

# Projeto CodifiKids: ensinando computação em escolas públicas

MARQUES, J.<sup>1</sup>, MATSUMOTO, A.<sup>1</sup>, NAZAR, J. G. M.<sup>1</sup>, ROSENDO, I.<sup>1</sup>, FARIA, I. B.<sup>1</sup>, YASUO, E. H. H.<sup>1</sup>, CHAVES, P. G. S.<sup>1</sup>, SIMOES, E. D. V.<sup>1</sup>

<sup>1</sup>Instituto de Ciências Matemáticas e de Computação – Universidade de São paulo (USP)

Avenida Trabalhador São-carlense, 400 - Centro CEP: 13566-590 - São Carlos - SP

{juanmarquesjordao, andrematsumoto02, jotagenazar, isaach, isabela\_bfarias, haradayasuo2010, pietragull}@usp.br, simoes@icmc.usp.br

**Resumo.** O projeto CodifiKids visa ensinar conceitos básicos de programação para crianças do ensino fundamental em escolas públicas de São Carlos, SP, utilizando a linguagem de programação Scratch. O projeto utiliza atividades lúdicas e práticas, como origami, jogos de tabuleiro e torres de lápis, para introduzir conceitos de algoritmos, variáveis, condicionais e laços de repetição. Essas atividades são adaptadas para refletir o cotidiano das crianças e são complementadas pelo ambiente visual e intuitivo do Scratch. O projeto demonstrou aumentar o engajamento dos alunos e melhorar a compreensão dos conceitos de programação. Essa abordagem visa contribuir para melhorar o ensino de programação nas escolas públicas e envolver mais alunos na era digital.

**Abstract.** The CodifiKids project aims to teach basic programming concepts to elementary school children in public schools in São Carlos, SP, using the Scratch programming language. The project employs playful and hands-on activities, such as origami, board games, and pencil tower building, to introduce concepts such as algorithms, variables, conditionals, and loops. These activities are adapted to reflect the children's daily lives and are complemented by Scratch's visual and intuitive environment. The project has demonstrated increased student engagement and improved understanding of programming concepts. This approach seeks to contribute to enhancing programming education in public schools and to involve more students in the digital era.

## 1. Introdução

O uso crescente de dispositivos eletrônicos nas escolas evidencia a presença marcante da tecnologia no cotidiano, tornando-se essencial à medida que crianças estão cada vez mais imersas em ambientes digitais [Bianchessi, 2020]. Nesse contexto, o ensino de Ciência da Computação na educação básica emerge como uma necessidade. No entanto, no Brasil, essa prática ainda é pouco difundida nas escolas públicas, resultando no desconhecimento do tema e em um possível desinteresse dos alunos, o que pode contribuir para a baixa adesão a carreiras na área tecnológica. Iniciativas internacionais, como o *Model Curriculum for K-12 Computer Science* dos Estados Unidos, buscam padronizar o ensino de computação para todas as idades e esferas sociais [SIGCSE, 2014].

No Brasil, o conhecimento em computação como ciência permanece restrito, abrangendo principalmente alunos que optam por cursos específicos na área [Gomes, 2010]. Contudo, é

fundamental que todos compreendam ao menos os conceitos básicos de computação, especialmente diante da crescente demanda por profissionais de tecnologia. Segundo dados do Banco Nacional de Empregos (BNE), o número de vagas na área de Tecnologia da Informação (TI) cresceu 79,6% entre janeiro e outubro de 2023, em comparação com o mesmo período do ano anterior, evidenciando um crescimento significativo e uma carência de mão de obra qualificada para suprir essas vagas [BNE, 2023].

