

**Pré-Algoritmos – Ações de Apoio à Melhoria do Ensino de Graduação****Pre-Algorithms - Actions to Support the Improvement of Undergraduate Education**

DOI:10.34117/bjdv6n3-213

Recebimento dos originais: 10/02/2020

Aceitação para publicação: 16/03/2020

**Reudismam Rolim de Sousa**

Departamento 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@ufersa.edu.br

**Felipe Torres Leite**

Departamento de Engenharias e Tecnologia  
Universidade Federal Rural do Semi-Árido (UFERSA)  
Rodovia BR-226, s/n, Pau dos Ferros - RN, 59900-000 - Brasil  
felipe.leite@ufersa.edu.br

**Ádller de Oliveira Guimarães**

Departamento de Engenharias e Tecnologia  
Universidade Federal Rural do Semi-Árido (UFERSA)  
Rodovia BR-226, s/n, Pau dos Ferros - RN, 59900-000 - Brasil  
adller.guimaraes@ufersa.edu.br

**Assunaueny Rodrigues de Oliveira**

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

**RESUMO**

Os cursos na área de computação requerem que os estudantes resolvam problemas por meio de algoritmos. Logo, os alunos precisam desenvolver o raciocínio lógico para resolver estes tipos de problemas, o que pode se tornar um desafio para alguns estudantes. Desta forma, várias abordagens foram desenvolvidas para minimizar este problema, tais como o uso de juízes online e linguagens de programação por blocos. Porém, não há uma solução consolidada para ensinar algoritmos. Neste trabalho, é proposta a criação de uma disciplina denominada “Pré-Algoritmos” para auxiliar os estudantes com dificuldades no componente curricular “Algoritmos”. Como resultado, foi identificado que os alunos aprovados em Pré-Algoritmos obtiveram um melhor desempenho em Algoritmos.

**Palavras-chave:** Algoritmos, Educação, Computação

**ABSTRACT**

Courses in the computing area require that students solve problems using algorithms. Therefore, 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

consolidated 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.

**Keywords:** Algorithms, Education, Computação

## 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. Desta forma, eles necessitam desenvolver o raciocínio lógico e a capacidade de abstração, o que se torna um desafio para alguns estudantes [Moreira et al. 2018]. Em especial, os discentes do Bacharelado Interdisciplinar em Tecnologia da Informação (BTI) da Universidade Federal Rural do Semi-Árido (UFERSA), campus Pau dos Ferros, vêm apresentando dificuldades na compreensão teórico-prática do componente curricular denominado de “Algoritmos”. Isto vem causando problemas de repetência, evasão e desmotivação para o componente curricular. Notadamente, há um alto grau de insucessos, ao longo dos semestres letivos, como pode ser visto na Tabela 1.

**Tabela 1. Insucessos em Algoritmos**

<b>Período</b>	<b>Insucessos</b>	<b>Discentes</b>	<b>%</b>
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

Como pode ser visto na Tabela 1, a taxa de insucesso ao longo de dois anos e meio (cinco semestres) atinge mais de 50% dos discentes matriculados na referida disciplina. Por ser um componente curricular de primeiro período no referido bacharelado, uma das possíveis causas para estes altos índices é alguma insuficiência de conhecimento referente à formação da educação básica apresentada por alguns ingressantes no curso. Acrescenta-se a isto 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 as suas dificuldades, utilizando

recursos de aprendizagem complementares oferecidos pela universidade, tais como a monitoria ou mesmo estudar com a frequência desejada [Moreira et al. 2018].

Diante deste cenário, é necessário buscar soluções para reduzir estas taxas de insucessos e este trabalho propõe um projeto com ações de apoio à melhoria do ensino de graduação, buscando novas estratégias de ensino de programação visando contemplar os alunos ingressantes em BTI, na tentativa de reduzir estes problemas.

Neste projeto, criou-se um curso intitulado de “Pré-Algoritmos”, que funciona como uma monitoria, porém, similar a um componente curricular tradicional. Nele, são abordadas propostas de ensino-aprendizagem que buscam desenvolver a lógica de programação, com respeito aos conteúdos já abordados no componente curricular Algoritmos, com enfoque em atividades práticas. Pré-Algoritmos ocorre concomitantemente a Algoritmos e, para obter sucesso no curso, os alunos são avaliados pela frequência e cumprimento de tarefas, ao invés de uma nota média, como nos componentes tradicionais.

Como metodologia de ensino-aprendizagem, o curso incentiva os estudantes a resolverem problemas de programação de juízes online, como, por exemplo, o URI [BEZ e TONIN 2014], desenvolvido pela Universidade Regional Integrada do Alto Uruguai e das Missões (URI), o que também os motivam a participarem de competições de programação. Além disso, inicialmente há um enfoque no uso de linguagem de blocos como *Scratch* [MIT 2019] e *Code.org* [Code.org 2019], no momento em que o aluno possui pouco ou nenhum contato com programação, que são comumente utilizadas nesse contexto [Santos et al. 2018].

