

## **Análise comparada de linguagens de programação para resolução de problemas matemáticos: Uma experiência no ensino médio integrado**

**Comparative analysis of programming languages for solving mathematical problems: An experience in integrated high school**

**Análisis comparativo de lenguajes de programación para la resolución de problemas matemáticos: Una experiencia en la escuela secundaria integrada**

Recebido: 30/12/2023 | Revisado: 07/01/2024 | Aceitado: 08/01/2024 | Publicado: 10/01/2024

### **Edison Garreta de Andrade**

ORCID: <https://orcid.org/0009-0001-0644-3932>  
Instituto Federal de Educação, Ciência e Tecnologia do Pará, Brasil  
E-mail: edison.andrade@ifpa.edu.br

### **Welyrson Carlos Coutinho Silva**

ORCID: <https://orcid.org/0009-0007-7139-6806>  
Instituto Federal de Educação, Ciência e Tecnologia do Pará, Brasil  
E-mail: welyrsoncarlos110905@gmail.com

### **Felipe Kailan Oliveira Lacerda**

ORCID: <https://orcid.org/0009-0005-7768-621X>  
Instituto Federal de Educação, Ciência e Tecnologia do Pará, Brasil  
E-mail: kailanlacevda@gmail.com

### **Paulo Vitor dos Santos Ferreira**

ORCID: <https://orcid.org/0009-0000-2404-4268>  
Instituto Federal de Educação, Ciência e Tecnologia do Pará, Brasil  
E-mail: pauloferreirasantos06@gmail.com

### **Marcos Eduardo Torres de Abreu**

ORCID: <https://orcid.org/0009-0005-0034-8458>  
Instituto Federal de Educação, Ciência e Tecnologia do Pará, Brasil  
E-mail: marcos.eduardo130406@gmail.com

### **João Victor Barbosa da Silva**

ORCID: <https://orcid.org/0009-0000-2364-806X>  
Instituto Federal de Educação, Ciência e Tecnologia do Pará, Brasil  
E-mail: joaobarbos29@gmail.com

### **Vitória Maria Corrêa de Sena Pereira**

ORCID: <https://orcid.org/0009-0009-6764-5843>  
Instituto Federal de Educação, Ciência e Tecnologia do Pará, Brasil  
E-mail: vitoriamcspereira@gmail.com

### **Resumo**

O presente trabalho visa investigar a possibilidade de uso de linguagens de programação para melhoria na aprendizagem de matemática, a partir da aplicação de recursos de programação para a resolução de problemas matemáticos, influenciados pelos repertórios pessoais de alunos, interações e colaborações com a equipe de trabalho. Foram utilizados problemas adaptados das Olimpíadas Brasileiras de Informática, aplicados em anos anteriores, para resolução nas linguagens C++, Java, JavaScript e Python, a critério dos participantes. A estratégia de resolução aplicada, bem como o uso ou não de materiais externos de consulta foi de livre escolha, com o incentivo à colaboração entre alunos e uso de mais de uma linguagem de programação para resolução de cada problema. Ao final, foram escolhidos dois problemas para análise quantitativa, com base nas métricas números de linhas de código, tempo de execução e número de chamadas a funções no código fonte. Foi realizada, também, análise qualitativa do processo de resolução dos problemas. Observou-se o uso de diferentes estratégias para a resolução dos problemas, influenciados sobretudo por experiências anteriores dos alunos e recursos próprios das linguagens disponíveis. Notou-se aumento de nível nas habilidades de programação e de resolução de problemas e melhoria na percepção das próprias habilidades em ambas as áreas.

**Palavras-chave:** Resolução de problemas; Modelagem; Linguagens de programação; Aprendizagem colaborativa.

### **Abstract**

The present work aims to investigate the possibility of using programming languages to improve mathematics learning, based on the application of programming resources to solve mathematical problems, influenced by students' personal repertoires, interactions and collaborations with the work team. Problems adapted from the Brazilian

Informatics Olympiads, applied in previous years, were used for resolution in the languages C++, Java, JavaScript and Python, at the discretion of the participants. The solution strategy applied, as well as the use or not of external materials for consultation, was free choice, being encouraged collaboration between students and the use of more than one programming language to solve each problem. In the end, two problems were chosen for quantitative analysis, based on the metrics number of lines of code, execution time and number of calls to functions in the source code. A qualitative analysis of the problem-solving process was also carried out. The use of different strategies to solve problems was observed, mainly influenced by students' previous experiences and resources specific to the available languages. An increase in the level of programming and problem-solving skills and an improvement in the perception of one's own skills in both areas were noted.

