

Uma Intervenção Metodológica para Auxiliar a Aprendizagem de Programação Introdutória: um estudo experimental

Wallace Duarte de Holanda¹, Jarbele Cássia da Silva Coutinho¹,
Laysa Mabel de Oliveira Fontes¹

¹ Centro Multidisciplinar de Pau dos Ferros (CMPF)
Universidade Federal Rural do Semi-Árido (UFERSA)
BR 226, Km 405 – São Geraldo, Pau dos Ferros, RN | CEP: 59900-000

wallace.holanda@alunos.ufersa.edu.br, jarbele.coutinho@ufersa.edu.br,
mabel.fontes@ufersa.edu.br

Abstract. *The teaching and learning process of introductory programming has been associated with high rates of failure and avoidance in computer courses. Given this context, this work evaluates the use of a methodological proposal, with the purpose of stimulating the logical reasoning and motivating the learning of the main concepts inherent to the learning of programming logic. This approach carried out methodological interventions through the inclusion of a complementary practical activity, with the presentation of scenarios and the availability of tips categorized by the level of difficulty. The results suggest the continuity of the application of these methodological interventions.*

Resumo. *O processo de ensino e aprendizagem de programação introdutória tem sido associado aos altos índices de reprovação e de evasão em cursos de computação. Diante desse contexto, este trabalho avalia a utilização de uma proposta metodológica, com a finalidade de estimular o raciocínio lógico e motivar a aprendizagem dos principais conceitos inerentes ao aprendizado de lógica de programação. Essa abordagem realizou intervenções metodológicas através da inclusão de uma atividade prática complementar, com a apresentação de cenários e disponibilização de dicas categorizadas pelo nível de dificuldade. Os resultados obtidos sugerem a continuidade da aplicação dessas intervenções metodológicas.*

1. Introdução

O ensino e aprendizagem de programação de computadores nas universidades tem sido uma problemática abordada em frequentes discussões de trabalhos científicos, tais como Rocha et al. (2010), Giraffa e Mora (2013), Soares e Carvalho (2017) e Souza, Batista e Barbosa (2016). Tais trabalhos são motivados, sobretudo, pela importância dos conceitos de programação na vida acadêmica dos cursos de computação [Rocha et al., 2010] e relatam grande preocupação com os elevados índices de retenção existentes nesses cursos, em virtude das taxas de insucesso dos discentes nas disciplinas iniciais de programação, onde os aspectos básicos de algoritmos e lógica de programação são ensinados.

Existem várias possibilidades de origem das dificuldades no ensino de programação. Pereira Júnior et al. (2005) destacam alguns dos principais problemas que contribuem com o insucesso dos estudante, a citar: (i) dificuldade de interpretação do próprio problema; (ii) dificuldade de identificar os pré-requisitos (lógica matemática, por exemplo) necessários para o desenvolvimento das competências de construção de algoritmos e programas; (iii) dificuldade do aluno em aplicar habilidades prévias, criando fonte de medo e frustração; (iv) necessidade de definir o paradigma de programação utilizado; e (v) dificuldades e/ou falhas advindas do ensino médio.

Entretanto, um problema não destacado, mas crucial ao desenvolvimento de habilidades e ao aprendizado de programação, é a diversidade de ritmos de aprendizagem dos estudantes, conjugada, na maioria dos casos, com turmas muito numerosas. Questiona-se, portanto: **como ministrar atendimento mais direcionado ao estudante, respeitando as individualidades e, ao mesmo tempo, buscando um trabalho cooperativo?** Para Santiago et al. (2017), existe uma necessidade evidente de novas propostas de metodologias instrucionais, as quais não sejam puramente baseadas em código textual.

