

Utilização de Desafios para o Desenvolvimento do Pensamento Computacional no Ensino Superior: um relato de experiência

Carolina Moreira Oliveira, Roberto Pereira, Ludmilla F. O. Galvão,
Leticia Mara Peres, Ermelindo Schultz

Programa de Pós-Graduação em Informática (PPGInf)
Universidade Federal do Paraná (UFPR)
Curitiba – PR – Brasil

{carolmoliveiraa, ludmillagalvaol, ermelindo.schultz}@gmail.com

{rpereira, lmperes}@inf.ufpr.br

Abstract. *This paper presents a challenge-based methodology for the development of Computational Thinking. Four challenges were proposed to practice Computational Thinking skills, such as abstraction, decomposition, pattern recognition, algorithms, data representation, simulation, among others. This methodology aims at stimulating the ability of students entering Computer Science to solve problems, considering autonomy, motivation and teamwork. The learned lessons came from observations made during the conducted challenges and show the potentialities of Computational Thinking.*

Resumo. *Este artigo apresenta uma metodologia baseada em desafios para o desenvolvimento do Pensamento Computacional. Quatro desafios foram propostos com o objetivo de exercitar habilidades como abstração, decomposição, reconhecimento de padrões, algoritmos, representação dos dados, simulação, entre outras. O objetivo da metodologia é estimular nos alunos ingressantes em Ciência da Computação a capacidade de resolver problemas, considerando a autonomia, motivação e o trabalho em equipe. As lições aprendidas surgiram de observações realizadas durante as aulas em que os desafios foram conduzidos e revelam as potencialidades do Pensamento Computacional.*

1. Introdução

Em 1980, Seymour Papert já via o computador como um estimulador de ideias e visualizava as possibilidades de utilizá-lo para inovar o sistema educacional. Em seu trabalho, Papert (1985) afirmava que a interação era guiada pelas crianças ensinando o computador a realizar as tarefas, e nunca no sentido contrário. Assim, o computador torna-se um aliado na construção do conhecimento e influenciador do pensamento humano.

Décadas depois, ideias tais como as de Papert são rerepresentadas sob o termo “Pensamento Computacional” (PC). Em 2006, Wing define-o como um conjunto de habilidades utilizadas para resolver problemas de maneira eficiente, fornecendo ferramentas mentais encontradas no campo da Ciência da Computação. Portanto, pensar computacionalmente é visualizar um mesmo problema em diferentes níveis, e encontrar estratégias para reformular um problema difícil em problemas menores e mais fáceis de serem resolvidos – refletindo na solução de maneira lógica, algorítmica, abstrata, recursiva e paralela.

Essa abordagem inclui explorar diferentes aspectos do problema, considerando a sua complexidade e projetando soluções com os recursos disponíveis [Wing 2008]. O PC se assemelha ao pensamento lógico e matemático, sendo uma espécie de pensamento analítico sob as lentes computacionais [Xu e Tu 2011]. Ainda, é considerado um pensamento universal porque pode ser aplicado em outros campos do interesse humano, porém, refere-se à maneira de pensar que é típica da Ciência da Computação.

Abstração, decomposição, reconhecimento de padrões e desenho algorítmico são os quatro pilares do Pensamento Computacional (Figura 1), sendo a abstração considerada como a essência desse tipo de pensamento. Segundo Wing (2008), um algoritmo é a abstração de uma sequência de passos para processar entradas e produzir as saídas desejadas. Consequentemente, é o processo de abstração que estabelece quais detalhes são importantes e quais podem ser ignorados, e permite identificar qual será a melhor ferramenta a ser utilizada dentre as disponíveis para resolver um problema específico.

