



## Utilizando a Computação Tangível como Estratégia para Facilitar o Ensino e Aprendizagem de Lógica de Programação

*Using Tangible Computing as a Strategy to Facilitate Teaching and Learning Programming Logic*

Giani Augusto Braga<sup>1</sup>



<https://orcid.org/0000-0002-5422-4494>



<http://lattes.cnpq.br/5237784514383473>

Anubis Graciela de Moraes Rossetto<sup>2</sup>



<https://orcid.org/0000-0001-8657-2816>



<http://lattes.cnpq.br/9641784472463009>

### RESUMO

As disciplinas iniciais de algoritmos e lógica de programação nos cursos de Computação frequentemente se tornam um desafio para os estudantes. A falta de familiaridade com o pensamento computacional pode ser um fator significativo, levando, em muitos casos, à reprovação ou até mesmo ao abandono do curso. Outro obstáculo é a escassez de acesso a equipamentos, como smartphones ou computadores, para estudos extraclasse. Diante disso, é necessário que os professores adotem metodologias que rompam com o formato de aula tradicional, possibilitando avanços no processo de aprendizagem. Desse modo, o presente texto resulta de um processo de ensino e aprendizagem em que foi desenvolvida e aplicada uma ferramenta educacional tangível em uma turma do curso de Desenvolvedor Mobile: peças de quebra-cabeça impressas em uma impressora 3D para o ensino de algoritmos e lógica de programação. Os resultados demonstram-se promissores, pois mais da metade da turma obteve nota superior a seis, nas avaliações que envolveram esses conhecimentos e aproximadamente 80% demonstraram compreensão dos conceitos básicos de algoritmos após a realização da atividade.

**Palavras-chave:** Quebra-cabeça; Algoritmos; Ferramenta Educacional.

### ABSTRACT

*The initial disciplines of algorithms and programming logic in Computing courses often become a challenge for students. Lack of familiarity with computational thinking can be a significant factor, leading, in many cases, to failure or even abandonment of the course. Another obstacle is the lack of access to equipment, such as smartphones or computers, for extracurricular studies. Given this, it is necessary for teachers to adopt*

<sup>1</sup> Instituto Federal Sul-Rio-Grandense (IFSUL) Passo Fundo, RS – Brasil. E-mail: [giani.braga80@gmail.com](mailto:giani.braga80@gmail.com)

<sup>2</sup> Instituto Federal Sul-Rio-Grandense (IFSUL) Passo Fundo, RS – Brasil. E-mail: [anubisrossetto@ifsul.edu.br](mailto:anubisrossetto@ifsul.edu.br)



*methodologies that break with the traditional class format, enabling advances in the learning process. Thus, this text results from a teaching and learning process in which a tangible educational tool was developed and applied in a Mobile Developer course class: puzzle pieces printed on a 3D printer for teaching algorithms and logic of programming. The results are promising, as more than half of the class obtained a grade above six in the assessments that involved this knowledge and approximately 80% demonstrated understanding of the basic concepts of algorithms after carrying out the activity.*

**Keywords:** Puzzle; Algorithms; Educational Tool.

## 1. INTRODUÇÃO<sup>3</sup>

Nos cursos de Desenvolvedor Mobile, assim como em outros cursos na área da computação, a base inicial é estabelecida por disciplinas de algoritmos e lógica de programação, fundamentais na formação dos estudantes no processo de aprendizagem. No entanto, frequentemente, esses tópicos se transformam em obstáculos para os estudantes. Isso pode acontecer por diferentes fatores, sendo um deles a falta de familiaridade com os novos padrões de pensamento demandados por essas disciplinas, resultando na ausência das habilidades necessárias. Uma consequência desse aspecto é que muitos estudantes ao enfrentarem dificuldades dessa natureza, lamentavelmente, acabam reprovando nessas disciplinas e, em casos mais extremos, desistem do curso (Santos *et al.*, 2015).

É notória a relevância dessas disciplinas, pois são a base para a programação e pré-requisito para disciplinas sequenciais, sendo assim, são significativas nos primeiros semestres. Estudos recentes apontam um aumento significativo na taxa de reprovação em disciplinas introdutórias de algoritmos e lógica de programação (Vieira *et al.*, 2015; Schorr; Bercht, 2018), indicando que é necessário rever os processos de ensino e aprendizagem. Como afirmam Cabello e Chagas (2021, p. 110), “[...] o efeito de reprovações é bastante negativo para a taxa de evasão e esse efeito é maior para aqueles que reprovaram no início do curso, sugerindo que políticas públicas que busquem reduzir a evasão devem focar, principalmente, os primeiros anos de curso”. Desse modo, fica evidente a barreira gerada para quem não consegue assimilar o conteúdo, e como consequência tem uma reprovação, o quanto isso dificulta a progressão no curso.

O SENAI (Serviço Nacional de Aprendizagem Industrial) oferece cursos de aprendizagem básica e de nível técnico gratuitos para jovens que buscam sua inserção no mercado de trabalho. Dentre os cursos ofertados está o de Desenvolvedor Mobile, que é constituído por turmas com estudantes de idades distintas, variando entre 15 e 24 anos. Um aspecto a ser considerado é a disparidade de conhecimento entre os estudantes que constituem essas turmas, uma vez que suas experiências de ensino médio variam consideravelmente. Outro ponto relevante é a predominância de estudantes de baixa renda, muitos dos quais não têm acesso a *smartphones* ou computadores, e em alguns casos, não há nenhum desses dispositivos. Essa falta de acesso impacta significativamente o aprendizado, pois na área tecnológica, a prática além do momento da aula é essencial para aprimorar os estudos. Portanto, identifica-se uma dificuldade no reconhecimento dos passos necessários para

<sup>3</sup> Esta pesquisa foi aprovada pelo Comitê de Ética em Pesquisa, conforme o Certificado de Apresentação de Apreciação Ética (CAAE) nº 74090223.0.0000.5342 e Parecer Consubstanciado nº 6.326.920, emitido em 27 de setembro de 2023.



solucionar algoritmos. Em outras palavras, os estudantes enfrentam obstáculos para identificar os dados, o resultado esperado e os métodos necessários para resolver os problemas.

