

Estimando Habilidades de Pensamento Computacional em Ingressantes de Cursos de Computação

Ana Karina Carvalho do Nascimento¹, Victor Manoel Augusto Soares¹,
Ana Liz Souto Oliveira¹, Wilkerson L. Andrade²

¹Departamento de Ciências Exatas (DCX) – Universidade Federal da Paraíba (UFPB)

²Laboratório de Prática de Software (SPLab) –
Universidade Federal de Campina Grande (UFCG)
analiz@dcx.ufpb.br, wilkerson@computacao.ufcg.edu.br

Abstract. *This paper discusses computational thinking skills assessed in undergraduate students in Computing courses before starting programming. We compared groups of students who declared themselves to have or did not have prior programming knowledge. We designed a quasi-experiment to estimate algorithm and pattern recognition skills. As a result, we observe a statistical difference in the average scores between students who have no prior programming knowledge and those who know two or more programming languages.*

Resumo. *O objetivo deste trabalho é discutir habilidades de Pensamento Computacional em estudantes ingressantes dos cursos de Computação, antes de iniciarem a disciplina introdutória de programação. Comparamos grupos de estudantes que autodeclararam ter ou não ter conhecimentos prévios em programação. Delineamos um quasi-experimento para estimar habilidades de algoritmo e reconhecimento de padrões nesses grupos. Como resultado, é possível observar que, comparando estatisticamente os grupos, podemos considerar uma diferença nas médias de acertos entre estudantes que não têm conhecimento em programação e estudantes que têm conhecimento de duas ou mais linguagens de programação.*

1. Introdução

Desde a introdução do termo Pensamento Computacional por Wing em 2006, há discussões a respeito da sua definição e das suas habilidades obrigatoriamente relacionadas. Entendemos que Pensamento Computacional “envolve resolver problemas, projetar sistemas e compreender o comportamento humano, aproveitando os conceitos fundamentais da ciência da computação” [Wing, 2006]. No contexto brasileiro, os quatro pilares do pensamento computacional foram disseminando, incluindo as habilidades de abstração, algoritmos, decomposição e reconhecimento de padrões [Brackmann *et al*, 2016].

A literatura apresenta abordagens diversas para se observar e avaliar pensamento computacional. Oliveira & Pereira (2023) desenvolveram um artefato para auxiliar a avaliação do pensamento computacional em graduandos, considerando um conjunto de 9 habilidades: abstração, decomposição, reconhecimento de padrões, coleta e análise dos dados, algoritmo, generalização, representação da solução, simulação/teste, otimização. Já Mooney e Lockwood (2020) utilizam instrumentos para estimar

pensamento computacional em graduandos que envolviam as habilidades de abstração, decomposição, algoritmo, generalização e avaliação. Uma revisão sistemática de literatura em língua inglesa recente apontou algoritmos como uma das habilidades mais avaliadas no contexto de pensamento computacional em aprendizagem colaborativa, com destaque para Ambiente Virtual de Aprendizagem e programação em blocos [Nascimento & Castro, 2023].

Observamos assim pluralidade para definir quantas e quais habilidades precisam ser consideradas em uma abordagem de pensamento computacional. Relacionando a aprendizagem de programação, carecemos de mais evidências e reflexões sobre os seus efeitos nas habilidades de pensamento computacional que dependem (e não) de codificação [da Silva, & Falcão, 2020]. Outro estudo recente apontou que, de forma geral, existe uma baixa relação entre pensamento computacional e as ementas de disciplinas de introdução à programação de cursos superiores [Mélo *et al.*, 2023].

Nessa vertente, este artigo propõe-se a discutir e estimar as habilidades de pensamento computacional em discentes ingressantes dos cursos de Computação antes de iniciarem a primeira disciplina de programação. Nesta pesquisa exploratória, investigamos se os estudantes que se declararam ter conhecimento prévio em programação obteriam maiores médias de acertos em um teste de resolução de problemas quando comparados a estudantes sem conhecimento prévio em programação. No nosso contexto, exploramos especificamente as habilidades de algoritmos e reconhecimento de padrões.

As questões de pesquisa (QP) que norteiam esta pesquisa são: **QP1** - Os estudantes ingressantes dos cursos de computação já iniciam o curso demonstrando habilidades de pensamento computacional?; **QP2** - Os estudantes que autodeclararam ter conhecimento prévio em programação obtiveram maiores médias de acertos em um teste de resolução de problemas quando comparados a estudantes sem conhecimento prévio em programação?; **QP3** - Qual o tamanho do efeito da diferença da média de acerto quando se compara o desempenho dos estudantes entre questões que abordam algoritmos e questões que abordam reconhecimento de padrões?

