

## **Pensamento Computacional e a Ação Computacional por Ensino Remoto: Um relato de experiência de uso do AppInventor em meio a pandemia de COVID-19**

**Éder Jacques P. Farias<sup>1</sup>, Windson Viana de Carvalho<sup>2</sup>, Alysson Diniz dos Santos<sup>2</sup>, Maria Elisa Gomes de Matos<sup>1</sup>, Jerry Macedo Castro<sup>1</sup>, George do Amaral Rodrigues<sup>1</sup>**

<sup>1</sup>Universidade Estadual Vale do Acaraú (UVA) – Sobral, CE – Brasil

<sup>2</sup>Universidade Federal do Ceará (UFC) – Fortaleza, CE – Brasil

**Abstract.** *This paper presents a proposal for teaching Computational Thinking, combining the use of Computational Action to Design Thinking. We evaluated it in a course taught to students in the first semester of Computer Science, which also used the concepts of Emergency Remote Education. At the end of the course, each student developed a mobile application, which had as its central theme: “ Fighting the pandemic of COVID 19 ”. The applications implemented in the course were designed using Design Thinking principles and developed on the block programming platform App Inventor. The results indicate that after the course, the students showed a visible increase in knowledge about Computational Thinking and improved their motivation to learn more about the subject.*

**Resumo.** *Este artigo apresenta uma proposta de ensino de Pensamento Computacional (PC) mesclando o uso de Ação Computacional e Design Thinking. Ela foi avaliada em um curso ministrado para alunos de primeiro semestre de Ciência da Computação que também usou os conceitos do Ensino Remoto Emergencial. Ao final, cada aluno desenvolveu um aplicativo para smartphone que tinha como tema central: “O combate a pandemia do COVID 19”. Os aplicativos implementados foram projetados usando Design Thinking e desenvolvidos na plataforma de programação em blocos App Inventor. Os resultados do curso indicam que os estudantes apresentaram um visível aumento no conhecimento a respeito do PC e na motivação em aprender mais sobre o assunto.*

### **1. Introdução**

Com o aumento da inserção da tecnologia digital nos diversos setores da economia mundial, a procura por profissionais de Ciência da Computação é intensa. De acordo com a Fundação *Code.org*, por exemplo, até o fim de 2020 haverá mais de um milhão de vagas para programadores em aberto apenas nos Estados Unidos. A necessidade da formação de um grande número de profissionais dessa área continua, portanto, em evidência.

No entanto, o Curso de Ciência da Computação conta historicamente com elevadas taxas de desistência nos semestres iniciais [Vihavainen et al. 2011], o que indica que o ensino dos conteúdos básicos do Curso não é tarefa fácil. Alguns pesquisadores que já abordaram o tema atribuem tal cenário ao foco excessivo na sintaxe das linguagens de programação [Castoldi and Polinarski 2009] e ao uso de materiais didáticos pouco interessantes [Gomes and Mendes 2015].

Neste contexto, o Pensamento Computacional (PC), pode ser entendido como uma abordagem prática na resolução de problemas, habilidade de projetar sistemas e entender a relação entre o pensamento humano e os conceitos que fundamentam a Ciência da Computação [Román-González et al. 2017]. Dessa forma, ensinar PC para alunos iniciantes de Ciência da Computação apresenta-se como uma estratégia interessante, visto que pode melhorar a capacidade de reconhecer, analisar, compreender e solucionar problemas.

O aprendizado do PC requer que o estudante adquira um conjunto de conhecimentos, desenvolva habilidades e internalize atitudes [Wang et al. 2017]. Uma abordagem metodológica voltada especificamente para o ensino de PC é a Ação Computacional, que consiste na proposta de atividades de ensino situadas na realidade das vidas e comunidades dos estudantes [Tissenbaum et al. 2019].