Logo, torna-se essencial implementar uma metodologia ou ferramenta capaz de proporcionar efetivamente o desenvolvimento do pensamento computacional (capacidade de resolver problemas de forma estruturada, a concepção de sistemas e a compreensão do comportamento humano, usando princípios da ciência da computação (Wing, 2021)) nos estudantes, impulsionando seu avanço no curso. Diante dessa necessidade, a computação tangível emerge como uma abordagem capaz de promover um pensamento computacional diferenciado, ultrapassando os métodos tradicionais que envolvem apenas teclados, mouses e telas (De Mattos Vogel; Pereira, 2023). A introdução de objetos físicos proporciona uma interação mais intuitiva e imersiva com a tecnologia (Lima *et al.*, 2023). Assim como afirma Piaget (1996, p. 71), “[...] a primeira função do conhecimento é ser uma *assimilação*, no sentido precisamente de uma interação entre o sujeito e o objeto”. Dessa forma, surge a proposta de empregar quebra-cabeças como ferramenta no ensino de algoritmos e lógica de programação. Essa abordagem visa oferecer aos educandos uma experiência prática e tangível que facilita a compreensão de conceitos complexos, estimulando sua participação e promovendo um aprendizado mais significativo.

Este trabalho foi uma pesquisa-ação que envolveu o desenvolvimento de um quebra-cabeça fabricado com o suporte de uma impressora 3D. Cada peça desse quebra-cabeça contém uma parte do algoritmo, desafiando os estudantes a montá-lo para resolver o problema proposto. O objetivo principal foi explorar o uso de uma ferramenta educacional tangível (quebra-cabeça) para facilitar o ensino de algoritmos e lógica de programação em uma turma de aprendizagem industrial básica no SENAI CFP-Passo Fundo (Centro de Formação Profissional Jorge Barbieux). Buscou-se compreender as motivações e desafios dos estudantes, especialmente daqueles mais propensos à evasão escolar e à reprovação nessas disciplinas. Além disso, almejava-se analisar o impacto da introdução de novas metodologias de ensino no desempenho e na compreensão dos estudantes. O trabalho também se propôs a desenvolver estratégias e avaliar o impacto do uso do quebra-cabeça como ferramenta para promover o desenvolvimento do conhecimento dos estudantes no conteúdo de lógica de programação.

O artigo está organizado da seguinte forma: Na seção 2, é apresentado o referencial teórico e os estudos que serviram como base para esse trabalho. Na seção 3, são apresentadas as etapas que foram necessárias para o desenvolvimento do trabalho. Na seção 4 são discutidos os resultados provenientes da aplicação da atividade e as respostas dos questionários. Por fim, na seção 5 é apresentado a conclusão e a proposição de trabalhos futuros.

## 2. TRABALHOS RELACIONADOS

Com o avanço contínuo da computação ao longo dos anos, torna-se fundamental introduzir o pensamento computacional para a aprendizagem das crianças e adolescentes, especialmente abordando os princípios fundamentais da ciência da computação (Araujo; Andrade; Guerrero, 2016; Grebogy; Santos; Castilho, 2021). Para o ensino de lógica de programação, várias ferramentas



gráficas (programação em blocos) foram desenvolvidas, como o Scratch (Resnick *et al.*, 2009), Blockly (Fraser, 2015), App Inventor (Mackellar *et al.*, 2011), entre outras, as quais são frequentemente empregadas para a aprendizagem nessa área. Essas ferramentas geralmente combinam ações de arrastar e soltar blocos, permitindo a execução de funções específicas e tornando-as de fácil utilização pelos estudantes.

Porém, a inserção de crianças e adolescentes na área da computação não é algo tão recente. Por exemplo, Seymour Papert, em 1980, antes mesmo do surgimento dessas ferramentas gráficas, tentando inserir as crianças e adolescentes na computação, apresentou algo inesperável para a época, um robô em forma de “tartaruga” que poderia ser programado por crianças para desenhar figuras geométricas em folhas de papel, podendo ser visualizado na Figura 1. Assim, esse fato pode ser considerado o início do que veio a se chamar, posteriormente, de computação tangível. Os autores Ishii e Ullmer (1997), conceituaram tangível como uma classe de interfaces de computador que empregam objetos físicos e superfícies como um meio de manipular e representar informações digitais. Ou seja, algo que a criança ou adolescente possa tocar, manusear e com isso auxiliar na sua aprendizagem.

**Figura 1** – Robô programável.



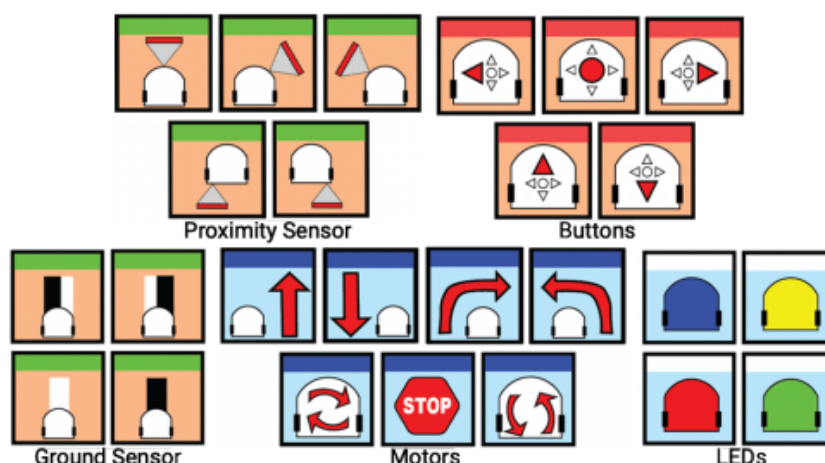
Fonte: Negrini (2018) e Fuste (2018).

Recentemente, diversos estudos têm explorado o campo da computação tangível ou programação tangível (Goyal *et al.*, 2016; Gardeli; Vosinakis, 2019; Im; Rogers, 2021). Os autores Mehrotra *et al.* (2020), desenvolveram uma plataforma chamada PaPL (Paper-based Programming Languages), destinada a criar linguagens de programação baseadas em papel. Essa plataforma utiliza as *webcams* de computadores e as câmeras de *tablets* para processar algoritmos de visão computacional, oferecendo uma estrutura para adaptar linguagens de programação existentes para versões em papel. Basicamente, a plataforma interpreta blocos de papel contendo código de programação, os quais são lidos pela *webcam* e executam ações específicas em um robô. Na Figura 2, é possível visualizar um exemplo desses papéis usados para controlar os movimentos do robô.



A pesquisa sobre a eficiência do PaPL foi conduzida ao longo de dois dias, envolvendo mais de cem estudantes do ensino médio, com idades entre 18 e 19 anos (Mehrotra *et al.*, 2020). Os resultados revelaram que tanto o tamanho do grupo quanto as restrições de uso exercem influência significativa na interação dos estudantes com a plataforma. Os autores ressaltam que a aprendizagem em duplas teve um efeito positivo. Ademais, os estudantes alcançaram pontuações elevadas nos testes do Parson's Puzzle.