Desse modo, a partir da identificação da problemática em questão, percebeu-se a oportunidade de propor e aplicar uma abordagem metodológica direcionada a facilitar o aprendizado de conceitos introdutórios de programação de computadores. Assim, este trabalho avalia a utilização de uma proposta metodológica junto a uma turma de discentes iniciantes em programação de computadores do curso de Bacharelado em Tecnologia da Informação da Universidade Federal Rural do Semi-Árido, *Campus* Pau dos Ferros, com a finalidade de estimular o raciocínio lógico e motivar a aprendizagem dos principais conceitos inerentes ao aprendizado de lógica de programação. Essa abordagem realizou intervenções metodológicas através da inclusão de uma atividade prática complementar, com a apresentação de descrições textuais de problemas de programação, denominadas de cenários, e disponibilização de dicas categorizadas pelo nível de dificuldade, destacado por Moreira et al. (2018).

Espera-se estimular nos discentes uma maneira diferenciada de aprendizagem dos conceitos introdutórios de programação, possibilitando o acompanhamento da execução da resolução do algoritmo/programa de maneira mais intuitiva e direcionada com a sua dificuldade.

O presente trabalho encontra-se organizado da seguinte forma: a Seção 2 caracteriza o problema abordado neste estudo; a Seção 3 aborda trabalhos correlatos; a Seção 4 descreve os procedimentos metodológicos adotados; na Seção 5, os resultados são apresentados e discutidos; e, por fim, a Seção 6 apresenta as considerações finais.

2. Caracterização do Problema

A disciplina de Algoritmos constitui a base para o ensino de programação e aborda os princípios do desenvolvimento do raciocínio lógico, com o objetivo de desenvolver a capacidade de análise e resolução de problemas dos alunos através da descrição das soluções na forma de algoritmos. Além disso, esta disciplina faz parte do projeto pedagógico de vários cursos na área de tecnologia, com a finalidade de introduzir os conceitos de programação, exigindo dos alunos determinadas habilidades e desenvolvendo competências. Entre essas, pode-se mencionar a capacidade de pensar

logicamente, a fim de organizar as ideias em forma de código e resolver problemas, tendo como propósito a construção de soluções algorítmicas [Lopes et al. 2017].

Os estudantes, na maioria das vezes, não estão habituados com as novas formas de pensar requeridas por esta disciplina, bem como, não possuem certas aptidões necessárias, encontram dificuldades que, por vezes, culminam na reprovação e evasão [Barbosa 2011]. Tal problemática tem sido motivo de preocupação dos alunos e dos próprios docentes [Vieira, Lima Junior e Vieira 2015].

Segundo Ramos (2011), estudar lógica, racionalizar os conceitos lógicos e, mais precisamente, a lógica de programação, assim como, criar algoritmos e conceitos computacionais, desenvolver aplicações e implementações na forma de software, tornou-se hoje uma área de conhecimento extremamente importante.

De acordo com Souza, Batista e Barbosa (2016), a aprendizagem dos conceitos de programação constitui uma das maiores dificuldades dos alunos de computação e afins. O discente que possui muita dificuldade em programação acaba desistindo das disciplinas ou até evadindo dos cursos.

Diante das opiniões expostas, corrobora-se 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, objetivando minimizar os problemas enfrentados pelos alunos nas disciplinas de programação introdutória.

3. Trabalhos Relacionados

As dificuldades de aprendizagem de programação introdutória têm sido associadas aos altos índices de reprovação e de evasão em cursos de computação e têm sido uma temática muito discutida na literatura acadêmica de Educação em Informática [Giraffa e Mora 2013; Souza, Batista, e Barbosa 2016], como os exemplos discutidos a seguir.

Em [Nunes et al. 2017], foi apresentada uma pesquisa-ação com o objetivo de motivar os alunos a desenvolverem suas aprendizagens de programação de computadores no ensino superior, particularmente na transição da programação de nível iniciante para a programação avançada. Para isto, foi desenvolvida uma abordagem motivacional, denominada SimProgramming. Segundo os autores, a partir das reflexões sobre o processo da pesquisa, pode-se concluir que o SimProgramming, em sua aplicação ao ensino de programação de computadores em turmas intermediárias, é promissor e ainda apresenta potencial para ser usado em outros contextos educacionais.

