

## UM ESTUDO DE CASO DA FERRAMENTA BEECROWD COMO TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO NA APRENDIZAGEM DE ALGORITMOS<sup>1</sup>

Bruno Ferreira<sup>2</sup>  
Camilly Leal Silva  
Felipe de Castro Leal Ribeiro

### RESUMO

Esse artigo delinea um estudo de caso abordando a aplicação de Tecnologia da Informação e Comunicação (TIC), no processo de ensino e aprendizagem de algoritmos. O estudo foi realizado em um curso de nível técnico de informática no Instituto Federal de Minas Gerais. Este trabalho concentra-se na utilização de softwares que propõem desafios algorítmicos e são capazes de analisar as soluções sugeridas pelos usuários. Ao empregar essas ferramentas, os estudantes têm a capacidade de verificar suas ideias e conceitos praticamente em tempo real. O texto lista algumas alternativas de software, seleciona a ferramenta denominada Beecrowd e faz uma análise da percepção de um grupo de alunos voluntários em relação à escolha. Os resultados mostram que a ferramenta pode ser uma boa opção para ajudar no ensino de conceitos básicos de programação.

**Palavras-chave:** TIC. Aprendizagem de algoritmo. Educação.

### A CASE STUDY OF THE BEECROWD TOOL AS INFORMATION AND COMMUNICATION TECHNOLOGY IN ALGORITHM LEARNING

### ABSTRACT

This article outlines a case study addressing the application of Information and Communication Technology (ICT) in the teaching and learning process of algorithms. The study was conducted in a computer science technical course at the Federal Institute of Minas Gerais. This work focuses on the use of software that proposes algorithmic challenges and is capable of analyzing solutions suggested by users. By employing these tools, users have the ability to verify their ideas and concepts virtually in real-time. The text explores various software alternatives, selects the tool named Beecrowd, and concludes with an analysis of the perception of a group of volunteer students regarding this choice. The results indicate that the tool can be a suitable choice to assist in teaching basic programming concepts.

**Keywords:** ICT. Algorithm's learning. Education.

---

<sup>1</sup>**Como citar este trabalho:** FERREIRA, Bruno; SILVA, Camilly Leal; RIBEIRO, Felipe de Castro Leal. Um estudo de caso da ferramenta Beecrowd como tecnologia da informação e comunicação na aprendizagem de algoritmos. ForScience, **Formiga**, v. 13, n. 2, e1313, jul./dez. 2025. DOI: [10.29069/forscience.2025v13n2.e1313](https://doi.org/10.29069/forscience.2025v13n2.e1313).

<sup>2</sup>**Autor correspondente:** Bruno Ferreira, e-mail: [bruno.ferreira@ifmg.edu.br](mailto:bruno.ferreira@ifmg.edu.br).

## 1 INTRODUÇÃO

A sociedade está imersa em uma era de transformações aceleradas, marcada por uma crescente digitalização. Este fenômeno permeia todos os aspectos da vida, desde a comunicação até a educação, o comércio e as relações interpessoais. A ascensão da tecnologia digital não apenas alterou a forma como interagimos com o mundo, mas também redefiniu a própria natureza das conexões humanas. A ubiquidade dos dispositivos digitais, a proliferação de plataformas *online* e a crescente integração de inteligência artificial são marcos evidentes dessa revolução digital. Neste cenário dinâmico, a sociedade se adapta constantemente a novas formas de interação e produção de conhecimento, desafiando-nos a repensar conceitos tradicionais e a abraçar a inovação em todos os setores da vida cotidiana.

Nesse contexto, os algoritmos emergem como protagonistas cruciais, moldando as dinâmicas contemporâneas. Essas sequências de instruções lógicas desempenham um papel central na filtragem de informações, na personalização de experiências *online* e na automação de uma miríade de processos. Desde as recomendações personalizadas em plataformas de *streaming* até os algoritmos de busca que influenciam as decisões de consumo, os algoritmos permeiam cada interação digital.

Contudo, a dificuldade de aprendizagem de algoritmos constitui um desafio complexo no panorama educacional contemporâneo. O entendimento dessas estruturas lógicas, essenciais na sociedade digital, frequentemente se depara com obstáculos variados. Os algoritmos, por sua natureza abstrata e a necessidade de uma compreensão sólida da lógica computacional, podem representar um entrave para muitos estudantes. A complexidade inerente a esses conceitos, aliada à diversidade de estilos de aprendizagem, torna crucial a implementação de abordagens pedagógicas inovadoras e inclusivas. As dificuldades na assimilação de algoritmos não apenas impactam o desempenho acadêmico, mas também têm implicações diretas no desenvolvimento de habilidades cruciais para a sociedade digital.

Timmermann (2015, p. 14) apresenta uma pesquisa realizada pela Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação (Brasscom), que aponta o assustador percentual de 82% na evasão escolar de cursos superiores na área de tecnologia no Brasil. A autora menciona ainda uma outra pesquisa realizada pela Sociedade Brasileira de Computação (SBC), que confirma o alto número de reprovação e conclui que não vê indícios de melhora nos índices de formação de novos profissionais na Computação nos próximos anos. Outro exemplo, citado por Giraffa e Moura (2012), que também destacam os altos índices de evasão e repetência. Os autores apontam preocupação com o abandono precoce dos cursos de

Computação, principalmente no primeiro ano. As principais dificuldades enfrentadas pelos alunos estão nas disciplinas de Cálculo e Programação. Essa situação contribui para a escassez de profissionais na área e traz impactos negativos para o mercado e a sociedade.

