ISSN: 2358-8829

ROBÓTICA E COMPUTAÇÃO DESPLUGADA PARA O ENSINO DO PENSAMENTO COMPUTACIONAL EM ESCOLAS DO ENSINO FUNDAMENTAL E MÉDIO DO SERTÃO PARAIBANO

Gleysla Monaliza da Silva Pereira [1]
Jhuan Dantas de Sousa Soares [2]
Roger Natan Carvalho Silva [3]
Iuri Kauã Simão de Oliveira [4]
Jonas Fernandes da Silva [5]

RESUMO

O pensamento computacional pode ser definido como a abstração de problemas, divisão em etapas e segmentação em passos para então implementação posterior em algoritmo. Com os recentes avanços tecnológicos, a capacidade de decomposição e resolução de problemas tornou-se uma habilidade cada vez mais valorizada no mercado de trabalho e em áreas da pesquisa acadêmica. O pensamento computacional, portanto, e os conhecimentos relacionados à computação passaram a ser obrigatórios nas etapas iniciais da educação básica alinhada às propostas da BNCC. Dada a necessidade de apresentação dessas propostas de ensino fundamental e médio, o objetivo deste trabalho foi apresentar formas de ensinar pensamento computacional e lógica de programação seguindo princípios de computação desplugada. Nesse sentido, foram aplicadas atividades na temática mencionada, baseadas nos materiais de Crema (2020) e Raabe *et al.* (2020), autores que constituíram parte do referencial teórico do trabalho, envolvendo uso de *Scratch*, cartões com condicionais (*if else, elif*), atividades com programação de Arduino e robótica, entre outras atividades. Quanto à metodologia de pesquisa, foi realizado um trabalho de natureza primária, objetivo descritivo e procedimentos técnicos observacionais. A coleta de dados aconteceu a partir de entrevistas com os alunos que participaram das atividades propostas. Em relação à abordagem, optou-se por uma abordagem qualitativa para análise das opiniões dos alunos. Os resultados indicaram que a implementação das atividades envolvendo computação desplugada e robótica contribuiu para o aumento do engajamento e do interesse dos alunos pelas temáticas abordadas. Dessa forma, conclui-se que tais estratégias foram fundamentais para o desenvolvimento do pensamento computacional e da lógica de programação na educação básica.

[1] Graduando do Curso de Técnico de Informática do Instituto Federal da Paraíba - IFPB, monaliza.gleysla@academico.ifpb.edu.br;
[2] Graduado pelo Curso de Técnico de Informática do Instituto Federal da Paraíba - IFPB, jhuan.dantas@academico.ifpb.edu.br;
[3] Graduando pelo Curso de Ciência da Computação da Universidade Estadual da Paraíba - UEPB, roger.silva@aluno.uepb.edu.br;
[4] Graduando do Curso de Engenharia Civil do Instituto Federal da Paraíba - IFPB, iuri.kaua@academico.ifpb.edu.br;
[5] Professor orientador: Mestre, Universidade Federal da Paraíba - UFPB, fernandes.jonas@ifpb.edu.br.

ISSN: 2358-8829

INTRODUÇÃO

O pensamento computacional é uma forma de composição de resolução de problemas que envolve: abstração de problemas, divisão em etapas, reconhecimento de padrões e segmentação em passos para então implementação posterior em formato de algoritmo, não necessariamente envolvendo código bruto, mas podendo ser passo a passo. Com os recentes avanços tecnológicos, essas habilidades tornam-se cada vez mais valorizadas no mercado de trabalho e em áreas da pesquisa acadêmica (Geraldes, 2026).

Conforme a importância da tecnologia foi crescendo, os conhecimentos relacionados à computação passaram a ser obrigatórios nas etapas iniciais da educação básica, alinhada às propostas da Base Nacional Comum Curricular (BNCC). Dada a necessidade de apresentação dessas propostas, o objetivo deste trabalho foi apresentar formas de ensinar pensamento computacional e lógica de programação seguindo princípios de computação desplugada e desplugada para alunos de ensino fundamental e médio de escolas do sertão paraibano.