Este trabalho detalha a concepção, aplicação e avaliação de uma abordagem de ensino de PC embasada na metodologia da Ação Computacional. Essa abordagem foi avaliada durante um curso de construção de aplicativos para *smartphone* usando o *APP Inventor* seguindo princípios do *Design Thinking*. 19 estudantes de primeiro semestre de Ciência da Computação participaram do curso. O curso foi avaliado - em caráter exploratório - quantitativamente, e foram encontrados resultados positivos com relação ao seu impacto no conhecimento e na motivação dos estudantes acerca do PC, bem como na boa qualidade dos códigos realizados pelos estudantes ao final do curso. Espera-se que os resultados detalhados neste artigo possam ajudar a inspirar pesquisadores e professores a analisar ou mesmo adotar o ensino de PC em semestre iniciais de Ciência da Computação, utilizando metodologias emergentes como a Ação Computacional.

O curso estava planejado para ser realizado de forma presencial. Com a pandemia de COVID-19, o mesmo foi adaptado para a modalidade remota. Os ajustes realizados seguiram os conceitos e ferramentas propostos pelo Ensino Remoto de Emergência (ERE) [Hodges et al. 2020]. Ressalta-se que o formato no qual o curso foi efetivado abre espaço para a discussão de ampliá-lo para um público mais amplo no futuro já que diminuíram as restrições físicas (i.e., laboratório) de execução do mesmo.

O restante deste trabalho está organizado na seguinte forma: a seção 2 apresenta a metodologia da pesquisa utilizada neste trabalho e em 3 são apresentados e discutidos os resultados obtidos. A seção 4 apresenta pesquisas relacionadas ao presente trabalho e, por fim, a seção 5 apresenta as conclusões extraídas, destacando os principais resultados obtidos, ameaças à validade e propondo alguns trabalhos futuros.

## 2. Metodologia

Este artigo relata a análise de um caso prático de uso da abordagem proposta em um curso de ensino de Pensamento Computacional. A inovação aqui proposta consiste em integrar conceitos de Ação Computacional e princípios de *Design Thinking* no ensino de PC para estudantes iniciantes de Ciência da Computação. A Ação Computacional, propõe que, ao mesmo tempo em que aprendem sobre computação, os jovens também devem ter oportunidade de fazer com que a computação tenha impacto direto em suas vidas e em suas comunidades [Tissenbaum et al. 2019]. Já o *Design Thinking* é um processo iterativo não linear usado para entender os usuários, desafiar suposições, redefinir problemas e criar, prototipar e testar soluções inovadoras [Lockwood 2010]. Ao mesclar essas duas estratégias, podemos tornar o ensino do PC mais inclusivo e motivador para os jovens

aprendizes. A avaliação deste curso baseou-se em uma estratégia quantitativa de pesquisa, de caráter exploratório. Esta seção detalha os procedimentos metodológicos adotados.

### 2.1. Questões de Pesquisa

Para guiar a concepção do curso, foram definidas as seguintes questões de pesquisa:

**Q1.** A abordagem proposta é capaz de impactar no conhecimento dos estudantes sobre o pensamento computacional?

**Q2.** A motivação dos participantes em conhecer mais sobre o pensamento computacional é modificada após o uso dessa abordagem no curso?

**Q3.** Qual a qualidade dos códigos apresentados pelos participantes do curso?

Desta forma, as atividades propostas e a avaliação realizada visaram coletar indícios para investigar estas perguntas levantadas.

### 2.2. Contexto e Perfil dos participantes

O curso foi ministrado para estudantes de uma turma do primeiro semestre do Curso de Ciência da Computação (CC) de uma universidade pública brasileira de uma cidade do interior do Brasil. É importante salientar que o curso foi aplicado antes que os estudantes tivessem contato com os conceitos básicos de programação no curso de CC. Inscreveram-se inicialmente para o curso um total de 25 estudantes, contudo, apenas 19 estudantes passaram por todas as etapas e apenas esses serão levados em consideração para a avaliação dos resultados. A média de idade dos estudantes que finalizaram o curso é igual a 18,23 anos, com desvio padrão de 0,6. Sendo 68,42% de homens e 31,58% de mulheres.