Reconhecendo a relevância desse tema, o Instituto de Ciências Matemáticas e de Computação (ICMC-USP) desenvolveu um curso de extensão com o objetivo de promover o ensino de lógica de programação para estudantes do ensino fundamental da rede pública de São Carlos, SP. Destinado a alunos do 5º e 6º anos, o curso adotou uma abordagem metodológica inspirada no método Paulo Freire, permitindo a aproximação de conceitos de computação por meio de dinâmicas simples e conectadas à realidade dos alunos [Weyh, 2020]. A aprendizagem prática foi facilitada pelo uso da plataforma Scratch, que oferece um ambiente lúdico e interativo para o desenvolvimento de habilidades em programação [John, 2010].

## 2. Metodologia

Os resultados apresentados neste artigo foram coletados ao longo de oito anos, abrangendo 16 edições de cursos de extensão realizados com 462 alunos do 5º e 6º ano do ensino fundamental, provenientes de 17 escolas públicas diferentes. O curso consiste em seis encontros semanais, cada um com duração de 1h10min, totalizando 7 horas de conteúdo. O projeto foi desenvolvido pelo Programa de Educação Tutorial (PET Computação-USP), que forneceu todo o apoio necessário, inclusive suporte financeiro.

As aulas seguem o cronograma apresentado na Figura 1:

Cronograma	Aula
1º	Introdução ao Computador
2º	Algoritmos
3º	Condicional
4º	Laços
5º	Variáveis
6º	Encerramento

Figura 1. Cronograma das aulas.

Durante a realização do curso, adotou-se a metodologia de ensino de Paulo Freire, que visa alcançar resultados significativos em um curto espaço de tempo [Biazzo, 2015]. Essa abordagem destaca-se por utilizar atividades e materiais intimamente relacionados à realidade dos alunos, tornando o aprendizado mais envolvente e prazeroso. No ensino de

computação para crianças, essa metodologia mostra-se especialmente eficaz [Feitosa, Sonia Couto Souza, 1999]. Todas as aulas e tópicos abordados são inicialmente apresentados por meio de atividades lúdicas, simples e acessíveis, adaptadas às diferentes realidades sociais dos alunos. Essa abordagem permite que as crianças se sintam mais engajadas e motivadas desde o início do processo de aprendizagem [Machado, K., 2024].

Após as dinâmicas, as crianças são conduzidas ao ambiente virtual, no qual têm a oportunidade de desenvolver o mesmo conteúdo utilizando a plataforma Scratch. Essa transição suave entre atividades presenciais e virtuais facilita a assimilação dos conceitos aprendidos, promovendo uma experiência de aprendizagem mais completa e eficiente [Bordin, A., 2023].

O curso conta com monitores voluntários, estudantes da Universidade de São Paulo, campus de São Carlos. A presença dos monitores proporciona uma atenção mais individualizada aos alunos, permitindo uma abordagem mais próxima e eficaz — especialmente considerando que crianças de 11 a 13 anos tendem a apresentar melhor desempenho quando se sentem ouvidas e confiantes, em um ambiente mais acolhedor e seguro [Cohen, 1994].

Antes de iniciarem suas atividades, os monitores passam por uma reunião inicial de capacitação, abordando tanto aspectos de interação com crianças quanto metodologias de ensino adequadas para esse público [Silva, 2023]. Como analisado por Ferreira e Souza (2022), ao final de cada aula é realizada uma reunião de feedback, na qual os monitores compartilham experiências, discutem desafios enfrentados e destacam aspectos positivos. Com base nesse retorno, são planejadas capacitações específicas para a aula seguinte, visando aprimorar as abordagens pedagógicas e maximizar os resultados de aprendizagem [Oliveira e Mendes, 2021]. Essa prática contínua de feedback e capacitação personalizada contribui significativamente para o aprimoramento da atuação dos monitores, garantindo suporte pedagógico de qualidade aos alunos [Pereira, 2020], promovendo um ambiente de aprendizagem mais enriquecedor e eficaz [Rodrigues, 2019].