**Keywords:** Problem solving; Modeling; Programming languages; Collaborative learning.

### Resumen

El presente trabajo tiene como objetivo investigar la posibilidad de utilizar lenguajes de programación para mejorar el aprendizaje de las matemáticas, a partir de la aplicación de recursos de programación para la resolución de problemas matemáticos, influenciados por los repertorios personales de los estudiantes, las interacciones y colaboraciones con el equipo de trabajo. Se utilizaron problemas adaptados de las Olimpiadas Brasileñas de Informática, aplicados en años anteriores, para ser resueltos en los lenguajes C++, Java, JavaScript y Python, a criterio de los participantes. La estrategia de solución aplicada, así como el uso o no de materiales de consulta externos, fue de libre elección, fomentándose la colaboración entre los estudiantes y el uso de más de un lenguaje de programación para resolver cada problema. Al final, se eligieron dos problemas para el análisis cuantitativo, en función de las métricas número de líneas de código, tiempo de ejecución y número de llamadas a funciones en el código fuente. También se llevó a cabo un análisis cualitativo del proceso de resolución de problemas. Se observó el uso de diferentes estrategias para la resolución de problemas, influenciadas principalmente por las experiencias previas de los estudiantes y los recursos específicos de los lenguajes disponibles. Se observó un aumento en el nivel de habilidades de programación y resolución de problemas y una mejora en la percepción de las propias habilidades en ambas áreas.

**Palabras clave:** Solución de problemas; Modelado; Lenguajes de programación; Aprendizaje colaborativo.

## 1. Introdução

Os conhecimentos em computação se tornaram imprescindíveis ao cidadão do século XXI, sendo tão importantes para se viver na sociedade contemporânea quanto os conhecimentos de outras disciplinas comumente estudadas na escola (Pereira, 2020). Isto implica não somente no uso de alguma linguagem de programação, mas, principalmente, no estímulo ao desenvolvimento do pensamento computacional.

Guedes et al. (2020) destacam, ao analisar o ensino em cursos de graduação, que alunos com dificuldades em disciplinas de algoritmos e linguagens de programação também apresentam, com muita frequência, baixo desempenho na área de matemática e sugere que as duas áreas sejam trabalhadas de maneira integrada, a fim de "aproximar as zonas de conhecimento dos estudantes, facilitando assim o aprendizado". Diversos países já inseriram o estudo de programação nas grades curriculares da educação básica, seja como disciplina específica ou como estratégia empregada em outros componentes curriculares, em especial a matemática (Kilhamm et al., 2021). A publicação do parecer CNE/CEB nº 2/2022 fortaleceu essa perspectiva para a educação brasileira, ao complementar a Base Nacional Curricular Comum (BNCC) com a descrição de competências e habilidades de computação a serem desenvolvidas ao longo de todos os anos e níveis da Educação Básica (Brasil, 2022).

No contexto brasileiro, Ferreira e Duarte (2019) destacam os Institutos Federais, que compõem a Rede Federal de Educação Profissional, Científica e Tecnológica - RFEPT, dentre as principais instituições públicas do país a investirem no ensino de programação na educação básica. Dados extraídos da Plataforma Nilo Peçanha (PNP) Edição 2023 indicam a prevalência de oferta de cursos de nível médio na RFEPT no eixo tecnológico Informação e Comunicação, correspondendo a 23,36% dos cursos ofertados em toda a rede, 22,44% do total de matrículas no nível médio e 20,97% das vagas de nível médio ofertadas na rede em todo o país em 2022 (dados extraídos pelos autores de <https://www.gov.br/mec/pt-br/pnp>). Conclui-se que um número cada vez maior de estudantes de nível médio vem sendo colocado em contato com conhecimentos específicos do domínio da computação, entre eles, linguagens de programação das mais variadas.