Como dito na Introdução, inicialmente o curso tinha um escopo presencial, com o uso de um laboratório de informática para o apoio ao desenvolvimento dos APPs. Contudo, por conta da pandemia do Covid-19, o projeto de intervenção precisou passar por algumas adaptações com base nos conceitos do Ensino Remoto Emergencial [Hodges et al. 2020]. No ERE, as aulas remotas de maneira síncrona são priorizadas (como no ensino presencial), usando, por exemplo, sistemas de webconferência. Os materiais são disponibilizados e as atividades e avaliações seguem usando um ambiente virtual de aprendizagem (AVA) de forma assíncrona. O uso do ERE, que inicialmente não estava planejado, possibilitou a utilização de tecnologias digitais em todo o curso, o que oferece uma perspectiva mais escalável da proposta para as suas próximas versões.

### 2.3. Intervenção

O curso teve duração de oito semanas e foi ministrado nos meses de Maio e Junho de 2020. O organograma mostrado na Figura 1 apresenta graficamente como o Ensino Remoto Emergencial e a Ação Computacional foram utilizados no projeto de intervenção proposto. Usou-se *Google Meet* para as aulas síncronas, o *Google Classroom* como AVA, o *Code.org* como plataforma inicial para programação em blocos, o *Design Thinking* para organização das ideias e prototipação dos projetos e o *APP Inventor* como plataforma de programação dos aplicativos para *smartphone*. A Figura 1 ressalta a relação entre os objetivos de cada tecnologia, o fluxo de aplicação e as ferramentas usadas.

A Tabela 1 apresenta um resumo de como se deu a organização do curso e descreve os encontros síncronos que aconteceram durante as oito semanas. Como pode-se observar,

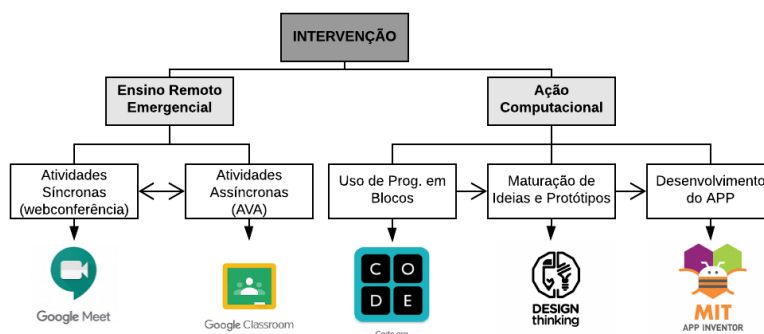


Figura 1. Arquitetura e ferramentas utilizadas no projeto de intervenção

foram feitos dois encontros em cada semana, com duração média de duas horas cada. Todas as aulas foram gravadas e dispostas para a turma no *Google Classroom*, assim como os slides e materiais de apoio, como vídeos externos, links para sites e artigos. O AVA também foi usado para o recebimento das tarefas que eram propostas no decorrer da semana, e para receber o código do aplicativo desenvolvido por cada estudante.

Na primeira semana, apresentou-se aos estudantes a formatação do curso e os conceitos iniciais sobre o que é o PC. Nas semanas 2 e 3, foi proposto aos estudantes que solucionassem desafios que tinham relação direta com o aprendizado dos principais aspectos do PC (Pensamento Algorítmico, Abstração, Divisão de Problemas e Reconhecimento de Padrões) [Grover and Pea 2013]. Neste período, com o objetivo de familiarizar os estudantes com a sintaxe de uma linguagem de programação em blocos, foi proposta a participação em um curso disponibilizado na plataforma *Code.org*. Em resumo, as 3 semanas iniciais, viu-se as bases teórico-práticas requeridas para a continuação do curso.

Da semana 4 até a semana 8, o curso se concentrou na criação de um projeto e na implementação de aplicativos para *smartphone*. Seguindo os conceitos da Ação Computacional, foi conduzida a construção de protótipos em torno da temática: “Enfrentamento da Pandemia causada pelo Covid-19”. Nos projetos, cada um dos estudantes desenvolveu e apresentou um aplicativo para ajudar no enfrentamento da pandemia do novo coronavírus, inserido na realidade da suas comunidades. A Figura 2 mostra as telas de alguns aplicativos desenvolvidos pelos estudantes no decorrer do curso.