Nesse sentido, foram aplicadas atividades na temática mencionada, baseadas nos materiais de Crema (2020) e Raabe *et al.* (2020), envolvendo uso de Scratch, cartões com condicionais (if else, elif), atividades com programação de Arduino e robótica, entre outras atividades.

Quanto à metodologia de pesquisa, foi realizado um trabalho de natureza primária, objetivo descritivo e procedimentos técnicos de levantamento (Wazlawick, 2021). A coleta de dados aconteceu a partir de entrevistas com os alunos que participaram das atividades propostas. Em relação à abordagem, optou-se por uma abordagem qualitativa para análise das opiniões dos alunos.

O presente trabalho está estruturado em introdução, com um panorama sobre a temática e objetivos; a metodologia, com o protocolo científico, o referencial teórico, que apresentou os trabalhos correlatos; os resultados e discussão apresentando as atividades realizadas e o detalhamento sobre como os estudantes encararam a proposta; por fim, as considerações finais do trabalho.

METODOLOGIA

A metodologia deste artigo seguiu as proposições de Wazlawick (2021), que divide a pesquisa em computação quanto à natureza, aos objetivos e aos procedimentos técnicos. Assim, trata-se de um trabalho primário, em que a produção científica se deu a partir de ações empíricas no ambiente estudado, pensamento computacional em sala de aula.

Quanto ao objetivo, enquadra-se como exploratório e descritivo, ao quais foram elaboradas atividades e, a partir delas, realizou-se a descrição das contribuições e dificuldades encontradas. Por fim, em relação aos procedimentos técnicos, foi realizada uma pesquisa de levantamento, em que se analisou as respostas dos alunos entrevistados.

FUNDAMENTOS DE ALGORITMO E PROGRAMAÇÃO NO CONTEXTO DO SCRATCH

ISSN: 2358-8829

ALGORITMO E LÓGICA DE PROGRAMAÇÃO

O conceito de algoritmo constitui um dos fundamentos estruturantes da Ciência da Computação e do pensamento computacional. Formalmente, um algoritmo pode ser definido como um procedimento finito, ordenado e não ambíguo, composto por instruções precisamente definidas que transformam entradas em saídas por meio de etapas executáveis (Cormen *et al.*, 2009). Entre suas propriedades essenciais destacam-se a finitude, o determinismo e a efetividade, garantindo que sua execução produza resultados previsíveis em tempo limitado.

A relevância do algoritmo ultrapassa sua dimensão técnica, pois representa a formalização do raciocínio aplicado à resolução de problemas. Nesse sentido, sua construção exige organização sequencial, clareza lógica e coerência estrutural.

A lógica de programação refere-se à capacidade de estruturar esse raciocínio de maneira sistemática, convertendo ideias abstratas em instruções executáveis por um sistema computacional. Tal processo envolve a decomposição de problemas complexos em partes menores, a definição da ordem adequada das ações e a utilização apropriada de estruturas de decisão e repetição. A ausência de consistência lógica compromete a corretude, a legibilidade e a manutenibilidade do programa.

Essa perspectiva converge com a definição de pensamento computacional proposta por Wing (2006), que o descreve como a habilidade de formular problemas e estruturar soluções de modo que possam ser executadas por agentes computacionais. Assim, algoritmo e lógica de programação configuram-se como instrumentos cognitivos fundamentais para a organização formal do pensamento.

O SCRATCH COMO AMBIENTE DE PROGRAMAÇÃO EDUCACIONAL

O Scratch, desenvolvido pelo Lifelong Kindergarten Group do MIT Media Lab e lançado publicamente em 2007, constitui um ambiente de programação visual projetado para fins educacionais. Sua proposta central é democratizar o acesso aos conceitos fundamentais da programação, especialmente entre estudantes da educação básica.