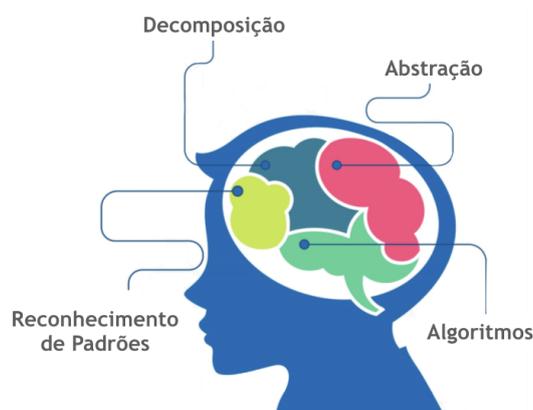


Figura 1. Pilares do Pensamento Computacional. Adaptado de [Shoop et al. 2016].

Ao desenvolver o Pensamento Computacional, uma pessoa torna-se primeiro resolvidora de problemas antes de codificadora, uma vez que desenvolve habilidades para a construção e representação da solução, focando na semântica em vez de uma sintaxe específica. Sendo possível, ao mesmo tempo, desenvolver a confiança, a persistência, a capacidade de lidar com contratempos, a comunicação e o trabalho em equipe [ISTE e CSTA 2011]. Consequentemente, pensar computacionalmente é pensar maneiras diferentes de resolver o problema por meio de uma perspectiva computacional.

Mesmo que existam diferentes interpretações acerca do termo Pensamento Computacional na literatura, há um entendimento sobre a importância e relevância do desenvolvimento desse “novo” pensar. Gouws et. al (2013) acreditam que, por meio de uma boa base no Pensamento Computacional, os alunos são capazes de desenvolver modelos mentais mais adequados e, consequentemente, poderão ser melhores Cientistas da Computação. Avila et al. (2017) mostra que, embora as pesquisas em PC se concentrem na Educação Básica, estudantes de graduação também podem se beneficiar do PC principalmente na área de Ciência da Computação e cursos afins.

Este artigo apresenta um conjunto de quatro desafios elaborados e aplicados com o objetivo de exercitar habilidades como abstração, decomposição, reconhecimento de

padrões, algoritmos, representação dos dados, simulação, entre outras. O propósito desta metodologia baseada na resolução de desafios é estimular nos alunos ingressantes nos cursos de Ciência da Computação a capacidade de resolver problemas por meio das habilidades do Pensamento Computacional.

O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta o Pensamento Computacional como um modelo de solução de problemas. A Seção 3 descreve uma metodologia de resolução de desafios para o desenvolvimento do Pensamento Computacional no Ensino Superior em Ciência da Computação. Em seguida, a Seção 4 traz as lições aprendidas das observações realizadas durante as aulas em que os desafios foram conduzidos, e a Seção 5 discorre as considerações finais.

2. Pensamento Computacional como Modelo de Resolução de Problema

O Pensamento Computacional possibilita uma aprendizagem mais ampla, favorecendo o desenvolvimento dos saberes necessários para apoiar o ensino e aprendizado dos alunos. Quando alinhado a uma metodologia focada em soluções de problemas reais, é possível através desse meio, menos abstrato, estimular a construção do conhecimento e o engajamento dos alunos. Segundo Ortiz et. al (2018), é importante considerar o contexto dos alunos para tornar a aprendizagem mais significativa, permitindo que eles trabalhem em atividades que lhes são mais atrativas e, conseqüentemente, alcançar um maior aproveitamento do conteúdo e melhorar o comprometimento deles com o processo.

Para Giraffa e Müller (2017), o aluno deve estar envolvido no processo de construção do seu conhecimento, tornando-o o principal responsável pelo seu aprendizado. As autoras afirmam que a metodologia utilizada possui impactos significativos nos alunos, de modo que as atividades devem ser diversificada para permitir a aplicação dos conceitos trabalhados. Essa abordagem torna-se útil para contornar as deficiências de formação que os alunos ingressantes apresentam, relacionadas desde a escrita e interpretação de textos, hábitos de estudo até aos aspectos de conteúdo básicos de formação.

Diante das possibilidades do Pensamento Computacional, é possível verificar que ele é altamente requerido no repertório de habilidades a serem desenvolvidas de maneira explícita pelos alunos de Computação. Mesmo que o termo “computacional” remeta ao computador e permita a analogia com a “programação”, eles não podem ser compreendidos como sinônimos. O Pensamento Computacional favorece o entendimento sobre os sistemas computacionais, como eles funcionam e como eles são programados [Barr e Stephenson 2011].