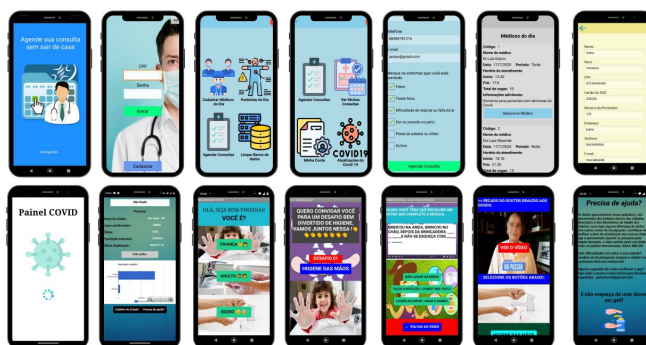


Figura 2. Telas dos projetos desenvolvidos pelos alunos

**Tabela 1. Organização do Curso**

	CONTEÚDO	OBJETIVOS
SEM. 1	Aula 1 – Apresentação do Curso	● Apresentar o formato do curso para os estudantes
	Aula 2 - Conceitos de Pensamento Computacional	● Entender o conceito de pensamento computacional
SEM. 2	Aula 3 - Programação em Blocos e desafios práticos de pensamento computacional.	● Entendimento teórico e prática da programação em blocos
	Aula 4 - Demonstração prática de programação em blocos	● Adquirir competências básicas do pensamento computacional
SEM. 3	Aula 5 - Resolução dos desafios da semana 2 e tira dúvida sobre Code.org	● Sedimentar o entendimento e a prática do pensamento computacional
	Aula 6 - Apresentação de conceitos de Algoritmos (Sequência, Seleção e Repetição)	● Esclarecer conceitos básicos do pensamento algorítmico
SEM. 4	Aula 7 - <i>Computational Action</i>	● Entendimento do conceito do <i>Computational Action</i>
	Aula 8 - Conceitos de App Inventor	● Apresentar a ferramenta de desenvolvimento do APP
SEM. 5	Aula 9 - APP Inventor - Switch de Ferramentas	● Conhecer quais ferramentas o APP inventor disponibiliza
	Aula 10 - <i>Design Thinking</i>	● Facilitar a organização das ideias
SEM. 6	Aula 11 - Reunião de Criação de Ideias para APP	● Possibilitar a maturação das ideias e a criação dos protótipos
	Aula 12- Workshop de co-criação	
SEM. 7	Aula 13 - Recebimento de protótipos	● Realizar o desenvolvimento dos Aplicativos
	Aula 14 - Tira dúvidas sobre a produção dos APP	
SEM. 8	Aula 15 - Reunião de finalização dos APP	
	Aula 16 - Entrega dos Aplicativos e fechamento do curso	

## 2.4. Materiais e Métodos

Para investigar as questões de pesquisa **Q1** e **Q2**, utilizou-se um questionário adaptado de [Oliveira et al. 2019], que foi concebido para avaliar o conhecimento e a motivação de estudantes acerca do PC. Este questionário - cujas respostas são expressas em Escala Likert - foi aplicado no início e no fim do curso com o objetivo de detectar se o conhecimento ou a motivação dos estudantes foi alterada.

Para a análise dos dados, é importante entender que o questionário é composto por uma categoria relacionada ao conhecimento do estudante (logo, relacionado à Q1) que é “definição e sala de aula” e outras três categorias relacionadas à motivação do estudante (logo, relacionado à Q2), que são: “confiança”, “interesse” e “utilidade”.

- **Definição e sala de aula** inclui itens que forneciam definições de PC e da visão do seu uso futuro em sala de aula [Yadav et al. 2014];
- A **confiança** dos estudantes nas suas capacidades de aprender habilidades relacionadas a Ciência da Computação [Hoegh and Moskal 2009];
- O **interesse** dos estudantes na Ciência da Computação [Hoegh and Moskal 2009];
- A crença dos estudantes na **utilidade** de aprender Ciência da Computação [Hoegh and Moskal 2009].