Em [Lopes et al. 2017], foi apresentada uma proposta de solução pedagógica que auxilia no atendimento às dificuldades de aprendizagem dos alunos iniciantes das disciplinas de Algoritmos e Programação. Os autores utilizaram um sistema de monitoramento a partir da integração da ferramenta CFacil [Gomes e Amaral 2016] em um ambiente de rede cliente e servidor, que permitiu ao docente o acompanhamento, em tempo real, das tarefas propostas aos estudantes em sala de aula, garantindo, desta forma, a proposição e ajuste das estratégias didáticas adotadas, visando um aprimoramento da experiência de ensino e aprendizagem de programação.

Em [Pereira, Oliveira e Oliveira, 2017], foi proposto um método para inferir a zona de aprendizagem de alunos de turmas de Introdução à Programação em ambientes

de correção automática de código. Segundo os autores, foi construído um perfil de programação baseado nos dados deixados pelos estudantes à medida que eles resolvem exercícios nesses sistemas. Os alunos que tiraram notas inferiores a 5 foram classificados em uma zona de dificuldade, do contrário em uma zona de expertise. Utilizou-se algoritmos de aprendizagem de máquina para fazer a predição. Ainda segundo os autores, o modelo preditivo proposto obteve 78,3% de acurácia já nas duas primeiras semanas de aula, o que ultrapassa os resultados de pesquisas que foram conduzidas em cenários semelhantes.

Em [Gul et al. 2017], foi apresentada uma proposta baseada no uso de mapas mentais para auxiliar os alunos na aprendizagem de programação. O estudo discutiu vários fatores que podem causar o insucesso de um aluno na resolução de problemas computacionais, sendo um deles, o fato de que os alunos geralmente evitam utilizar algum tipo de representação do problema antes de codificá-lo. Mediante a esse fato, um estudo experimental foi conduzido, utilizando a programação baseada em texto e em blocos. Segundo os autores, os resultados mostraram que a metodologia baseada em mapas mentais funcionou bem com linguagens de programação baseadas em texto e em blocos para melhorar a aprendizagem dos alunos.

Em [Henrique e Tedesco 2017], foram apresentados os resultados de uma revisão sistemática da literatura sobre os trabalhos relacionados ao processo de ensino e aprendizagem de programação, publicados nos últimos 4 anos no cenário nacional, no SBIE e WalgProg (nas edições de 2015 e 2016). Segundo os autores, foi possível identificar conhecimentos, habilidades, atitudes e competências a partir dos trabalhos analisados.

Por fim, em [Barbosa, Costa e Brito 2017], foi proposto o uso de algoritmos de agrupamento como meio para minimizar o esforço despendido na avaliação de códigos em disciplinas introdutórias de programação. De acordo com os autores, as concordâncias obtidas nos resultados variaram de razoável a perfeita, considerando as avaliações semiautomáticas obtidas com os agrupamentos e as avaliações de especialistas, o que sugere que é possível minimizar o esforço de avaliação despendido.

Em uma perspectiva diferente, esse artigo avalia a utilização de uma proposta metodológica junto a uma turma de programação introdutória do curso de Bacharelado em Tecnologia da Informação da Universidade Federal Rural do Semi-Árido, *Campus Pau dos Ferros*, com a finalidade de estimular o raciocínio lógico e motivar a aprendizagem dos principais conceitos inerentes ao aprendizado de lógica de programação. Essa abordagem realizou intervenções metodológicas através da inclusão de uma atividade prática complementar, com a apresentação de descrições textuais de problemas de programação, denominadas de cenários, e disponibilização de dicas categorizadas com o nível de dificuldade.