Para Barr e Stephenson (2011), outro ponto a ser considerado é a organização. Eles afirmam que para resolver um problema complexo deve-se gerar e testar sistematicamente as soluções, indiferente do domínio. Como é inviável prever todas as possibilidades, é importante selecionar os parâmetros corretos e diversificar bem os dados. Durante a modelagem da solução, é necessário realizar a depuração e a correção dos erros encontrados e, na sequência, formalizar o algoritmo para o problema em questão.

3. Proposta de Desafios de Pensamento Computacional no Ensino Superior

Com o objetivo de estimular e verificar o raciocínio lógico dos estudantes ingressantes na graduação em Ciência da Computação, foram elaborados, aplicados e analisados um

conjunto de quatro desafios. O objetivo foi provocar os alunos a pensar no processo de resolução de problemas, desenvolver habilidades do Pensamento Computacional e *soft skills*¹ de comunicação, liderança, trabalho em equipe e pensamento crítico.

Os desafios apresentados nessa seção foram realizados durante o primeiro semestre de 2019 na disciplina de Introdução à Ciência da Computação do curso de Bacharelado em Ciência da Computação da Universidade Federal do Paraná. A disciplina possuía duas turmas, com 49 alunos matriculados em cada uma delas, totalizando 98 alunos (91 homens e 7 mulheres), todos os inscritos na disciplina eram ingressantes do curso. Os desafios foram iniciados em sala de aula e finalizados à distância utilizando a plataforma *Moodle*, com duração aproximada de 20h em sala (dois encontros semanais de duas horas-aula).

Os quatro desafios foram: a *lista de chamada*, *ordenação*, *busca* e *celebridade*. Os desafios possuem como objetivo exercitar as habilidades do Pensamento Computacional, considerando abstração, decomposição, reconhecimento de padrões, formulação de algoritmos, simulação e otimização. Para a resolução, não foi necessária a utilização de nenhuma ferramenta computacional, utilizando apenas papel e caneta.

As soluções deveriam ser representadas em sequências de instruções simples e diretas em língua portuguesa, uma vez que ocorreria a interpretação literal das instruções ao executá-las. Após a resolução individual, todos os desafios possuíam um momento para socialização da solução. Em um primeiro momento, cada estudante deveria executar o algoritmo de um colega e fornecer *feedback* sobre a solução, considerando a correteza e facilidade de execução. Em um segundo momento, as propostas de soluções deveriam ser discutidas, comparadas e melhoradas pelos alunos. Se eles julgassem necessário, era possível criar uma nova solução utilizando o melhor de cada uma das resoluções iniciais.

Cada desafio era acompanhado pela contextualização e descrição do problema, pelo enunciado do desafio e das restrições a serem respeitadas no momento de construção da solução. O segundo momento do desafio representou um processo de amadurecimento da solução de maneira colaborativa, uma vez que os alunos avaliavam, argumentavam e apontavam correções entre eles. No final, os grupos de alunos conseguiam apresentar uma solução superior àquela entregue na primeira parte do desafio.

Assim, mesmo que os quatro desafios fossem diferentes, eles possuíam uma estrutura básica que deveria ser respeitada durante o processo de resolução dos desafios. Para cada desafio, os alunos deveriam: (1) caracterizar e representar o problema; (2) elaborar uma solução genérica; (3) representar a solução em passos a serem seguidos; (4) experimentar a solução para um problema específico (e.g., um conjunto de n elementos); e (5) socializar as soluções. A Figura 2 ilustra o processo para resolução dos desafios. As particularidades de cada desafio são apresentadas nas próximas seções.

