

Pré-Algoritmos – Ações de Apoio à Melhoria do Ensino de Graduação

Reudismam Rolim de Sousa¹, Felipe Torres Leite¹,
Ádller de Oliveira Guimarães¹, Assunaueny Rodrigues de Oliveira¹

¹Departamento de Engenharias e Tecnologia
Universidade Federal Rural do Semi-Árido (UFERSA)
Rodovia BR-226, s/n, Pau dos Ferros - RN, 59900-000 – Brasil

{reudismam.sousa, adller.guimaraes, felipe.leite}@ufersa.edu.br,
assunauenyufersa@gmail.com

Abstract. *Courses in the computing area require that students solve problems using algorithms. To this end, students need to develop the logical thinking to solve these types of problems, which can become a challenge for some students. Thus, several approaches have been proposed to address this problem, such as the use of online judges and block-based programming languages. However, there is no closed solution to teach algorithms. In this work, it is proposed to create a class entitled “Pré-Algoritmos” to assist students with the class “Algoritmos”. As a result, it was identified that the students that were approved in Pré-Algoritmos obtained better performance in Algoritmos.*

Resumo. *Os cursos na área de computação requerem que os estudantes resolvam problemas por meio de algoritmos. Para isso, os alunos precisam desenvolver o raciocínio lógico para resolver esses tipos de problemas, o que pode se tornar um desafio para alguns estudantes. Dessa forma, várias abordagens foram desenvolvidas para minimizar esse problema, tais como o uso de juízes online e linguagens de programação por blocos. Porém, não há uma solução pronta para ensinar algoritmos. Neste trabalho, propõe-se a criação de uma disciplina denominada “Pré-Algoritmos” para auxiliar os estudantes com dificuldades no componente curricular Algoritmos. Como resultado, identificou-se que os alunos que participaram de Pré-Algoritmos obtiveram um melhor desempenho em Algoritmos.*

1. Introdução

Ao entrar em um curso superior na área de computação, os alunos precisam desenvolver a habilidade de resolver problemas por meio de linguagens de programação. Dessa forma, eles necessitam desenvolver o raciocínio lógico e a capacidade de abstração, o que se torna um desafio para alguns alunos [Moreira et al. 2018]. Em especial, os alunos do Bacharelado em Tecnologia da Informação (BTI) da Universidade Federal Rural do Semi-Árido (UFERSA), Centro Multidisciplinar de Pau dos Ferros – RN, estão apresentando dificuldades na compreensão do componente curricular PEX1236 - Algoritmos, o que vem causando problemas de repetência, evasão e desmotivação nesse componente curricular. Notadamente, vem aumentando o número de reprovados, ao longo dos períodos letivos, como pode ser visto na Tabela 1.

Tabela 1. Insucessos em Algoritmos.

Período	Insucessos	Discentes	%
2015.2	64	79	81 %
2016.1	64	115	56 %
2016.2	97	111	87 %
2017.1	91	130	70 %
2017.2	99	150	66 %

Fonte: Autoria própria

Por ser um componente curricular de primeiro período, em BTI na UFERSA – Pau dos Ferros – RN, uma das possíveis causas para este aumento é o “baixo” nível na formação da educação básica apresentado por alguns ingressantes no curso. Acrescenta-se a isso, a dificuldade de interpretação de exercícios que podem ser resolvidos por meio de algoritmos [Queiroz et al. 2018]. Tais problemas dificultam o desenvolvimento do raciocínio lógico dos alunos, indispensável à criação de algoritmos. Outra possível causa é a desmotivação dos discentes para superar suas dificuldades, utilizando recursos de aprendizagem complementares oferecidos pela universidade, tais como os monitores de componentes curriculares ou mesmo estudar com a frequência desejada [Moreira et al. 2018].