Ao se modelar um problema matemático utilizando uma linguagem de programação, o algoritmo, ao ser traduzido para uma ou mais linguagens, pode apresentar características diferentes, em virtude das particularidades de cada uma. O número de linhas de código necessárias para resolver um problema em uma linguagem pode ser consideravelmente diferente da quantidade de linhas empregadas se outra linguagem for utilizada. Mais que isso, um mesmo problema pode ser solucionado de maneiras diferentes, seja por disponibilidade de recursos ou pelo emprego de raciocínios diversos, fazendo com que os códigos fontes gerados para modelar diferentes abordagens para a solução de um mesmo problema também apresentem diferenças consistentes.

Para Bassanezi (2015), modelagem consiste em “um processo de criação de modelos em que são definidas as estratégias de ação do indivíduo sobre a realidade”. Soistak (2016), ao realizar experiências com modelagem matemática no ensino médio profissionalizante, avalia que a resolução de problemas matemáticos com aplicações contextualizadas às áreas específicas do curso profissionalizante, a partir de uma abordagem baseada em modelagem, apresentou resultados positivos tanto para o enriquecimento curricular da área específica, quanto da matemática. Por sua vez, Carvalho e Klüber (2021) descrevem os resultados satisfatórios de uma atividade de modelagem matemática utilizando programação de computadores (ainda que não utilizem diretamente uma linguagem de programação), concluindo que os debates ocorridos nos momentos de resolução das questões e a forma como os participantes enxergaram as tarefas estavam entre os pontos mais relevantes da experiência.

O presente artigo visa investigar de que maneira estudantes de nível médio com diferentes repertórios e construtos pessoais (Kelly, 1963) utilizam recursos disponíveis em linguagens de programação para resolução de problemas matemáticos, a partir da comparação entre as soluções propostas por eles com base em métricas estabelecidas, e realizar análise qualitativa das soluções, além de autoavaliação quanto à evolução das próprias habilidades de programação e matemática.

Neste contexto escolheu-se como local de pesquisa o campus Ananindeua, do Instituto Federal de Educação, Ciência e Tecnologia do Pará (IFPA), localizado na Região Metropolitana de Belém. O IFPA Campus Ananindeua oferta cursos técnicos integrados ao ensino médio nas áreas de Informática e Meio Ambiente, com um total de 35 vagas anuais para cada curso.

O presente trabalho tem por objetivo investigar a possibilidade de uso de linguagens de programação para melhoria na aprendizagem de matemática, a partir da aplicação de recursos de programação para a resolução de problemas matemáticos, influenciados pelos repertórios pessoais de alunos, interações e colaborações com a equipe de trabalho

## **2. Metodologia**

Conforme perspectiva apresentada por Proetti (2017), o método empregado apresenta tanto caráter quantitativo, quanto qualitativo, visto coletar tanto dados subjetivos das interações entre os sujeitos, quanto dados numéricos das soluções encontradas para os problemas. Caracteriza-se como uma pesquisa-ação, haja vista a centralidade das interações e cooperações entre pesquisador e sujeitos pesquisados, essenciais para a condução da pesquisa, análise e compreensão dos resultados (Franco, 2005). Thiollent (2018) destaca a importância deste tipo de pesquisa, sobretudo na área da Educação, ao considerar que os métodos participativos se aproximam mais das situações reais do contexto educacional, quando comparados a outros modelos.

A pesquisa consistiu, inicialmente, na seleção e adaptação de problemas de edições anteriores das Olimpíadas Brasileiras de Informática, promovidas pela Universidade de Campinas. Foram selecionados 20 problemas matemáticos com níveis de dificuldade diversos, abordando conteúdos como números inteiros e racionais, plano cartesiano, estatística, sequências, análise combinatória, matrizes e geometria plana. Disponibilizou-se as linguagens de programação C++, Java,

Python e JavaScript (esta última sugerida pelos alunos participantes da pesquisa), por serem linguagens bem documentadas e de vasto uso comercial, acadêmico e científico.