Para responder a essas questões de pesquisa, empregamos o design de pesquisa *quasi*-experimento, no qual desejamos observar se a variável *conhecimento prévio em programação dos estudantes* poderia influenciar *a média de acertos em um teste de resolução de problemas envolvendo as habilidades do pensamento computacional*. Selecionamos um teste de pensamento computacional que estima habilidades de algoritmo e reconhecimento de padrões desenvolvido por Palt *et al.* (2018). Aplicamos o teste numa amostra de ingressantes de cursos de computação e analisamos os dados obtidos por meio de estatística inferencial e descritiva.

As demais seções do artigo estão organizadas conforme descrito a seguir: a Seção 2 apresenta as habilidades de pensamento computacional; a Seção 3 descreve trabalhos relacionados; a Seção 4 aborda os procedimentos metodológicos; a Seção 5 mostra os resultados; a Seção 6 apresenta as discussões que conduzem às questões de pesquisa e as ameaças à validade. Por fim, a Seção 7 apresenta as considerações finais e trabalhos futuros.

2. Habilidades do Pensamento Computacional

Para parte da comunidade acadêmica da Europa, as principais habilidades do pensamento computacional são abstração, pensamento algorítmico, decomposição, avaliação, e generalização [Csizmadia *et al.* 2015]. Definindo essas habilidades, podemos compreender que: (i) Abstração: trata da capacidade de concentra-se apenas em informações relevantes, ignorando completamente detalhes desnecessários; (ii) Avaliação: Consiste em garantir que a solução apresentada seja a mais adequada e eficiente; (iii) Decomposição: Corresponde ao processo de dividir um problema complexo em soluções menores, a fim de gerenciá-las mais facilmente; (iv) Generalização: Está ligada a identificação de padrões/semelhanças em problemas já existentes e solucionados. É uma forma de otimizar o processo, abstraindo soluções tratadas anteriormente e focando em resultados para novos problemas; e por último (v) Pensamento Algorítmico: Permite desenvolver uma solução seguindo uma série de passos ordenados para resolver o problema.

Dentre as habilidades relacionadas ao pensamento computacional no Brasil, é comum adotar o termo "quatro pilares do pensamento computacional" para se referir às habilidades de abstração, algoritmo, decomposição e reconhecimento de padrões. Esse termo foi adotado por Brackmann (2016), onde para ele, "os pilares formam a base do PC, e além de ter grande importância, são interdependentes durante o processo de formulação de soluções computacionalmente viáveis" [Brackmann, 2016, p.1]

Diferente dos quatro pilares citados por Brackmann (2016), Csizmadia *et al.* (2015) emprega uma habilidade a mais: a avaliação. Isso porque, interpretamos nesta pesquisa, que as habilidades de generalização e reconhecimento de padrões podem estar intimamente interligadas, e ambas podem ser compreendidas como uma forma de reconhecer padrões em problemas já definidos, cujas soluções já foram estabelecidas e podem ser reaplicadas fazendo adaptações. Assim, entendemos a necessidade de descrever quais habilidades estão sendo abordadas quando nos referimos à pesquisa na área de pensamento computacional.

3. Trabalhos Relacionados

Nesta seção apresentamos alguns estudos no qual também foram avaliadas as habilidades do pensamento computacional em estudantes, tanto de escolas como do ensino superior, no tocante a conhecimentos prévios em programação.

Karalar *et al.* (2021) avaliaram pensamento computacional em estudantes da oitava série na Turquia através do teste CTSCLE adaptada para uma versão turca. Além de dados relacionando a gênero, posse de computador, acesso à internet, eles relacionaram os resultados do teste no tocante aos estudantes possuem ou não conhecimento prévio em programação. Os autores notaram uma diferença estatisticamente significativa nas habilidades de algoritmo, avaliação e generalização em estudantes que já haviam tido um conhecimento prévio em programação quando comparados aos estudantes sem conhecimento em programação.

O estudo desenvolvido por Bubica *et al.* (2021) criou um modelo de avaliação para as habilidades de abstração, decomposição e pensamento algorítmico, utilizando uma abordagem de Design Centrado em Evidências. A ferramenta foi validada para estudantes de ensino médio de escolas Croatas. Os resultados mostram que o modelo apresentado foi adequado para avaliar a compreensão dos conceitos de abstração e

pensamento algorítmico, independentemente da experiência anterior com linguagens de programação. Entretanto a avaliação da habilidade de decomposição necessita ser reformulada.

Mooney e Lockwood (2020) acompanharam um grupo de graduandos durante o primeiro ano de graduação em curso de Computação. Por meio de pré e pós-teste baseado em questões do Desafio Bebras, eles avaliaram as habilidades de pensamento computacional no início e no final desse primeiro ano letivo. Como resultado, foi apontado que os alunos que tiveram bons resultados em matemática e programação tiveram um desempenho melhor em relação ao grupo que não tiveram bom desempenho em matemática e não tinham conhecimento prévio em programação.