Para tornar o curso mais atrativo, todas as atividades desenvolvidas no Scratch são baseadas em jogos, visando ao engajamento dos alunos. Eles têm liberdade para pensar em suas próprias soluções ou seguir as instruções dos monitores, promovendo tanto a criatividade quanto o aprendizado guiado. Além disso, os alunos são incentivados a explorar o código e a plataforma do Scratch após a conclusão do conteúdo da aula, estimulando a autonomia e a descoberta individual [Bransford, 1999]. Como forma de reconhecimento pelo esforço e progresso dos participantes, é oferecido um certificado ao término do curso, valorizando o tempo e o empenho dedicados pelos alunos.

## **2.1. Dinâmicas**

As dinâmicas desempenham um papel fundamental na execução das aulas, sendo o primeiro momento em que os alunos têm contato com o mundo da computação de forma “desplugada”, conforme explorado por Garcez (2024). Cada dinâmica é cuidadosamente planejada para abordar o tema da aula de maneira simples e lúdica, proporcionando uma experiência ao mesmo tempo divertida e educativa para as crianças [Vieira, 2013].

As dinâmicas e seus respectivos conceitos foram organizados conforme cada aula, conforme listado abaixo:

1. Aula um - Introdução ao Computador: as dinâmicas desta primeira aula visam familiarizar os alunos com o uso de computadores ou notebooks. Os alunos são convidados a explorar jogos dentro do ambiente do Scratch, desenvolvidos especialmente para esta atividade. Os jogos incluem:
  - a. Digitação: neste jogo, o aluno interage com o personagem "Petico" respondendo suas perguntas por meio de digitação. O objetivo é promover a prática da digitação de forma lúdica e interativa.
  - b. Uso do Mouse: neste jogo, os alunos são desafiados a clicar na maior quantidade de tomates que aparecem na tela no menor tempo possível. Essa atividade visa familiarizá-los com o uso do mouse.
2. Aula dois - Algoritmos: nesta segunda aula, o foco está em atividades que envolvem a aplicação de algoritmos, muitas vezes de forma sutil, sem que as crianças percebam de imediato. A primeira dinâmica proposta é a construção de um origami (Figura 2), uma atividade simples que segue uma sequência lógica de passos — ou seja, um algoritmo de montagem. Ao apresentar o origami dessa forma, os alunos conseguem abstrair esse conceito fundamental da computação de maneira prática e tangível.

A segunda dinâmica é o jogo "Mestre Mandou". Nele, os monitores realizam ações simples, como passar manteiga em um pão. No entanto, para que a ação seja executada, os alunos devem escrever um pseudocódigo detalhado que será seguido literalmente pelo monitor. A atividade só é concluída com sucesso quando os alunos conseguem descrever com precisão cada etapa necessária, reforçando a importância de instruções claras e estruturadas na lógica computacional.



**Figura 2. Dinâmica do origami.**

3. Aula três - Condicional: nesta aula, é proposta a atividade de um jogo de tabuleiro (Figura 3a), cujo objetivo é percorrer todas as casas até alcançar o destino final. A dinâmica é simples: a cada rodada, os jogadores avançam uma casa. Quando param em uma casa de cor específica, devem pegar uma carta correspondente àquela cor (Figura 3b) e seguir as instruções nela contidas. Cada carta apresenta condições que devem ser cumpridas, oferecendo duas opções: "se" ou "senão", como o funcionamento das condicionais em algoritmos.



Figura 3a. Dinâmica do tabuleiro.

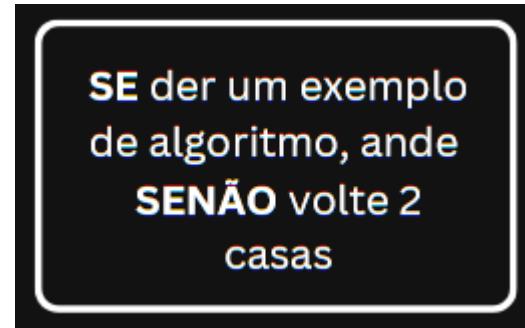


Figura 3b. Exemplo de carta.