4. Metodologia

Para o experimento realizado neste trabalho, adotou-se a proposta metodológica de [Rocha 2010 *apud* Ferreira e Monteiro, 2009], onde são definidos 14 níveis para o estudo de programação imperativa. Como este experimento se deu em uma disciplina denominada Algoritmos, cujo componente curricular é composto de conceitos iniciais de lógica de programação, seguido do estudo e aplicação desses conhecimentos à

linguagem de programação C, adotou-se um roteiro contemplando 6 dos níveis destacados por Rocha (2010). Cada nível foi representado no experimento através da descrição textual de um problema de programação, denominado cenário, conforme apresentado a seguir:

- **Cenário 1:** Bases de programação imperativa, usando a Linguagem C;
- **Cenário 2:** Estruturas de seleção;
- **Cenário 3:** Estruturas de repetição, combinadas com estruturas de seleção;
- **Cenário 4:** Vetores;
- **Cenário 5:** Matrizes;
- **Cenário 6:** Programação Modular.

Para a resolução de cada cenário, foram disponibilizadas três dicas de apoio ao desenvolvimento do código. As dicas foram elaboradas e divididas em três categorias, seguindo as dificuldades de aprendizagem de programação apontadas por Moreira et al. (2018), sendo elas: (i) interpretação do problema; (ii) compreensão da sintaxe; e (iii) estruturação da lógica de programação. A solicitação das dicas poderia ser realizada sempre que o discente apresentasse uma dificuldade de entendimento da resolução do cenário. A Tabela 1 apresenta o conteúdo relacionado a cada dica.

Tabela 1. Conteúdos relacionados a cada dica

Dica 1	Incluía a explicação do cenário, ou seja, os comandos de entrada necessários, os comandos de saída esperados e o processamento a ser realizado.
Dica 2	Apontava a sintaxe (do principal conteúdo abordado na questão) indicada para o desenvolvimento do código referente ao cenário.
Dica 3	Retratava a lógica de programação indicada para a resolução do problema, em forma de descrição narrativa.

O procedimento para a realização dessa abordagem segue o fluxo de atividades ilustrado na Figura 1.



Figura 1. Fluxo de atividades da abordagem adotada

A abordagem adotada e exposta anteriormente foi realizada com 25 discentes de uma turma de Algoritmos do curso de Bacharelado em Tecnologia da Informação da Universidade Federal Rural do Semi-Árido, *Campus* Pau dos Ferros. Dentre estes discentes, quatorze deles estavam cursando a disciplina pela primeira vez, sete pela segunda vez e quatro pela terceira vez. Os cenários foram apresentados e as dicas foram dispostas, de modo que, a cada dificuldade, o discente pudesse consultá-las. Ao final do desenvolvimento de cada cenário, o discente foi direcionado a responder um

questionário fechado com questões sobre: (i) a quantidade de dicas solicitadas para desenvolver o problema apresentado no respectivo cenário; e (ii) o nível de dificuldade da resolução do cenário. Por fim, o discente informou suas sugestões e impressões gerais sobre a abordagem adotada para auxiliar a aprendizagem de programação introdutória, em um último questionário semiaberto. Os dados coletados são apresentados e discutidos na seção a seguir.

5. Resultados e Discussão

Os dados obtidos foram analisados sob uma perspectiva individual e grupal. Observou-se que, no geral, os discentes estudam programação através de algumas práticas comuns, como: estudo individual (31,91%), estudo em grupo (10,63%), por meio de livros e/ou apostilas (10,63%), através de videoaulas (10,63%), na monitoria da disciplina (25,53%) e alguns não estudam com frequência (10,63%).

Em relação ao desempenho geral dos discentes, observou-se que, na maioria dos cenários, o índice de códigos corretos superou o índice de códigos incorretos, parcialmente corretos ou não enviados. Entretanto, percebeu-se que, no cenário 3, apenas 32% dos códigos desenvolvidos foram considerados corretos. Destaca-se, também, o crescimento exponencial dos códigos não enviados ao longo do experimento. Possivelmente, o aumento da dificuldade, de acordo com o avanço nos cenários, pode ter contribuído para esta ocorrência. O Gráfico 1 expõe o desempenho dos discentes em cada cenário.

É válido ressaltar que a avaliação do código de programação desenvolvido em cada cenário se deu da seguinte maneira: (i) Correto: código que compila e executa, de acordo com o cenário; (ii) Parcial: código que compila e executa, mas não de forma coerente com o cenário; e (iii) Incorreto: código não compila e não executa.