Diante deste cenário, precisa-se buscar soluções para reduzir os índices de repetência, evasão e desmotivação no referido componente curricular. Dessa forma, neste trabalho, propõe-se um projeto para ações de apoio à melhoria do ensino de graduação, que busca novas estratégias de ensino de programação, visando contemplar os alunos ingressantes no curso de BTI, na tentativa de reduzir as taxas de insucessos em Algoritmos.

Nesse projeto, criou-se uma disciplina, intitulada Pré-Algoritmos, que funciona como uma monitoria, mas similar a um componente curricular tradicional, abordando propostas de ensino-aprendizagem que buscam desenvolver a lógica de programação, com respeito aos conteúdos já abordados no componente curricular Algoritmos. Pré-Algoritmos ocorre paralelamente a Algoritmos e, para obter sucesso na disciplina, os alunos são avaliados pela frequência, ao invés de uma avaliação somativa, como nos componentes curriculares tradicionais. Como metodologia de ensino-aprendizagem, a disciplina incentiva os estudantes a resolverem problemas de programação de juízes *online*, tais como o URI [BEZ and TONIN 2014], desenvolvido pela Universidade Regional Integrada do Alto Uruguai e das Missões (URI), o que também os motivam a participarem dessas competições. Além disso, há um enfoque no uso de linguagem de blocos como Scratch [MIT 2019] e Code.org [Code.org 2019], nas aulas iniciais da disciplina em que o aluno possui pouco ou nenhum contato com programação, que são comumente utilizadas nesse contexto [Santos et al. 2018].

1.1. Objetivo Geral

Pretende-se por meio deste projeto fornecer para os alunos do curso de BTI do Centro Multidisciplinar de Pau dos Ferros – RN os fundamentos teóricos e práticos básicos necessários para um melhor desempenho no componente curricular Algoritmos.

1.2. Objetivos Específicos

1. Reduzir o número de insucessos no componente curricular Algoritmos;

2. Desenvolver no aluno bolsista a prática docente;
3. Elaborar materiais didáticos complementares para utilização no componente curricular Algoritmos.

2. Pré-Algoritmos

A disciplina Pré-Algoritmos objetiva desenvolver o raciocínio lógico dos alunos iniciantes no componente curricular Algoritmos do curso de BTI. Durante a disciplina, os alunos solucionam diversos problemas de acordo com os conteúdos vistos em sala de aula no decorrer do componente curricular Algoritmos. Os problemas são selecionados de uma plataforma com problemas típicos de maratonas de programação, com níveis de complexidade variados, do nível iniciante ao nível avançado. Por tratar-se de uma disciplina introdutória, usou-se apenas problemas de nível iniciante.

2.1. Metodologia

1. Inicialmente, os participantes do projeto se reuniram com os docentes que ministraram ou ministram o componente curricular Algoritmos na UFERSA do Centro Multidisciplinar de Pau dos Ferros – RN para identificar as dificuldades enfrentadas pelos alunos. Também foram analisados os principais problemas apresentados pela literatura para as dificuldades dos estudantes [Moreira et al. 2018, Queiroz et al. 2018];
2. Para sanar as dificuldades dos alunos, foram desenvolvidas iniciativas para ensino de algoritmos que buscaram o desenvolvimento do raciocínio lógico, utilizando Tecnologias de Informação e Comunicação em Educação – TICs de apoio ao aprendizado, tal como a linguagem Scratch e o ambiente de aprendizado Code.org, bem como novas metodologias de ensino;
3. Os participantes do projeto lecionaram a disciplina Pré-Algoritmos, durante quatro horas semanais, totalizando sessenta horas;
4. Durante a disciplina, são propostos aos alunos problemas de programação, que podem ser respondidos em uma linguagem de programação, tal como VisualG [VisualG 2019] e C, ou em uma linguagem de blocos;
5. Foram determinados horários de atendimento aos alunos beneficiados pelo projeto para poderem tirar dúvidas com um docente associado ao projeto;
6. O aluno bolsista auxiliou o professor de Pré-Algoritmos na aplicação de listas de exercícios em sala de aula.