Atualmente, existem diversos cursos *online* e gratuitos (*YouTube*, *+IFMG*, *Udemy*) que ensinam algoritmos. Estes buscam facilitar a aprendizagem e deixar o estudo menos burocrático. Porém nesse caso, os estudantes assumem um papel passivo na aprendizagem, eles não praticam e não sabem se estão evoluindo, isso pode gerar uma insegurança ou desmotivar o estudante, pois ele não vê o progresso. Foi com esse nicho de mercado que surgiram as ferramentas de correção de algoritmos. Elas têm cadastradas questões/problemas e casos de testes, assim elas confrontam as soluções apresentadas para sinalizar aos usuários a porcentagem de acertos, além de outros recursos possíveis.

No entanto, este estudo introduz potenciais ferramentas de correção de algoritmos, implementando especificamente a aplicação *Web* conhecida como *Beecrowd* a um grupo de estudantes do primeiro ano do curso Técnico Integrado em Informática do Instituto Federal de Minas Gerais (IFMG) do campus Formiga. O objetivo principal consistiu em colher as percepções dos alunos durante o uso dessa ferramenta.

## 2 REFERENCIAL TEÓRICO

O termo algoritmo não é novo e foi sendo formulado desde o surgimento do sistema de notação numérico hindu-arábico e da escrita dos cálculos algébricos pelo árabe Al-Kharazmi, contudo, foi a partir de 1250 d.c. que o matemático e astrônomo John Halifax apresentou conceitos mais concretos. Hoje esses conceitos e sua lógica são as bases para a Ciência da Computação (FILHO, 2007, p. 33). Cormen et al. (2002, p. 22) descrevem que um algoritmo é uma sequência finita de instruções bem definidas e não ambíguas.

De acordo com Souza (2008, p.6), é importante para a escrita de algoritmos, um bom raciocínio lógico-dedutivo, ou seja, a lógica que precisa ser desenvolvida é aquela que “diga como as pessoas devem raciocinar” para escrever sequência de comandos que resolvam uma determinada tarefa com sucesso. Na mesma linha de pensamento, Setti (2009, p.11), explica que os maiores obstáculos epistemológicos à aprendizagem de algoritmos são justamente, a forma de pensar ou a forma de raciocinar sobre a solução de um determinado problema e, aponta que o domínio sobre os conceitos da Matemática Discreta e da Lógica Matemática (ensino das proposições lógicas) pode ser importante em conjunto com metodologias de ensino adequadas. Todavia, esse grau de conhecimento matemático tende a ser uma das principais causas dos altos

índices de reprovação dos alunos do ensino técnico e superior (SANTOS; COSTA, 2006; BARCELOS, TAROUÇO; BERCHT, 2009).

O aprendizado baseado em problemas, o uso de TICs na educação e mesmo a junção dos dois conceitos em sala de aula, não são uma novidade. Existem diversos experimentos e relatos de uso dessas técnicas em várias áreas do conhecimento e na Computação não é diferente. Por exemplo, Santos e Costa (2005), apresentam uma ferramenta com uma abordagem um pouco diferente, propondo um software educacional de simulação gráfica, o qual representa passo a passo as instruções do algoritmo com o objetivo de complementar o material teórico acerca da manipulação das estruturas de dados tradicionais.

Outro exemplo do uso de TIC no ensino de programação é a ferramenta Alice. De acordo com Aktunc (apud VALASKI; PARAÍSO, 2015), essa ferramenta apresenta potencial motivador ao permitir a criação de animações e mundos virtuais por meio de uma interface simplificada, favorecendo o desenvolvimento da autoconfiança dos estudantes na exploração dos conceitos de programação. No entanto, o autor adverte que, por si só, a ferramenta não é suficiente para proporcionar uma aprendizagem completa dos fundamentos da programação, especialmente no que se refere à orientação a objetos. Complementando essa perspectiva, Valaski e Paraíso (2015), por meio da realização de oficinas semestrais com 48 estudantes, observaram que a Alice se mostra particularmente eficaz para alunos com dificuldades na abstração de conceitos introdutórios de programação, embora tenham sido identificadas limitações em sua aplicação.

## 2.1 Ferramentas de correção de algoritmos

Existem diversas opções de ferramentas para a correção de algoritmos. Hemming e Franzen (2018) apresentam várias delas, com destaque para a plataforma da *Code School*<sup>1</sup>. Ela oferece a possibilidade para o aluno testar o código diretamente no navegador Web, outra característica é a utilização da experiência de gamificação<sup>2</sup> no aprendizado, encorajando o aluno a continuar. As outras ferramentas são a *Khan Academy*<sup>3</sup> e a *Code Chef*<sup>4</sup>. A primeira dessas duas disponibiliza exercícios, vídeos e um sistema de aprendizado personalizado que

<sup>1</sup> <https://hsmuniversity.com.br/code-school/> Acessado em 01/05/2023

<sup>2</sup> <https://mundoekoa.com/2020/04/14/gamificacao-na-educacao-tantos-beneficios-nao-podem-fazer-mal/> Acessado em 25/04/2023.

<sup>3</sup> <https://pt.khanacademy.org/> Acessado em 25/04/2023.