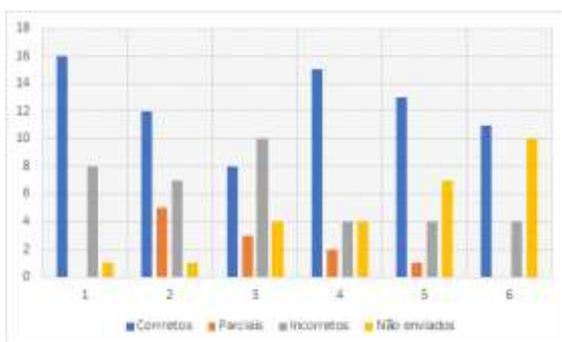


Gráfico 1. Desempenho por cenário

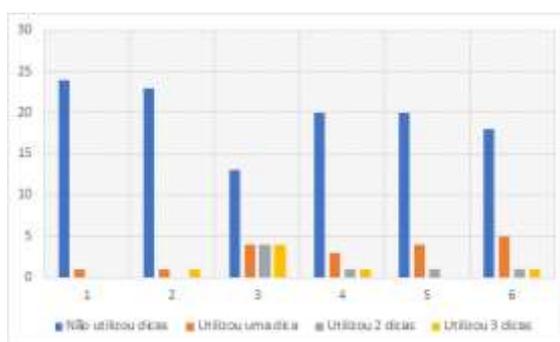


Gráfico 2. Solicitação de dicas por cenário

O Gráfico 2 apresenta a quantidade de dicas solicitadas em cada cenário. Observou-se que boa parte dos discentes não solicitaram dicas e percebeu-se, também, uma frequência de solicitações por dicas pelos mesmos discentes em alguns cenários.

Para uma análise mais específica, dividiu-se os participantes do experimento em dois grupos: (i) Grupo 1: discentes ingressantes na disciplina (14 discentes); e (ii) Grupo 2: discentes repetentes na disciplina (11 discentes). A partir desses agrupamentos, categorizou-se em quatro subgrupos, sendo eles: (i) Subgrupo 1: discentes que desenvolveram mais de 50% de códigos de programação corretos; (ii) Subgrupo 2:

discentes que desenvolveram até 50% de códigos de programação corretos; (iii) Subgrupo 3: discentes que não conseguiram desenvolver o código de programação corretamente em nenhum dos cenários; e (iv) Subgrupo 4: discentes que não desenvolveram o código de programação do cenário. A Tabela 2 apresenta os índices de cada subgrupo.

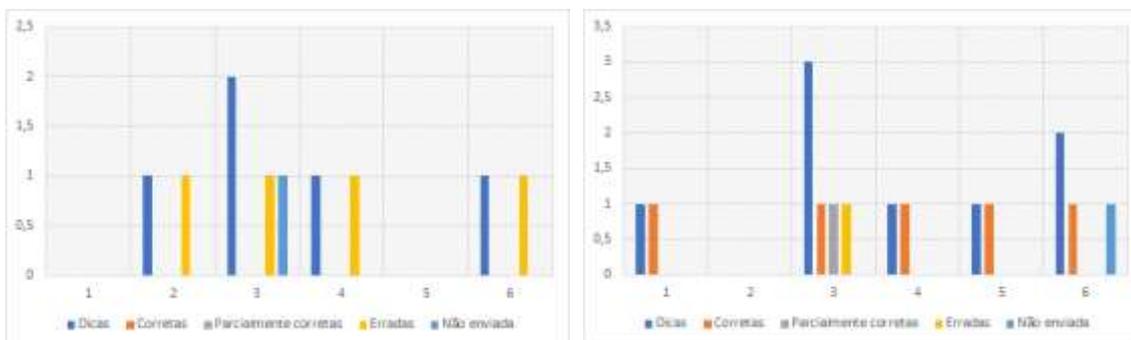
Tabela 2. Análise dos índices dos resultados de cada subgrupo

