratch* operam segundo a lógica do “jardim de infância digital” (kindergarten approach), estruturada no ciclo imaginar, criar, brincar, compartilhar e refletir, que aproxima o programar da experiência lúdica e exploratória e trata o erro como parte natural da aprendizagem. Bers (2018), por sua vez, sustenta que o contato precoce com a programação favorece não apenas competências técnicas, mas também habilidades socioemocionais, como perseverança, resolução colaborativa de problemas e pensamento sequencial, e defende os ambientes visuais como via democrática de acesso a essas competências. A plataforma conta, ainda, com uma comunidade on-line (Scratch Online Community), na qual os projetos podem ser publicados, remixados e comentados, ainda que, na experiência aqui relatada, o uso tenha se concentrado no ambiente local, em razão da infraestrutura escolar.

2.3 Aprendizagem por projetos, mediação e inclusão

Na aprendizagem por projetos (Project-Based Learning), os conceitos deixam de ser conteúdos isolados a memorizar e passam a ser meios para concretizar um produto autoral definido pelo próprio estudante. A proposta dialoga com o construcionismo e favorece a motivação intrínseca, em especial entre adolescentes em fase de consolidação de identidade e autonomia. Ela se alinha, ainda, à concepção freireana de que ensinar “não é transferir conhecimento, mas criar as possibilidades para a sua própria produção ou a sua construção” (FREIRE, 1996, p. 25).

O papel de quem media esse processo encontra respaldo na formulação de Vygotsky sobre a zona de desenvolvimento proximal:

> A aprendizagem humana pressupõe uma natureza social específica e um processo através do qual as crianças penetram na vida intelectual daqueles que as cercam. O aprendizado escolar introduz elementos básicos de uma maneira nova na experiência da criança. A distância entre o nível de desenvolvimento real, que se costuma determinar através da solução independente de problemas, e o nível de desenvolvimento potencial, determinado através da solução de problemas sob a orientação de um adulto ou em colaboração com companheiros mais capazes, define a zona de desenvolvimento proximal (VYGOTSKY, 1998, p. 112).

Essa compreensão sustenta tanto a mediação docente quanto a aprendizagem entre pares. Quando o horizonte é a inclusão, o Desenho Universal para a Aprendizagem acrescenta que os ambientes devem oferecer múltiplos meios de representação, de ação e expressão e de engajamento, para que a variabilidade dos estudantes seja tratada como regra e não como exceção (CAST, 2018). O caráter modular dos blocos do *Scratch*, que admite tentativa, erro e reorganização sem penalização, aproxima a ferramenta desse princípio, sobretudo quando associada a estratégias de ensino estruturado voltadas a estudantes neurodivergentes.

3 METODOLOGIA

A intervenção ocorreu no âmbito do PIBID, em duas turmas do Ensino Fundamental II, uma de 7º e outra de 8º ano, de uma escola da rede municipal de Afogados da Ingazeira (PE), e foi conduzida pelas bolsistas com orientação docente. Os estudantes apresentavam repertórios distintos de contato prévio com tecnologias digitais, e entre eles havia um estudante neurodivergente, cujo acompanhamento recebeu registro específico. Em razão das limitações de infraestrutura e de conectividade da escola, as atividades com computador utilizaram o *Scratch* em ambiente local, sem recurso à comunidade on-line da plataforma.

3.1 Etapa 1: atividades desplugadas

A primeira etapa foi realizada sem o uso de computadores e sem contato inicial com a sintaxe do *Scratch*, priorizando atividades práticas para introduzir os fundamentos do pensamento computacional. Foram propostas situações como a organização de cartões com etapas de uma receita, a dinâmica do colega robô, na qual os estudantes elaboravam comandos para orientar os movimentos de um colega, e a resolução de labirintos por meio de setas que representavam direções. Também foram exploradas sequências de instruções com erros, desafiando os estudantes a identificar e corrigir os comandos para alcançar o resultado esperado.

As atividades tiveram como finalidade desenvolver o raciocínio lógico, a construção de sequências ordenadas, a compreensão dos comandos e a identificação de erros. Além disso, permitiram demonstrar que a execução de uma tarefa depende da clareza e da organização das instruções, estabelecendo uma relação inicial com o

ISSN: 2358-8829

conceito de algoritmo. Essa abordagem buscou consolidar conhecimentos fundamentais antes da introdução à programação visual, evitando a sobrecarga cognitiva decorrente do contato simultâneo com um novo conceito e uma nova ferramenta (BRACKMANN, 2017).

3.2 Etapa 2: apropriação da plataforma

Na segunda etapa, os bolsistas realizaram uma apresentação guiada da interface do *Scratch*, composta por palco, sprites e categorias de blocos. Em seguida, os estudantes resolveram exercícios dirigidos que exploravam os blocos de sequência, repetição e variáveis. Os exercícios tinham objetivos curtos e verificáveis no palco, de modo que o estudante pudesse comparar o comportamento obtido com o esperado e ganhar familiaridade com o ambiente antes de assumir um projeto próprio.