3.1. Desafio 1 – Lista de Chamada

O desafio da *lista de chamada* busca resolver um problema enfrentado em sala de aula: a demora para todos os alunos assinarem a lista de presença. Para as aulas da disciplina, as turmas A e B frequentemente estavam unidas em uma mesma sala, que ficava lotada com 98 alunos. A presença na aula é verificada pela assinatura em uma lista de chamada, e cada uma das turmas possui uma lista específica. Cada estudante presente deve encontrar

¹*Soft Skills* é um termo em inglês utilizado para se referir as habilidades comportamentais.



Figura 2. Processo de Resolução dos Desafios

o seu nome em uma das listas e assinar na linha correspondente. Na sala, normalmente há n alunos, sentados em carteiras que estão distribuídas em x filas de tamanhos variados.

O desafio era propor um procedimento para passar as listas de modo que o tempo fosse otimizado – muitas vezes, a aula terminava e ainda havia pessoas que não assinaram a lista. Isso pode tanto causar problemas de atraso para a próxima aula que precisa usar a sala, quanto para a pessoa que pode esquecer de assinar a lista e receber falta. A solução precisa respeitar o fato de que cada aluno senta onde preferir, que a lista precisa ser impressa e deve ser assinada pelas pessoas presentes na sala e matriculadas na disciplina.

Habilidades do Pensamento Computacional exercitadas: decomposição, abstração, representação dos dados, paralelização e simulação.

3.2. Desafios 2 e 3 – Ordenação e Busca

O desafio da *ordenação* tem o objetivo organizar um conjunto de n personas, inicialmente não ordenadas, com base em algum critério pré-definido, podendo ser: nome, data de nascimento, identificador, entre outras informações. A solução deve respeitar que cada persona ocupa apenas uma posição na matriz unidimensional (vetor), e não pode ser colocada em cima de outra e nem ocupar duas posições ao mesmo tempo. Portanto, o desafio consiste em definir uma sequência de passos para ordenar as personas.

O desafio da *busca* visa propor uma solução para verificar se uma determinada persona está presente no conjunto ou para encontrar uma persona que atenda um critério específico. Para esse desafio, pode-se considerar que o vetor estará ou não ordenado, e esse fator impactará na escolha da estratégia para resolução do problema. Ainda, é possível realizar a ordenação antes de iniciar a busca, porém, é importante verificar se essa é a abordagem mais indicada.

Ambas as soluções devem ser generalizadas ao ponto de funcionar para qualquer

grupo de elementos, e otimizadas para não realizar comparações e trocas em excesso. Para o desafio da *Ordenação*, espera-se como entrada um vetor e um critério de ordenação, e como saída o vetor ordenado. O desafio da *Busca* tem como entrada um vetor de elementos e o elemento específico a ser localizado. Ao final da execução, será possível retornar a posição do elemento ou a informação que ele não está presente no vetor.

Para os desafios de *Ordenação* e *Busca* foram utilizados como material um conjunto de personas que os alunos produziram em uma atividade anterior, na qual cada discente criou uma persona que representa uma pessoa importante na história da Computação. As personas foram representadas na forma de cartão com as seguintes informações: foto, nome, ano de nascimento e falecimento (se for o caso), cidade e país de origem, formação acadêmica, instituição de atuação, área da Computação de maior atuação, principais feitos e contribuições para a Computação e curiosidades. A Figura 3 apresenta 4 exemplos de personas que foram criadas pelos estudantes.



Figura 3. Personas criadas pelos alunos.

Habilidades do Pensamento Computacional exercitadas: Representação dos dados, reconhecimento de padrões, generalização, algoritmos e simulação.

3.3. Desafio 4 — Celebridade

O desafio da *Celebridade* tem como o objetivo determinar se existe uma celebridade em uma festa e nomeá-la, caso exista uma. Para uma pessoa ser considerada uma celebridade, ela precisa ser conhecida por todas as pessoas presentes na festa mas não pode conhecer ninguém (por exemplo, a Ada Lovelace no SBIE).