<sup>4</sup> <https://www.codechef.com/> Acessado em 01/05/2020.

habilita os estudantes a aprender no seu próprio ritmo, com a abordagem de assuntos como: busca binária, ordenação, algoritmos recursivos e grafos. O *Code Chef* é uma plataforma de competição de desafios em programação, atualmente com suporte a mais de 50 linguagens, contando com uma grande comunidade de estudantes e profissionais.

Na mesma linha existe o *Neps Academy*, o qual disponibiliza cursos que vão desde a introdução à programação até conceitos de Inteligência Artificial. O portal eletrônico apresenta também uma ferramenta de correção de exercícios em tempo real, com o objetivo de treinar os alunos para competições na área da Computação (NEPS ACADEMY, 2023).

O *Run Codes* é outra opção, ela é uma plataforma gratuita, que oferece uma gama de ferramentas para desenvolvedores de software e programadores. Disponibiliza uma ferramenta de submissão e correção de códigos em diversas linguagens de programação como C/C++, Java, R e outras.

Ainda sobre essa última ferramenta, o usuário consegue se cadastrar como professor ou aluno, visto que professores conseguem postar exercícios, criar turmas, acompanhar o desempenho e evolução dos alunos, por meio de estatísticas e verificar a similaridade. Segundo o site, os exercícios são corrigidos em segundos e o aluno pode verificar a nota e os detalhes da correção. Já os alunos conseguem ver as disciplinas, próximas entregas, entrar em turmas por meio de código e responder exercícios. O site é todo em português, educativo e de fácil acesso. Porém, mesmo com tantas características favoráveis para uso, a plataforma foi recentemente desligada por motivos de falta de dedicação dos autores e custos.

Outra ferramenta que parece promissora é a *Beecrowd*, ela é gratuita, tem tradução para o português, é intuitiva e atualizada, ela tem a opção da escolha de linguagem na qual o estudante deseja trabalhar. Existe suporte para dezenas de linguagens. O site tem várias funcionalidades interessantes, como por exemplo, encontrar oportunidades de trabalho, painel com gráfico de progresso, quantos problemas foram resolvidos, pontuação, etc. Também são fornecido cursos, e os problemas são agrupados por tipos. No modo professor, há a possibilidade de fazer listas e problemas personalizados.



**Figura 1-** Tela de exercício do *Beecrowd*  
**Fonte:** imagem retirada do site da ferramenta Beecrowd

Depois de analisar artigos como: Hemming e Franzen (2018), Valente (2011) e Vieira Junior (2018), além de sites, blogs e redes sociais na área de programação de computadores, levantaram-se possíveis ferramentas para serem adotadas neste trabalho, as quais são listadas no Quadro I e caracterizadas mais abaixo no texto.

Nome	Endereço eletrônico
Bee Crowd	<a href="https://www.beecrowd.com.br">https://www.beecrowd.com.br</a>
Run Codes	<a href="https://run.codes/">https://run.codes/</a>
Neps Academy	<a href="https://neps.academy/br/login">https://neps.academy/br/login</a>
Code Academy	<a href="https://www.codecademy.com/">https://www.codecademy.com/</a>
Code War	<a href="https://www.codewars.com/">https://www.codewars.com/</a>
Grass Hopper	<a href="https://grasshopper.app/pt_br/">https://grasshopper.app/pt_br/</a>
Mimo	<a href="https://mimo.org/">https://mimo.org/</a>
Code Chef	<a href="https://www.codechef.com/">https://www.codechef.com/</a>

**Quadro 1** - Ferramentas analisadas (URLs acessadas em 09/12/2023)  
**Fonte:** próprio autor.

Para esse estudo julgou-se importante que a ferramenta a ser adotada possa responder as perguntas abaixo da melhor forma possível:

- A ferramenta é gratuita?
- A ferramenta é em português ou tem suporte para essa língua?
- A ferramenta permite a criação de listas customizadas de exercícios?
- A ferramenta permite programar em Java?
- A ferramenta permite programar em outras linguagens de programação?
- A ferramenta é mantida de forma atualizada?

- A ferramenta tem módulos para professores e alunos?
- A ferramenta possui algum diferencial ou observação relevante?

Depois de analisar as oito ferramentas listadas no Quadro 1, pôde-se montar uma tabela com a resposta das perguntas-chave para a seleção da ferramenta (Quadro 2). Uma ferramenta que se destacou foi a *Beecrowd*, ela é gratuita, tem a separação professor/aluno, permite que o professor crie seus próprios exercícios e está traduzida para português. Além disso, tem suporte ao Java e a outras linguagens como C e Python. Isso deu o indicativo que essa poderia ser a ferramenta escolhida para esse trabalho e a qual foi adotada para o experimento.