Para selecionar os alunos participantes, foi realizada uma chamada pública, efetuada pela Secretaria das Coordenações de Curso, buscando atender o máximo de alunos.

2.2. Bolsista

O projeto recebeu um bolsista no último semestre de execução, com dedicação de doze horas semanais e as seguintes atribuições:

1. Atender aos alunos beneficiados pelo projeto, apoiando e tirando dúvidas referentes aos conteúdos ministrados em sala de aula;
2. Auxiliar o coordenador na elaboração de listas de exercícios, materiais didáticos e resumos;

3. Auxiliar na criação de um sítio *web* para divulgar o projeto e os materiais desenvolvidos pela equipe;
4. Aplicar e resolver listas de exercícios com os alunos beneficiados pelo projeto;
5. Comunicar semanalmente ao coordenador do projeto as principais análises sobre os alunos e eventuais solicitações dos mesmos.

O bolsista foi cedido pelo projeto ações de apoio à melhoria do ensino de graduação – AAMEG, da UFERSA. O coordenador do projeto avaliou o bolsista semanalmente mediante o cumprimento das atividades propostas, tendo ele obtido êxito na execução dessas atividades.

2.3. URI Online Judge


O URI é um juiz *online* parte de um projeto sendo desenvolvido pelo Departamento de Ciência da Computação da URI, tendo como principal objetivo o de promover a prática de programação e o compartilhamento de conhecimento [BEZ and TONIN 2014]. O URI possui problemas em diferentes categorias, tais como problemas para iniciantes, problemas matemáticos ou problemas mais complexos envolvendo grafos.

Os problemas do URI foram escolhidos, pois são descritos de forma detalhada, usando diferentes formas de explanação. Por exemplo, uma questão do URI apresenta uma descrição geral do problema, uma descrição da entrada e da saída esperada e exemplos de entradas e saídas para o problema. Na Figura 1, pode ser visto um exemplo de uma questão do URI.

Figura 1. Exemplo de uma questão do URI

URI Online Judge | 1001

Extremamente Básico

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

Leia 2 valores inteiros e armazene-os nas variáveis **A** e **B**. Efetue a soma de **A** e **B** atribuindo o seu resultado na variável **X**. Imprima **X** conforme exemplo apresentado abaixo. Não apresente mensagem alguma além daquilo que está sendo especificado e não esqueça de imprimir o fim de linha após o resultado, caso contrário, você receberá "Presentation Error".

Entrada

A entrada contém 2 valores inteiros.

Saída

Imprima a mensagem "X = " (letra X maiúscula) seguido pelo valor da variável **X** e pelo final de linha. Cuide para que tenha um espaço antes e depois do sinal de igualdade, conforme o exemplo abaixo.

Exemplos de Entrada	Exemplos de Saída
10 9	X = 19
-10 4	X = -6

Fonte: Adaptada de URI [BEZ and TONIN 2014]

2.4. Linguagens de Blocos

Nas aulas iniciais da disciplina, linguagens de blocos são apresentadas aos alunos. Uma linguagem de bloco permite que alunos com pouco ou nenhum conhecimento com linguagens de programação resolvam problemas, pela composição de blocos visuais que podem ser traduzidos para uma linguagem de programação convencional [MIT 2019, Code.org 2019].

Uma das linguagens de blocos apresentadas foi o Scratch, que foi projetado para ser usado por alunos de 8 a 16 anos, mas pode ser utilizado por todas as idades [MIT 2019]. Esse ambiente é utilizado em mais de cento e cinquenta países e disponível em mais de quarenta idiomas diferentes [MIT 2019].

Na Figura 2, pode ser visto um exemplo do ambiente Scratch. Para solucionar um problema, o aluno pode selecionar os blocos mais apropriados (Área 1), arrastar esses blocos para compor a solução do problema (Área 2). Quando executado, o personagem realiza as ações descritas pelo programa (Área 3).