Do ponto de vista pedagógico, Holanda, Coutinho & Fontes (2018) ressaltam que “o processo de ensino e aprendizagem de programação introdutória é complexo e constituído por muitas e diferentes variáveis e, portanto, faz-se necessário investigar tais variáveis e propor novas metodologias e práticas”. Deste modo, optou-se por utilizar uma abordagem que propiciasse a aprendizagem colaborativa, conforme discutido por Dillenbourg (1999), na perspectiva da teoria sociocultural de Vygotsky (2007), visto que, para este, a aprendizagem ocorre a partir da interação e com o propósito maior de se compartilhar os significados construídos, concluindo que o ensino só é de fato consumado quando este compartilhamento é efetivado. Utiliza-se, ainda, as perspectivas de Gowin (1981), visto o objetivo do processo de ensino-aprendizagem ser encontrar congruências de significados entre professor, alunos e materiais pedagógicos empregados, Freire (2019), ao compreender que a colaboração e interação entre os sujeitos proporciona novas visões e incorporações a conhecimentos já construídos ou em construção, e Rogers (1973), visto que propomos uma experiência de aprendizagem não apenas mediada, mas facilitada pelas relações interpessoais. Além disso, Pontes et al. (2022) relatam aumento no engajamento e motivação de estudantes ao realizarem experiências de resolução de problemas, quando apresentados como desafios aos alunos.

Foram selecionados discentes do Curso Técnico em Informática Integrado ao Ensino Médio, cursando o 1º ou 2º anos, com conhecimentos de lógica de programação e algum contato, ainda que superficial, com pelo menos uma das linguagens selecionadas. Deu-se total liberdade aos participantes para escolha das linguagens utilizadas (com a recomendação de que cada aluno selecionasse pelo menos duas), elaboração de estratégias próprias para a resolução dos problemas propostos, troca de ideias, cooperação e consulta a materiais impressos ou digitais. Além disso, no decorrer das resoluções foram direcionados questionamentos individuais e coletivos por parte do pesquisador.

Após as resoluções, foram selecionados dois problemas principais para análise:

- Problema 1 - resolução de equações do 2º grau: dada uma equação do 2º grau na forma  $ax^2 + bx + c = 0$ , determinar suas raízes e identificar os casos em que não existem raízes reais;
- Problema 2 - expressões de adição e subtração de números racionais: dada uma expressão de somas e subtrações de números racionais na forma fracionária, conforme Figura 1, retornar o resultado da expressão como uma fração irredutível.

**Figura 1** - Modelo de expressão de somas e subtrações de números racionais na forma fracionária.

$$\pm \frac{a_1}{b_1} \pm \frac{a_2}{b_2} \pm \dots \pm \frac{a_n}{b_n}$$

Fonte: Autoria própria.

Para cada problema, foram especificadas as devidas restrições: no Problema 1, deve-se ter o coeficiente  $a \neq 0$ ; no Problema 2, deve-se ter, nos termos da figura 1, todos os denominadores diferentes de zero.

Conforme sugere Zapalowski (2011), foram utilizadas as métricas número de linhas de código (LC) e tempo de execução (TE) para comparação dos algoritmos. Foi criada, também, uma terceira métrica denominada chamadas a funções (CF), para verificar o uso de recursos próprios de cada linguagem e como isso influencia na solução encontrada.

Para a métrica LC, realizou-se um processo de adaptação de cada código fonte, com a retirada de linhas vazias (sem código) e comentários, além de alterar a posição de chaves de início e fim de blocos de códigos, para linguagens que as utilizam, de tal modo que estes caracteres ficassem ao final de uma linha com comandos. O objetivo desta adaptação é contabilizar somente as linhas que de fato executam comandos, eliminando questões puramente sintáticas.

A medição do tempo de execução (TE) foi realizada a partir da manipulação do tempo de início e fim da execução de cada código, coletados do sistema com comandos específicos de cada linguagem. Para isto, foram criados, utilizando-se um script em Python, arquivos com número crescente de casos de teste:  $10^2$ ,  $10^4$ ,  $10^6$  e  $10^7$  casos. Cada arquivo de teste foi executado 50 vezes para cada código, nas mesmas condições de processamento e utilizando o mesmo equipamento, calculando-se a média aritmética simples do tempo de execução com precisão de até seis casas decimais.