**Figura 2** – Blocos de programação do PaPL.



Fonte: Mehrotra *et al.* (2020).

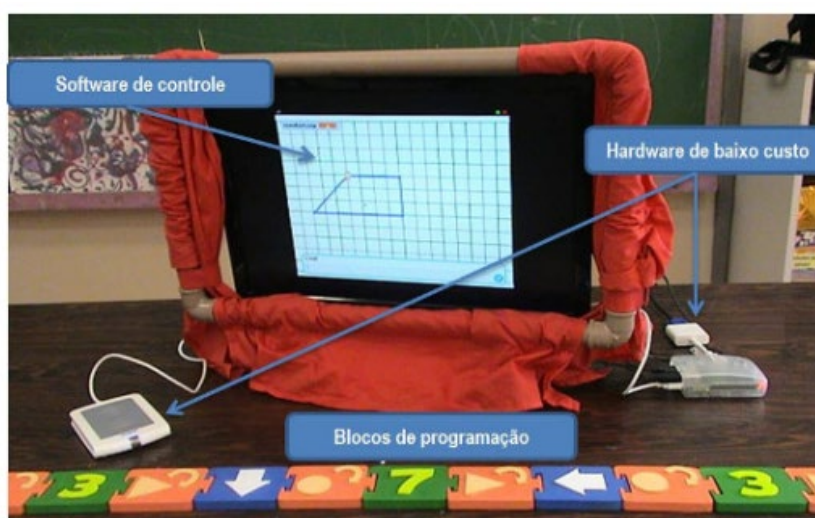
Carbajal e Baranauskas (2016) apresentam o TaPrEC (Tangible Programming Environment for Children), como um ambiente que permite as crianças criarem programas por meio de objetos tangíveis. Esse ambiente opera no contexto do programa Scratch, onde a arquitetura se baseia em um *Raspberry Pi*, que é um minicomputador com recursos suficientes para a execução da proposta. O TaPrEC também faz uso de RFID (Radio Frequency Identification), um método de identificação automática por sinais de rádio frequência que recupera e armazena dados remotamente por meio de etiquetas RFID. A identificação RFID pelo *Raspberry Pi* ocorre por meio de blocos de programação, compostos por peças de madeira que se assemelham a blocos de quebra-cabeça, como está demonstrado na Figura 3. Cada bloco possui um símbolo em alto-relevo representando sua função, com cores e formas projetadas para serem facilmente compreendidas por qualquer pessoa. Quanto à forma, existem três tipos de blocos: de início, de ação e de fim. Cada um com sua funcionalidade específica.

A pesquisa dos autores Carbajal e Baranauskas (2016) envolvendo o TaPrEC foi conduzida em uma escola de ensino fundamental, contando com a participação de oito professoras e vinte crianças com idades entre 8 e 10 anos. Foram realizadas treze oficinas experimentais ao longo de seis semanas. Conforme relatado pelos autores, a ferramenta mostrou-se apropriada para a programação tangível. A análise dos resultados obtidos indicou que o ambiente conseguiu cativar as crianças, as quais não enfrentaram dificuldades ao manipular os objetos tangíveis, aprendendo de forma eficaz os conceitos básicos fundamentais de programação.



Com base na contribuição desses autores e Piaget (1996, p. 71) que afirma que: “Os conhecimentos não partem, com efeito, nem do sujeito [...] nem do objeto [...], mas das interações entre sujeito e objeto”, torna-se importante possibilitar que os estudantes manipulem objetos tangíveis. Pois, é a interação do sujeito com o objeto, que provoca um desequilíbrio e estimula uma resposta ao objeto. Esse desequilíbrio leva o sujeito a fazer adaptações para restaurar o equilíbrio, resultando em uma nova ação sobre o objeto, que contribui para a construção do conhecimento (Flach; Becker, 2016).

**Figura 3** – Ambiente de estudos TaPrEC.



Fonte: Carbajal e Baranauskas (2016).

Portanto, é proposto um projeto que estimule o pensamento computacional dos educandos, para isso foi criado um quebra-cabeça como objeto tangível, proporcionando ao estudante a aprendizagem de conceitos básicos de algoritmos e lógica de programação, provocando o desequilíbrio na construção do conhecimento. Dessa forma, almejou-se aumentar a compreensão, a motivação, e diminuir o índice de evasão dos estudantes no curso.

### 3. METODOLOGIA

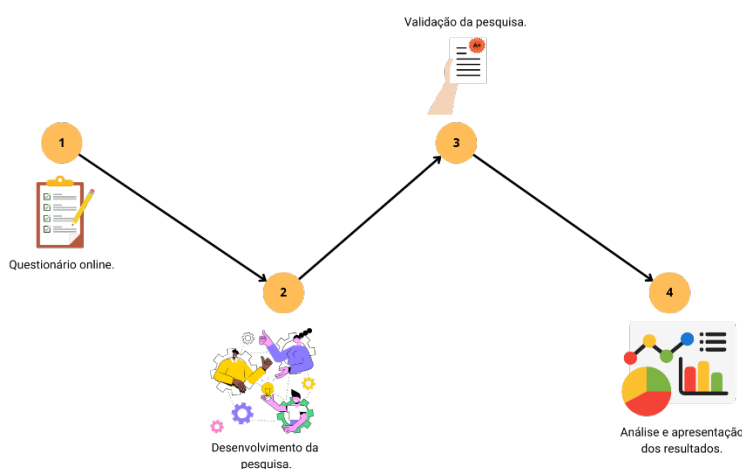
O presente texto resulta de uma pesquisa-ação que adotou uma abordagem qualitativa, aplicado e analisado no contexto de uma turma composta de vinte e quatro estudantes do curso de aprendizagem industrial básica. Durante a pesquisa, uma sequência de etapas foi seguida para sua realização (podendo ser observadas na Figura 4), sendo elas:

- O primeiro passo diz respeito à aplicação de um questionário *online*, visando analisar e identificar os conhecimentos prévios dos estudantes. Essa avaliação permitiu estabelecer uma base para compreender o ponto de partida dos educandos, adaptando o ensino de acordo com suas necessidades específicas;



- O segundo passo compreendeu o desenvolvimento da pesquisa em si, em que a ferramenta educacional foi empregada como recurso principal para o ensino e aprendizagem de algoritmos e lógica de programação. Essa abordagem teve como objetivo promover uma aprendizagem interativa e prática, possibilitando aos estudantes a construção de algoritmos por meio de peças de quebra-cabeça. Essa metodologia visou facilitar o desenvolvimento de habilidades e a compreensão dos conceitos essenciais de lógica de programação;

**Figura 4** – Etapas seguidas durante o desenvolvimento.



Fonte: Elaborada pelos próprios Autores.