Como forma de reforçar a resposta à **Q2**, foi aplicado, ao fim do curso, um questionário motivacional que também foi utilizado em [Oliveira et al. 2019]. Neste questionário - cujas respostas também usavam Escala Likert - as perguntas estavam divididas em 4 categorias: Atenção, Relevância, Confiança e Satisfação.

- A capacidade de captar e sustentar a **atenção** dos estudantes;
- A percepção da **relevância** do conteúdo por parte dos estudantes;
- A **confiança** dos estudantes nas suas capacidades;
- A **satisfação** dos estudantes em ver seus objetivos atingidos.

Para a aplicação e coleta de dados dos questionários, utilizou-se a ferramenta *Google Forms*. Os questionários usados na pesquisa podem ser encontrados em <https://cutt.ly/4dwOj8x>. A confiabilidade e consistência interna dos questionários foram medidas com o auxílio do coeficiente Alfa de Cronbach [Cronbach 1951].

Com relação à **Q3**, o software *Code Master* propiciou suporte a avaliação dos códigos apresentados pelos estudantes. O *Code Master* é um aplicativo gratuito que permite avaliar e classificar automaticamente projetos programados com o *App Inventor* [Von Wangenheim et al. 2018]. A escolha de uma ferramenta de análise automática de código é importante, pois descarta qualquer tipo de subjetividade e enviesamento na avaliação. A opção pelo *Code Master* ocorreu também por este realizar a verificação dos códigos submetidos levando em conta os conceitos do PC. A ferramenta utiliza rubricas que analisam indiretamente as competências com base na medição de indicadores do resultado da aprendizagem do PC [Von Wangenheim et al. 2018].

### 3. Resultados e Discussões

#### 3.1. Conhecimento sobre o PC

A questão de pesquisa **Q1** foi investigada nos 06 itens do pré/pós-teste que referem-se à categoria Definição e Sala de Aula. Neste contexto, ressalta-se que os questionários de pré/pós-teste apresentaram um valor de Alpha Cronbach de 0,8949 e 0,9344, o que indica confiabilidade da consistência interna dos questionários, ou seja, os itens são suficientemente diferentes e medem elementos distintos [Cronbach 1951].

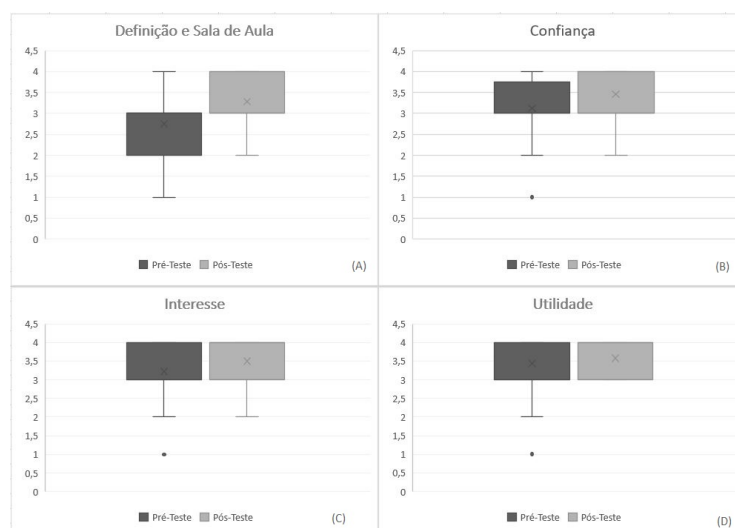


Figura 3. Declarações de Atitudes - Pré e Pós-teste

Como pode-se verificar na Figura 3A, houve uma melhora no entendimento dos participantes do curso a respeito da definição e do conceito de PC. O pré-teste apresentou mediana igual a 3,0, média igual a 2,7560 e Amplitude Interquartílica (AI) igual a