4. Aula quatro - Laços de repetição: nesta dinâmica, os alunos recebem um conjunto de lápis e uma folha contendo um algoritmo previamente criado na plataforma Scratch (Figura 4). O desafio consiste em compreender o código apresentado e replicá-lo de forma autônoma. Durante a atividade, os monitores apenas intervêm quando realmente necessário, incentivando a autonomia dos alunos. Ao final da execução do algoritmo, os alunos devem ter construído uma torre de lápis, conforme especificado no código. A estrutura principal do programa é baseada em laços de repetição, com ênfase no comando "enquanto" (while), permitindo que os estudantes entendam, na prática, como funcionam as repetições em um algoritmo.

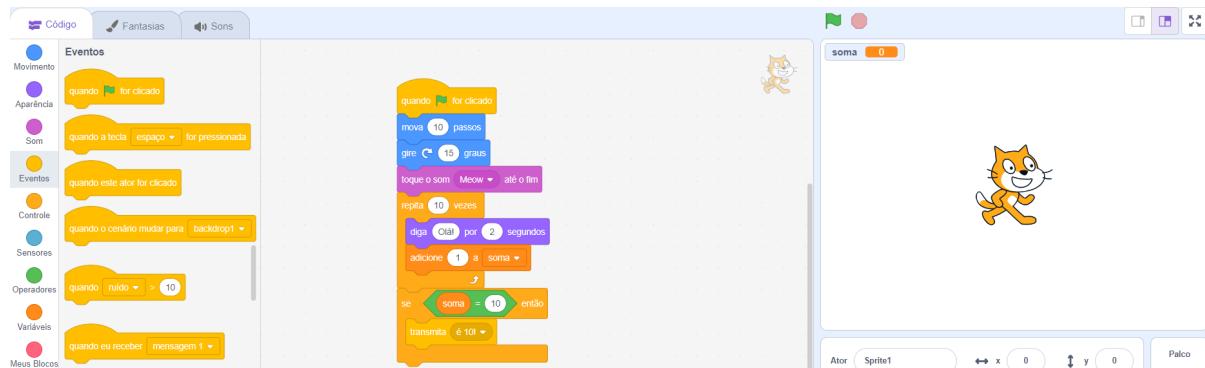


Figura 4. Trecho do algoritmo da torre de lápis.

5. Aula cinco - Variáveis: os alunos trabalham com o objetivo de memorizar as palavras ditas e repetí-las em sequência aumentando a dificuldade com o tempo até que surja a necessidade de armazenar essas informações em pedaços de papel como se fossem variáveis armazenadas na memória do computador.
6. Aula seis - Revisão: nesta aula, é selecionado um jogo que a maioria dos alunos aprecia. A partir disso, busca-se exemplos do jogo relacionados aos conceitos abordados no curso. Por exemplo, a lógica de abrir e fechar uma porta em uma casa no jogo "Minecraft" ou a construção de uma base, condicionada à disponibilidade de recursos, no jogo "Roblox".

### 3. Scratch

O scratch é uma linguagem de programação projetada para ensinar de forma acessível e divertida. No lugar de digitar linhas de código e se preocupar com uma sintaxe, os usuários fazem seu código por meio de blocos gráficos e uma interface simplificada de arrastar e soltar. Cada bloco representa uma ação e comandos, sendo separados em grupo como por exemplo, movimento, controle e som. Cada conjunto de bloco possui a mesma cor e formato, permitindo uma melhor visualização das opções (Figura 5). Dessa forma, o aluno só precisa se concentrar em desenvolver sua lógica de programação e seu algoritmo e não se preocupa com a sintaxe da linguagem de programação e seus eventuais problemas. Isto é ideal para os primeiros contatos com a programação, seja para crianças ou para ingressantes em áreas de computação [Brenda, F. 2024].



**Figura 5. Exemplo de código no Scratch.**

É notável durante as aulas que a capacidade de pensamento lúdico e criativo proporciona uma vantagem significativa ao utilizar o Scratch. Diante do cenário atual no Brasil, é importante apoiar a adoção dessa ferramenta em todas as etapas da educação nas quais são introduzidos os conceitos de computação, o que foi demonstrado por [Fagerlund, 2020].