- O terceiro passo consistiu na validação da pesquisa. Para avaliar o aprendizado dos estudantes, optou-se por realizar atividades de sistematização e trabalhos como forma de avaliação. Essas atividades permitiram a coleta de dados para avaliar o nível de compreensão e domínio dos conceitos trabalhados sobre lógica de programação. Essa abordagem possibilitou identificar *insights* valiosos sobre o progresso dos estudantes e apontar os ajustes pedagógicos necessários para melhorar a eficácia durante o processo de ensino e aprendizagem;
- O quarto passo refere-se a análise e apresentação dos resultados adquiridos. Os dados foram obtidos por meio de uma avaliação diagnóstica realizada antes e após a intervenção pedagógica. Essa abordagem possibilitou avaliar o progresso dos estudantes durante o período de aprendizado de algoritmos e lógica de programação, identificando oportunidades de aprimoramento no ensino e verificando a eficácia da estratégia educacional implementada. A análise dos resultados pré e pós-intervenção está mais bem detalhada na Seção 4.

Durante a etapa de produção das peças de quebra-cabeça para o projeto, decidiu-se utilizar uma impressora 3D para confeccionar os modelos. Porém, no decorrer do processo de impressão, ocorreram alguns contratemplos, resultando em uma parte ser impressa em papel para poder dar sequência na pesquisa. As representações dos algoritmos foram feitas em pseudocódigo, permitindo



a criação de algoritmos que abordam conceitos fundamentais de lógica de programação. Essa abordagem estratégica garante que o quebra-cabeça seja uma ferramenta educacional capaz de estimular o desenvolvimento do pensamento computacional nos estudantes (Silva *et al.*, 2020).

### 3.1 ALGORITMOS UTILIZADOS

Para que os estudantes entendessem o processo e o funcionamento de algoritmos, foram criadas algumas questões que envolvem o pensamento lógico por parte dos educandos e a partir disso, eles posam resolver as questões. As representações das respostas desses algoritmos foram feitas em pseudocódigo.

De acordo com Dourish (2022), o pseudocódigo oferece uma representação abstrata e geral de um algoritmo. O pseudocódigo tem a capacidade de ser convertido para qualquer linguagem de programação, superando as peculiaridades de cada uma delas. Funciona como uma espécie de esboço do algoritmo, pronto para ser transformado em programas específicos que ainda não foram escritos, por meio da mediação de um código particular.

No total, foram criados dez algoritmos para a resolução do quebra-cabeça e da atividade avaliativa, abrangendo desde problemas simples, como a soma de idades, até questões mais desafiadoras que envolvem cálculos de porcentagem, como aumento salarial ou desconto em produtos. Esses algoritmos podem incentivar os estudantes a montarem o quebra-cabeça, bem como promover o desenvolvimento do raciocínio lógico e do pensamento computacional. Abaixo estão listadas as questões elaboradas que foram respondidas pelos estudantes:

- Faça um algoritmo que leia a idade de duas pessoas, em seguida calcule a soma dessas idades. Por fim exiba na tela o seu resultado.
- Faça um algoritmo que leia um número inteiro e no final exiba o seu antecessor e o seu sucessor.
- Faça um algoritmo que leia os valores para três variáveis A, B, C. Inicialmente, o algoritmo deve trocar os valores entre B e C. Logo após, a variável A deve receber o valor inicial de B. No final, mostrar as 3 variáveis.
- Faça um algoritmo que leia o preço de um produto, calcule e mostre o seu preço promocional, com 5% de desconto.
- Faça um algoritmo que leia o salário de um funcionário, calcule e mostre o seu novo salário, com 15% de aumento.

### 3.2 CRIAÇÃO DAS PEÇAS

Algumas pesquisas recentes exploram o uso de ferramentas que permitem aos estudantes interagirem ativamente em seu processo de aprendizado o (Gardeli; Vosinakis, 2019; Im; Rogers, 2021). Além disso, a programação em blocos, como no caso do Scratch (Resnick *et al.*, 2009), também é amplamente utilizada para o ensino de lógica de programação. Uma outra ferramenta que combina a manipulação de peças com o uso da programação é a robótica Lego. Ela possibilita



aos estudantes montar seus próprios robôs e programar seus movimentos por meio de programação em blocos (Antonello *et al.*, 2020).

Com o objetivo de oferecer uma abordagem semelhante aos estudantes, surgiu a ideia de utilizar um quebra-cabeça como metodologia de ensino e aprendizagem, proporcionando uma experiência tangível que facilite o aprendizado dos estudantes. Mesmo estando em um mundo em constante evolução tecnológica (Haldorai; Murugan; Ramu, 2021), essa realidade não se aplica a todos os estudantes do curso, especialmente por serem de baixa renda, muitos dos quais não têm acesso a computadores ou *smartphones* para estudos além da sala de aula. Considerando esse cenário, as peças foram desenvolvidas de modo a permitir que os estudantes aprendam durante as aulas e, se necessário, levem-nas para atividades extraclasse.

A Figura 5 ilustra um exemplo do quebra-cabeça impresso. Os algoritmos foram impressos contendo aproximadamente 15 peças, em cada uma das peças há um trecho do pseudocódigo para que os estudantes possam memorizar o “esqueleto” de um algoritmo. A estrutura permanece na mesma sequência, variando apenas nas instruções a serem executados. A elaboração das peças foi planejada para estimular o raciocínio lógico e a reconstrução de algoritmos de forma lúdica e tangível, permitindo que os alunos identifiquem a sequência correta das instruções e compreendam o papel de cada comando dentro do fluxo lógico.

**Figura 5** – Algoritmo impresso na impressora 3D.



Fonte: Elaborada pelos próprios Autores.

### 3.3 PESQUISA DIAGNÓSTICA

Para avaliar os estudantes, foram realizadas pesquisas diagnósticas antes e depois do desenvolvimento da atividade. O questionário inicial foi organizado na plataforma *Google Forms*, incluindo as perguntas abaixo apresentadas, nas quais as respostas foram de múltipla escolha.

- Antes do início deste curso, você teve alguma experiência prévia em algoritmos ou lógica de programação?
- Você tem interesse em seguir uma carreira na área de Tecnologia da Informação?



- Você possui acesso a alguma ferramenta tecnológica para estudos, como um smartphone ou computador?