Para descobrir se em uma festa existe uma celebridade é necessário realizar perguntas a todas as pessoas que estão presentes e realizar anotações para conseguir identificar quem é a celebridade, caso exista. Um aspecto importante é pensar na otimização, pois será necessário realizar perguntas a todos os presentes. Como restrição, o entrevistador poderia realizar apenas uma pergunta por vez (e.g., Fulano conhece Ciclano?), não podendo perguntar se conhece um conjunto de pessoas.

Habilidades do Pensamento Computacional exercitadas: abstração, representação dos dados, decomposição do problema, algoritmos.

3.4. Avaliação

A avaliação dos desafios foi realizada com base nas soluções entregues pelos alunos. Devido ao fato da grande quantidade de alunos matriculados na disciplina, e não ser possível

realizar correções e apontamentos para todos durante as aulas, cada estudante registrava a soluções do desafio em um tópico de fórum na plataforma *Moodle*. As entregas dos desafios eram divididas em duas partes: a primeira era uma entrega individual e a segunda em grupo. Ambas as entregas deveriam conter a caracterização do problema e a sequência de passos para resolver o desafio. Como as entregas eram incrementais, primeiramente os alunos realizavam o desafio, na sequência socializavam com um colega. A fase de socialização é importante para identificar quais pontos da solução inicial não estavam claros, quais estavam errados e quais deviam ser melhorados.

Após o momento de socialização, os grupos deveriam propor uma nova solução, podendo considerar o que cada uma das soluções possuía de melhor, ou ainda, uma nova solução alternativa. Para os primeiros três desafios (*Lista de chamada*, *Busca* e *Ordenação*), o requisito de entrega era uma sequência de passos em língua portuguesa ou o fluxograma da solução, enquanto para o desafio da *Celebridade* foi solicitado que a solução fosse representada por meio de fluxograma. Cada desafio exercita um conjunto de habilidades do Pensamento Computacional que pode ser relacionado com os passos necessários para solucioná-los. A Tabela 1 apresenta essas relações.

Tabela 1. Relação das habilidades do PC com os passos dos desafios.

Habilidades do PC	Passos dos Desafios
Abstração	Realizar interpretação do problema. Identificar quais são as entradas e as saídas do problema. Eliminar os detalhes desnecessários.
Decomposição	Identificar as partes de um problema e subdividi-lo quando necessário. Reaproveitar soluções desenvolvidas para resolver o mesmo problema.
Reconhecimento de Padrões	Identificar os padrões do problema e da solução. Reaproveitar as instruções por meio de repetições e condições. Transferir ideias entre desafios.
Algoritmos	Formalizar a sequência de passos necessários para resolver o problema. Refinar a solução.
Generalização	Propor soluções que possam ser utilizadas com diferentes quantidade de dados.
Paralelismo	Executar uma mesma solução simultaneamente.
Simulação	Simular as resoluções de acordo com o cenário do problema. Executar a resolução do colega e propor correções. Comparar o desempenho de soluções semelhantes.

4. Lições Aprendidas

Nesta seção, apresentamos as principais lições aprendidas com os desafios realizados a partir da experiência de propor desafios para desenvolver as habilidades de Pensamento Computacional em alunos ingressantes em um curso de Ciência da Computação. Essas lições ajudarão a refinar os desafios e inspirar a replicação dessas atividades em outros

contextos (universidades e/ou cursos). Serão apresentados, também, alguns desafios que precisam ser considerados em aplicações futuras.

A primeira lição aprendida foi sobre a importância de deixar as aulas mais dinâmicas e participativas. O primeiro desafio foi realizado bem no início do semestre, os alunos não conheciam uns aos outros e possuíam vergonha de conversar com os colegas ou expor seus apontamentos em sala. Porém, com o passar do tempo, foi possível perceber que eles começaram a se engajar melhor tanto na execução dos desafios quanto com os colegas no momento de criação da solução. Durante o momento da socialização, foi possível perceber que as discussões começaram a ser mais longas e profundas.