A métrica CF foi contabilizada a partir de todas as funções utilizadas em cada código, com exceção daquelas empregadas para realizar entrada e saída de dados, comandos condicionais e de repetição (outras funções utilizadas em conjunto com estes comandos, como parâmetros, foram contabilizadas). Nesta métrica foram computadas, ainda, as chamadas a funções implementadas pelos próprios alunos.

Além disso, para verificar questões qualitativas quanto à resolução dos problemas propostos, foram aplicados questionários para detectar a melhoria nas habilidades de programação, melhoria na autopercepção quanto às próprias habilidades para resolução de problemas e contribuição da metodologia empregada para o aprendizado de matemática.

### 3. Resultados e Discussão

A linguagem C++ não foi escolhida por nenhum participante, visto não ser utilizada em seus contextos particulares. Observou-se que metade dos participantes optou por trabalhar com a linguagem Java, por já estarem familiarizados com sua estrutura, enquanto a outra metade optou por se aprofundar somente no estudo da linguagem Python. Dentre os que escolheram a linguagem Java, 67% também utilizaram a linguagem JavaScript para resolução dos problemas, enquanto 33% escolheram Python como segunda linguagem. Para a realização das análises quantitativas, optou-se pelo uso das soluções que aplicaram estratégias diferentes em cada linguagem, sendo selecionada uma solução para cada par problema/linguagem.

A contagem do número de linhas de código (LC) não apresentou dados conclusivos, visto haver grande variação a depender da linguagem e do problema estudado, conforme Quadro 1.

**Quadro 1 - Métrica Número de Linhas de Código – LC.**

Linguagem	Problema 1	Problema 2
Java	53	64
JavaScript	75	48
Python	92	67

Fonte: Autoria própria.

Observa-se no Quadro 1 que, para o Problema 1, Python apresentou 73,6% de linhas de código a mais que a linguagem Java, enquanto no Problema 2 as duas linguagens apresentam quantidades similares. No Problema 2, JavaScript apresentou o menor número de linhas de código dentre as três linguagens, o que não ocorre no Problema 1, não sendo possível estabelecer uma associação entre linguagem e número de linhas de código para a situação em questão.

Ao se analisar o tempo de execução (TE), observou-se comportamentos diferentes para cada linguagem:

**Quadro 2 - Métrica Tempo de Execução – TE (em segundos).**

Linguagem	$10^2$ casos		$10^4$ casos		$10^6$ casos		$10^7$ casos	
	Problema 1	Problema 2	Problema 1	Problema 2	Problema 1	Problema 2	Problema 1	Problema 2
Java	0,004	0,01	0,061	0,112	1,203	1,889	8,0872	15,004
JavaScript	0,00088	0,00264	0,1518	0,04354	0,67894	3,10864	6,3399	11,682
Python	0,000262	0,00654	0,051605	0,81598	5,865155	59,98896	47,47342	-

Fonte: Autoria própria.

Depreende-se do Quadro 2 que há correlação entre a linguagem utilizada, o número de casos processados e o tempo de execução de cada algoritmo. A linguagem JavaScript apresentou menor tempo de execução na maioria dos cenários, enquanto Python apresentou elevado tempo de execução a partir de  $10^6$  casos, comparada às demais linguagens. O tempo de execução do algoritmo desenvolvido em Python para o Problema 2 excedeu 2 minutos, motivo pelo qual os testes foram interrompidos para este cenário. Realizando ajustes de curvas com a aplicação do método dos mínimos quadrados aos dados do Quadro 2 para cada linguagem e problema, utilizando o modelo de regressão exponencial de Gompertz, conforme descrito por Hoffman (2016), para encontrar uma aproximação para os dados de tempos de execução, obtemos os modelos dispostos no Quadro 3.