Figura 2. Linguagem de blocos Scratch



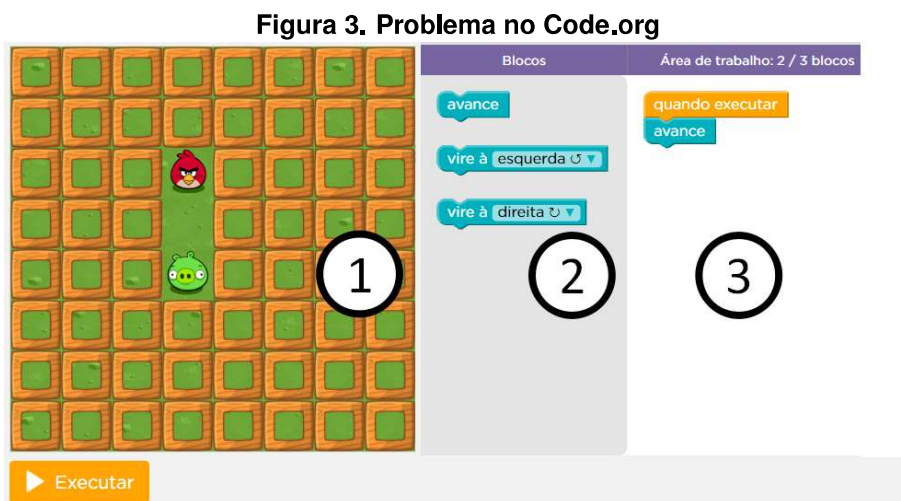
Fonte: Adaptada de Scratch [MIT 2019]

Outra linguagem de blocos apresentada aos alunos é a disponível pela organização sem fins lucrativos Code.org [Code.org 2019]. Essa organização dedica-se a expandir o acesso à Ciência da Computação em escolas e aumentar a participação das mulheres e das minorias não representadas [Code.org 2019]. A organização é apoiada por companhias, tais como Amazon, Facebook, Google e Microsoft [Code.org 2019]. Na Figura 3, pode ser visto um problema de programação no ambiente Code.org. O problema apresenta uma descrição visual (Área 1). O aluno deve selecionar um conjunto de blocos (Área 2) e montar uma solução para o problema (Área 3).

Além dessas linguagens de bloco, o URI permite que o estudante submeta problemas usando linguagens de blocos.

3. Trabalhos Relacionados

Queiroz et al. [Queiroz et al. 2018] investigou os aspectos motivantes no processo de ensino-aprendizado em programação em um estudo de caso com estudantes do curso de BTI da UFERSA, no Centro Multidisciplinar de Pau dos Ferros-RN. Eles identificaram que 60% dos alunos apresentam dificuldades na compreensão da lógica de programação. Alguns dos fatores apontados pelos estudantes são não compreender conteúdos complexos em pouco tempo, não se interessar por aulas teóricas e erros contínuos no código.



Fonte: Modificada de Code.org [Code.org 2019]

Por sua vez, Moreira et al. [Moreira et al. 2018] realizou um *survey* com 110 estudantes universitários de programação introdutória na UFERSA, para identificar os principais desafios enfrentados pelos estudantes para aprender programação e os principais conteúdos que os alunos apresentam dificuldades. Como resultado, os autores identificaram que a maior dificuldade dos alunos é em desenvolver a lógica de programação. Além desta, os alunos apontaram outras dificuldades, tal como o entendimento da sintaxe, a falta de tempo para se dedicar a disciplina e a interpretação dos problemas propostos.

No cenário internacional, Hawi [Hawi 2010] investigou as principais causas que levam ao sucesso e ao fracasso os alunos de uma disciplina introdutória de programação da Universidade Notre Dame, Louaize. Foram selecionados 45 alunos de sexos distintos, sendo a maioria de sexo masculino, para elicitare as possíveis causas de insucessos. As principais causas apontadas foram a habilidade, esforço, dificuldade na tarefa e a sorte.