Perguntas	Ferramentas							
	Beecrowd	Run Codes	Neps Academy	Codecademy	Code War	Grasshopper	Mimo	Code Chef
É gratuita?	Sim	Sim	Parcialmente	Não	Sim	Sim	Não	Parcialmente
Permite criar listas customizadas?	Sim	Sim	Sim	Não	Não	Não	Não	Não
Suporte à linguagem Java	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim
Suporte a outras linguagens?	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Está em português?	Sim	Sim	Sim	Não	Não	Sim	Não	Não
É mantida atualizada?	Sim	Não	Sim	Sim	Sim	Não	Sim	Sim
Tem módulos para professor e aluno?	Sim	Sim	Não	Não	Não	Não	Não	Não
Diferencial ou observação relevante?	Sim. Integração com Moodle	Não	Não	Sim, disponibiliza projetos de exemplo.	Não.	Não	Não	Sim, modo competitivo

**Quadro 2** - Análise das ferramentas selecionadas  
**Fonte:** próprio autor.

### 3 METODOLOGIA

Esse artigo é fruto do trabalho desenvolvido e financiado obedecendo o edital interno de extensão nº 44/2022 do IFMG – Campus Formiga. O projeto teve como orientador um professor efetivo do campus e dois alunos do segundo ano do curso Técnico em Informática (autores do artigo). Os alunos executaram os trabalhos em sete etapas mensais conforme listadas abaixo:

1. Pesquisou-se por TICS disponíveis e selecionou-se a mais promissora;

2. Os alunos aprenderam a usar a ferramenta escolhida;
3. Selecionou-se problemas relacionados aos conceitos básicos no aprendizado de algoritmos;
4. Inseriu-se os problemas/questões na ferramenta, os casos de testes e os *feedbacks* ao usuário/estudante foram configurados;
5. Foi criado um vídeo explicativo para estudantes e professores sobre a ferramenta;
6. Aplicou-se problemas cadastrados na ferramenta para alguns alunos voluntários do curso Técnico Integrado em Informática do IFMG;
7. Avaliou-se a percepção dos alunos que usaram a ferramenta através de um questionário.

Para esse projeto, os alunos bolsistas trabalharam em dupla, isso justamente para que um aluno não teste o próprio problema/questão que inseriu na plataforma. Julgou-se importante que outra pessoa veja se o enunciado, as dicas e os *feedbacks* cadastrados são suficientes e se contribuem para a aprendizagem.

#### **4 RESULTADOS E DISCUSSÃO**

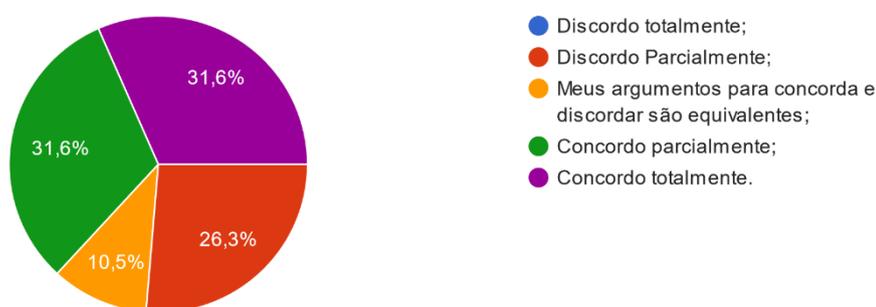
Os alunos do primeiro ano do curso Técnico Integrado em Informática do IFMG – Campus Formiga foram convidados a testar a ferramenta *Beecrowd* em um laboratório de informática da escola. O experimento foi realizado em duas horas e contou com 18 alunos atraídos pelo tema. O experimento teve os seguintes passos:

- 1) Contextualização do tema, apresentação da ferramenta e cadastro dos alunos na mesma;
- 2) Foi criada uma lista com apenas dois exercícios e os responsáveis pelo experimento mostraram aos voluntários como era o uso da ferramenta utilizando essa lista;
- 3) Os voluntários tiveram cerca de 90 minutos para usar a ferramenta, a fim de resolver uma outra lista com 4 questões abordando assuntos básicos e já lecionados em sala de aula como: instruções sequenciais, condicionais e de repetição.
- 4) Por fim, nos últimos 10 minutos, os voluntários responderam a um questionário com perguntas de múltipla escolha e duas questões discursivas.

O objetivo desse último passo era fazer uma consulta qualitativa através de um questionário eletrônico para compreender a percepção dos participantes perante o uso da ferramenta para a aprendizagem de algoritmos. As questões de múltipla escolha foram baseadas na escala proposta por Likert (1932). Nesta escala são dispostas afirmações e cada afirmação

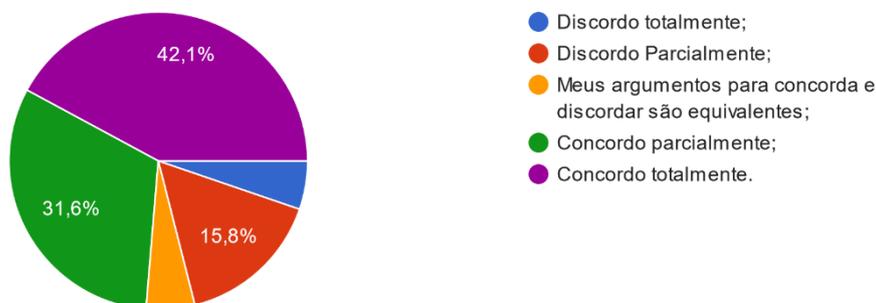
deve ser avaliada em uma escala de concordância ou discordância (VIEIRA E DALMORO, 2008; JÚNIOR E COSTA, 2014).