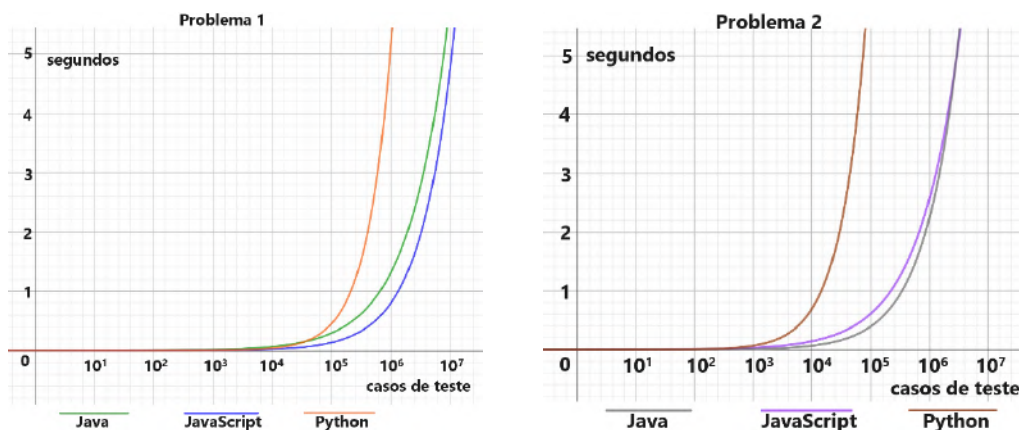
**Quadro 3 - Funções de Regressão para cada Linguagem x Problema.**

Linguagem	Problema 1	Problema 2
Java	$y = 0,00017 e^{1,492 \log x}$	$y = 0,00046 e^{1,437 \log x}$
JavaScript	$y = 0,0000191 e^{1,774 \log x}$	$y = 0,0000662 e^{1,737 \log x}$
Python	$y = 0,00000244 e^{2,43 \log x}$	$y = 0,0000763 e^{2,277 \log x}$

Fonte: Autoria própria.

Nas funções do Quadro 3,  $x$  representa o número de casos de teste, enquanto  $y$  representa o tempo de execução do algoritmo (em segundos). Observa-se que, para a situação em que são utilizados  $10^7$  casos de teste com o problema 2 resolvido na linguagem Python, a função de regressão nos retorna  $y = 637,89$  segundos, o que indica que, para esta situação, o programa levaria pouco mais de 10 minutos para finalizar o processamento do teste, tornando inviável a continuação do teste para este cenário. Os gráficos dessas funções podem ser vistos na Figura 2.

**Figura 2 - Gráficos das funções de regressão.**



Fonte: Autoria própria.

A Figura 2 mostra que, para o problema 1, aproximadamente a partir de  $10^5$  casos de teste o tempo de execução em Python apresenta um crescimento muito superior às demais linguagens. Para o problema 2, observa-se que o tempo de execução do problema resolvido em Python já apresenta taxa de crescimento superior para pouco mais de  $10^3$  casos.

A contagem de funções utilizadas (CF) em cada resolução consta no Quadro 4:



**Quadro 4 - Métrica Chamadas a Funções – CF.**

Linguagem	Problema 1	Problema 2
Java	10	10
JavaScript	11	5
Python	19	15

Fonte: Autoria própria.

Observa-se no Quadro 4 que, em todos os cenários, a linguagem Python apresentou o maior número de funções utilizadas nas resoluções. Isto pode ser explicado pelo interesse dos discentes em explorar mais recursos da linguagem, visto ter sido a linguagem escolhida para aprofundamento de estudos pela maioria dos participantes.

Analisando os aspectos qualitativos da pesquisa, observou-se o uso de diferentes estratégias de resolução para um mesmo problema, influenciados tanto pela linguagem escolhida, quanto por repertórios e construtos pessoais anteriores ao projeto. Quanto à resolução de um mesmo problema em diferentes linguagens por um mesmo aluno, foram observadas duas estratégias:

- Tradução *ipsis litteris* de uma linguagem para outra, utilizado principalmente por discentes que optaram por Java e JavaScript;
- Estratégias diferentes de resolução do problema, a fim de aproveitar melhor os recursos específicos de cada linguagem, utilizado sobretudo por discentes que resolveram os problemas em Python.