Por meio desse questionário, foi possível obter dados para identificar os conhecimentos prévios ou mesmo se eles tinham ou não alguma compreensão sobre lógica de programação e, também, o nível de interesse na área, além de caracterizar os estudantes que possuem ou não ferramentas tecnológicas (como *smartphones* e computadores) para estudos além daqueles desenvolvidos em sala de aula.

Do mesmo modo, o questionário respondido pelos estudantes após a atividade foi criado na plataforma *Google Forms*, cujo as perguntas estão abaixo apresentadas e, também, as respostas eram constituídas por múltipla escolha.

- Você conseguiu entender conceitos de lógica de programação por meio da utilização do quebra-cabeça?
- O seu interesse mudou em seguir uma carreira na área de Tecnologia da Informação?
- Qual é o seu grau de compreensão em relação a algoritmos e lógica de programação?
- Como você avalia a atividade em termos de sua experiência e percepção?
- Você tem algum comentário ou feedback que gostaria de compartilhar sobre a atividade? Se sim, por favor, escreva abaixo.

A aplicação do questionário após o desenvolvimento da atividade pelos estudantes, permitiu identificar como os educandos vivenciaram essa experiência. Isso permitiu reconhecer aqueles que internalizaram o conhecimento sobre algoritmos, mapear os que ainda enfrentam dificuldades e possibilitar o planejamento de ações para garantir avanços na compreensão, também, para esses estudantes. Além disso, por meio dos *feedbacks* obtidos, foi possível avaliar se esse formato de atividade pode ser benéfico, para os estudantes, ao longo do curso.

## 4. RESULTADOS

Nesta seção, são discutidos os principais resultados decorrentes da atividade do quebra-cabeça, resolvida pelos próprios estudantes, assim como dos questionários respondidos antes e depois da realização da prática.

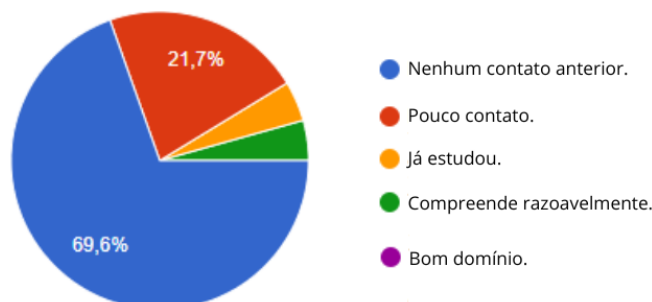
### 4.1 DIAGNÓSTICO PRÉVIO

No diagnóstico prévio foram obtidas 23 respostas dos estudantes. Na Figura 6 estão representados os resultados (em percentual) referentes a primeira pergunta *Antes do início deste curso, você teve alguma experiência prévia em algoritmos ou lógica de programação?*, considerando as alternativas que estavam apresentadas para escolha pelos respondentes. A alternativa que teve o maior percentual de escolha foi aquela em que 69,6% dos estudantes afirmaram não ter tido “nenhum contato anterior” com os conceitos básicos de algoritmo ou lógica de programação. Portanto, isso indica que a proposição da atividade utilizando peças de quebra-cabeça para introduzir esses



conceitos foi adequado, pois fica demonstrado a sua relevância para aqueles que não tiveram ou experienciaram pouco contato com algoritmos.

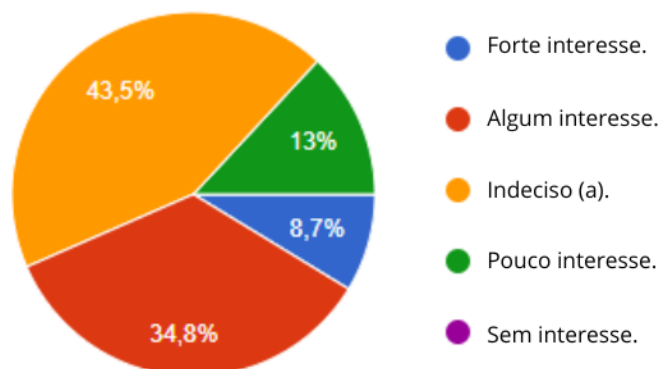
**Figura 6** – Antes do início deste curso, você teve alguma experiência prévia em algoritmos ou lógica de programação?



Fonte: Elaborada pelos próprios Autores.

A Figura 7 ilustra os resultados da segunda pergunta *você tem interesse em seguir uma carreira na área de Tecnologia da Informação?* O maior percentual de respostas foi em que cerca de 44% dos estudantes indicaram estar indecisos no momento, expressando o desejo de aprender mais sobre a área e, então, decidir se esse é um campo de seu interesse para seguir como carreira profissional. Portanto, observa-se que, a realização da atividade com o quebra-cabeça foi o primeiro contato dos estudantes com algoritmos, em que eles puderam compreender o funcionamento de uma aplicação simulada, com essa aplicação eles podem mudar e futuramente passarem a demonstrar seu interesse pela área. Muitas vezes, a escolha profissional decorre de um ato individual, de responsabilidade do próprio indivíduo, em que o seu contexto de vivência não é considerado. Entende-se que uma escolha decorre, em muitos aspectos, das vivências e experiências do sujeito, logo, se essas, em relação a uma determinada área são inexistentes ou traumáticas, muito provavelmente não será opção de escolha como profissão. Desse modo, torna-se imperioso um trabalho que possa apresentar conceitos da área da computação de forma que contribua para a compreensão por parte dos estudantes e, assim, ela torna-se uma opção como campo profissional.

**Figura 7** – você tem interesse em seguir uma carreira na área de Tecnologia da Informação?



Fonte: Elaborada pelos próprios Autores.



Os resultados referentes à terceira pergunta: *Você possui acesso a alguma ferramenta tecnológica para estudos, como um smartphone ou computador?* Alguns dados interessantes surgiram: 65,2% dos estudantes possuem *smartphones*, 26,1% têm tanto *smartphone* quanto computador, e 8,7% possuem apenas computador, sem ter acesso a um *smartphone*. Esses dados revelam potenciais desafios, já que não ter um computador em algum momento do curso ou da carreira profissional pode ser limitante. Da mesma forma, a falta de acesso a um *smartphone* pode restringir os estudos, dependendo do contexto e do módulo que o estudante estiver.

## 4.2 QUEBRA-CABEÇA

A resolução do quebra-cabeça foi conduzida pelos estudantes individualmente. Primeiramente, os exercícios foram distribuídos e, após a sua resolução, uma correção era realizada. Em todos os casos, os estudantes conseguiram resolver os quebra-cabeças com sucesso. As fotos dos grupos trabalhando na atividade estão ilustradas nas Figura 9 e 10, pode-se notar que em determinados momentos os alunos interagiram para conseguir resolver os desafios propostos.