Mélo et al (2023) aplicaram questões do Desafio Bebras em estudantes de graduação de cursos de Computação. O intuito foi avaliar se havia diferença significativa no número de acertos gerais das questões quando se comparou grupos de ingressantes sem conhecimento prévio em programação e grupos de graduandos após cursarem uma e duas disciplinas de programação. Os autores apontaram como resultado que não encontraram diferenças significativas de desempenho entre os grupos avaliados, carecendo de mais investigação.

4. Procedimentos Metodológicos

Iniciamos a pesquisa com um levantamento bibliográfico, a fim de proporcionar conhecimento do estado da arte no tema. Nossa pesquisa se caracteriza por ser de cunho exploratório, uma vez que nos permite conhecer o fenômeno que desejamos investigar. Adotamos o design de pesquisa *quasi*-experimento por ser adequado ao que desejamos observar: se a variável *conhecimento prévio em programação dos estudantes* poderia influenciar *a média de acertos em um teste* de resolução de problemas envolvendo as habilidades do pensamento computacional.

Nas seções seguintes, apresentamos as considerações sobre ética na pesquisa, os participantes, o instrumento utilizado e os procedimentos realizados na coleta e análise dos dados.

4.1. Considerações sobre Ética na Pesquisa

Antes de executarmos esse estudo, realizamos a submissão do projeto de pesquisa ao Comitê de Ética da Universidade Federal da Paraíba. Após passar pelos trâmites, o projeto foi aprovado na Plataforma Brasil sob número CAAE 31632220.0.0000.5188.

4.2. Participantes

A amostra foi composta por 53 estudantes ingressantes de duas turmas dos cursos Licenciatura em Ciência da Computação e Bacharelado em Sistemas de Informação da Universidade Federal da Paraíba, matriculados em disciplinas de introdução à programação, correspondente ao período acadêmico iniciado em julho de 2023 na referida universidade. A amostra foi obtida por conveniência e não randômica, o qual foi um dos motivos para a escolha do design de pesquisa *quasi*-experimento [Arranha & Reis, 2020]. Apenas participaram da pesquisa os discentes que se voluntariaram e não houve bonificação na nota da disciplina para os participantes.

4.3. Instrumento

O instrumento adotado neste trabalho foi o apresentado no artigo de Palts & Pedaste (2019) por ser um instrumento validado para estimar habilidades específicas de algoritmo e reconhecimento de padrões. Palts & Pedaste (2019) apontaram um instrumento de 10 questões para avaliar as habilidades de algoritmo e reconhecimento de padrões. Esse instrumento foi construído utilizando questões do Desafio Bebras. As questões originais encontram-se no idioma inglês, por isso precisaram passar por um processo minucioso de tradução e adaptação. Este processo contou com a ajuda de colaboradores externos à pesquisa, que revisaram cada questão para que não perdessem sua coerência textual. As questões passaram por um teste piloto antes de serem aplicadas na amostra atual e novas adaptações foram realizadas para versão final do instrumento em português.

O instrumento é composto por 10 questões, sendo 7 questões objetivas e 3 discursivas de respostas curtas. Destas, 5 questões trabalham a habilidade de algoritmo e 5 que trabalham a habilidade de reconhecimento de padrões. A Tabela 1 apresenta o nome das questões e respectiva habilidade abordada. O instrumento pode ser visualizado na íntegra no [link](#).

Tabela 1. Questões do instrumento por habilidade [Palts & Pedaste, 2019]

Questão	Habilidade	Questão	Habilidade
1 - Operação de Guindaste	Algoritmo	6 - Almoço de Castor	Reconhecimento de padrões
2 - Popularidade		7 - Jogo de Botões	
3 - Troca de Palavras		8 - Decorando com Chocolate	
4 - Geolocalização		9 - Alinhamento dos Lápis	
5 - Sistema de Irrigação		10 - Construindo um Chip	

4.4. Execução e coleta de dados

Após tradução e adaptação do instrumento, colocamos os itens no formato de exibição do *Google Forms*, uma vez que foi a plataforma escolhida para coleta de dados, por ser de acesso aos pesquisadores e participantes de forma gratuita. Após isso, realizamos um teste piloto com uma amostra reduzida. O intuito foi eliminar possíveis falhas, ajustar informações e verificar se os dados estavam sendo armazenados corretamente.

A coleta de dados foi realizada na primeira semana de aula do semestre acadêmico iniciado em julho de 2023 na universidade. Foi realizada uma visita às duas turmas da disciplina de Introdução à Programação (turma de ingressantes) antes do início da ministração do conteúdo das aulas de programação. Após explicar as questões éticas e a importância da pesquisa no âmbito acadêmico, os estudantes foram convidados a participar da pesquisa de forma voluntária, respondendo o Termo de Consentimento e Livre Esclarecido. O instrumento foi aplicado com tempo de resposta de até 40 minutos. Foram obtidas 53 respostas/participantes.