Inspirado na linguagem LOGO, concebida por Seymour Papert, o Scratch fundamenta-se em princípios construcionistas que defendem a aprendizagem por meio da criação de artefatos significativos (Papert, 1980). A interface baseada em blocos gráficos encaixáveis reduz a complexidade sintática presente em linguagens textuais, permitindo que o foco da aprendizagem recaia sobre a estrutura lógica do algoritmo.

Essa característica favorece a internalização de conceitos como sequência, condição, repetição, variáveis e operadores, promovendo um ambiente propício ao desenvolvimento do pensamento computacional. Ao possibilitar a criação de animações, jogos e simulações interativas, o Scratch integra dimensão técnica e expressiva, articulando lógica formal e criatividade.

VARIÁVEIS DE ENTRADA E SAÍDA

As variáveis constituem abstrações fundamentais na programação, correspondendo a espaços simbólicos de armazenamento capazes de reter valores durante a execução de um algoritmo. Sua utilização permite representar dados, controlar estados e viabilizar o processamento de informações. No Scratch, as variáveis apresentam tipagem dinâmica, dispensando declaração prévia de tipo, o que simplifica sua aplicação em contextos pedagógicos.

ISSN: 2358-8829

As variáveis de entrada são responsáveis por receber informações externas ao sistema, frequentemente fornecidas pelo usuário por meio de interações ou eventos. Esses dados são incorporados ao fluxo de execução do algoritmo e influenciam diretamente as decisões subsequentes. Por sua vez, as variáveis de saída armazenam os resultados produzidos após o processamento, sendo comunicados ao usuário por meio de recursos visuais ou textuais disponíveis na plataforma.

OPERADORES MATEMÁTICOS E LÓGICOS

Os operadores matemáticos — adição, subtração, multiplicação e divisão — desempenham papel estruturante na execução algorítmica, pois viabilizam o processamento numérico e a avaliação de condições.Assim, essas estrutura permite a realização de cálculos sobre variáveis e constantes, obedecendo às regras de precedência e associatividade que garantem a coerência das expressões avaliadas (Forbellone; Eberpãcher, 2005). Sua aplicação assegura precisão nos resultados e viabiliza a modelagem quantitativa de situações-problema.

Os operadores lógicos, fundamentados na álgebra booleana, operam sobre proposições que assumem valores binários (verdadeiro ou falso). Entre os principais operadores destacam-se AND (E), OR (OU) e NOT (NÃO), amplamente utilizados em estruturas condicionais para determinar diferentes caminhos de execução (Manzano, 2010). Esses operadores possibilitam a formalização de decisões automatizadas baseadas em critérios previamente estabelecidos.

ESTRUTURAS CONDICIONAIS E DE REPETIÇÃO

Outro ponto importante é a utilizadção de estruturas condicionais, que possibilitam a tomada de decisão com base na avaliação de expressões booleanas. No Scratch, são representadas pelos blocos “se <condição> então” e “se <condição> então / senão”. Essas estruturas viabilizam comportamentos adaptativos, permitindo que o programa execute diferentes ações conforme os resultados das condições avaliadas. As estruturas de repetição, ou loops, possibilitam a execução iterativa de instruções.

Ainda sobre o Scratch, destacam-se os blocos “repita ( )”, “repita até ( )” e “para sempre”. Tais estruturas promovem eficiência algorítmica, evitando redundâncias e permitindo a modelagem de processos contínuos ou cíclicos. A compreensão dessas estruturas é fundamental para o desenvolvimento do raciocínio algorítmico, pois permite estruturar decisões e controlar a execução de tarefas de forma sistemática, coerente e eficiente.

CONSIDERAÇÕES TEÓRICAS INTEGRADORAS

A articulação entre algoritmo, lógica de programação, variáveis, operadores e estruturas de controle constitui a base conceitual necessária para a formação do pensamento computacional. Esses elementos permitem organizar dados, estruturar processos, tomar decisões e modelar comportamentos dinâmicos, configurando-se como bases da programação e da automação e pensamento computacional.