### 4. Resultados

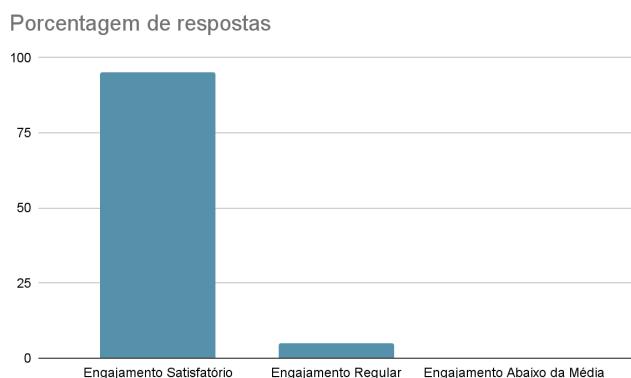
Observou-se que, ao incorporar vivências do cotidiano dos alunos, houve um aumento significativo no engajamento e no interesse pelo conteúdo das aulas. Ao longo dos anos,

constatou-se que métodos de ensino mais estáticos, como apresentações em slides, não eram bem recebidos pelas crianças, que se distraíam com facilidade. Para contornar esse problema, a turma passou a ser subdividida em pequenos grupos, cada um acompanhado por um monitor designado, com o objetivo de promover um acompanhamento mais individualizado e eficaz do processo de aprendizagem.

Como esperado, os grupos passaram a trabalhar de forma colaborativa, favorecendo questionamentos e discussões entre os alunos e o monitor. Após a formulação de um conceito lógico durante as dinâmicas, este era posteriormente implementado na plataforma Scratch, conforme observado por Martins (2015).

Os resultados alcançados foram satisfatórios, evidenciando a eficácia das dinâmicas aplicadas e sua contribuição para uma assimilação mais significativa dos conteúdos. A combinação entre os conceitos abordados nas atividades e sua aplicação prática no Scratch revelou-se essencial para o entendimento sólido da lógica computacional. O envolvimento dos monitores teve papel central nesse processo: por meio de sua atuação ativa, foi possível desenvolver novas abordagens, o que contribuiu ainda mais para a melhoria do ensino e da compreensão dos alunos.

Para avaliar a assimilação dos conteúdos, adotou-se uma métrica baseada na observação do desempenho dos estudantes durante as atividades propostas. Considera-se um desempenho satisfatório quando o aluno conclui todas as atividades dentro do tempo disponível, demonstrando uma compreensão eficaz dos conceitos. Um desempenho regular é atribuído quando o aluno consegue finalizar apenas uma das etapas — seja a dinâmica ou o projeto no Scratch —, indicando uma assimilação parcial do conteúdo. Por fim, um desempenho abaixo da média é identificado quando o aluno não consegue concluir nem as dinâmicas nem o projeto no Scratch, o que sugere dificuldades na compreensão do material trabalhado.



**Figura 6: Média do engajamento dos alunos como relatado pelos monitores.**

Considerando as respostas dos *feedbacks* dos monitores (Figura 6), 95% dos monitores relataram um desempenho satisfatório dos alunos, uma vez que não houve alunos em seus cursos que não conseguiram realizar as atividades propostas durante as aulas. Apenas 5% dos monitores relataram um desempenho regular. Assim, pode-se concluir que o curso

atingiu seu objetivo no quesito de possibilitar um ensino adequado de computação para as crianças.

## **5. Considerações Finais**

A iniciativa permitiu que os alunos explorassem a construção de algoritmos aplicados à resolução de problemas do cotidiano, além de compreenderem estruturas básicas de código e desenvolverem o pensamento computacional. Por meio dessas atividades, os estudantes foram incentivados a considerar a área da computação como uma opção viável para seu futuro acadêmico e profissional. Também foi destacada a importância do conhecimento em computação na atualidade, equiparando-o a outras áreas das ciências tradicionais. Dessa forma, o projeto não apenas ofereceu uma introdução significativa à computação, como também ampliou a percepção dos alunos sobre a relevância e as possibilidades dessa área em um contexto mais amplo.