O primeiro desafio também mostrou que o tamanho dos grupos influencia bastante no engajamento dos alunos. Como o primeiro desafio tinha objetivo resolver um problema enfrentado pelos alunos em sala, foi planejado que as socializações seriam incrementais: primeiro os alunos propunham uma solução individual, depois em duplas, em grupos de 4 pessoas, 8 e 16 pessoas. À medida que o grupo aumentava, menos estudantes participavam e uma quantidade pequena de participantes interagiam entre si, sendo possível perceber que o engajamento era proporcionalmente inverso ao tamanho do grupo: quanto maior o grupo menos engajados estavam os estudantes.

Outra lição aprendida foi a respeito da quantidade de entregas realizadas. Como os desafios possuíam geralmente duas partes e quase sempre não era possível finalizá-los em sala, os alunos precisavam investir um tempo extraclasse tanto para representar a solução de forma clara, quanto para formalizar a entrega. Nesse sentido, um dos grandes desafios é motivá-los a aproveitar o tempo em sala e ajustar a quantidade de entregas necessárias, pois solicitar muitas entregas via *Moodle* pode desmotivar os alunos ou fazer com que enviem qualquer resposta apenas para registrar a entrega.

Foi notado que alguns estudantes se distraíam com muita facilidade, principalmente os que sentavam mais ao fundo da sala. Isso pode ter ocorrido devido ao fato de que a sala utilizada precisava ser muito grande e alguns deles ficavam muito distantes dos professores, mesmo que os professores estivessem constantemente circulando pela sala no momento da realização dos desafios. Portanto, consegue-se observar que existe uma relação do nível de atenção com o local em que esse aluno está posicionado na turma.

No momento de elaboração dos desafios, foram pensadas maneiras de como aproximar os alunos aos desafios. O desafio da *Lista de chamada* foi um problema real, enfrentado em sala, os desafios de *Ordenação* e *Busca* utilizaram como material as personas criadas pelos alunos, e o desafio da *Celebridade* trouxe a própria turma como contexto do problema (exemplo, uma festa dos calouros). Usar exemplos reais e contextualizados ajuda a promover além do engajamento e motivação uma aprendizagem mais significativa.

O fato dos alunos precisarem formalizar as suas soluções de forma que o colega conseguisse executar o algoritmo sem poder realizar inferências fez com que eles pensassem mais no momento de resolução dos desafios. A necessidade de avaliar crítica e construtivamente a solução dos colegas, apontando os erros, diagnosticando os equívocos e sugerindo melhorias à solução fez com eles desenvolvessem habilidades como argumentação e avaliação do próprio conhecimento.

No último desafio (*Celebridade*), foi possível verificar que os alunos estavam mais engajados e aproveitando melhor o tempo em sala, mostrando uma postura mais madura.

Eles ainda conseguiram desenvolver a habilidade de trabalho em equipe, utilizando de forma explícita as habilidades do Pensamento Computacional: dividindo o problema em problemas menores, mais fáceis de serem resolvidos; simplificando e identificando as reais necessidades à medida que as soluções eram desenvolvidas; negociando desde o momento de dividir as tarefas até ao mesclar as soluções ou defender as próprias ideias; e na construção da solução, trabalhando de maneira colaborativa para representar a solução.

5. Considerações Finais

Este artigo discutiu a realização de desafios para exercitar as habilidades do Pensamento Computacional e relatou lições aprendidas de uma experiência prática. Os desafios propostos apresentam uma possibilidade de aprendizagem por meio de resoluções de problemas reais e contextualizados, portanto, potencialmente interessante aos alunos. Além dos ganhos com o desenvolvimento do Pensamento Computacional, foi possível que os alunos exercitassem também as *soft skills* de colaboração, comunicação, pensamento criativo, trabalho em equipe, negociação e administração do tempo.

Exercitar cada uma das habilidades do Pensamento Computacional ajuda o aluno a reconhecer a importância de pensar de diferentes formas, considerando diferentes perspectivas no momento de propor uma solução. Os momentos de trabalho em equipe exigiam a colaboração entre os alunos à medida que eles se envolviam com as soluções dos colegas. Assim, de maneira colaborativa, os alunos avançavam na construção de modelos mentais mais significativos para resolução de problemas.