No contexto educacional, o Scratch apresenta-se como um ambiente adequado para introdução desses conceitos, pois reduz barreiras sintáticas e privilegia a construção da lógica estrutural. Desse modo, a programação deixa de ser apenas domínio técnico e passa a constituir instrumento formativo para o desenvolvimento de competências cognitivas superiores relacionadas à abstração, decomposição e modelagem de problemas.

ISSN: 2358-8829

COMPUTAÇÃO DESPLUGADA NO DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL

A computação desplugada constitui uma abordagem metodológica que promove o desenvolvimento de conceitos fundamentais da Ciência da Computação sem a utilização direta de dispositivos digitais. Tal perspectiva fundamenta-se na premissa de que os princípios estruturantes do pensamento computacional, como decomposição, reconhecimento de padrões, abstração e elaboração de algoritmos, são independentes de tecnologias específicas, podendo ser trabalhados por meio de atividades concretas, lúdicas e colaborativas.

O termo "computação desplugada" foi amplamente difundido a partir do projeto Computer Science Unplugged, desenvolvido na University of Canterbury, que sistematizou atividades pedagógicas voltadas à compreensão de conceitos computacionais por meio de jogos, dinâmicas corporais e materiais manipuláveis. Conforme Bell, Witten e Fellows (2011), a proposta demonstra que fundamentos como algoritmos, codificação de dados e estruturas de controle podem ser ensinados de forma concreta, favorecendo a compreensão conceitual antes da formalização sintática.

Ao tratar de conceitos como variáveis, operadores, condicionais e laços de repetição por meio de simulações corporais ou jogos de regras, a computação desplugada favorece a internalização de estruturas lógicas antes de sua representação formal em blocos de programação. Wing (2006) argumenta que o pensamento computacional não se restringe à programação, mas corresponde a uma forma específica de organizar o raciocínio para resolver problemas de maneira sistemática. Assim, a abordagem desplugada contribui para consolidar essa forma de pensar, reduzindo a sobrecarga cognitiva associada à interface tecnológica.

No contexto da robótica educacional, a computação desplugada assume papel estratégico ao permitir que estudantes compreendam previamente a lógica de sensores, atuadores e fluxos condicionais por meio de dramatizações ou atividades com cartões de comando. Antes mesmo da implementação em plataformas digitais ou robóticas, os discentes experimentam a lógica sequencial ("faça isto, depois aquilo"), condicional ("se ocorrer X, execute Y") e iterativa ("repita até que"), internalizando os princípios que posteriormente serão formalizados em ambientes como o Scratch.

Pedagogicamente, tal abordagem favorece a aprendizagem significativa (Ausubel, 2003), pois conecta novos conceitos a estruturas cognitivas já existentes, além de estimular a colaboração, a comunicação e o raciocínio lógico. Em contextos escolares com limitações tecnológicas, a computação desplugada também amplia o acesso à educação em computação, democratizando o ensino de conceitos fundamentais.

Portanto, a computação desplugada não deve ser compreendida como etapa isolada, mas como componente articulado a práticas digitais e robóticas. No presente projeto, essa abordagem funciona como estratégia introdutória e estruturante para o ensino de variáveis, operadores, condicionais e loops no Scratch, fortalecendo a base conceitual necessária ao desenvolvimento do pensamento computacional e à resolução de problemas em robótica educacional.

ARDUINO

O Arduino é amplamente reconhecido como uma plataforma de prototipagem eletrônica de código aberto, que integra hardware e software de maneira flexível e

ISSN: 2358-8829

acessível. Concebido para facilitar o desenvolvimento de projetos interativos, o Arduino democratizou o acesso à eletrônica e à programação, permitindo que indivíduos com diferentes níveis de conhecimento técnico possam criar e implementar sistemas eletrônicos (ARDUINO, [s.d.]a).

