

Scratch e o Processo de Aprendizagem na Perspectiva de Vygotsky e Papert

Resumo:

Este relato descreve a experiência na disciplina de Introdução à Lógica de Programação de um curso de Licenciatura em Matemática, utilizando o Scratch como ferramenta de desenvolvimento. A disciplina visa desenvolver o pensamento lógico e algorítmico, integrando programação e matemática, promovendo o pensamento computacional. O trabalho explora as teorias do Sociointeracionismo de Vygotsky e do Construcionismo de Papert. A experiência foi estruturada em roteiros progressivos, com o professor atuando como mediador e utilizando estratégias de scaffolding. Foram desenvolvidos projetos que consistiram na criação de uma animação e de um jogo interativo. Conclui-se que a combinação das abordagens sociointeracionista e construcionista promove um ambiente de aprendizagem dinâmico, colaborativo e eficaz, preparando os alunos para enfrentar desafios complexos e desenvolver soluções criativas.

Palavras-chave: Pensamento Computacional. Scratch. Sociointeracionismo. Construcionismo. Ensino de Programação.

1 Introdução

A Este relato tem como objetivo compartilhar a experiência vivida na disciplina de Introdução à Lógica de Programação, do curso de Licenciatura em Matemática da Universidade Federal do Recôncavo da Bahia, usando como ferramenta o Scratch do Instituto de Tecnologia de Massachusetts como ambiente de desenvolvimento de programas.

A programação desempenha um papel fundamental no desenvolvimento do pensamento lógico e algorítmico, exigindo a organização de ideias e a estruturação de soluções passo a passo para resolver problemas de forma eficiente. Segundo Santos e Machado (2019), a algoritmização é essencial para a programação, pois permite a criação de códigos estruturados e organizados, facilitando sua compreensão e execução pelo computador.

Além disso, a programação se relaciona diretamente com a matemática, especialmente no desenvolvimento de algoritmos, que contribuem para a aprendizagem matemática. Como destacado

Jaylson Teixeira


Universidade Federal do Recôncavo de Bahia
Amargosa, BA – Brasil

 <https://orcid.org/0000-0003-3149-7155>

✉ jaylsont@ufrb.edu.br

Edinilson Teixeira de Oliveira

Universidade Federal do Recôncavo de Bahia
Amargosa, BA – Brasil

 <http://orcid.org/0000>

✉ email@email.com.br

Jose Allyson Divino Santos

Universidade Federal do Recôncavo de Bahia
Amargosa, BA – Brasil

 <https://orcid.org/0009-0000-7714-9928>

✉ allysondivino@aluno.ufrb.edu.br

Marielen de Almeida Silva

Universidade Federal do Recôncavo de Bahia
Amargosa, BA – Brasil

 <http://orcid.org/0000>

✉ marielensilva11@gmail.com

Ronaldo Leal dos Santos

Universidade Federal do Recôncavo de Bahia
Amargosa, BA – Brasil

 <https://orcid.org/0009-0001-9152-9836>

✉ ronaldo.rgr24@gmail.com

Recebido • 04/04/2025

Aprovado • 05/06/2025

Publicado • 08/08/2025

Comunicação Científica

por Oliveira e Gomes (2017), o objetivo dos algoritmos é formular um problema e encontrar sua solução, podendo ser amplamente utilizados na resolução de problemas matemáticos. Tudo isso é fundamental para o ensino do componente "Introdução à Lógica de Programação", que visa desenvolver habilidades essenciais no processo de aprendizado.

A Base Nacional Comum Curricular (BNCC) incentiva o ensino do pensamento computacional na educação básica, permitindo que um professor de Matemática bem preparado integre a programação ao ensino de funções, estatística e modelagem matemática. De acordo com a BNCC, o pensamento computacional está alinhado a essa abordagem, destacando a aplicação de métodos e recursos matemáticos, incluindo tecnologias digitais, para representar e solucionar problemas do cotidiano, do contexto social e de diferentes áreas do conhecimento, avaliando a eficácia das estratégias e dos resultados (BRASIL, 2016).

Este trabalho explora a relação entre o Sociointeracionismo de Vygotsky e o Construcionismo de Papert, ambos baseados na ideia de que a aprendizagem ocorre por meio da interação e da experiência ativa. Vygotsky destaca a importância do ambiente social e da mediação de indivíduos mais experientes no desenvolvimento cognitivo, enquanto Papert, inspirado no Construtivismo de Piaget, enfatiza que o conhecimento se consolida quando o aluno constrói algo significativo. Ambos reconhecem que o aprendizado não é um processo isolado, mas sim colaborativo, sendo impulsionado tanto pela interação social quanto pela experimentação ativa, especialmente quando apoiado pelo uso de tecnologias e ferramentas.