1, com primeiro quartil (Q1) igual a 2 e terceiro quartil (Q3) igual a 3. Já o pós-teste, apesar de também apresentar mediana igual a 3 e uma AI igual a 1, apresentou uma média de 3,2778, Q1 igual a 3 e Q3 igual a 4. Esse resultado demonstra uma evolução de *score* na categoria avaliada. Outra informação relevante é que o pré-teste apresentou o valor mínimo para as respostas igual a 1 (que é o valor mais baixo utilizado na escala Likert adotada), demonstrando pouco conhecimento de alguns participantes a respeito da definição de PC. Este resultado indica preliminarmente uma resposta positiva à Q1, necessitando entretanto de validações estáticas com um número maior de estudantes e de aplicações.

O gráfico apresentado na Figura 4A, ajuda a elucidar os resultados. Pode-se notar que houve uma migração das respostas para categorias mais altas na escala Likert adotada. No pré-teste, por exemplo, 53% das respostas foram “Concordo”, já no pós-teste, a 80% das respostas foram “Concordo Fortemente”.

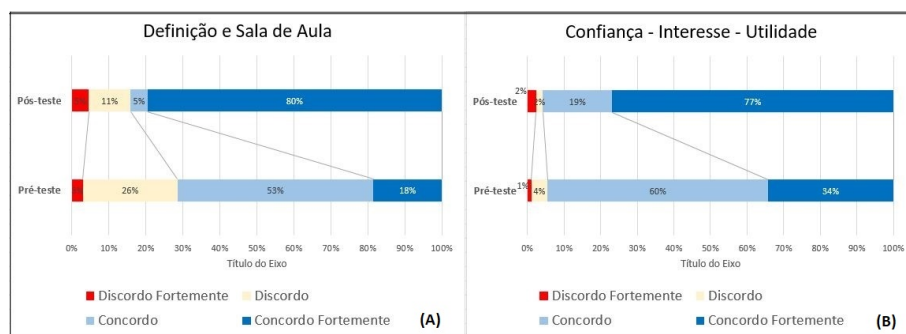


Figura 4. Respostas aos questionários Pré/Pós-teste (Escala Likert)

### 3.2. Motivação sobre o PC

Em relação a Q2, a motivação dos estudantes foi medida por meio das categorias Confiança (08 itens), Interesse (10 itens) e Utilidade (06 itens) do questionário pré/pós teste, bem como pelos 36 itens questionário motivacional aplicado ao final do curso.

Com relação ao questionário pré/pós-teste, apesar da mediana dos resultados da Confiança dos estudantes serem iguais a 3 em pré e pós-teste (Figura 3B), houve pequena melhora na média que passou de 3,1205 para 3,4583, e no no valor do Q3, que passou de 3,75 para 4. As avaliações de Interesse (Figura 3C) e Confiança (Figura 3D), tiveram o valor da mediana aumentada de 3 para 4. Nota-se também a ausência de *outliers* de valores baixos para os pós-testes, o que indica que nenhum estudante obteve pontuações negativamente fora do padrão. O gráfico apresentado na Figura 4B, faz uma compilação das respostas dadas aos questionários pré/pós-teste nas categorias Confiança, Interesse e Utilidade. No pré-teste 60% das respostas estavam concentradas em “Concordo”, já no pós-teste, essa maior concentração se deu na categoria mais alta da escola, o “Concordo Fortemente”, que obteve 77% das respostas.

O formulário de declarações motivacionais aplicado ao fim do curso, apresentou confiabilidade com um valor de Alpha Cronbach de 0,7989. Como pode ser observado na figura 5 (A), as categorias Atenção, Relevância e Satisfação obtiveram mediana igual a 5 (que é a pontuação máxima na escala Likert utilizada para este questionário) e a categoria confiança obteve mediana igual a 4. Os resultados apresentados tanto pelo pré/pós-teste

quanto pelo questionário motivacional indicam que o curso pode ter afetado positivamente a motivação dos participantes. Conjectura-se que os resultados expressivos observados - especialmente pelo questionário motivacional - tem relação com a metodologia adotada da Ação Computacional que visa especificamente em motivar os estudantes com atividades de ensino situadas na realidade das suas vidas cotidianas.