A plataforma é composta por uma placa microcontroladora (hardware) e um Ambiente de Desenvolvimento Integrado (IDE) baseado em Processing/Wiring (software), que simplifica a escrita e o upload de código. Sua natureza open-source fomenta uma vasta comunidade de usuários e desenvolvedores, resultando em farta documentação, bibliotecas e exemplos que aceleram o processo de prototipagem e aprendizado (ARDUINO, [s.d.]a).

RESULTADOS E DISCUSSÃO

Durante o percurso desse trabalho realizaram-se momentos formativos em oficinas e atividades com arduino, scratch e computação desplugada. Assim, o público alvo foram estudantes do ensino fundamental e médio. As atividades aconteceram nas seguintes instituições de ensino: Dionísio da Costa - Patos - PB (2 encontros), ECIT Sebastião Guedes da Silva - Teixeira - PB (1 encontro) e Instituto Federal do Sertão da Paraíba - Patos - PB (4 encontros). Descritas a seguir, estão dispostas as atividades realizadas:

ATIVIDADE - O LABIRINTO

Essa atividade foi titulada como “O Labirinto”, pensada para demonstrar, de maneira lúdica, como o computador responde aos comandos do usuário por meio do sistema operacional. Para isso, foi construído um trajeto em forma de labirinto, utilizando fita crepe para demarcar o caminho e objetos como cadeiras e marcações em fita para servir de obstáculos.

A dinâmica ocorre em duas fases. Na primeira fase, o labirinto foi preparado sem a presença dos alunos e, posteriormente, cada grupo — de 3 pessoas — indicou um integrante para participar como líder. Esse aluno teve os olhos vendados e precisou percorrer o caminho seguindo apenas as orientações verbais de um dos colegas. Caso encostasse algum obstáculo ou ultrapassasse os limites do percurso, deveria retornar ao início. O terceiro participante ficou responsável por registrar as instruções fornecidas. No final, com as instruções registradas, os alunos montam um algoritmo, parecido com o Scratch, usando os “blocos de programação” feitos em MDF.

Cada percurso foi realizado com controle de tempo e apenas um grupo tem acesso ao ambiente por vez, enquanto os outros aguardam do lado de fora, para evitar que obtenham vantagem por verificar o percurso previamente. A proposta é auxiliar na compreensão do funcionamento do computador de forma prática e interativa.

ATIVIDADE - CARTÕES DE CONDIÇÃO

A atividade denominada “Cartões de Condição” serve para promover a compreensão de estruturas condicionais (SE/ENTÃO/SENÃO) e o desenvolvimento do pensamento lógico, por meio de uma abordagem de Computação Desplugada. A dinâmica visa demonstrar, de forma concreta, como sistemas computacionais executam ações com base em condições estabelecidas pelo usuário.

ISSN: 2358-8829

Foram elaborados cartões contendo condições, ações e operadores lógicos (E/OU/NÃO). Os alunos, organizados em grupos, utilizaram os cartões para construir regras condicionais coerentes, seguindo a estrutura lógica básica: SE (condição) → ENTÃO (ação) → SENÃO (ação alternativa). Posteriormente, os grupos trocaram seus conjuntos de regras para realizar uma verificação cruzada, avaliando a consistência e a lógica das condicionais produzidas por outra equipe.

Na etapa final, os alunos foram desafiados a elaborar condicionais mais complexas, integrando operadores lógicos e múltiplas condições, e registraram suas construções em papel para posterior representação em fluxogramas ou blocos de programação. A atividade permitiu a compreensão prática de como decisões são tomadas em algoritmos, evidenciando a importância da precisão lógica na programação.

ATIVIDADE - CALCULADORA DE ADIÇÃO

A proposta didática envolve a criação de um projeto interativo no qual o usuário insere dois valores numéricos e o sistema retorna o resultado da soma. Embora o problema matemático seja simples, sua construção no ambiente de programação permite explorar fundamentos estruturantes da lógica computacional.