Ao finalizar a coleta de dados, as questões foram corrigidas como certas ou erradas de acordo com as respostas do instrumento original. Os resultados foram tabulados em planilhas para serem analisados por estatística inferencial e descritiva.

5. Resultados

Nesta seção, apresentamos os resultados referentes a caracterização da amostra e os testes estatísticos de hipótese que executamos para responder as questões de pesquisa.

5.1. Caracterização dos participantes

Dos 53 participantes, 44 se declararam do sexo masculino, 8 do sexo feminino, e uma pessoa preferiu não se identificar. Com relação ao curso, 17 estudantes pertencem à turma do curso de Sistema de Informação e 36 da turma do curso de Ciência da Computação. A idade média da amostra é de 18,45 anos.

Com relação ao número de acerto do instrumento, a média geral de acertos das 10 questões foi de 5,8 e o desvio padrão de 1,90. A Figura 1 representa a quantidade de acertos obtidos pelos estudantes em cada questão. Pode-se observar que a maior parte dos alunos acertaram 4 questões, enquanto uma pequena parte, conseguiram acertar mais de 7 questões, e nenhum conseguiu acertar todas.

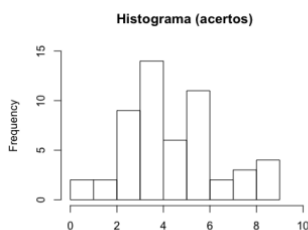


Figura 1. Quantidade de acertos por questão

Iniciamos a análise primeiramente investigando se a amostra segue ou não uma distribuição normal. Para isso, utilizamos o teste Shapiro-Wilk, com nível de significância de 5%. Adotamos como hipótese nula H_0 : *Os dados seguem uma distribuição normal*, e H_1 : *Os dados não seguem uma distribuição normal*, como hipótese alternativa. Como resultado, obtivemos $p\text{-valor} = 0.02549$. Assim, como o $p\text{-valor}$ foi inferior a 0,05, a hipótese alternativa foi aceita, significando que as amostras não seguem uma distribuição normal.

Para considerarmos as respostas oriundas dos estudantes do curso de Sistemas de Informação ($n = 17$) junto com os estudantes do curso de Ciência da Computação ($n = 36$) numa mesma base de dados, precisamos saber se podemos considerar que essas amostram (respostas do curso de Sistemas de Informação e respostas do curso de Ciência da Computação) vem da mesma população. Para isso, utilizamos o teste de Wilcoxon para a comparação das médias amostrais de acertos entre as duas amostras, (com um nível de significância de 5%), uma vez que não se segue uma distribuição normal e a amostra é não pareada.

Então, comparamos as médias de acertos dos estudantes por curso. Consideramos como hipótese nula H_0 : *A média de acertos dos estudantes dos cursos são as mesmas*. E hipótese alternativa H_1 : *A média de acertos dos estudantes dos cursos são diferentes*. Segundo o teste, foi obtido $p\text{-valor} = 0,2408$, e como este valor foi acima de 0,05, a hipótese nula foi aceita. Logo, consideramos que as médias dos estudantes que cursam Sistemas de Informação são as mesmas dos estudantes que cursam Ciência da Computação. Portanto, podemos considerar que amostra vem da mesma população, e

consequentemente considerar as respostas de ambos os cursos numa mesma base de dados.

5.2. Teste de comparação por conhecimento prévio em programação

No início da coleta de dados, junto com os dados sociodemográfico, perguntamos aos participantes sobre os conhecimentos prévios em programação. Solicitamos que os estudantes autodeclarassem se possuíam conhecimento prévio em programação. Eles deveriam selecionar no questionário inicial uma dentre as alternativas: (a) **não** possuo conhecimento prévio em programação; ou (b) possuo conhecimento em **uma** linguagem de programação (independente de qual seja); ou (c) possuo conhecimento em **duas ou mais** linguagens de programação.

Com o intuito de examinar mais detalhadamente o efeito da autodeclaração que os estudantes fizeram sobre seus conhecimentos em programação, comparamos as médias de acertos dos três subgrupos. Os três subgrupos foram: (G1) estudantes que autodeclararam **não** ter conhecimento prévio em programação, (G2) estudantes que autodeclararam ter conhecimento em **uma** linguagem, e (G3) estudantes que autodeclararam que já programam em **duas ou mais** linguagens.