Figura 5. Declarções Motivacionais - Pós-teste (A) e Notas do Code Master(B)

### 3.3. Qualidade dos códigos

A Figura 5 (B) apresenta o gráfico *Box Plot* da pontuação de programação atribuída pelo *Code Master* aos códigos dos APPs desenvolvidos pelos alunos no fim do curso. O gráfico, responde de maneira positiva a Q3. Nota-se que, apesar de ter sido o primeiro aplicativo desenvolvido por todos os participantes do curso, o resultado final foi positivo. A média e mediana das notas atribuídas aos códigos é igual a 6,3225 e 6,145, respectivamente, com desvio padrão igual a 1,4586, o que podem ser consideradas boas notas para um primeiro código, pois o resultado apresentado pelo *Code Master* possui uma escala de 0 a 10. É válido salientar que o gráfico ainda apresentou o terceiro quartil igual a 7,71 e valor máximo igual a 8,29.

Pode-se ainda afirmar que os códigos desenvolvidos pelos estudantes possuem características que remetem ao aprendizado do PC. Essa afirmação decorre do fato de o *Code Master* medir a complexidade dos programas com base nas várias dimensões do PC, como abstração, sincronização, paralelismo, noções algorítmicas de controle de fluxo, interatividade do usuário e representação de dados [Von Wangenheim et al. 2018]. A avaliação completa obtida pelos estudantes pode ser encontrada em <https://bit.ly/30iIvkW>.

## 4. Trabalhos Relacionados

Em [Oliveira and Pereira 2019], o autor apresenta uma proposta de melhorar o aprendizado do PC utilizando uma abordagem baseada na resolução de problemas reais. Essa pesquisa teve como público alvo alunos iniciantes do curso de Ciência da Computação, tentando compreender e diminuir a alta taxa de evasão do curso. Por se tratar apenas de uma proposta esse estudo não apresentou resultados.

Já [Karling and Lisbôa 2019] dispõe de um estudo voltado ao desenvolvimento do PC em discentes do Curso de Licenciatura em Computação, utilizado também a plataforma *App Inventor*, com a finalidade de avaliar a construção de aplicativos. Nesse estudo, optou-se pela avaliação por pares, onde os próprios alunos julgaram os trabalhos dos colegas. Adicionalmente, os alunos do 6º semestre puderam aprimorar a habilidade da docência durante 4 semanas de aulas com vinte alunos do 1º período. O estudo apresentou resultados positivos, tanto para os estudantes de 1º quanto para os do 6º semestre.



A principal diferença entre os trabalhos acima citados e o presente estudo, está na metodologia do ensino do PC. Uma vez que neste trabalho, utilizou-se a Ação Computacional e o *Design Thinking* para estimular o desenvolvimento de projetos de aplicativos móveis como maior significado para os estudantes. Além disso, sua aplicação se deu por meio de ensino remoto.

## 5. Considerações Finais

O ensino do PC envolve fazer com que os estudantes adquiram a habilidade de resolver problemas, projetar sistemas e entender o comportamento humano, baseando-se nos conceitos fundamentais de CC [Wing 2006]. Neste sentido, pode-se concluir que a abordagem proposta de ensino de Pensamento Computacional com a utilização da Ação Computacional obteve resultados promissores. Essa conclusão decorre do fato de que todas as avaliações propostas apresentaram resultados satisfatórios, juntamente com a qualidade dos APPs desenvolvidos durante o processo. Em resposta as questões de pesquisa apresentadas na seção 2.1, pode-se afirmar que:

- Os conhecimentos dos participantes em relação ao pensamento computacional foram alteradas positivamente;
- Os participantes estão motivados em conhecer mais sobre a pensamento computacional depois do curso;
- Os aplicativos desenvolvidos pelos estudantes durante o curso, apresentaram, de acordo com os instrumentos utilizados, uma boa qualidade de código.