Durante a realização da atividade puderam ser feitas algumas observações. Por exemplo, a cooperação dos estudantes para a montagem do quebra-cabeça, cada um tinha o seu próprio quebra-cabeça, com a quantidade exata de peças para resolvê-lo, porém em determinado momento, quando algum dos educandos identificou uma dificuldade para montá-lo, passou a auxiliar o colega do lado na resolução, fazendo uso até mesmo dos cadernos para uma possível solução do algoritmo antes da montagem do quebra-cabeça. Outra observação foram alguns comentários dos estudantes, questionando se os quebra-cabeças poderiam ser levados por eles para estudos extraclasses. Esse comentário foi algo surpreendente, pois um dos objetivos propostos para a aplicação dessa atividade seria justamente que os educandos utilizassem o quebra-cabeça como uma ferramenta para estudos além da sala de aula.

**Figura 8** – Grupo montando o quebra-cabeça.



Fonte: Elaborada pelos próprios Autores.



Como conclusão da pesquisa os estudantes foram desafiados a responder um trabalho avaliativo, contendo quatro questões, que contemplaram os conceitos básicos aprendidos desenvolvendo o quebra-cabeça. Abaixo são apresentadas as questões a qual os educandos tiveram que solucionar.

- **Questão 1 (peso 2):** A locadora de carros precisa de ajuda para cobrar seus serviços. Escreva um programa que pergunte a quantidade de Km percorridos por um carro alugado e a quantidade de dias pelos quais ele foi alugado, Faça um algoritmo que calcule o preço total a pagar, sabendo que o carro custa R\$ 90,00 por dia e R\$ 0,20 por Km rodado.
- **Questão 2 (peso 3):** Associe as letras dos tipos de dados com os respectivos tipos de dados.

**Figura 9** – Grupo montando o quebra-cabeça.



Fonte: Elaborada pelos próprios Autores.

- **Questão 3 (peso 3):** O custo ao consumidor de um carro novo é a soma do custo de fábrica com a porcentagem do distribuidor e dos impostos, ambos aplicados ao custo de fábrica. Supondo que a porcentagem do distribuidor seja 28% e dos impostos 45%, faça um algoritmo que leia o custo de fábrica de um carro e após calcular, mostrar o custo ao consumidor.
- **Questão 4 (peso 2):** Faça um algoritmo que leia o valor da hora aula e a quantidade de horas aulas ministradas por um professor no período de um mês. Calcular o salário bruto, o desconto do imposto correspondente a 27% do salário bruto, e o salário líquido. No final, mostrar o salário bruto, o valor do desconto, e o salário líquido.

Na Tabela 1 são apresentados os dados referentes as notas obtidas pelos estudantes da turma atual e da turma anterior, após as correções dos trabalhos avaliativos, realizada na mesma disciplina. Porém na turma anterior não foi promovida essa atividade com o quebra-cabeça. Quando comparadas os dados das duas turmas, os resultados mostram uma diferença significativa. Da turma



que realizou a atividade com o quebra-cabeça, sete estudantes atingiram uma nota maior que 8, em contraste, a turma anterior somente 3 educandos conseguiram essa nota. Desse modo, fica evidente que a utilização dessa ferramenta teve uma influência positiva na assimilação e construção do conhecimento desses estudantes, possibilitando que o uso dessa ferramenta seja algo recorrente nas disciplinas de algoritmos e lógica de programação no curso.

Outros dados interessantes de serem analisados é que, conforme a Tabela 1, na turma atual, cinco estudantes alcançaram notas entre 6 e 8. Em contraste, na turma anterior, apenas um estudante obteve nota entre 6 e 8. Considerando todos os educandos que conquistaram notas acima de 6, são mais da metade da turma. Portanto, isso reforça a importância da ferramenta na construção do conhecimento desses educandos, porém ainda existe um número significativo de estudantes que ficaram com notas abaixo de 6, o que demanda uma análise criteriosa para compreender as razões e dificuldades enfrentadas por eles durante o processo de ensino e aprendizagem.

**Tabela 1** – Notas dos trabalhos por turmas.

| Turma Atual  |           |           |           |           |     | Turma Anterior |     |
|--------------|-----------|-----------|-----------|-----------|-----|----------------|-----|
| Estudantes   | Questão 1 | Questão 2 | Questão 3 | Questão 4 | NF  | Estudantes     | NF  |
| Estudante 1  | 0         | 2.4       | 0         | 0         | 2.4 | Estudante 1    | 7.8 |
| Estudante 2  | 0.7       | 2.4       | 3         | 2         | 8.1 | Estudante 2    | 9.5 |
| Estudante 3  | 1.5       | 1.8       | 2.2       | 1         | 6.5 | Estudante 3    | 10  |
| Estudante 4  | 0         | 2.4       | 0         | 0         | 2.4 | Estudante 4    | 1.3 |
| Estudante 5  | 0.7       | 2.4       | 3         | 2         | 8.1 | Estudante 5    | 1   |
| Estudante 6  | 1.5       | 1.8       | 2.2       | 1         | 6.5 | Estudante 6    | 1.5 |
| Estudante 7  | 1         | 2.7       | 2.5       | 2         | 8.1 | Estudante 7    | 2.1 |
| Estudante 8  | 0.7       | 2.4       | 3         | 2         | 8.1 | Estudante 8    | 1.3 |
| Estudante 9  | 0.5       | 2.7       | 1.5       | 2         | 6.7 | Estudante 9    | 9.6 |
| Estudante 10 | 1         | 2.7       | 2.5       | 2         | 8.1 | Estudante 10   | 2.8 |
| Estudante 11 | 1         | 2.7       | 3         | 2         | 8.6 | Estudante 11   | 1.4 |
| Estudante 12 | 1         | 2.1       | 1.5       | 0         | 4.5 | Estudante 12   | 3.8 |
| Estudante 13 | 0.75      | 2.8       | 2.5       | 2         | 8   | Estudante 13   | 3.2 |
| Estudante 14 | 0.5       | 3         | 0         | 0         | 3.5 | Estudante 14   | 2.2 |
| Estudante 15 | 0.7       | 2.1       | 1         | 0         | 3.8 | Estudante 15   | 1.9 |
| Estudante 16 | 1.5       | 1.8       | 2.2       | 1         | 6.5 | Estudante 16   | 2.2 |
| Estudante 17 | 1         | 2.1       | 1.5       | 0         | 4.5 | Estudante 17   | 0.9 |
| Estudante 18 | 0         | 2.4       | 1.5       | 1         | 4.9 | Estudante 18   | 0.9 |
| Estudante 19 | 0         | 2.4       | 0         | 0         | 2.4 | Estudante 19   | 4.3 |
| Estudante 20 | 0.7       | 2.7       | 2.5       | 0         | 5.9 | Estudante 20   | 1.3 |
| Estudante 21 | 0.5       | 3         | 0         | 0         | 3.5 |                |     |
| Estudante 22 | 0         | 2.4       | 0         | 0         | 2.4 |                |     |

Fonte: Elaborada pelos próprios Autores.