2 Aportes teóricos

Nesta seção serão exploradas as teorias do Sociointeracionismo de Vygotsky e do Construcionismo de Papert

2.1 Teoria Sóciointeracionista

A Teoria Sócio-Interacionista, proposta por Lev Vygotsky (1896-1934), evidencia que a aprendizagem ocorre por meio da interação social. Ao contrário das teorias que dizem que o aprendizado é um fenômeno exclusivamente centrado no indivíduo, Vygotsky argumenta que o conhecimento é construído em grupo, através da mediação de pessoas mais experientes e do uso de ferramentas culturais, como a linguagem (Oliveira, 1995). A linguagem, inicialmente considerada como linguagem falada, pode ser estendida a outros tipos como a linguagem escrita, a linguagem dos surdos, a linguagem audiovisual etc (Teixeira, 2016).

Um dos conceitos centrais dessa teoria é a **mediação**, que ocorre quando um indivíduo (como um professor ou colega mais experiente) auxilia o aprendiz na construção do conhecimento, utilizando diferentes recursos para facilitar a compreensão. Esse suporte se torna essencial, especialmente quando o aluno ainda não possui autonomia para resolver determinadas tarefas de maneira independente (Oliveira, 1995).

Outro conceito fundamental é a Zona de Desenvolvimento Proximal (ZDP), definida como a diferença entre o que o aluno já consegue realizar sozinho e o que ele pode aprender com auxílio.

Segundo Vygotsky (1984), o ensino eficaz deve focar nessa zona, proporcionando desafios que estejam além da capacidade atual do aluno, mas que possam ser superados com apoio. Esse suporte, quando bem estruturado, possibilita que o estudante avance progressivamente até alcançar a autonomia (Oliveira, 1995). Este auxílio do mais apto ao menos apto também é chamado de scaffolding.

A tradução literal de scaffolding seria andaime, que auxilia os trabalhadores a chegar em lugares mais altos durante uma construção ou reforma de prédios. Mas no contexto da educação, scaffolding se refere a um suporte temporário que auxilia o aluno a superar etapas do aprendizado, sendo retirado aos poucos conforme o estudante adquire maior domínio sobre o conteúdo ensinado. Neste processo o estudante participa ativamente da sua aprendizagem, garantindo que ele adquira confiança para solucionar problemas até se tornar independente na realização de uma tarefa (Gonulal; Loewen, 2018). Este auxílio chamado de scaffolding pode vir não apenas do professor mas também de outros estudantes que estejam trabalhando em pares ou auxiliando um colega (Gonulal; Loewen, 2018).

A utilização dessa abordagem é especialmente relevante na introdução de conceitos computacionais a alunos que possuem pouca familiaridade com a área. A mediação do professor e o uso de estratégias de suporte permitem que esses estudantes evoluam progressivamente nos fundamentos da programação, estimulando uma aprendizagem mais significativa e alinhada às diretrizes da BNCC, que priorizam o uso da tecnologia como uma ferramenta essencial para o desenvolvimento do pensamento lógico e matemático.

2.2 Construcionismo de Papert

O termo construcionismo foi criado por Seymour Papert, com base no construtivismo de Jean Piaget. Uma das principais características dessa teoria é que o aprendizado ocorre por meio da ação do fazer, colocando o aluno como protagonista do processo de aprendizagem (Teixeira, 2016).

Papert foi um dos primeiros pensadores a perceber o impacto transformador que os computadores e suas tecnologias associadas teriam na sociedade. Enquanto, na época, os computadores pessoais eram vistos principalmente como ferramentas para jogos, lazer, compras, transações bancárias e comunicação, Papert (1985) enxergava neles um grande potencial educativo, refletindo sobre sua aplicação no ensino. Anos depois, suas previsões se confirmaram, evidenciadas pelas profundas mudanças na forma como as pessoas pensam, trabalham, se comunicam, se divertem e aprendem.

Em seus estudos, Papert (1985) defendia que o computador não deveria ser utilizado de forma passiva, apenas transmitindo conhecimento ao aluno. Pelo contrário, ele acreditava que o estudante deveria "ensinar" o computador, programando-o. Segundo Papert, programar significa aprender uma linguagem compreendida pelo computador para estabelecer comunicação com ele.

Papert mudou a visão sobre o uso dos computadores na educação, defendendo que os professores deveriam utilizá-los como ferramentas para apoiar a aprendizagem. Nesse contexto, o

educador poderia promover um ambiente cooperativo, onde cada participante contribui com seu conhecimento, e inclusive o professor atuaria como aprendiz. Esse modelo valoriza a participação ativa no ensino e na aprendizagem, permitindo que se aprenda tanto sobre o processo quanto por meio dele (Teixeira, 2016).