Referente ao questionário, na Figura 2, o gráfico apresenta que 36,8% dos estudantes não acharam o processo de inscrição no site intuitivo, em conversas com os alunos, eles relataram que ficaram confusos devido à quantidade de informações solicitadas, como por exemplo, tamanho da camisa (campo usado para eventos patrocinados). Além disso, a ferramenta oferece oportunidades de emprego, então exige dados como faixa salarial pretendida. Contudo, 63,2% ainda avaliaram como boa, a etapa inicial no uso do *Beecrowd*. É importante ressaltar que essa etapa é feita apenas uma vez, o que não impacta diretamente no uso da ferramenta.



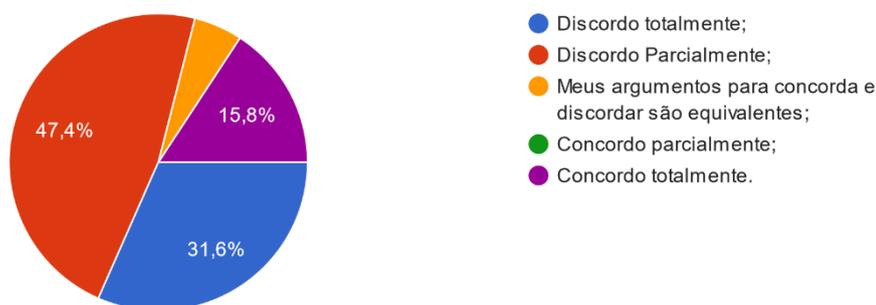
**Figura 2** – “Eu achei o processo de inscrição e uso da ferramenta fáceis.”  
**Fonte:** Próprio autor.

A Figura 3 apresenta a percepção dos respondentes quanto à melhora no desempenho na disciplina se usasse o *Beecrowd*. Neste caso, 63,7% dos entrevistados concordam totalmente ou parcialmente que seria interessante o uso para fins motivacionais, os voluntários se mostraram desafiados a ver seus algoritmos sendo aprovados pelos testes da ferramenta.



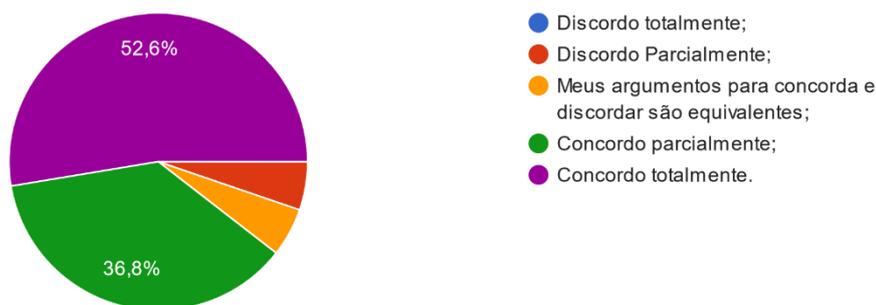
**Figura 3** – “O uso da ferramenta pode ajudar a aumentar o comprometimento das atividades da disciplina de Programação em sala”  
**Fonte:** Próprio autor.

As Figuras 4 e 5 mostram um certo temor inicial dos voluntários, isso em relação ao uso da ferramenta para atividades avaliativas, ou seja, as provas tradicionais. Veja que eles apoiam o uso da ferramenta em atividades referentes à lista de exercícios. Os autores deste artigo chegaram à conclusão que os voluntários têm medo da correção de algoritmos de forma binária, ou seja, certo ou errado. Eles demonstram preferência por uma avaliação que considere não apenas o resultado final do programa, mas também a estrutura e a lógica do algoritmo desenvolvido por eles. Dessa forma, tendo uma pontuação proporcional à parte implementada corretamente. Esse pensamento fica evidente nas respostas discursivas que são apresentadas mais abaixo.



**Figura 4** – “A ferramenta *Beecrowd* pode ser usada como uma atividade avaliativa em sala de aula (Prova)”  
**Fonte:** Próprio autor.

(Listas de exercício)”.



**Figura 5** – “A ferramenta *Beecrowd* pode ser usada como uma atividade prática em sala de aula  
**Fonte:** Próprio autor.

Do ponto de vista do professor/tutor, a correção binária também não é interessante, mas a ferramenta minimiza esse problema. Veja na Figura 6, o exemplo de *feedback* que o *Beecrowd* disponibilizou depois do experimento. Pode-se observar que em verde tem-se as questões corretas para cada aluno. As marcações em vermelho indicam que o algoritmo criado pelo estudante não chegou aos mesmos resultados dos casos de teste cadastrados.

A ferramenta pode ajudar no dia a dia dos professores/tutores, pois corrige as respostas dos alunos e em caso de falha, basta clicar em cima do quadrado vermelho, para ver a resposta do estudante. O professor pode analisar apenas as questões marcadas com incorretas.