#### 4.3 DIAGNÓSTICO PÓS

Ao término da atividade, os estudantes responderam um questionário, totalizando 23 respostas. Os resultados da primeira pergunta (*Você conseguiu entender conceitos de lógica de programação por meio da utilização do quebra-cabeça?*). A maioria dos estudantes, correspondendo a 35%, afirmou que "Sim, conseguiram compreender os conceitos de lógica de programação de forma clara com a ajuda da ferramenta". Enquanto 30% responderam que "Sim, em partes, a ferramenta me auxiliou na compreensão, mas ainda tenho algumas dúvidas". Apenas um estudante indicou que a atividade não contribuiu para sua compreensão em lógica de programação. Ao somar as duas primeiras respostas, constata-se que 65% dos estudantes conseguiram, de alguma forma, compreender os conceitos básicos de algoritmos e lógica de programação. Portanto, esses dados são interessantes de serem comparados com os resultados do trabalho avaliativo, pois mais da metade da turma obteve nota superior a 6, comprovando a internalização do conteúdo pelos estudantes.

A segunda pergunta (*O seu interesse mudou em seguir uma carreira na área de Tecnologia da Informação?*), foi semelhante a utilizada no questionário prévio. Observa-se um aumento positivo nas respostas: o número de estudantes que afirmaram ter um forte interesse na área passou de 8,7% para 10%. Aqueles que demonstraram algum interesse na área passaram de 34,8% para 40%. Além disso, os estudantes indecisos, que anteriormente representavam 43,5%, passaram a ser 25%. Esses dados sugerem que, além de auxiliar os estudantes a compreenderem o conteúdo, a ferramenta contribuiu para um leve aumento no interesse deles, podendo auxiliar em uma possível escolha para seguir a profissão na área da tecnologia.

Os resultados da terceira pergunta (*Qual é o seu grau de compreensão em relação a algoritmos e lógica de programação?*), 40% dos estudantes demonstraram ter um bom entendimento, 40% relataram uma compreensão básica e apenas 20% enfrentaram dificuldades em compreender completamente os conceitos. Levando em conta as duas primeiras respostas, conclui-se que 80% dos estudantes conseguiram, ao menos, alcançar uma compreensão dos conceitos básicos. Esse é um resultado positivo para a pesquisa, pois considerando essa compreensão do conteúdo os estudantes podem começar a desenvolver o pensamento computacional e, por consequência, avançar para as próximas disciplinas do curso, além de ser possível ingressar na carreira profissional na área da tecnologia.

A quarta pergunta (*Como você avalia a atividade em termos de sua experiência e percepção?*), a grande maioria dos estudantes (70%) considerou a atividade extremamente útil e esclarecedora. Cerca de 15% mostraram neutralidade em relação à atividade, enquanto somente 10% avaliaram que a atividade foi boa, mas com espaço para melhorias. Esses resultados evidenciam o grande interesse dos estudantes pela atividade, além de reforçar a importância da computação tangível como uma metodologia de ensino e aprendizagem, podendo ser aprimorada para diferentes contextos.

A última pergunta (*Você tem algum comentário ou feedback que gostaria de compartilhar sobre a atividade? Se sim, por favor, escreva abaixo.*) se refere aos feedbacks dos estudantes. Na Figura 10, alguns desses comentários são apresentados. É notável a presença de comentários interessantes, como por exemplo: "A atividade foi divertida, pois trocamos ideias sobre o assunto para



conseguirmos realizá-la, e consegui compreendê-la de forma clara”. Isso evidencia que a atividade além de ter sido uma experiência divertida para os estudantes, também permitiu interações entre colegas, o que segundo o Moreira (2009, p.19): “[...] Na perspectiva de Vygotsky, o desenvolvimento cognitivo é a conversão de relações sociais em funções mentais. Não é através do desenvolvimento cognitivo que o indivíduo torna-se capaz de socializar, é através da socialização que se dá o desenvolvimento dos processos mentais superiores”. Dessa forma, é fundamental que os educandos interajam socialmente durante a realização das atividades, contribuindo para a aprendizagem significativa.

A pesquisa evidencia que a ferramenta tangível, ao estabelecer uma mediação entre o sujeito e o objeto de conhecimento, conforme a teoria de Piaget (1996), pode ser uma estratégia eficaz para a construção do conhecimento, a superação de desafios e o aumento da motivação dos estudantes em disciplinas que historicamente apresentam altas taxas de reprovação e evasão.

**Figura 10** – Quinta pergunta do diagnóstico pós atividade.

A atividade ajudou na compreensão sobre a base dos algoritmos, mas ainda há alguns pontos que poderia melhorar.

A atividade foi divertida, pois trocamos ideias sobre o assunto para conseguir realizá-lo, e consegui compreendê-lo de forma clara.

Muito incrível o ensino, e o que estamos aprendendo, um pouco difícil de entender no início mas é muito legal o quebra cabeça!

Fiquei confuso em algumas partes mais tudo bem

Eu acho muito útil

Sem comentários.

Muito bom o ensino, e o que estamos aprendendo, só um pouco difícil de aprender no início, mas é muito legal a dinâmica do quebra-cabeça.

Fonte: Elaborada pelos próprios Autores.

## 5. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho propôs a implementação de uma ferramenta tangível, um quebra-cabeça, no contexto do curso de Desenvolvedor Mobile no SENAI CPF - Jorge Barbieux, com foco nas disciplinas de algoritmos e lógica de programação, sendo aprovado pelo comitê de ética. A ferramenta revelou-se eficaz para o processo de ensino e de aprendizagem, visto que mais da metade da turma atingiu a nota superior a seis na realização do trabalho avaliativo, o que pode impactar positivamente na permanência desses estudantes no curso, possibilitando o avanço para os módulos seguintes e,



evitando a reprovação ou evasão escolar. Além disso, demonstraram um aumento de interesse na área, podendo, futuramente, ser uma escolha para seguir como carreira profissional.