Com a utilização de desafios para o desenvolvimento das habilidades do Pensamento Computacional, pôde-se observar que os alunos estavam mais engajados durante a realização das atividades. Também foi possível perceber evolução entre o primeiro desafio realizado e o último, principalmente na postura de cada aluno durante o trabalho em equipes e a responsabilidade com o tempo em sala destinado à tarefa. Ainda foi possível perceber que as soluções possuíam um certo amadurecimento entre as entregas.

As lições aprendidas demonstram positivamente que o método aplicado auxilia o aluno a desenvolver o hábito de pensar computacionalmente durante a resolução dos problemas. Em relação às dificuldades durante o processo de solução dos desafios, reconhece-se a importância da socialização e simulação para direcionar os alunos ao refinamento e correção das soluções. O uso de papel e caneta para criar soluções e a língua portuguesa para representá-las, reduziu preocupações para a utilização do conhecimento do Pensamento Computacional, como a sintaxe de uma linguagem de programação ou o domínio de uma ferramenta computacional.

Por isso, é possível afirmar que os estudantes de Ciência da Computação precisam desenvolver desde o primeiro momento as habilidades do pensar computacionalmente. Dessa primeira experiência, as lições aprendidas confirmam a percepção de que um curso de Pensamento Computacional, sem o uso de uma linguagem de programação, traz diferentes benefícios para estudantes de Ciência da Computação.

Como trabalhos futuros, surge a necessidade de propor mais desafios e investigar outras metodologias para o desenvolvimento do Pensamento Computacional no ensino superior. Além disso, é preciso verificar se ações para o desenvolvimento do Pensamento Computacional têm impacto no desempenho dos alunos em outras disciplinas, especialmente nas disciplinas de programação. Ainda, ao analisar o desempenho, é possível

verificar possíveis efeitos de médio e longo prazo no desempenho desses alunos. Outra possibilidade é realizar as mesmas atividades em turmas diferentes, permitindo a comparação de resultados.

Agradecimentos

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Os autores agradecem todos os alunos da disciplina de Introdução à Ciência da Computação.

Referências

- Avila, C. M. O., Bordini, A., Cavalheiro, S. A. d. C., Reiser, R. H. S., Weissahn, Y., Aguiar, M. S., da Cunha, M. M., e Foss, L. (2017). Computação na Educação Básica no Brasil: o estado da arte. *Revista de Informática Teórica e Aplicada*, 23(2):210.
- Barr, V. e Stephenson, C. (2011). Bringing Computational Thinking to K12: what is involved and what is the role of the Computer Science Education. 2(1):48–54.
- Giraffa, L. M. M. e Müller, L. (2017). Metodologia baseada em sala de Aula invertida e Resolução de Problemas Relacionado ao Cotidiano dos Estudantes: uma proposta para ensinar programação para iniciantes. *International Journal on Computational Thinking (IJCThink)*, 1(1):52.
- Gouws, L. A., Bradshaw, K., e Wentworth, P. (2013). Computational Thinking in Educational Activities: An Evaluation of the Educational Game Light-bot. pages 10–15.
- International Society for Technology in Education (ISTE) e Computer Science Teachers Association (CSTA) (2011). Operational Definition of Computational Thinking: for k-12 education.
- Ortiz, J. S. B., Oliveira, C. M., e Pereira, R. (2018). Aspectos do Contexto Sociocultural dos Alunos estão Presentes nas Pesquisas para Ensinar Pensamento Computacional? Em *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 7, page 520.
- Papert, S. (1985). *Logo: Computadores e Educação*.
- Shoop, R., Flot, J., e McKenna, J. (2016). Helping Students Build Conceptual Models. Carnegie Mellon University.
- Wing, J. M. (2006). Computational Thinking. 49(3):33–35.
- Wing, J. M. (2008). Computational Thinking and Thinking About Computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881):3717–3725.
- Xu, Z. W. e Tu, D. D. (2011). Three New Concepts of Future Computer Science. *Journal of Computer Science and Technology*, 26(4):616–624.