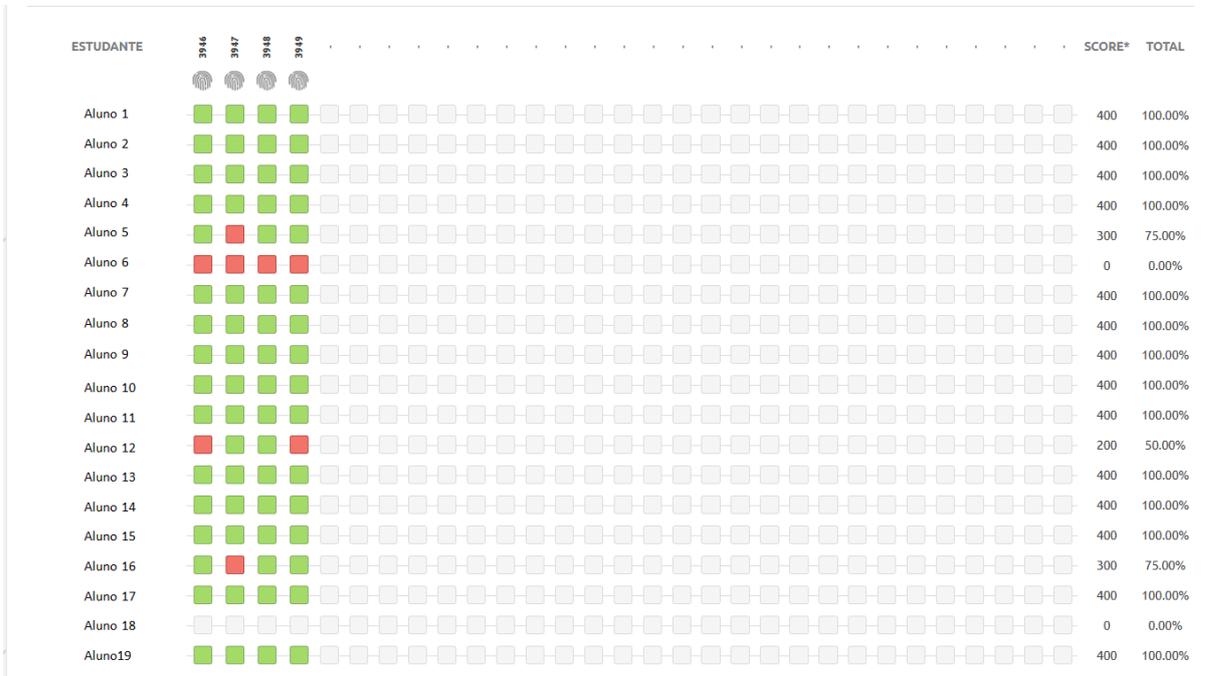


Figura 6 – “Visualização dos resultados da lista de exercícios do experimento”  
 Fonte: Próprio autor.

Outras informações importantes são as estatísticas das submissões (Figura 7). Para a primeira questão houve 16 respostas corretas após 64 tentativas, ou seja, uma média de 3,6 tentativas até que a resposta fosse considerada correta.

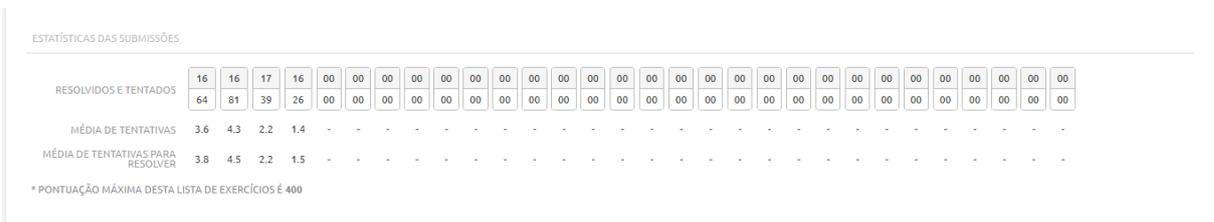


Figura 7 – “Visualização da estatística da lista de exercícios do experimento”.  
 Fonte: Próprio autor.

Contudo, voltando à preocupação dos alunos, veja o exemplo do “Aluno 6”. Ele aparentemente errou as quatro questões, no entanto ao analisar as respostas clicando no quadrado vermelho, o professor pode fazer uma reavaliação e inserir uma nova nota. A Figura 8 mostra a solução apresentada pelo “Aluno 6” para a questão 1 da lista. O código para o algoritmo proposto pode ser considerado correto, a pessoa apenas esqueceu de obedecer a uma

regra da própria ferramenta, a qual foi explicada no início do experimento: a classe deve ter o nome “Main” e o nome dado foi “ProjetoPontoExtra”, ainda assim a solução algorítmica está correta.



SUBMISSÃO 811781 NOVO COMENTÁRIO

LINGUAGEM JAVA 14

RESPOSTA Compilation error

RUNTIME 0.000s

**CÓDIGO FONTE**

```
1 import java.util.Scanner;
2
3 public class ProjetoPontoExtra {
4
5     public static void main(String[] args) {
6
7         Scanner entrada = new Scanner (System.in);
8
9         float salario = entrada.nextFloat();
10        float porcentagem = entrada.nextFloat();
11        float soma = porcentagem*(salario/100);
12        System.out.println("O novo salário é de " + soma + " reais!");
13    }
```

**Figura 7** – “Exemplo de solução inicialmente considerada como incorreta pelo Beecrowd”  
**Fonte:** Próprio autor.

Referente à primeira pergunta discursiva: “Por favor, nos conte o que você gostou ao usar uma ferramenta de correção automática de exercícios?”. Essa pergunta foi respondida pelos 19 alunos e apenas uma resposta não demonstrou um sentimento exclusivamente positivo:

*“É boa para corrigir exercícios sem comprometimento, porém, pelo o código ter que ser *\*exatamente\** do jeito que a ferramenta acha que é o certo, prejudicaria os alunos em uma prova, por exemplo.”*

Três exemplos de respostas positivas e que refletem o sentimento dos demais alunos são:

*“Eu gostei, pois é uma ferramenta com uma interface amigável e pode auxiliar na correção de códigos em uma lista de exercícios, por exemplo. A praticidade da ferramenta também auxilia muito com uma correção rápida e dicas para o que pode estar errado no seu código.”*

*“Fiquei surpresa com a ferramenta, foi além das minhas expectativas. Gostei bastante da plataforma para utilizar na correção de exercícios, muito eficaz e precisa.”*

*“A ferramenta te dá a oportunidade de corrigir seu erro o que é bastante agradável e também as respostas de acerto e erro são precisas.”*

A segunda pergunta dizia: “Por favor, nos conte o que você NÃO gostou ao usar uma ferramenta de correção automática de exercícios?”, essa questão foi respondida pelos 19 alunos e as três listadas abaixo refletem a impressão dos demais voluntários:

*“A ferramenta acaba sendo muito exigente em relação à maneira como se deve imprimir o código.”*

*“Um único ponto negativo, é o fato do seu algoritmo estar certo, mas por causa de um simples espaço na resposta, já constar como erro.*

*“A ferramenta classificar o código como incorreto por conta de meros detalhes, que na execução no NetBeans não são erros consideráveis, prejudicando parcialmente um aluno em uma prova que fosse corrigida pela ferramenta.”*

Em relação ao módulo do professor da ferramenta, os autores perceberam que existem diversos fatores positivos ao usar a ferramenta, mas notaram as seguintes dificuldades, as quais precisariam de melhorias da equipe que mantém a ferramenta *Beecrowd*:

- A ferramenta permite usar o comando *Random*, mas não é possível formular casos de teste, pois toda vez que o programa for executado, ele criará novos números;
- A ferramenta não deixa o estudante criar mensagens iniciais. Então não é possível criar exercícios para exibição de menus, ou que mostrem frases iniciais;
- A ferramenta não dá a opção para verificar se o estudante usou alguma estrutura requisitada. Exemplo, crie um algoritmo com "*switch*", ao invés, de um "*if*";
- A ferramenta deixa criar classes internas e estáticas, além da classe fornecida. Mas ela não verifica se o aluno realmente criou uma classe ou um método requisitado;
- Existe um inconveniente de que o professor é obrigado a teclar um "*Enter*" para quebrar a linha ao criar os casos de teste para indicar uma quebra de linha na resposta;
- O sistema apresenta lentidão na maior parte do uso no módulo do professor;

## 5 CONCLUSÃO

O atual cenário do mercado de trabalho tem gerado uma crescente demanda por profissionais versáteis que possam contribuir em diversas fases do desenvolvimento de software. Entretanto, essa demanda não tem sido plenamente atendida, não por falta de instituições educacionais ou programas de treinamento nesse campo, mas, sim, devido às dificuldades e ao

desinteresse manifestados pelos aspirantes a essa área, especialmente quando confrontados com a necessidade de adquirir domínio sobre os conceitos fundamentais de algoritmos.

A constante incidência de reprovações em disciplinas acadêmicas relacionadas a linguagens de programação tem sido objeto de estudo por diversos autores, que destacam os métodos de ensino contemporâneos como um dos fatores determinantes. Diante desse contexto, este estudo propôs a implementação de uma categoria específica de Tecnologias da Informação e Comunicação (TICs) para superar os desafios no processo de ensino-aprendizagem, enfrentados por estudantes em cursos técnicos e de ensino superior. O objetivo central foi empregar uma ferramenta de correção de algoritmos, visando fomentar a criatividade, confiança e proatividade dos alunos no entendimento dos conceitos fundamentais de algoritmos.

Ao final das sete etapas propostas nesse trabalho, a aplicação de uma ferramenta de correção de algoritmos demonstrou fortes indícios de que pode ser mais um recurso para ajudar professores e alunos. Os alunos são desafiados e mostram engajamento ao praticar algoritmos até que as respostas fornecidas sejam aprovadas. Já o professor pode ganhar tempo corrigindo apenas as questões que foram reprovadas, além de aproveitar o ambiente favorável pela ferramenta, para tirar o máximo dos alunos durante a aprendizagem da lógica de programação.

Como uma ação de constante melhoria para motivar e melhorar o ensino-aprendizagem de algoritmos, pode-se citar como próximos passos: testes de outras ferramentas de correção de algoritmos, uso de gamificação e ferramentas que ensinam conceitos de algoritmos através de jogos e competições.

Com base nos resultados qualitativos obtidos nesse trabalho, com relação ao uso de TICs no ensino de algoritmos, os autores recomendam fortemente que docentes de diferentes instituições e modalidades de ensino, considerem a aplicação das metodologias modernas e apoiadas por TICs em disciplinas de programação.

## REFERÊNCIAS

BARCELOS, R.; TAROUCO, L.; BERCHT, M. O uso de mobile learning no ensino de algoritmos. **Renote – Revista Novas Tecnologias na Educação**, v. 7, n. 3, p. 327-337, 2009.

CORMEN, T. H. et al. **Algoritmos: teoria e prática**. Tradução da segunda edição [americana] por VANDENBERG, D. de Souza. Rio de Janeiro: Elsevier, 2002. Reimpressão.

FILHO, C. F. **História da computação [recurso eletrônico]**: o caminho do pensamento e da tecnologia. Porto Alegre: EDIPUCRS, 2007. 205 p.