Papert também mencionava que, ao construir modelos, os alunos se interessariam pelo aprendizado e aprenderiam mesmo com pouca ou nenhuma orientação do professor. Segundo ele, o construcionismo promove um ambiente de aprendizagem baseado em micromundos, nos quais as pessoas aprendem ativamente por meio da experimentação e exploração. A programação de computadores seria um meio para criar estes micromundos. Nesse contexto, Papert também contribuiu para a criação da linguagem LOGO no Instituto de Tecnologia de Massachusetts (MIT), o mesmo local que, anos depois, desenvolveria a linguagem Scratch, ambas voltadas ao desenvolvimento do pensamento computacional e à aprendizagem ativa.

3 Experiência vivida

O componente curricular de Introdução à Lógica de Programação na UFRB faz parte da Licenciatura em Matemática e pode ser desafiador para quem não tem familiaridade com computação. Para tornar o aprendizado mais acessível, o professor usa roteiros com dificuldades progressivas. Um exemplo disso é o algoritmo do contador, que no início foi explicado detalhadamente, mas depois passou a ser apenas solicitado nos demais roteiros. A colaboração entre os alunos, especialmente os que já têm mais experiência, se torna essencial, evidenciando o scaffolding, em que um estudante ajuda o outro antes de recorrer ao professor.

Os roteiros orientam a produção de um ou dois programas por semana, permitindo que os alunos compreendam o processo enquanto adquirem maior autonomia na programação. O professor incentiva que, após a implementação do programa conforme o roteiro, os alunos façam modificações, indo além das instruções iniciais e personalizando seu trabalho. Papert afirmava que “o construcionismo privilegia os caminhos individuais em vez do melhor caminho”, e essa ideia se refletia nas aulas. Após seguir os roteiros propostos, os alunos traçavam seus próprios caminhos e buscavam formas de aprimorar o material apresentado. Segundo Papert & Harel, (1991), “o conhecimento acontece no processo de construção, aprendendo enquanto se faz”, e essa abordagem era evidente no curso.

Ao final do curso são solicitados dois programas que devem ser criados sem roteiro para nos conduzir. O primeiro programa é uma animação com uma história numa estrutura linear com as cenas acontecendo sequencialmente até o fim do programa. O outro programa é um jogo com interação com o usuário de modo que as sequências das cenas dependem das opções selecionadas pelo usuário ou pelos resultados dos desafios encontrados nas cenas de jogos.

Para a criação da animação houve uma votação entre os alunos para a escolha do tema para toda a sala. Com o tema definido, criamos um storyboard representando as cenas que seriam programadas e o que aconteceria em cada cena. Para a criação do programa final o tema é livre

escolhido por cada dupla de trabalho. Assim como na animação, criou-se um storyboard, antes de partir para a programação final.

Nessa etapa final, os alunos precisavam usar a imaginação e a criatividade para a construção dos projetos. O desenvolvimento não se restringia a uma única aula, mas se estendia por 2 semanas, no caso da animação, e por 3 semanas, no caso do programa final. Durante esse período, os estudantes escolhiam o tema, utilizavam os roteiros já trabalhados como base e iniciavam a construção do projeto. Como afirma Teixeira (2016).

O construcionismo de Papert propõe que o aluno vá formando seus próprios conceitos na medida em que constrói um artefato externo que se possa mostrar. Nesta prática, o aluno levará algumas aulas pensando e repensando seu projeto, discutindo com colegas, mostrando o que já fez e o que pretende fazer (Teixeira, 2016, p.94).

Neste processo de revisão do próprio trabalho os alunos revêem seus conhecimentos e corrige hipóteses através dos erros na programação com Scratch e na conversa com seus pares e professores, construindo e reconstruindo o seu conhecimento.

O professor estimulava os alunos com um ponto extra para que no final do roteiro sempre mudasse um pouquinho o programa, isso fez com que os alunos sempre buscassem aquele extra, usando a criatividade e assim deixando os alunos mais autônomos, o que contribuiu para os projetos finais, os quais não havia roteiro e dependíamos da nossa criatividade.

A autonomia conquistada ao longo do curso foi essencial no momento final. Quando o professor anunciou que teríamos que criar um projeto sem roteiro, apenas com um tema escolhido pela turma, muitos ficaram receosos. No entanto, como já estávamos acostumados a fazer pequenas alterações nos programas dos roteiros, a tarefa se tornou mais natural.

No nosso caso, a sala escolheu o tema das Olimpíadas para o projeto de animação. No projeto final, o professor lançou um novo desafio de criar um jogo contendo pelo menos duas mecânicas de jogos, com tema livre. Na animação sentimos um pouco de insegurança, pois até então sempre tínhamos um roteiro para seguir. Já no projeto final do jogo tivemos mais tempo e, com a experiência da animação, o processo se tornou menos intimidador e mais acessível.