Ao investigar a melhoria de habilidades de programação, quanto ao nível de conhecimento dos participantes em recursos de lógica e linguagem de programação, houve relatos de melhoria após a participação no projeto, conforme dados do Quadro 5:

**Quadro 5 - Conhecimento de Estruturas de Programação Antes e Depois do Projeto (percentual %).**

Recurso	Excelente		Muito bom		Bom		Regular		Ruim		Não conhece	
	Antes	Depois	Antes	Depois	Antes	Depois	Antes	Depois	Antes	Depois	Antes	Depois
Entrada de dados	50,00	50,00	16,67	50,00	33,33	-	-	-	-	-	-	-
Saída de dados	33,33	66,67	33,33	33,33	33,33	-	-	-	-	-	-	-
Estruturas Condicionais	16,67	33,33	33,33	50,00	50,00	16,67	-	-	-	-	-	-
Estruturas de Repetição	-	33,33	33,33	16,67	33,33	-	16,67	50,00	16,67	-	-	-
Vetores	-	33,33	-	16,67	50,00	16,67	16,67	33,33	33,33	-	-	-
Matrizes	-	-	-	16,67	-	-	16,67	50,00	-	-	83,33	33,33

Fonte: Autoria própria.

Os dados do Quadro 5 revelam aumento no nível de conhecimento em todos os recursos de lógica de programação investigados. Observa-se que a grande maioria dos participantes já possuía algum domínio de estruturas de condição e repetição, o que não se repete para a manipulação de vetores e matrizes. Além disso, os dados indicam melhoria na percepção das próprias habilidades de programação, o que pode implicar em aumento de confiança e autoestima para resolução de problemas de forma computacional.

Em relação a habilidades específicas de resolução de problemas matemáticos, 85% dos participantes relatam ter

sentido maior facilidade para assimilar fórmulas da disciplina com o uso da metodologia do projeto, quando comparado à resolução de exercícios de modo tradicional em sala de aula. O mesmo percentual de participantes indica que a metodologia do projeto colaborou, muito ou em certa medida, na compreensão de problemas e percepção de estruturas lógico-matemáticas internas a estes.

#### 4. Conclusão

A escolha de uma linguagem de programação para fins acadêmicos ou educacionais perpassa por diversos critérios, incluindo preferências pessoais. Dentre estes, destaca-se a familiaridade dos envolvidos com os recursos da linguagem escolhida e a disponibilidade de material de consulta.

A análise apresentada no presente trabalho não pretende esgotar o assunto, nem orientar a opção por uma ou outra linguagem específica, mas tão somente ilustrar que as interfaces entre matemática e programação podem ser exploradas, para fins educacionais, em contextos quantitativos e qualitativos que extrapolem a mera aplicação de comandos isolados. Além disso, buscamos exemplificar uma possibilidade para o uso de uma linguagem mais abstrata e generalista na aplicação de problemas matemáticos para alunos de ensino médio.

Muitos dos problemas trabalhados no decorrer do projeto foram encarados pelos alunos, inicialmente, como de difícil solução, revelando dúvidas entre os participantes sobre suas próprias capacidades para resolvê-los. Isto não impediu a construção de soluções criativas, a colaboração e discussão de ideias entre os membros, criação de relações com outros problemas resolvidos anteriormente e situações vivenciadas em sala de aula ou a busca, de forma individual ou coletiva, por outros conhecimentos adicionais.

Considera-se que os objetivos do estudo foram alcançados, sendo observados avanços no tempo de resolução dos problemas, uso de raciocínios mais elaborados e aumento no repertório de estratégias de resolução dos participantes.

Os resultados encontrados sugerem que a metodologia aplicada no decorrer da realização deste trabalho colaborou para o fortalecimento da apreensão de estruturas matemáticas. Deste modo, pretende-se expandir esta abordagem para a resolução de problemas da física, visto que a disciplina também lida com forte aparato matemático. Faz-se necessário, ainda, verificar a aplicação da metodologia para alunos com pouco ou nenhum conhecimento de linguagens e lógica de programação, a fim de se investigar a viabilidade da abordagem deste trabalho para o aprendizado introdutório de algoritmos e linguagens de programação.