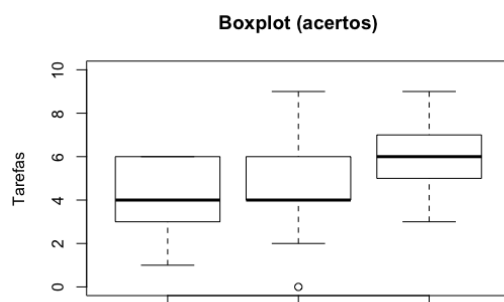


Figura 2. Boxplots das médias de acertos dos alunos divididos por subgrupos

A Figura 2 representa gráficos boxplots de acertos dos três subgrupos para uma percepção visual. Da esquerda para a direita, o primeiro boxplot retrata os acertos dos (G1) estudantes que autodeclararam não ter conhecimento prévio em programação. O segundo boxplot (do meio) representa os acertos dos (G2) estudantes que autodeclararam conhecimento em uma linguagem de programação. E o terceiro (mais a direita), ilustra os acertos dos (G3) estudantes que autodeclararam ter conhecimento em duas ou mais linguagens. Visualmente, percebemos uma diferença entre as médias de acerto do subgrupo (G3) que declarou conhecer mais de duas linguagens de programação em comparação aos outros dois grupos (que apresentam comportamentos visuais mais semelhantes).

Como há esse indício visual de diferença do subgrupo G3 frente aos demais, executamos testes de Wilcoxon, comparando hipóteses nula e alternativa, se há ou não diferença na média de acerto entre os subgrupos. Como hipótese nula, consideramos **H0**: As médias de acerto dos estudantes do subgrupo G_x e G_y são as mesmas. E como hipótese alternativa, consideramos **H1**: As médias de acertos dos estudantes do subgrupo G_x são diferentes dos estudantes do subgrupo G_y . As variáveis dos subgrupos x e y foram substituídas por G1, G2 e G3 e os valores de p-valor obtidos no teste de

Wilcoxon foram agrupados na Tabela 2, juntamente com a hipótese aceita, considerando 5% de significância.

Tabela 2. Informações de comparação entre grupos (Teste de Wilcoxon)

Grupos Comparados	P-valor	Hipótese Aceita
G1 e G2	0.4991	H0
G1 e G3	0.02894	H1
G2 e G3	0.09303	H0

G1 – Estudantes que autodeclararam **não** ter conhecimento prévio em programação
G2 – Estudantes que autodeclararam ter conhecimento em **uma** linguagem
G3 - Estudantes que autodeclararam ter conhecimento em **duas ou mais** linguagens

Observando os resultados consolidados na Tabela 2, podemos considerar a hipótese alterativa apenas para os grupos G1 e G3, ou seja, que as médias de acerto do subgrupo (G1) de estudantes que declararam não ter conhecimento prévio em programação é diferente da média de acertos dos (G3) estudantes que declararam ter conhecimento em duas ou mais linguagens de programação. Para as comparações entre os grupos G1 e G2 e entre G2 e G3, aceitamos a hipótese nula, que não há diferença significativa entre as médias de acerto. Esse resultado está em conformidade com a percepção visual dos bloxplots da Figura 2.

5.3. Testes de Comparação de Acertos por Habilidades

Neste segundo momento de análise dos dados, nos interessa investigar a média de acertos agrupada por habilidade. Lembramos que as questões de 1 a 5 avaliam a habilidade de **algoritmo** (Seção 5.3.1) e as questões de 6 a 10 avaliam a habilidade de **reconhecimento de padrões** (Seção 5.3.2).

5.3.1 Análise da Habilidade de Algoritmo

As primeiras cinco questões do instrumento avaliavam a habilidade de algoritmos. A Tabela 3 apresenta a média de acerto de cada subgrupo que está sendo investigado.

Tabela 3. Média de acerto das cinco questões de Algoritmos

Subgrupo	Média de acerto
G1 – Estudantes que autodeclararam não ter conhecimento prévio em programação	3,07
G2 – Estudantes que autodeclararam ter conhecimento em uma linguagem	3,01
G3 - Estudantes que autodeclararam ter conhecimento em duas ou mais linguagens	3,60

Com o propósito de investigar se existe diferença significativa para as médias de acerto, considerando as cinco questões de algoritmos, executamos testes de Wilcoxon, comparando hipóteses nula e alternativa, se há ou não diferença na média de acerto entre os três subgrupos, executando os cálculos com dois grupos de cada vez. Como hipótese nula, consideramos **H0**: *As médias de acertos dos estudantes do subgrupo Gx e Gy são as mesmas*. E como hipótese alternativa, consideramos **H1**: *As médias de acertos dos estudantes do subgrupo Gx são diferentes dos estudantes do subgrupo Gy*. As variáveis dos subgrupos x e y foram substituídas por G1, G2 e G3 e os valores de p-valor obtidos no teste de Wilcoxon foram agrupados na Tabela 4, juntamente com a hipótese aceita, considerando 5% de significância.