3.3 Etapa 3: projetos autorais e ampliação

Na terceira etapa, os estudantes desenvolveram projetos autorais na forma de animações e histórias interativas, cujos temas, personagens e mecânicas foram definidos por eles próprios. Ao longo dessa etapa, a mediação dos bolsistas combinou atendimento individualizado e acompanhamento em pequenos grupos, com a finalidade de esclarecer dúvidas e provocar a reflexão sobre o funcionamento dos scripts, sem substituir a autoria dos estudantes.

Em um segundo momento, parte dos estudantes evoluiu dos projetos iniciais para projetos de maior porte. Essa ampliação ocorreu em ritmos distintos e foi acompanhada de mediação voltada ao planejamento em subtarefas e à construção e ao teste incrementais de cada trecho antes de sua integração ao script final, razão pela qual nem todos os estudantes alcançaram esse momento no período da intervenção.

4 RESULTADOS E DISCUSSÃO

4.1 Projetos autorais e conceitos mobilizados

A maioria dos estudantes desenvolveu projetos que articulavam, de forma coordenada, sequência, eventos e repetição e, em menor grau, variáveis. O resultado confirma a adequação do *Scratch* como ferramenta de iniciação ao pensamento computacional nesse segmento e é coerente com a tese construcionista de que a construção de artefatos próprios sustenta a compreensão de conceitos abstratos (PAPERT, 1980; RESNICK, 2017). A aprendizagem por projetos favoreceu o protagonismo: à medida que avançavam nas etapas, os estudantes demonstraram maior autonomia na definição de temas, personagens e mecânicas de suas histórias e animações, passando da execução de exercícios dirigidos à tomada de decisões de projeto.

4.2 Dificuldades conceituais e operacionais

Duas dificuldades se destacaram, a primeira foi a compreensão do conceito de variável, sobretudo em sua função de armazenar e atualizar valores de forma dinâmica durante a execução do programa. A segunda foi a organização lógica de comandos mais

complexos, em especial quando laços de repetição se combinavam a condicionais aninhadas. Ambas são recorrentes na iniciação à programação e podem ser atribuídas ao grau de abstração que exigem: a variável não possui existência visível no palco, e as estruturas aninhadas demandam que o estudante mantenha simultaneamente várias condições em mente. Tais dificuldades foram parcialmente superadas por meio de mediação individualizada, analogias concretas e decomposição dos problemas em subtarefas menores.

No caso das variáveis, recorreu-se à analogia da "caixa com etiqueta", que guarda um valor passível de alteração, associada ao monitor de variáveis do Scratch, que exibe o valor no palco em tempo real, e iniciou-se o trabalho por variáveis de finalidade concreta e motivadora para a faixa etária, como a pontuação de um jogo ou o contador de vidas. No caso dos comandos complexos, cada bloco foi construído e testado de forma incremental antes de sua integração ao script final, e utilizaram-se fluxogramas ou storyboards em papel como planejamento prévio, de modo que a transposição para o ambiente só ocorresse depois de validada a lógica. A leitura do código em voz alta ("o que este bloco faz, e depois deste?") fortaleceu, ainda, a verbalização do raciocínio sequencial.

Uma análise conjunta dessas estratégias revela um padrão didático: as que mais contribuíram foram aquelas que tornaram visível o que o código mantém implícito. O monitor de variáveis explicita a mudança de valor, e o fluxograma externaliza a lógica antes de sua codificação. A decomposição, por sua vez, é a aplicação direta de um dos pilares do pensamento computacional (WING, 2006) à própria superação das dificuldades, o que indica que esse conceito opera, na prática docente, também como método de ensino.

4.3 Heterogeneidade e aprendizagem entre pares

Observou-se participação heterogênea, com diferentes níveis de envolvimento e ritmos de apropriação, compatível com a diversidade de conhecimento prévios e de contato com tecnologias digitais. Essa variação não foi tratada como obstáculo a ser eliminado, mas como recurso: a organização em pequenos grupos com papéis rotativos permitiu que estudantes com maior domínio apoiassem os colegas, mobilizando a mediação entre pares que a perspectiva sociointeracionista reconhece como produtora de desenvolvimento. A diferenciação de desafios por complexidade, mantido o tema do projeto, preservou o sentido coletivo da tarefa e evitou que a diferença de ritmo se convertesse em exclusão. Para Bers (2018), é nesse tipo de situação, em que a resolução de problemas é compartilhada, que se desenvolvem as habilidades socioemocionais associadas à programação.