#### Referências

- Bassanezi, R. C. (2015). *Modelagem matemática: teoria e prática*. Contexto.
- Brasil. (2022). Parecer CNE/CEB nº 2/2022, de 17 de fevereiro de 2022. Normas sobre computação na educação básica – complemento à Base Nacional Comum Curricular (BNCC). <http://portal.mec.gov.br/conselho-nacional-de-educacao/base-nacional-comum-curricular-bncc>
- Carvalho, F. J. R., & Klüber, T. E. (2021). Modelagem matemática e programação de computadores: uma possibilidade para a construção de conhecimento na educação básica. *Educação Matemática Pesquisa*, 23(1), 297-323. <http://doi.org/10.23925/1983-3156.2021v23i1p297-323>
- Dillenbourg, P. (1999). What do you mean by collaborative learning? In: Dillenbourg, P. (Ed.). *Collaborative learning: cognitive and computational approaches*. Elsevier.
- Ferreira, R. C., & Duarte, S. (2019). Ensino de programação: trajetória histórico-social e avanços na cultura digital no Brasil. *Revista Brasileira de Ensino de Ciência e Tecnologia*, 12(1), 386-408. <http://periodicos.utpr.edu.br/rbect/article/view/7532>
- Franco, M. A. S. (2005). Pedagogia da pesquisa-ação. *Educação e Pesquisa*, 31(3), 483-502. <http://doi.org/10.1590/S1517-97022005000300011>
- Freire, P. (2019). *Pedagogia da autonomia: saberes necessários à prática educativa*. (74a ed.). Paz & Terra.
- Gowin, D. B. (1981). *Educating*. Cornell University Press.
- Guedes, K., Meira, A. H., Beleti Junior, C. R., & Shirabayashi, J. V. (2020). Integração no ensino de disciplinas matemáticas e de programação: uma revisão



sistemática. *Revista de Ensino de Engenharia*, 39, 379-389.

Hoffman, R. (2016). *Análise estatística de relações lineares e não lineares*. ESALQ-USP.

Holanda, W. D., Coutinho, J. C. S., & Fontes, L. M. O. (2018). Uma intervenção metodológica para auxiliar a aprendizagem de programação introdutória: um estudo experimental. In: *Anais do VII Congresso Brasileiro de Informática na Educação*. <http://repositorio.ufc.br/handle/riufc/44088>

Kelly, G. A. (1963). *Theory of personality: the psychology of personal constructs*. Norton.

Kilhamn, C., Brating, K., & Rolandsson, L. (2021). Teachers' arguments for including programming in mathematics education. In: *Bringing Nordic mathematics education into the future: Proceedings of Norma 20*. <http://www.diva-portal.org/smash/get/diva2:1541426/FULLTEXT01.pdf>

Pereira Jr. H. N. (2021). *Matemática e programação: uma nova abordagem de ensino*. Dissertação (Mestrado) Programa de Pós-Graduação em Matemática. PUC/Rio de Janeiro, RJ.

Pontes, E. A. S., Silva, J. C. S., Santos, J. B., Ribeiro, R. L. A. O., Silva, J. Cerqueira, P. C., & Silva, B. H. M. S. (2022). Desafios matemáticos em sala de aula: uma prática metodológica para ensinar e aprender matemática através da resolução de problemas. *Research, Society and Development*, 11(8), e50711830901. <http://doi.org/10.33448/rsd-v11i8.30901>

Proetti, S. (2017). As pesquisas qualitativa e quantitativa como métodos de investigação científica: um estudo comparativo e objetivo. *Revista Lumen*, 2(4). <https://doi.org/10.32459/revistalumen.v2i4.60>

Rogers, C. (1973). *Liberdade para aprender: uma visão do que a educação pode se tornar*. (2a ed.). Interlivros.

Thiollent, M. (2018). *Metodologia da pesquisa-ação*. (18a ed.). Cortez.

Soistak, A. V. (2016). Uma experiência com a modelagem matemática no ensino médio profissionalizante. In: Brandt, C. F., Burak, D., Klüber, T. E. (Orgs.). *Modelagem matemática: perspectivas, experiências, reflexões e teorizações* (2a ed.). Editora UEPG.

Vygotsky, L. S. (2007). *A formação social da mente*. (7a ed.). Martins Fontes.

Zapalowski, V. (2011). *Análise quantitativa e comparativa de linguagens de programação*. Trabalho de Conclusão de Curso (Graduação) Bacharelado em Ciência da Computação. UFRGS/Porto Alegre, RS. <http://lume.ufrgs.br/handle/10183/31036>