Tabela 4. Informações de comparação entre grupos (Questões Algoritmos)

Grupos Comparados	p-valor	Hipótese Aceita
G1 e G2	0.7798	H0
G1 e G3	0.04508	H1
G3 e G2	0.06441	H0

De acordo com o p-valor, podemos considerar a hipótese alternativa, ou seja, que há diferença nas médias de acertos, especificamente na habilidade de algoritmo, quando comparamos o subgrupo G1 (que não tem conhecimento em uma linguagem) com o subgrupo G3 (que tem conhecimento em duas ou mais linguagens), considerando 5% de significância. Para os demais subgrupos, não podemos considerar que há diferença nas médias de acertos.

5.3.2 Análise da Habilidade de Reconhecimento de Padrões

As cinco últimas questões do instrumento abordavam problemas relacionados ao reconhecimento de padrões. A Tabela 5 sumariza as médias de acertos dos três subgrupos. Em um primeiro momento, já percebemos que todos os três subgrupos obtiveram menor número de acertos nas questões de reconhecimento de padrões, quando comparado aos números de acertos das questões que trabalhavam a habilidade de algoritmos.

Tabela 5. Média de acerto das cinco questões de Reconhecimento de Padrões

Subgrupo	Média de acerto
G1 – Estudantes que autodeclararam não ter conhecimento prévio em programação	1,77
G2 – Estudantes que autodeclararam ter conhecimento em uma linguagem	1,66
G3 - Estudantes que autodeclararam ter conhecimento em duas ou mais linguagens	1,96

Realizamos procedimento similar para investigar se existe diferença significativa para as médias de acerto, levando em consideração apenas as cinco questões de reconhecimento de padrões. Estabelecemos as hipóteses nula e alternativa, (se há ou não diferença na média de acerto entre os três subgrupos), e executamos os testes de Wilcoxon para dois grupos de cada vez. Como hipótese nula, consideramos **H0**: *As médias de acertos dos estudantes dos subgrupos G_x e G_y são as mesmas.* E como hipótese alternativa, consideramos **H1**: *As médias de acertos dos estudantes do subgrupo G_x são diferentes dos estudantes do subgrupo G_y .* Trocamos as variáveis x e y por 1,2,3 para caracterizar cada subgrupo. Os valores obtidos no teste de Wilcoxon e as hipóteses aceitas foram agrupados na Tabela 6 para cada comparação.

Tabela 6. Informações de comparação entre grupo (Questões Reconhecimento de padrões)

Grupos Comparados	P-valor	Hipótese Aceita
G1 e G2	0.4925	H0
G1 e G3	0.1258	H0
G2 e G3	0.2452	H0

Observando os resultados da Tabela 6, concluímos que não podemos afirmar que os grupos se diferenciam na quantidade de acertos das questões relacionadas a reconhecimento de padrões.

6. Discussão dos Resultados

Retornamos as questões de pesquisa para discutir os resultados e suas limitações.

6.1 (QP1) Os estudantes ingressantes dos cursos de computação já iniciam o curso demonstrando habilidades de pensamento computacional?

Considerando que o instrumento escolhido para realizar a avaliação poderia ser resolvido sem conhecimento prévio em programação, era esperado que o grupo que autodeclarou não saber programação pudesse usar de suas habilidades adquiridas ao longo da vida para resolver as questões. De fato, 98,11% dos estudantes conseguiram acertar pelo menos uma questão do instrumento, mesmo 28,30% destes autodeclarando não ter conhecimento prévio algum em programação. Com isso, consideramos que, em ao menos uma questão, a maioria dos estudantes conseguiram resolver problemas e pensar logicamente. Esse resultado está de acordo com o esperado, uma vez que essas competências cognitivas não são exclusivas da Computação, ou seja, são competências previstas que as pessoas possam desenvolver ao longo da sua vida.

6.2 (QP2) Os estudantes que autodeclararam ter conhecimento prévio em programação obtiveram maiores médias de acertos em um teste de resolução de problemas quando comparados a estudantes sem conhecimento prévio em programação?

Na Figura 2, os boxplots apontaram que a médias de acertos dos estudantes que tem conhecimento em duas ou mais linguagens de programação é ligeiramente maior quando comparados aos outros grupos (que tem conhecimento em uma linguagem ou não possuem conhecimento prévio). E pelos resultados do teste de Wilcoxon, podemos considerar que as médias de acerto do subgrupo (G1) de estudantes que autodeclararam não ter conhecimento prévio em programação é diferente da média de acertos dos (G3) estudantes que autodeclararam ter conhecimento em duas ou mais linguagens de programação.

Pelo conhecimento de literatura, espera-se que estudantes que tenham mais experiência em linguagens de programação estejam mais aptos a resolverem problemas que incluem algoritmo. Tal comportamento foi reportado pelas pesquisas de de Karalar *et al.* (2021), Bubica *et al.* (2021) e Mooney e Lockwood (2020). Entretanto, entre o subgrupo G1 que autodeclararam não ter conhecimento prévio em programação e subgrupo G2 que autodeclararam ter conhecimento em uma linguagem, não vemos diferença expressiva, nem nos bloxplots da Figura 2. Na nossa investigação, o próprio estudante autodeclarava seu conhecimento prévio em programação. Isso pode ter sido um viés de pesquisa e levado a estudantes que apenas tiveram uma pequena noção de como a programação funciona, a declararem que sabem programar em uma linguagem, quando na verdade não possuem experiência. Além disso, o número pequeno da amostra pode ter influenciado no resultado.