4. Resultados e Discussões

Para avaliar o projeto, aplicou-se a disciplina Pré-Algoritmos em dois semestres seguidos, 2018.1 e 2018.2 no curso superior de BTI da UFERSA, no Centro Multidisciplinar de Pau dos Ferros – RN. Para identificar a efetividade da disciplina, comparou-se o desempenho dos alunos que foram aprovados em Pré-Algoritmos com os demais alunos que foram matriculados em Algoritmos, mas que não fizeram parte de Pré-Algoritmos ou não obtiveram frequência mínima em Pré-Algoritmos (denominados de alunos tradicionais). Para ser aprovado em Pré-Algoritmos, adotou-se como critério de frequência mínima 50 %.

Na Tabela 2, podem ser vistos os dados referente aos dois semestres de execução do projeto para ações de apoio à melhoria do ensino de graduação. Em ambos os semestres, foram aprovados em torno de 20 alunos (22 no período 2018.1 e 20 no período 2018.2). Nos dois períodos participaram de Algoritmos em torno de 100 alunos (98 no período 2018.1 e 104 no período 2018.2). Nos dois períodos, o número de alunos que foram aprovados em Pré-Algoritmos e também em Algoritmos foi o mesmo, 16. Comparando os alunos aprovados em Pré-Algoritmos com os alunos tradicionais, percebe-se que dos alunos aprovados em Pré-Algoritmos, 76 % foram aprovados em Algoritmos, enquanto que apenas 36 % dos alunos tradicionais foram aprovados em Algoritmos.

Tabela 2. Resultado do projeto Pré-Algoritmos

Descrição	Valores
Aprovados em Pré-Algoritmos (2018.1)	22
Aprovados em Pré-Algoritmos e Algoritmos (2018.1)	16
Total de Alunos em Algoritmos (2018.1)	98
Aprovados em Algoritmos 2018.1	34
Aprovados em Pré-Algoritmos (2018.2)	20
Aprovados em Pré-Algoritmos e Algoritmos (2018.2)	16
Total de Alunos em Algoritmos (2018.2)	104
Aprovados em Algoritmos 2018.2	66
Porcentagem de Aprovados em Pré-Algoritmos e Algoritmos (2018.1)	73 %
Porcentagem de Aprovados em Pré-Algoritmos e Algoritmos (2018.2)	80 %
Porcentagem de Aprovados em Algoritmos (2018.1)	35 %
Porcentagem de Aprovados em Algoritmos (2018.2)	63 %
Porcentagem de Aprovados em Pré-Algoritmos e Algoritmos 2018.1 e 2018.2	76 %
Porcentagem de Aprovados em Algoritmos 2018.1 e 2018.2	50 %
Porcentagem de Alunos Tradicionais Aprovados em Algoritmos 2018.1 e 2018.2	36 %

Fonte: Autoria própria

4.1. Ameaças à Validade e Limitações

O trabalho apresenta ameaças com respeito a sua validade. Como ameaça à validade interna, tem-se que os alunos que participam de Pré-Algoritmos podem ter uma pré-disposição para o aprendizado de Algoritmos, em contraste com alunos que tenham mais dificuldade na disciplina. Para amenizar essa ameaça, realizou-se uma chamada pública, em que todos os alunos possuíam a mesma oportunidade de fazer parte da disciplina, o que pode incluir tanto alunos motivados para o aprendizado quando alunos que apresentavam dificuldades. Como ameaça à validade externa, a disciplina foi aplicada apenas na UFERSA e poderia não funcionar para outras instituições. Para mitigar essa ameaça, a disciplina foi aplicada a turmas de diferentes semestres de Algoritmos, além de ministrada por diferentes membros do projeto ao longo dos semestres, usando a mesma metodologia. Além disso, o projeto foi aplicado a disciplina de Algoritmos com três professores diferentes ao longo do semestre em que a disciplina Pré-Algoritmos foi aplicada.