GIRAFFA, L. M. M.; MORA, M. da C. Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. In: CONFERÊNCIA LATINO-AMERICANA SOBRE O ABANDONO NA EDUCAÇÃO SUPERIOR – CLABES, 3., 2015, Porto Alegre. **Anais [...]**. Porto Alegre: CLABES, 2015.

HEMMING, C.; FRANZEN, E. **Ferramenta de apoio ao ensino e aprendizagem de algoritmos e programação**. 2018. Monografia (Conclusão de Curso) – Centro de Ciências Exatas e Tecnológicas, Universidade do Vale do Taquari – UNIVATES, Lajeado, 2018.

JÚNIOR, S. D. S.; COSTA, F. J. Mensuração e escalas de verificação: uma análise comparativa das escalas de Likert e Phrase Completion. **PMKT – Revista Brasileira de Pesquisas de Marketing, Opinião e Mídia**, v. 15, p. 1-16, 2014.

LIKERT, R. **A technique for the measurement of attitudes**. New York: Columbia University Press, 1932. (Archives of Psychology).

NEPS ACADEMY. **Homepage da ferramenta**. Disponível em: <https://neps.academy/login?next=%2F>. Acesso em: 2 nov. 2023.

SANTOS, R. P. dos S.; COSTA, H. A. X. Análise de metodologias e ambientes de ensino para algoritmos, estruturas de dados e programação aos iniciantes em computação e informática.

INFOCOMP Journal of Computer Science, v. 5, n. 1, p. 41-50, 2006.

SANTOS, R. P.; COSTA, H. A. X. Desenvolvimento de um software educacional para o ensino de algoritmos, estruturas de dados e programação. In: CICESAL, 18.; SEMINÁRIO PIBIC/CNPq, 12.; SEMINÁRIO PBICT/FAPEMIG, 8.; CONGRESSO DE EXTENSÃO, 1., 2005, Lavras. **Anais [...]**. Lavras: [s.n.], 2005. v. 1, p. 113. [CD de Resumos].

SETTI, M. de O. G. **O processo de discretização do raciocínio matemático na tradução para o raciocínio computacional: um estudo de caso no ensino/aprendizagem de algoritmos**. 2009. Tese (Doutorado) – Universidade Federal do Paraná, Curitiba, PR, 2009.

SOUZA, J. N. de. **Lógica para ciência da computação: uma introdução concisa**. 2. ed. Rio de Janeiro: Elsevier, 2008. 218 p.

TIMMERMANN, G. L. K. **O desafio de aprender e ensinar algoritmos: mediações que professores e alunos estabelecem com o conteúdo no ensino superior**. 2015. Dissertação (Mestrado) – Universidade Regional do Noroeste do Estado do Rio Grande do Sul, 2015.

VALASKI, J.; PARAÍSO, E. C. O ensino de programação através da ferramenta Alice: um estudo sobre as práticas docentes no ensino médio. In: CONGRESSO INTERNACIONAL DE EDUCAÇÃO E TECNOLOGIAS E ENCONTRO DE PESQUISADORES EM

EDUCAÇÃO A DISTÂNCIA, v. 1, n. 1, 2015. **Anais [...]**, 2015. Disponível em: <https://doi.org/10.20396/ciet.v1i1.12062>. Acesso em: 28 maio 2025.

VALENTE, J. A. Diferentes usos do computador na educação. **Educação Pública**, 2011. Disponível em: <http://www.educacaopublica.rj.gov.br/biblioteca/tecnologia/0022.html>. Acesso em: 30 out. 2023.

VIEIRA JUNIOR, N. **Tecnologias e comunicação na educação**. Apostila da Pós-graduação em Docência – IFMG – Campus Arcos. ISSN 978-85-920655-3-9, 2018.

## **DADOS DO AUTOR:**

### **Bruno Ferreira**

**E-mail:** [bruno.ferreira@ifmg.edu.br](mailto:bruno.ferreira@ifmg.edu.br)

**Curriculum Lattes:** <http://lattes.cnpq.br/9437251245138635>

Possui graduação em Ciência da Computação pelo Centro Universitário de Formiga (2003), pós graduação em Redes de Computadores pelo Centro Universitário do Sul de Minas (2004), mestrado em Modelagem Matemática e Computacional pelo CEFET-MG (2008) e doutorado em Ciência da Computação pela UFMG (2016). Tem experiência na área de Ciência da Computação, com ênfase em Métodos Formais, Linguagens de Programação e Engenharia de Software. Atualmente é professor efetivo do Instituto Federal de de Minas Gerais - (IFMG - Campus Formiga).

### **Camilly Leal Silva**

**E-mail:** [camillyleals080@gmail.com](mailto:camillyleals080@gmail.com)

Técnica em Informática pelo Instituto Federal de Minas Gerais - IFMG Campus Formiga (2024). Graduanda em Ciência da Computação no IFMG - Campus Formiga com previsão de término em 12/2028.

### **Felipe de Castro Leal Ribeiro**

**E-mail:** [felipecastrolealribeiro@gmail.com](mailto:felipecastrolealribeiro@gmail.com)

**Curriculum Lattes:** <http://lattes.cnpq.br/7232737875767344>

Técnico em Informática pelo Instituto Federal de Minas Gerais - IFMG Campus Formiga (2024). Graduando em Ciência da Computação no IFMG - Campus Formiga com previsão de término em 12/2028.