Um outro fator são as características dos instrumentos de mensuração. No presente trabalho, o instrumento não exigia conhecimento prévio em programação e

abordava duas habilidades específicas: algoritmos e reconhecimento de padrões. Os instrumentos dos outros estudos podem avaliar outras habilidades de forma que fique mais propenso a pessoas que já sabem programar, a construírem a resposta correta em detrimento daqueles que ainda não se depararam com as habilidades que envolve programar além da capacidade explícita de construir algoritmos por meio de linguagem de programação.

6.3 (QP3) Qual o tamanho do efeito da diferença da média de acerto quando se compara o desempenho dos estudantes entre questões que abordam algoritmos e questões que abordam reconhecimento de padrões?

Os resultados estatísticos conseguiram identificar uma diferença nas médias de acertos nas questões envolvendo a **habilidade de algoritmo** entre o subgrupo G1 (de estudantes que declararam não ter conhecimento prévio em programação) e o subgrupo G3 (de estudantes que declararam ter conhecimento em duas ou mais linguagens de programação). Assim, para investigar a magnitude dessa diferença, escolhemos o teste D de Cohen para estimar o tamanho do efeito. Considerando um intervalo de 95% de confiança, obtivemos $D = 0.6621$, um valor considerado de magnitude de pequena a média.

Já com relação a mensuração da **habilidade de reconhecimento de padrões**, podemos constatar que não há diferença significativa na média dos subgrupos dos estudantes, nem mesmo nos estudantes que autodeclararam ter um conhecimento mais avançado em programação. Ainda constatamos que média de acerto foi menor quando comparado as questões de algoritmo, não atingindo nem a média de acerto de duas questões. Essa quantidade baixa de acerto pode ter acontecido pela forma como a habilidade de reconhecer padrões foi retratada e cobrada nas questões do instrumento. Seria importante realizar uma análise qualitativa nos itens para identificar se há algum viés.

Em particular, a pesquisa de Bublica *et al.* (2021) apontou que o instrumento criado foi apto a medir a habilidade de algoritmo; entretanto, para a habilidade de decomposição, ele precisa de adaptações. Já o estudo de Mélo *et al.* (2023) não apresentou diferença significativa de acertos no instrumento que avaliava pensamento computacional, quando comparados graduandos sem conhecimento em programação (ingressantes) com graduados que já cursaram duas ou três disciplinas de programação. Fazendo um paralelo com nossa pesquisa, o instrumento utilizado aqui conseguiu captar (alguma) diferença na habilidade de algoritmo, mas não em reconhecimento de padrões. Isso pode apontar alguma fragilidade nos respectivos instrumentos para habilidades específicas, ou ainda levar a refletir sobre a própria natureza da habilidade e suas associações. Enquanto pesquisadores, almejamos conseguir medir as habilidades isoladamente, mas para o contexto de pensamento computacional, no sentido de resolução de problemas, faz mais sentido quando as habilidades (de algoritmo, abstração, decomposição, reconhecimento de padrões) estão indissociadamente interligadas para construir uma solução, seja um problema que exija programação ou não.

6.4 Limitações da Pesquisa e Ameaças à validade

Como toda pesquisa, nosso estudo possui ameaças à validade. No tocante à validade interna, ressaltamos que, embora selecionamos um instrumento validado para mensurar habilidades de algoritmo e reconhecimento de padrões, realizamos uma tradução e adaptação, e por isso, não há garantia da validação para o português. Também ressaltamos que, para esse estudo, não consideramos outros fatores externos e vieses que podem ter influência nos dados coletados e nos resultados. Com relação à validade externa, consideramos nosso estudo explorativo, e neste estágio da pesquisa, o tamanho da amostra ainda é pequeno para generalizações. Entretanto, acreditamos que os resultados já fomentam discussões e aprendizado para próximos experimentos que abordem o mesmo problema de avaliação de pensamento computacional.

7. Considerações finais

Esta pesquisa exploratória investigou se os estudantes que autodeclararam ter conhecimento prévio em programação obteriam maiores médias de acertos em um teste de pensamento computacional, quando comparados a estudantes sem conhecimento prévio em programação. Empregamos o design de pesquisa *quasi*-experimento, observando se a variável *conhecimento prévio em programação dos estudantes* poderia influenciar a *média de acertos* no teste. O teste abordava questões de resolução de problemas envolvendo, primordialmente, as habilidades de algoritmos e reconhecimento de padrões.