	Grupo 1	Grupo 2
Subgrupo 1	57,1%	45,5%
Subgrupo 2	14,3%	27,3%
Subgrupo 3	28,6%	18,2%
Subgrupo 4	-	9%

No geral, em relação ao Grupo 1, percebeu-se que os discentes ingressantes (57,1%) desenvolveram mais de 50% de códigos de programação corretos. Enquanto que, 14,3% deles conseguiram obter até 50% de acertos e, os outros 28,6% desenvolveram códigos de programação incorretos. Em relação ao Grupo 2, notou-se que, dentre os discentes repetentes, 45,5% deles conseguiram desenvolver mais de 50% de códigos de programação corretos. Enquanto que, 27,3% conseguiram obter até 50% de acertos e, os outros, 18,2% não apresentaram códigos corretos em nenhum dos cenários. Destacamos que um dos discentes desistiu de participar da atividade - equivalente a 9% do total.

5.1 Aproveitamento de Dicas pelos Subgrupos

Com intuito de verificar a eficiência das dicas, uma análise foi realizada em cada um dos subgrupos citados anteriormente. O objetivo principal dessa verificação foi compreender se as dicas podem, ou não, auxiliar o discente a encontrar uma solução concreta (correta ou parcial) para um determinado problema. Dessa forma, analisou-se a resposta enviada pelo discente em quatro possibilidades: correta, parcialmente correta, incorreta ou não enviada. A Figura 2 apresenta o aproveitamento por dicas dos discente ingressantes.



(a) Acertaram até 50%

(b) Acertaram mais de 50%

Figura 2. Aproveitamento de dicas pelos discentes ingressantes

De acordo com a Figura 2(a), os discentes desse subgrupo solicitaram dicas nos cenários 2, 3, 4 e 6. Nos cenários 2, 4 e 6, apenas um discente requisitou o auxílio, mas não obteve êxito em nenhuma delas. No cenário 3, dois discentes solicitam, mas um código estava incorreto e o outro não foi enviado. Nesse caso, o aproveitamento por dicas não se mostrou satisfatório, já que nenhum discente conseguiu obter sucesso.

Observando a Figura 2(b), os discentes desse subgrupo solicitaram dicas nos cenários 1, 3, 4, 5 e 6. Nos cenários 1, 4 e 5, apenas um discente requisitou o recurso e conseguiu obter sucesso em todos eles, ou seja, 100% de aproveitamento. No cenário 3, três discentes solicitaram, no qual, um deles conseguiu desenvolver o cenário corretamente e o outro parcialmente, atingindo 66,6% de aproveitamento, e o terceiro errou. Por último, no cenário 6, dois discentes requisitam o auxílio, sendo que um acertou e o outro não enviou, atingindo assim 50% de aproveitamento. Dessa forma, pode-se concluir que as dicas foram muito importantes para esse subgrupo, atingindo um valor mínimo de 50%, além de 100% de aproveitamento em 3 cenários. A Figura 3 apresenta o aproveitamento por dicas dos discentes repetentes.