O tema do nosso projeto final foi a história de dois irmãos, em que um tentava salvar o outro de um demônio extremamente poderoso. Um dos principais desafios foi a criação de um mundo em que o mapa se movimentava em vez do próprio personagem, além de garantir que o boneco caminhasse apenas pela estrada, mesmo com o controle livre do jogador.

O trabalho foi feito em dupla, o que permitiu a troca de conhecimentos e a ajuda mútua, especialmente nas partes em que um tinha mais dificuldade e o outro podia contribuir. Essa dinâmica está diretamente ligada ao conceito de Zona de Desenvolvimento Proximal (ZDP) e a técnica de scaffolding, que destaca como a interação social é fundamental para o aprendizado. Como explica Vygotsky (1978), primeiro aprendemos com os outros, e só depois consolidamos esse conhecimento individualmente. Nesse sentido, a parceria foi essencial para o avanço dos projetos.

Além disso, como as ZDPs são construídas de forma colaborativa, algumas duplas costumavam me procurar para tirar dúvidas e pedir ajuda em partes específicas de seus projetos.

Esse processo reforça a ideia sociointeracionista sobre o papel do ambiente social e cultural na construção do conhecimento (Cole & Wertsch, 1996). Assim, a experiência do projeto demonstrou, na prática, como a aprendizagem colaborativa contribui para o crescimento tanto individual quanto coletivo.

4 Conclusão

O sociointeracionismo e o construcionismo se complementam, pois ambos destacam a importância da interação social e da construção ativa do conhecimento no processo de aprendizagem. O sociointeracionismo, com a ideia de mediação (scaffolding), mostra como o apoio do professor e dos colegas é fundamental para que os alunos avancem no aprendizado, especialmente na programação em Scratch. Já o construcionismo reforça a ideia de que o aprendizado se torna mais eficaz quando os alunos têm a oportunidade de criar e personalizar seus próprios programas, tornando o processo de ensino mais significativo e engajador.

Na experiência do componente curricular de Introdução a Lógica de Programação foi possível reconhecer estas duas abordagens no processo didático vivenciado. No ensino de Introdução a Lógica de Programação, a Zona de Desenvolvimento Proximal se manifesta na mediação do professor, na interação entre os alunos e na evolução gradual das atividades. A interação entre os estudantes também fortalece a aprendizagem, pois aqueles com maior familiaridade com o computador auxiliam os colegas, favorecendo o desenvolvimento coletivo. Esse processo se intensifica nos projetos finais, como a animação e o jogo, quando os alunos utilizam os conceitos aprendidos com mais autonomia.

O conhecimento é socializado, isso ocorre por meio da interação entre os indivíduos, na troca de ideias, no diálogo e na colaboração. A mediação do professor desempenha um papel fundamental ao guiar os alunos no processo de aprendizagem, fornecendo suporte por meio de explicações, questionamentos e desafios progressivos. Além disso, o aprendizado colaborativo entre os próprios estudantes contribui para a socialização do conhecimento. Dessa forma, observou-se na prática, as abordagens sociointeracionista e o construcionista que se complementam, contribuindo para um ensino mais dinâmico, colaborativo e eficaz.

Referências

COLE, M., & Wertsch, J. V. (1996). Beyond the Individual-Social Antinomy in Discussions of Piaget and Vygotsky. *Human Development*, 39(5), 250-256.

EDUCAÇÃO É A BASE.[s.l: s.n.]. Disponível em: .Aces so em: fev. 2025.

GONULAL,Talip; LOEWEN, Shawn. Scaffolding technique. **The TESOL encyclopedia of English language teaching**, [s. l.], p. 1–5, 2018.

OLIVEIRA, E. C.; GOMES, M. L. Desenvolvendo o pensamento computacional por meio da resolução de problemas matemáticos. *Anais do XVI Simpósio Brasileiro de Informática na Educação*, Recife, 2017.

OLIVEIRA, Marta Kohl de. Vygotsky: aprendizado e desenvolvimento: um processo sócio histórico. [s. l.], 1995.

PAPERT, S. A máquina das crianças: repensando a escola na era da informática. Porto Alegre, RS: Artes Médicas, 2008.

PAPERT, S. LOGO: computadores e educação. São Paulo, SP: Brasiliense, 1985.

PAPERT, S.; HAREL, I. Constructionism. New Jersey, Norwood: Ablex Publishing, 1991. ROGOFF, B. (1990). Apprenticeship in Thinking: Cognitive Development in Social Context. New York: Oxford University Press.

SANTOS, L. M. dos; MACHADO, G. F. Introdução à Programação: Conceitos, Algoritmos e Linguagens. Elsevier, 2019

TEIXEIRA, Jaylson. **Contribuições para o ensino de programação de computadores a futuros professores de matemática**. 2016. PhD Thesis– Universidade do Minho (Portugal), 2016.

VYGOTSKY, L. S. (1978). Mind in Society: The Development of Higher Psychological Processes. Cambridge, MA: Harvard University Press