Como resultado geral, observamos que os estudantes com conhecimento em mais de uma linguagem de programação tiveram diferença significativa na média de acerto quando comparados a estudantes sem conhecimento prévio em programação. Esse resultado geral está em conformidade com o estado da arte, que aponta que o conhecimento em programação influencia na resolução de problemas. Entretanto, não conseguimos observar diferenças no grupo que autodeclarou ter conhecimento em uma linguagem. Em particular, essa presença ou ausência de diferença entre os grupos estava mais diretamente relacionada a habilidade de algoritmo do que a habilidade de reconhecimento de padrões. Acreditamos que esse resultado pode estar relacionado a própria autodeclaração do estudante ter conhecimento em uma linguagem de programação, mas ter pouca experiência em, de fato, resolver problemas programando.

Esta pesquisa contribui para a discussão da mensuração de habilidades de pensamento computacional e sua relação com o conhecimento em programação. Também disponibilizamos como artefato o instrumento que foi traduzido e adaptado para o português para que possa ser empregado em outros contextos de pesquisa.

Como trabalhos futuros, pretendemos investigar outras variáveis que podem influenciar na mensuração de pensamento computacional e programação, bem como outras métricas e instrumentos que podem ser empregados para auxiliar a discussão da aquisição de pensamento computacional. Planejamos um estudo longitudinal para acompanhar esses estudantes em sua trajetória no aprendizado em programação, tanto de disciplinas iniciais como de disciplinas mais avançadas.

Disponibilidade de Artefato

O instrumento traduzido e adaptador está disponível no [link](#).

Referências

- Aranha, Eduardo; Reis, Thiago. (2020). Delineamentos Experimentais em Informática na Educação. In: Jaques, Patrícia Augustin; Siqueira; Sean; Bittencourt, Ig; Pimentel, Mariano. (Org.) *Metodologia de Pesquisa Científica em Informática na Educação: Abordagem Quantitativa*. Porto Alegre: SBC, 2020.
- Boucinha, R. M. (2017). *Aprendizagem do Pensamento Computacional e Desenvolvimento do Raciocínio*. Tese. Universidade Federal do Rio Grande do Sul.
- Brackmann, C. P., Casali, A., Barone, D. A. C., & Hernández, S. (2016). *Pensamento computacional: Panorama nas américas. XVIII Simpósio Internacional de Informática Educativa, SIIE, 2016*, 197.
- Bubica, N., Boljat, I. (2021). *Assessment of Computational Thinking – A Croatian Evidence-Centered Design Model*. Informatics in Education.
- de Melo, N. M. T., Campos, L. S., & de Araujo, E. C. (2023). *Uma visão sobre Pensamento Computacional no Ensino Superior do Brasil: características e desafios*. In Anais do XXXIV Simpósio Brasileiro de Informática na Educação (pp. 1465-1476). SBC.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., and Woollard, J. (2015). *Computational thinking: A guide for teachers*. Google Scholar.
- Grover, S. (2017). *Assessing Algorithmic and Computational Thinking in K-12: Lessons from a Middle School Classroom*.
- Karalar, H. and Alpaslan, M. M. (2021). *Assessment of Eighth Grade Students' Domain-General Computational Thinking Skills*. In International Journal of Computer Science Education in Schools, August 2021, Vol.5, No.1.
- Mélo, Á. M., Ferreira, S. D. N., Oliveira, A. L. S., & Andrade, W. L. (2023). Avaliação do Pensamento Computacional em Graduandos de Cursos de Computação: uma Disciplina de Programação faz Diferença?. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação* (pp. 47-57).
- Nascimento, E. S. O., & de Castro, M. F. (2023). *Estratégias de aprendizagem colaborativa aplicadas no ensino e avaliação do pensamento computacional: uma revisão sistemática*. *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, 1591-1604.
- Oliveira, C. M., & Pereira, R. (2023). *Coleta de Evidências do Exercício do Pensamento Computacional no Ensino Superior em Computação: um artefato de apoio*. In Anais do III Simpósio Brasileiro de Educação em Computação (pp. 300-309). SBC.
- Palts, T. and Pedaste, M. (2019) *Tasks for Assessing Computational Thinking Skills at Secondary School Level*. In Innovative Technologies and Learning (pp.216-226).
- Silva, E. L. O., e Falcão, T. P. (2020) *O Pensamento Computacional no Ensino Superior e seu Impacto na Aprendizagem de Programação*. In Anais do Workshop sobre Educação em Computação. SBC. (WEI, 2020).

Silva, I. S. F., Júnior, J. D. A. e Falcão, T. P. (2022). *Panorama Sobre Iniciativas para Promover o Pensamento Computacional no Ensino Superior Brasileiro*. Em Anais do Simpósio Brasileiro de Educação em Computação (EDUCOMP, 2022).

Wing, J. M. (2006). *Computational thinking*. Communications of the ACM, 49(3):33–35.