4.4 O percurso do estudante neurodivergente

Nas etapas iniciais, o estudante neurodivergente acompanhado apresentou dificuldades na organização sequencial dos comandos, manifestas na dispersão da ordem lógica dos blocos e na dificuldade de planejar previamente a animação. O suporte diferenciado consistiu na fragmentação das instruções em passos menores e sequenciais, apoiados por checklists visuais que indicavam a ordem das etapas do projeto; na oferta

de tempo adicional e de atendimento individualizado nos momentos de maior dificuldade; e na redução de estímulos concorrentes durante a atividade. Com esse apoio, o estudante evoluiu de forma progressiva até estruturar com êxito uma animação simples. Duas características da ferramenta contribuíram para esse percurso. A primeira é a eliminação do erro de sintaxe, que retira uma fonte de frustração própria das linguagens textuais. A segunda é a possibilidade de reorganizar blocos sem penalização, que transforma o erro em etapa do processo e é compatível com o ensino estruturado indicado para estudantes neurodivergentes. Em consonância com o Desenho Universal para a Aprendizagem (CAST, 2018), os checklists visuais e a fragmentação das instruções ofereceram representações alternativas da mesma tarefa, ao passo que o tempo adicional e o ambiente com menos estímulos flexibilizaram as condições de execução. Como se trata de um único caso, o resultado deve ser lido como indício do potencial da ferramenta quando associada a mediação intencional, e não como generalização.

4.5 Infraestrutura e inclusão digital

A escassez de equipamentos e a carga horária restrita do laboratório de informática limitaram a frequência e a continuidade das atividades, e é por isso que a inclusão digital aparece aqui em dupla acepção: como acesso aos equipamentos e como desenvolvimento das competências para utilizá-los de forma criativa e autônoma. O uso do *Scratch* em ambiente local atenuou a dependência de conectividade, mas teve como contrapartida o afastamento da Scratch Online Community, que poderia ampliar o compartilhamento e a autoria coletiva. O rodízio de equipamentos, acompanhado de atividades desplugadas para quem aguardava, manteve todos os estudantes em tarefas com sentido pedagógico, o que reforça a utilidade da primeira etapa também como recurso de gestão do tempo em contextos de recursos limitados.

5 CONSIDERAÇÕES FINAIS

A experiência relatada mostra que o *Scratch* é uma ferramenta adequada à introdução do pensamento computacional no Ensino Fundamental II, por conciliar acessibilidade de interface, fundamentação construcionista e possibilidade de personalização das aprendizagens. O percurso que vai das atividades desplugadas aos projetos autorais favoreceu a compreensão de sequência, eventos e repetição e o protagonismo dos estudantes, enquanto as variáveis e as estruturas aninhadas exigiram mediação mais cuidadosa. O principal aprendizado, contudo, diz respeito ao papel da mediação: a tecnologia, por si só, não garante aprendizagem, e foram a decomposição de problemas, a diferenciação de estratégias e a atenção às singularidades dos estudantes, incluindo o estudante neurodivergente, que converteram o uso da plataforma em prática educativa inclusiva. Para a comunidade científica e para a formação docente, o relato oferece um conjunto de estratégias replicáveis e evidencia o valor do PIBID como espaço de articulação entre formação inicial e escola básica. Reconhecem-se limites: trata-se de um relato qualitativo de uma experiência localizada, com acompanhamento de um único caso de neurodivergência e sem instrumentos padronizados de avaliação da aprendizagem. Persistem, por fim, os desafios de inclusão

digital em escolas com acesso limitado a recursos tecnológicos, que demandam políticas de ampliação da infraestrutura e a continuidade de programas de iniciação à docência.

AGRADECIMENTOS

Ao PIBID, ao IFPE – Campus Afogados da Ingazeira, à escola parceira e aos estudantes, que tornaram possível a experiência relatada.

REFERÊNCIAS

BERS, M. U. Coding as a playground: programming and computational thinking in the early childhood classroom. New York: Routledge, 2018.

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. Ministério da Educação. Base Nacional Comum Curricular. Brasília: MEC, 2018.

BRASIL. Conselho Nacional de Saúde. Resolução nº 510, de 7 de abril de 2016. Dispõe sobre as normas aplicáveis a pesquisas em Ciências Humanas e Sociais. Diário Oficial da União, Brasília, 24 maio 2016.

CAST. Universal Design for Learning Guidelines version 2.2. Wakefield: CAST, 2018.

FREIRE, P. Pedagogia da autonomia: saberes necessários à prática educativa. 68. ed. Rio de Janeiro: Paz e Terra, 1996.

PAPERT, S. Mindstorms: children, computers, and powerful ideas. New York: Basic Books, 1980.

RESNICK, M. Lifelong kindergarten: cultivating creativity through projects, passion, peers, and play. Cambridge, MA: MIT Press, 2017.

RESNICK, M. et al. Scratch: programming for all. Communications of the ACM, New York, v. 52, n. 11, p. 60-67, 2009.

VYGOTSKY, L. S. A formação social da mente: o desenvolvimento dos processos psicológicos superiores. 6. ed. São Paulo: Martins Fontes, 1998.

WING, J. M. Computational thinking. Communications of the ACM, New York, v. 49, n. 3, p. 33-35, 2006.