Os questionários aplicados após a atividade indicaram que o quebra-cabeça contribuiu significativamente para a compreensão dos conceitos básicos de lógica de programação e algoritmos pelos estudantes, auxiliando na construção do conhecimento. Os *feedbacks* coletados após o desafio da montagem do quebra-cabeça foram predominantemente positivos, evidenciando que a utilização de ferramentas tangíveis estimula a motivação e interação social dos estudantes para realização das atividades, promovendo uma aprendizagem significativa.

Durante a produção das peças, surgiram contratempos na impressão com a impressora 3D. Para contornar esse problema e garantir o cumprimento do prazo, foi optado por imprimir o restante das peças em papel.

Nos trabalhos futuros, a intenção é ampliar a confecção das peças 3D do quebra-cabeça, abrangendo mais conceitos de lógica de programação, como as estruturas condicionais (if, else) e laços de repetição. Além disso, planeja-se avaliar o ensino e aprendizagem dos estudantes diante dessas novas perspectivas de conteúdo.

## 5. REFERÊNCIAS

ANTONELLO, Ricardo et al. Robôs como Ferramenta Extensionista: a experiência do projeto de robótica pedagógica com a tecnologia de robôs Lego Mindstorms® do IFC-Campus Luzerna. **Extensão Tecnológica: Revista de Extensão do Instituto Federal Catarinense**, n. 12, p. 42-61, 2020.

ARAUJO, Ana Liz; ANDRADE, Wilkerson; GUERRERO, Dalton. Um mapeamento sistemático sobre a avaliação do pensamento computacional no Brasil. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2016. p. 1147.

CABELLO, A. F.; TM, Chagas. Reprovações e evasão no ensino superior-uma análise com base na metodologia do INEP. **Rev Temas Educ.** 2021; 30: 1-18

CARBAJAL, Marleny Luque; BARANAUSKAS, Maria Cecília Calani. Taprec: um ambiente de programação tangível de baixo custo para crianças. **SIMTEC-Simpósio dos Profissionais da UNICAMP**, n. 6, p. 77-77.

DE MATTOS VOGEL, Bruno Gonzaga; PEREIRA, Claudia Pinto. Trabalhando o pensamento computacional através de uma interface tangível para programação de robôs. **Revista Brasileira de Informática na Educação**, v. 31, p. 117-148, 2023.

DOURISH, Paul. The stuff of bits: An essay on the materialities of information. MIT press, 2022.

FLACH, Pâmela Ziliotto Sant'anna; BECKER, Fernando. Biologia, conhecimento e consciência: articulações possíveis na construção da aprendizagem. **Educação**, v. 39, n. 1, p. 74-82, 2016.



FRASER, Neil. Ten things we've learned from Blockly. In: **2015 IEEE blocks and beyond workshop (blocks and beyond)**. IEEE, 2015. p. 49-50.

FUSTÉ LLEIXÀ, Anna. Hypercubes: learning computational thinking through embodied spatial programming in augmented reality. 2018. Tese de Doutorado. Massachusetts Institute of Technology.

GARDELI, Anna; VOSINAKIS, Spyros. ARQuest: A tangible augmented reality approach to developing computational thinking skills. In: **2019 11th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)**. IEEE, 2019. p. 1-8.

GOYAL, Sidhant et al. Code bits: an inexpensive tangible computational thinking toolkit for K-12 curriculum. In: **Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction**. 2016. p. 441-447.

GREBOGY, Elaine Cristina; SANTOS, Icleia; CASTILHO, Marcos Alexandre. Mapeamento das iniciativas de promoção do pensamento computacional no ensino fundamental. In: **Simpósio Brasileiro de Informática na Educação (SBIE)**. SBC, 2021. p. 965-975.

HALDORAI, Anandakumar; MURUGAN, Suriya; RAMU, Arulmurugan. Evolution, challenges, and application of intelligent ICT education: An overview. **Computer Applications in Engineering Education**, v. 29, n. 3, p. 562-571, 2021.

IM, Hyejin; ROGERS, Chris. Draw2code: Low-cost tangible programming for creating ar animations. In: **Proceedings of the 20th Annual ACM Interaction Design and Children Conference**. 2021. p. 427-432.

ISHII, Hiroshi; ULLMER, Brygg. Tangible bits: towards seamless interfaces between people, bits and atoms. In: **Proceedings of the ACM SIGCHI Conference on Human factors in computing systems**. 1997. p. 234-241.

LIMA, Maria Adriana V. et al. Computação para Meninas: Pensamento Computacional com o apoio de interface tangível. In: **Women in Information Technology (WIT)**. SBC, 2023. p. 364-369.

MACKELLAR, Bonnie. App inventor for android in a healthcare IT course. In: **Proceedings of the 13th annual conference on Information technology education**. 2012. p. 245-250.

MEHROTRA, Aditya et al. Introducing a paper-based programming language for computing education in classrooms. In: **Proceedings of the 2020 ACM conference on innovation and technology in computer science education**. 2020. p. 180-186.

MOREIRA, Marco Antônio. Aprendizagem significativa: a teoria e textos complementares. Lf Editorial, 2023, p. 19.

NEGRINI, Lucio; BERNASCHINA, Sandra. **La robotica educativa nella scuola dell'obbligo ticinese**. Quaderni didattici del Dipartimento formazione e apprendimento, 2018.

NEGRINI, Lucio; BERNASCHINA, Sandra. **La robotica educativa nella scuola dell'obbligo ticinese**. Quaderni didattici del Dipartimento formazione e apprendimento, 2018.



PIAGET, J. **Biologia e conhecimento**: ensaio sobre as relações entre as regulações orgânicas e os processos cognoscitivos. 1996, p. 71. v. 2

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

SANTOS, Antunes et al. A importância do fator motivacional no processo ensino-aprendizagem de algoritmos e lógica de programação para alunos repetentes. In: **Workshop sobre Educação em Computação (WEI)**. SBC, 2015. p. 168-177.

SCHORR, Maria; BERCHT, Magda. Análise longitudinal do desempenho dos estudantes de Ensino Médio e estudantes de nível superior para Algoritmos e Programação. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2018. p. 550.

SILVA, Josivan P. et al. Turing project: An open educational game to teach and learn programming logic. In: **2020 15th Iberian Conference on Information Systems and Technologies (CISTI)**. IEEE, 2020. p. 1-6.

VIEIRA, Carlos Eduardo Costa et al. Dificuldades no processo de aprendizagem de algoritmos: uma análise dos resultados na disciplina de al1 do curso de sistemas de informação da faeterj–campus paracambi. **Cadernos UniFOA**, v. 10, n. 27, p. 5-15, 2015.

WING, Jeannette M. Pensamento computacional. **Educação e Matemática**, n. 162, p. 2-4, 2021.

**Submissão: 28/04/2024**

**Aceito: 06/10/2025**