É importante destacar que os resultados reportados são preliminares, e atendem apenas ao caráter exploratório desta pesquisa. Há limitações quanto à população e amostra, uma vez que não foram seguidos os critérios de aleatoriedade na escolha da amostra, e à ausência de um grupo de controle. Além disso, para dar maior liberdade aos alunos no momento em que estivessem respondendo aos questionários, optou-se pelo anonimato nas respostas, o que impossibilitou a rastreabilidade entre os testes individuais aplicados antes e depois do curso. Apesar destes fatores, espera-se que os dados analisados ajudem a inspirar pesquisadores a abordar o PC em seus cursos, utilizando metodologias emergentes como a Ação Computacional.

Este artigo apresentou um caso da aplicação da Ação Computacional em um curso para o semestre inicial de Ciência da Computação. A continuidade desta pesquisa pode ser contemplada com a adaptação e aplicação do curso para outros públicos-alvo. Para tanto, a estratégia planejada é incremental: aplicar para cursos de STEM (Ciências, Tecnologia, Engenharia e Matemática) exceto Computação, posteriormente estender para outros Cursos de Graduação de outras áreas. Esta estratégia está alinhada com pesquisas recentes que indicam que o uso do PC pode transformar o comportamento de diversas disciplinas, profissões e setores [Wing and Stanzone 2016].

## Referências

- Castoldi, R. and Polinarski, C. A. (2009). A utilização de recursos didático-pedagógicos na motivação da aprendizagem. *I Simpósio Nacional de Ensino de Ciência e Tecnologia*, 684.
- Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3):297–334.

- Gomes, A. J. and Mendes, A. J. (2015). À procura de um contexto para apoiar a aprendizagem inicial de programação. *Educação, Formação & Tecnologias-ISSN 1646-933X*, 8(1):13–27.
- Grover, S. and Pea, R. (2013). Computational thinking in k–12: A review of the state of the field. *Educational researcher*, 42(1):38–43.
- Hodges, C., Moore, S., Lockee, B., Trust, T., and Bond, A. (2020). The difference between emergency remote teaching and online learning. *Educause Review*, 27.
- Hoegh, A. and Moskal, B. M. (2009). Examining science and engineering students’ attitudes toward computer science. In *2009 39th IEEE Frontiers in Education Conference*, pages 1–6. IEEE.
- Karling, D. A. and Lisbôa, E. S. (2019). Desenvolvimento do pensamento computacional no ensino superior. *Olhares & Trilhas*, 21(1):58–69.
- Lockwood, T. (2010). *Design thinking: Integrating innovation, customer experience, and brand value*. Simon and Schuster.
- Oliveira, C. M. and Pereira, R. (2019). Desenvolvimento do pensamento computacional no ensino superior em ciência da computação. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 8, page 1502.
- Oliveira, E., Bittencourt, R., and Trindade, R. (2019). Designing and evaluating a computational thinking course for k-12 brazilian educators. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 8, page 1094.
- Román-González, M., Pérez-González, J.-C., and Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Computers in Human Behavior*, 72:678–691.
- Tissenbaum, M., Sheldon, J., and Abelson, H. (2019). From computational thinking to computational action. *Communications of the ACM*, 62(3):34–36.
- Vihavainen, A., Paksula, M., and Luukkainen, M. (2011). Extreme apprenticeship method in teaching programming for beginners. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 93–98.
- Von Wangenheim, C. G., Hauck, J. C., Demetrio, M. F., Pelle, R., da Cruz Alves, N., Barbosa, H., and Azevedo, L. F. (2018). Codemaster—automatic assessment and grading of app inventor and snap! programs. *Informatics in Education*, 17(1):117–150.
- Wang, L., Sy, A., Liu, L., and Piech, C. (2017). Learning to represent student knowledge on programming exercises using deep learning. *International Educational Data Mining Society*.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.
- Wing, J. M. and Stanzione, D. (2016). Progress in computational thinking, and expanding the hpc community.
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., and Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE)*, 14(1):1–16.