5. Conclusão

O ensino de disciplinas relacionadas à programação vem se tornando um desafio para alunos ingressantes em cursos superiores na área de computação. Em especial no curso de BTI na UFERSA, Centro Multidisciplinar de Pau dos Ferros – RN, vem aumentando o número de alunos reprovados a cada semestre letivo. Para amenizar esse problema, nesse trabalho propõe-se o projeto para ações de apoio à melhoria do ensino de graduação intitulado Pré-Algoritmos, que funciona como uma monitoria, mas similar a um componente curricular tradicional, abordando conteúdos que buscam desenvolver a lógica de programação e temas já abordados no componente curricular Algoritmos. Para avaliar o projeto para ações de apoio à melhoria do ensino de graduação, comparou-se o desempenho dos alunos que participaram da abordagem, com o desempenho dos alunos tradicionais (alunos que não participaram de Pré-Algoritmos ou que não obtiveram frequência

mínima em Pré-Algoritmos). Como resultado, identificou-se que dos alunos aprovados em Pré-Algoritmos, 76 % foram aprovados em Algoritmos. Em contraste, apenas 36 % dos alunos tradicionais foram aprovados em Algoritmos. Dessa forma, conclui-se que o projeto para ações de apoio à melhoria do ensino de graduação auxiliou na redução do número de insucessos na disciplina Algoritmos.

Como trabalho futuro, pretende-se realizar um estudo para rastrear o desempenho dos alunos, em outras disciplinas que tenham como pré-requisito o componente curricular Algoritmos, tais como Algoritmos e Estrutura de Dados I e II. Adicionalmente, pretende-se avaliar uma nova abordagem para apresentar os conteúdos antes de serem abordados em Algoritmos, de forma que os estudantes possam participar de Algoritmos com algum conhecimento sobre o conteúdo ministrado. Da forma como o curso é ministrado atualmente, os conteúdos são abordados apenas depois de mostrados em Algoritmos.

Agradecimentos

Agradecemos à UFERSA pela infraestrutura disponibilizada e por ceder um bolsista para o projeto, pelo Edital AAMEG, lançado pela PROGRAD, ao Instituto Metr pole Digital – IMD por ceder um laborat rio para condu o do projeto, aos os revisores an nimos pela sugest es e a todos que contribuíram para o projeto.

Refer ncias

- [BEZ and TONIN 2014] BEZ, J. L. and TONIN, N. A. (2014). URI online judge e a internacionaliza o da universidade. Dispon vel em <https://scratch.mit.edu/> Acessado em 15 de abril de 2019.
- [Code.org 2019] Code.org (2019). Code.org. Dispon vel em <https://code.org/> Acessado em 15 de abril de 2019.
- [Hawi 2010] Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education*, 54(4):1127–1136.
- [MIT 2019] MIT (2019). Scratch. Dispon vel em <https://scratch.mit.edu/> Acessado em 15 de abril de 2019.
- [Moreira et al. 2018] Moreira, G. L., Holanda1, W., da S. Coutinho, J. C., and Chagas, F. S. (2018). Desafios na aprendizagem de programac o introdut ria em cursos de ti da ufersa, campus pau dos ferros: um estudo explorat rio. In *Proceedings of the III Encontro do Oeste Potiguar, ECOP '18*, pages 90–96. ECOP.
- [Queiroz et al. 2018] Queiroz, J. V., Rodrigues, L. M., and Coutinho, J. (2018). Um relato dos fatores motivacionais na aprendizagem de programac o na perspectiva de alunos iniciantes em programac o da universidade federal rural do semi- rido campus pau dos ferros-rn. In *Proceedings of the III Encontro do Oeste Potiguar, ECOP '18*, pages 90–96. ECOP.
- [Santos et al. 2018] Santos, P. S. C., Araujo, L. G. J., and Bittencourt, R. A. (2018). A mapping study of computational thinking and programming in brazilian k-12 education. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–8.
- [VisualG 2019] VisualG (2019). VisualG 3. Dispon vel em <http://visualg3.com.br/> Acessado em 15 de abril de 2019.