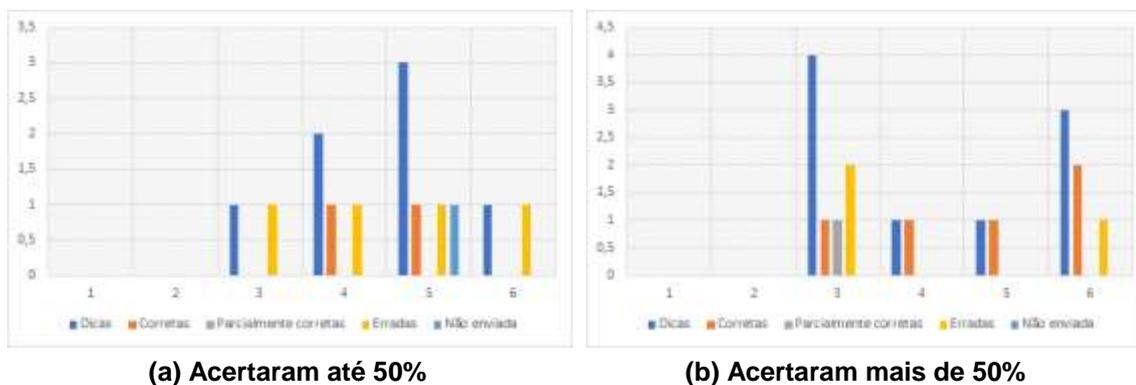


Figura 3. Aproveitamento de dicas pelos discentes repetentes

De acordo com a Figura 3(a), os discentes desse subgrupo requisitaram dicas nos cenários 3, 4, 5 e 6. Nos cenários 3 e 6, apenas um recurso foi solicitado, sendo que em ambos o código estava incorreto. No cenário 4, duas dicas foram requisitadas, onde um código estava correto e o outra incorreto, obtendo assim, um aproveitamento de 50%. No cenário 5, três dicas foram solicitadas, sendo que um código estava correto, outro incorreto e um código não foi enviado, resultando em um aproveitamento de 33,3%. Em meio a esses dados, as dicas se mostraram razoáveis, alcançando no máximo um aproveitamento de 50%.

Baseando-se na Figura 3(b), os discentes desse subgrupo solicitaram dicas nos cenários 3, 4, 5 e 6. Nos cenários 4 e 5, apenas um discente requisitou a dica e conseguiu obter sucesso em todos eles, ou seja, 100% de aproveitamento. Nos cenários 3, quatro discentes solicitaram, no qual um deles conseguiu acertar totalmente, outro parcialmente e dois erraram as respostas, atingindo 50% de aproveitamento. Por último, no cenário 6, três discentes requisitaram a dica, no qual, dois deles tiveram êxito no código desenvolvido e um deles errou a resposta, obtendo dessa forma, um aproveitamento de 66,6%. Assim, pode-se concluir que as dicas foram relevantes, pois atingiu-se um valor mínimo de 50% em dois dos cenários, além de 100% de aproveitamento nos outros dois cenários.

6. Considerações Finais

Este trabalho descreveu a aplicação de uma abordagem metodológica direcionada a facilitar o aprendizado de conceitos introdutórios de programação de computadores. Além disso, foi descrita a avaliação do impacto desta abordagem no rendimento dos alunos ao final do experimento.

De acordo com os resultados da avaliação e as opiniões dos participantes, percebeu-se que a abordagem aplicada é atrativa e, portanto, sugere-se a continuidade da aplicação dessas intervenções metodológicas. Entretanto, a abordagem apresentada não esgota, tampouco soluciona, em absoluto, o problema relacionado ao ensino e aprendizagem de programação introdutória em cursos superiores, mas contribui na exploração de estratégias metodológicas para acompanhar e avaliar o desempenho dos discentes na prática de abstração e solução de problemas de programação.

Como trabalhos futuros, pretende-se: (i) aplicar tal abordagem metodológica com outras turmas para uma análise mais abrangente de sua eficácia; (ii) aplicar técnicas de *Learning Analytics* para verificar possíveis correlações entre os dados obtidos neste trabalho, como, por exemplo, verificar se existe correlação entre o aproveitamento/sucesso do aluno e as dicas fornecidas a eles; e (iii) desenvolver um software educativo para implementar a ideia descrita na Figura 1, de modo a incentivar os alunos a aprenderem de forma intuitiva.