Uma das grandes conquistas nesse cenário foi a inclusão do ensino de computação no currículo oficial do Estado de São Paulo, com a adoção do Scratch como ferramenta pedagógica. Essa iniciativa reforça a necessidade de preparar os alunos desde cedo para um mundo cada vez mais digital, garantindo o desenvolvimento de competências fundamentais para o século XXI. Contudo, tão importante quanto oferecer esse conhecimento é educar os alunos sobre o uso consciente da tecnologia. A ausência de diretrizes claras sobre o controle e a utilização adequada de dispositivos móveis pode resultar em uso excessivo e prejudicial, especialmente entre crianças. Assim, além do ensino de programação, é essencial abordar o equilíbrio no tempo de tela e a construção de uma relação saudável com a tecnologia.

Outro importante legado deste projeto, ao longo de seus anos de execução, é a consolidação da ideia de que a computação deve ser considerada uma disciplina essencial no ensino fundamental, em consonância com as diretrizes da Base Nacional Comum Curricular (BNCC). Espera-se que a metodologia proposta contribua para o aprimoramento do ensino básico de programação nas escolas públicas, garantindo que um número cada vez maior de alunos tenha acesso qualificado e consciente à era digital.

O projeto possui perspectivas promissoras de expansão, por meio do Programa de Educação Tutorial (PET), inicialmente na Universidade de São Paulo. A partir de 2025, todos os grupos PET da USP terão como tema a ampliação do acesso de escolas públicas a materiais didáticos com enfoque tecnológico. Com a implementação desta nova fase, será possível convidar outros grupos PET de universidades federais para participar da iniciativa, visando a disseminação da metodologia em nível nacional.

Como continuidade deste trabalho no ensino fundamental, encontra-se em desenvolvimento um novo projeto voltado ao ensino médio de escolas públicas. Nessa nova etapa, a linguagem de programação adotada será o Python, mais robusta e amplamente utilizada no mercado de trabalho. Essa mudança visa não apenas aprofundar os conhecimentos dos alunos em computação, mas também prepará-los para inserção profissional de maneira mais competitiva.

Ao final do ciclo básico, espera-se que os alunos se tornem mais confiantes, críticos e preparados para se destacar em um cenário cada vez mais tecnológico e globalizado. Assim, este projeto representa um avanço significativo na democratização do acesso ao conhecimento computacional e na formação de cidadãos mais conscientes, autônomos e aptos a enfrentar os desafios e oportunidades do século XXI.

## Referências

BIANCHESI, Cleber. Nomofobia e a dependência tecnológica do estudante. Bagai, 2020. p. 13–15.

SIGCSE. Proceedings of the 45th ACM Technical Symposium on Computer Science Education. 2014. p. 493–498.

COHEN, E. G. Restructuring the classroom: Conditions for productive small groups. *Review of Educational Research*, v. 64, p. 1–35, 1994.

FEITOSA, Sonia Couto Souza. Método Paulo Freire - Princípios e Práticas de uma Concepção Popular de Educação. 1999. Disponível em: <https://acervo.paulofreire.org/handle/7891/4274>.

TOLOMEI, B. V. A gamificação como estratégia de engajamento e motivação na educação. *EaD em Foco*, v. 7, n. 2, 2017. Disponível em: <https://doi.org/10.18264/eadf.v7i2.440>.

BRANSFORD, J.; BROWN, A. L.; COCKING, R. R. *How people learn: Brain, mind, experience, and school*. Washington, D.C.: National Academy Press, 1999.

VIEIRA, A.; PASSOS, O.; BARRETO, R. Um relato de experiência do uso da técnica computação desplugada. In: *WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, XXI.*, 2013, Porto Alegre. Anais [...]. Porto Alegre: SBC, 2013. p. 671–680.