No Scratch, a atividade é desenvolvida por meio de:

- Uso de variáveis para armazenar os valores inseridos pelo usuário;
- Utilização do bloco operador aritmético "+" (adição);
- Emprego de blocos de entrada de dados ("pergunte e espere");
- Exibição do resultado com blocos de saída ("diga" ou exibição em variável);
- Organização sequencial dos blocos, evidenciando a estrutura de algoritmo.

ATIVIDADE - CALCULADORA COMPLETA

A proposta didática envolve a criação de um projeto interativo no qual o usuário escolhe qual operação deseja realizar (adição, subtração, multiplicação ou divisão) e, em seguida, insere dois valores numéricos. O sistema avalia a operação escolhida por meio de estruturas condicionais e executa apenas o cálculo correspondente.

No Scratch, a atividade é desenvolvida por meio de:

- Uso de variáveis para armazenar a operação escolhida e os números digitados;
- Utilização de blocos de entrada de dados ("pergunte e espere");
- Aplicação de operadores aritméticos (+, −, ×, ÷);
- Uso de blocos condicionais (se... então ou se... então... senão);
- Exibição do resultado com blocos de saída ("diga").

Embora matematicamente simples, a atividade exige organização lógica mais elaborada, pois o programa deixa de ser linear e passa a operar com fluxo de decisão.

ISSN: 2358-8829

ATIVIDADE - PAR OU ÍMPAR

A atividade proposta consiste no desenvolvimento de um algoritmo que identifique se um número inserido pelo usuário é par ou ímpar na plataforma Scratch, com o objetivo de introduzir conceitos de estruturas condicionais e operadores lógicos dentro da perspectiva do pensamento computacional.

Embora o conceito matemático seja simples, sua construção no ambiente de programação permite explorar fundamentos estruturantes da lógica computacional, particularmente a tomada de decisão baseada em condições, a utilização de

operadores matemáticos (resto de divisão) e a comunicação clara de resultados ao usuário.

No Scratch, o algoritmo é desenvolvido por meio dos seguintes blocos:

- Pergunte - Solicita um valor numérico do usuário e armazena em uma variável
- Operador % - Calcula o resto da divisão (operador modular)
- Se [condição] - Estrutura condicional que verifica se resto = 0 (par) ou ≠ 0 (ímpar)
- Diga - Exibe mensagem ao usuário informando se o número é par ou ímpar

ATIVIDADE - OFICINAS DE ARDUINO

O Arduino é uma ferramenta de prototipação que possibilita realizar a conexão e programação em código C++ voltada para utilização em sala de aula. Os alunos podem conectar os componentes de maneira simplificada e testar o encaixe com diversos componentes eletrônicos (sensores de distância, sensores de luminosidade, entre outros), encarando menores índices de dificuldades com princípios de eletrônica e no código.

Essa plataforma de programação permite que os estudantes façam um código capaz de perceber o ambiente ao seu redor, com os diversos sensores, e realize alterações no meio, utilizando atuadores como: LEDs, motores, relés, entre outros componentes. O objetivo da atividade realizada foi eninar o grupo de alunos sobre a principais características do Arduino: suas entradas e saídas de dados, controle de energia, método de criação do código, como testar os componentes.

Dessa maneira, os estudantes que participaram dessa oficina foram orientados para montagem de um sensor ultrasônico, que mede a distância de um obstáculo à frente calculando o tempo que a onda leva para ir e voltar. O valor encontrado foi armazenado em uma variável para ser trabalhado com condicionais para acionar 3 LEDs de acordo com o valor da distância.

Além disso, foram organizadas oficinas para montagem de kits 2 wheels drive (2WD) em que os alunos utlizaram os componentes: Motores de corrente alternada, sensor ultrasônico, Arduino, ponte H L298. O objetivo foi elaborar um robô capaz de andar sozinho (de acordo com a programação previamente feita) e desviar de obstáculos detectados pelo sensor.