Referências

- Barbosa, A. A., Costa, E. B. e Brito, P. H. (2017) “Uma abordagem adaptativa para gerar agrupamento de códigos em disciplinas de programação introdutória”, In: Anais do XXVIII Simpósio Brasileiro de Informática na Educação (SBIE), Recife.
- Barbosa, L. S. (2011). Aprendizado significativo aplicado ao ensino de algoritmos. 57f. Dissertação (Mestrado em Sistemas e Computação), Universidade Federal do Rio Grande do Norte, Natal.
- Giraffa, L. M. e Mora, M. C. (2013) “Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno”, In: Anais do III Conferencia sobre el Abandono en la Educación Superior (CLABES), Cidade do México.
- Gomes, M. S e Amaral, E. M. H. (2016) “Uma Proposta de Ferramenta para Simplificar a Depuração de Códigos em C, por Alunos Iniciantes”, In: Anais dos Workshops do V Congresso Brasileiro de Informática na Educação (WCBIE), Uberlândia.
- Gul, S., Asif, M., Ahmad, W. e Ahmad, U. (2017) “Teaching Programming: A Mind Map based Methodology to Improve Learning Outcomes”, In: International Conference on Information and Communication Technologies (ICICT), Karachi.
- Henrique, M. S. e Tedesco, P. C. A. R. (2017) “Uma Revisão sistemática da Literatura sobre conhecimentos, habilidades, atitudes e competências desejáveis para auxiliar a aprendizagem de programação”, In: Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação (WCBIE), Recife.
- Lopes, P. P., Gomes, M. S., Dantas, T. F. e Amaral, E. M. H. (2017) “Proposta de um Sistema para o Monitoramento das Atividades de Programação Para Alunos

- Iniciantes”, In: Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação (WCBIE), Recife.
- Moreira, G. L., Holanda, W., Coutinho, J. C. S. e Chagas, F. S. (2018) “Desafios na aprendizagem de programação introdutória em cursos de TI da UFERSA, campus Pau dos Ferros: um estudo exploratório”, In: Anais do III Encontro de Computação do Oeste Potiguar (ECOP), Pau dos Ferros.
- Nunes, R. R., Pedrosa, D., Morgado, L., Paredes, H., Martins, P., Cravino, J. e Barreira, C. (2017) “SimProgramming: uma abordagem motivacional para a aprendizagem de alunos intermediários de programação”, In: Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação (WCBIE), Recife.
- Pereira, F. D., Oliveira, E. H. T. e Oliveira, D. F. (2017) “Predição de Zona de Aprendizagem de Alunos de Introdução à Programação em Ambientes de Correção Automática de Código”, In: Anais do XXVIII Simpósio Brasileiro de Informática na Educação (SBIE), Recife.
- Pereira Júnior, J. C. R., Rapkiewicz, C. E., Delgado, C. e Xexeo, J. A. M. (2005) “Ensino de algoritmos e programação: uma experiência no nível médio”, In: Anais do XIII Workshop de Ensino em Informática (WEI), São Leopoldo.
- Ramos, R. A. D. O. (2011). O uso de mídias interativas na compreensão de conceitos da lógica computacional. 90f. Dissertação (Mestrado em Mídias Digitais), Pontifícia Universidade Católica de São Paulo, São Paulo.
- Rocha, S. P., Ferreira, B., Monteiro, D., Nunes, D. S. C. e Góes, H. C. N. (2010). Ensino e Aprendizagem de Programação: Análise de Proposta Metodológica Baseada no Sistema Personalizado de Ensino. In *Revista Novas Tecnologias na Educação*. v. 8, n. 3.
- Santiago, A. D. V. e Kronbauer, A. H. (2017). Um Modelo Lúdico para o Ensino de Conceitos de Programação de Computadores. In *Revista Brasileira de Informática na Educação*. v. 25, n. 3, p. 1-29.
- Soares, F. A. L. e Carvalho, R. B. (2017). Proposta de um Portal Educacional para estudantes de programação de computadores. In *Revista Abakós*. v. 5, n. 2, p. 36-58.
- Souza, D. M., Batista, M. H. S. e Barbosa, E. F. B. (2016). Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: Um Mapeamento Sistemático. In *Revista Brasileira de Informática na Educação*, v. 24, p. 39-52.
- Vieira, C. E. C., Lima Junior, J. A. T. e Vieira, P. P. (2015) Dificuldades no Processo de Aprendizagem de Algoritmos: uma Análise dos Resultados na Disciplina de AL1 do Curso de Sistemas de Informação da FAETERJ – Campus Paracambi. In *Cadernos UniFOA*, v. 10, n. 27, p. 5-15.