MARTINS, Patrícia Taquelim Da Cruz Furtado. O desenvolvimento de competências sociais na educação de infância: os contributos da brincadeira e das atividades de pequenos grupos. Lisboa: ESELx - Dissertações de Mestrado, 2013.

FAGERLUND, Janne; HÄKKINEN, Päivi; VESISENAHO, Mikko; VIIRI, Jouni. Computational thinking in programming with Scratch in primary schools: A systematic review. 2020.

GOMES, Anabela de Jesus. Dificuldades de aprendizagem de programação de computadores: contributos para a sua compreensão e resolução. Coimbra, 2010.

WEYH, L. F.; NEHRING, C. M.; WEYH, C. B. A educação problematizadora de Paulo Freire no processo de ensino-aprendizagem com as novas tecnologias. *Brazilian Journal of Development*, 2020.

MALONEY, J.; RESNICK, M.; RUSK, N.; SILVERMAN, B.; EASTMOND, E. The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, 2010. Disponível em: <https://doi.org/10.1145/1868358.1868363>.

SIMON, Cristiano Biazzo; PAGÈS BLANCH, Joan. Paulo Freire, ensino, história e os desafios da contemporaneidade. *Diálogos - Revista do Departamento de História e do Programa de Pós-Graduação em História*, v. 19, n. 1, p. 117–140, 2015.

GARCEZ, D.; AMORIM, T.; CAMPOS, C. Oficinas de computação desplugada e plugada: um relato de experiência da sinergia entre formação de licenciandos em computação e escola pública no DF. In: *WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO*, XXXII., 2024, Porto Alegre. Anais [...]. Porto Alegre: SBC, 2024. p. 10–20. DOI: 10.5753/wei.2024.1978.

MACHADO, K.; NEGREIROS, M.; MACHADO, C.; SOUSA, R. Ensino de programação e desenvolvimento do raciocínio lógico: um relato de experiência com estudantes do ensino fundamental. In: *WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO*, XXXII., 2024, Porto Alegre. Anais [...]. Porto Alegre: SBC, 2024. p. 46–56. DOI: 10.5753/wei.2024.2263.

BORDIN, A. S.; RODRIGUES, L. M.; CASAGRANDE, T. Ensino, pesquisa e extensão no ensino de engenharia de software: um relato de experiência. In: *WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO*, XXI., 2023. Anais [...], 2023.

BRENDA, F. et al. Um curso de programação para calouros com Scratch: um relato de licenciandos em computação. In: *WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO*, XXXII., 2024, Porto Alegre. Anais [...]. Porto Alegre: SBC, 2024. p. 251–262. DOI: 10.5753/wei.2024.2731.

SILVA, J. P.; SANTOS, M. L.; OLIVEIRA, R. T. A importância da capacitação inicial para monitores no ensino de computação infantil. *Revista Brasileira de Educação em Computação*, v. 16, n. 1, p. 45–58, 2023.

FERREIRA, A. L.; SOUZA, P. R. Reuniões de feedback como ferramenta de aprimoramento na monitoria de programação para crianças. *Anais do Simpósio Brasileiro de Informática na Educação*, v. 33, n. 1, p. 234–240, 2022.

OLIVEIRA, M. S.; MENDES, F. A. Planejamento de capacitações específicas baseado em feedback contínuo: um estudo de caso com monitores de robótica educacional. *Revista de Tecnologias Aplicadas à Educação*, v. 12, n. 3, p. 78–89, 2021.

PEREIRA, L. F.; LIMA, G. H.; ALMEIDA, S. R. O impacto da capacitação personalizada no desempenho de monitores de informática educativa. *Cadernos de Educação e Tecnologia*, v. 15, n. 2, p. 101–115, 2020.

RODRIGUES, C. A. Monitoria acadêmica: uma ferramenta de auxílio no processo de ensino-aprendizagem. *Anais do Congresso Nacional de Educação*, v. 8, n. 1, p. 321–330, 2019.