ISSN: 2358-8829

CONSIDERAÇÕES FINAIS

Durante as atividades práticas observadas, especialmente as propostas desplugadas e de robótica educacional, as anotações de campo registraram sinais de maior participação dos alunos em comparação com os momentos de exposição teórica acompanhados pelos pesquisadores. Essa percepção baseou-se em comportamentos como a formulação de perguntas, a solicitação de esclarecimentos e a colaboração entre colegas.

Considerando esses registros, as atividades práticas pareceram favorecer uma participação mais ativa naquele contexto. Esse achado pode ser discutido à luz da perspectiva construcionista de Papert (1980), que valoriza a aprendizagem por meio da criação e da exploração de artefatos. Assim, as atividades podem ter oferecido oportunidades para que os alunos participassem ativamente do processo.

Durante a proposta do projeto, houveram dificuldades para organizar os kits de robótica, causadas pela ausência de componentes específicos para a montagem dos protótipos, algo que ocasionou adiamentos e necessidade de refazer orçamento entre outros projetos para conseguir atender à demanda das oficinas de arduino. Além disso, realizar o transporte entre as escolas também foi um desafio. Outra lacuna desse trabalho foi que, apesar de comentadas, não foi possível abordar as atividades “Impar ou par” e “calculadora de adição” por questão de tempo durante os momentos de encontro com as escolas.

Para trabalhos futuros, pode-se considerar elaborar atividades baseadas em outros materiais e trabalhando com parcerias entre mais instituições de ensino, no presente artigo foram trabalhadas apenas 3 instituições. Como também ofertar momentos com maior tempo, buscando observar como as atividades podem ser apresentadas de maneira online, para atingir um público maior, oferecendo materiais como ebooks e cartilhas com os conteúdos abordados.

Palavras-chave: BNCC, Pensamento Computacional, Computação desplugada, Ensino de Computação.

AGRADECIMENTOS

Agradecimentos aos colaboradores que participaram da organização dos momentos formativos, aos professores das escolas que acolheram o grupo de pesquisa; aos estudantes que ficaram engajados com as propostas apresentadas e ao Congresso Nacional de Educação pela oportunidade de apresentar o trabalho e contribuir para o desenvolvimento científico.

REFERÊNCIAS

ARDUINO. **Analog Input Pins**. [S. l.: s. n.], [s.d.]a. Disponível em: https://www.arduino.cc/en/Tutorial/AnalogInputPins. Acesso em: 16 fev. 2026.

AUSUBEL, D. P. Aquisição e retenção de conhecimentos: uma perspectiva cognitiva. Lisboa: Plátano, 2003.

BELL, T.; WITTEN, I.; FELLOWS, M. **Computer Science Unplugged**: Ensinando Ciência da Computação sem o uso do computador. Christchurch: University of Canterbury, 2011.

CORMEN, T. H. *et al*. **Introduction to Algorithms**. 3. ed. Cambridge: MIT Press, 2009.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F.. **Lógica de programação**. São Paulo: Pearson Prentice Hall, 2005.

GERALDES, W. B. Pensamento Computacional para professores da EPT: da transversalidade curricular à elaboraçãode projetos pedagógicos com metodologias ativas. *In*: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 15, 2026, Goiânia; JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA NA EDUCAÇÃO, 15. 2026, Porto Alegre. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2026. p 1-21. Disponível em: https://books-sol.sbc.org.br/index.php/sbc/catalog/view/210/942/2051. Acesso em: 13 out. 2026.

MALONEY, J. *et al.* **The Scratch programming language and environment**. ACM Transactions on Computing Education, New York, v. 10, n. 4, p. 1–15, nov. 2010.

MANZANO, J. A. N. G. Algoritmos: Lógica para Desenvolvimento de Programação de Computadores. São Paulo: Érica, 2010

MIT MEDIA LAB. **Scratch – About**. Disponível em: https://scratch.mit.edu/about https://sol.sbc.org.br/index.php/wei/article/view/3507/3466

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

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

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