O projeto Pré-Algoritmos foi aplicado durante o ano letivo de 2018, totalizando dois semestres (2018.1 e 2018.2).

### 1.1 OBJETIVO GERAL

Pretende-se por meio deste projeto fornecer para os alunos do curso de BTI da UFERSA, campus Pau dos Ferros, os fundamentos teóricos e práticos básicos necessários para um melhor desempenho no componente curricular “Algoritmos”.

### 1.2 OBJETIVOS ESPECÍFICOS

- Reduzir o número de insucessos no componente curricular “Algoritmos”;
- Desenvolver no aluno bolsista à prática docente;
- Elaborar e aplicar exercícios complementares para auxiliar estudos para o componente curricular “Algoritmos”.

## 2 PRÉ-ALGORITMOS

O curso de Pré-Algoritmos objetiva desenvolver o raciocínio lógico dos alunos iniciantes em Algoritmos do curso de BTI. Durante o curso, os discentes solucionam diversos exercícios de acordo com os conteúdos vistos em sala de aula no decorrer do componente Algoritmos. As tarefas são selecionadas 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 um curso introdutório, foram usados apenas problemas de nível iniciante.

### 2.1 METODOLOGIA

Inicialmente, os participantes do projeto se reuniram com os docentes que ministraram ou ministram Algoritmos na instituição para identificar as dificuldades enfrentadas pelos alunos já identificadas. Também foram analisados os principais pontos apresentados pela literatura sobre as adversidades enfrentadas pelos estudantes [Moreira et al. 2018, Queiroz et al. 2018].

Para saná-las, 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, tais como a linguagem *Scratch* e o ambiente de aprendizado *Code.org*, bem como novas metodologias de ensino.

Os participantes do projeto lecionaram o curso Pré-Algoritmos, durante quatro horas semanais, totalizando sessenta horas, ao longo de um semestre. Durante o curso, são propostas aos alunos tarefas de programação, que podem ser respondidas em uma linguagem de programação, tais como *VisualG* [VisualG 2019] e C [Deitel e Deitel 2011], ou em uma linguagem de blocos.

Foram determinados horários de atendimento aos alunos beneficiados pelo projeto para poderem tirar dúvidas com um docente associado ao projeto. Além disso, 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 Cursos, buscando atender o máximo de alunos.

O projeto Pré-Algoritmos foi aplicado durante o ano letivo de 2018, totalizando dois semestres (2018.1 e 2018.2).

### 2.2 BOLSISTA

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

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

- Auxiliar na criação de um site *web* para divulgar o projeto e os materiais desenvolvidos pela equipe;
- Aplicar e resolver listas de exercícios com os alunos beneficiados pelo projeto;
- 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 e TONIN 2014]. O URI possui problemas em diferentes categorias, tais como problemas para iniciantes, matemáticos ou mais complexos envolvendo grafos.

Os exercícios 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 exercício, 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.

Como pode ser visto na Figura 1, o padrão de exercícios do URI possui o nível de dificuldade, uma descrição do o problema, o que é esperado como entrada e saída, além dos exemplos de entrada e saída, para que o estudante consiga responder o problema seguindo as regras propostas.

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 e TONIN 2014]

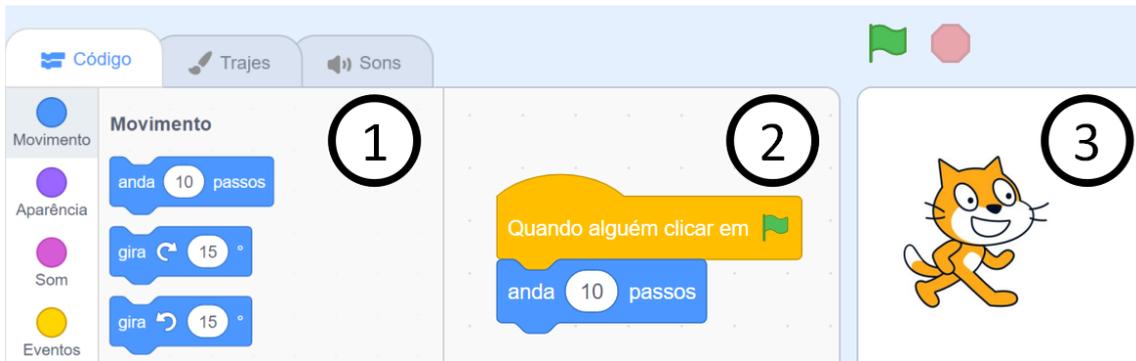
## 2.4 LINGUAGENS DE BLOCOS

Nas aulas iniciais de Pré-Algoritmos, 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*.

Figura 2. Linguagem de blocos Scratch



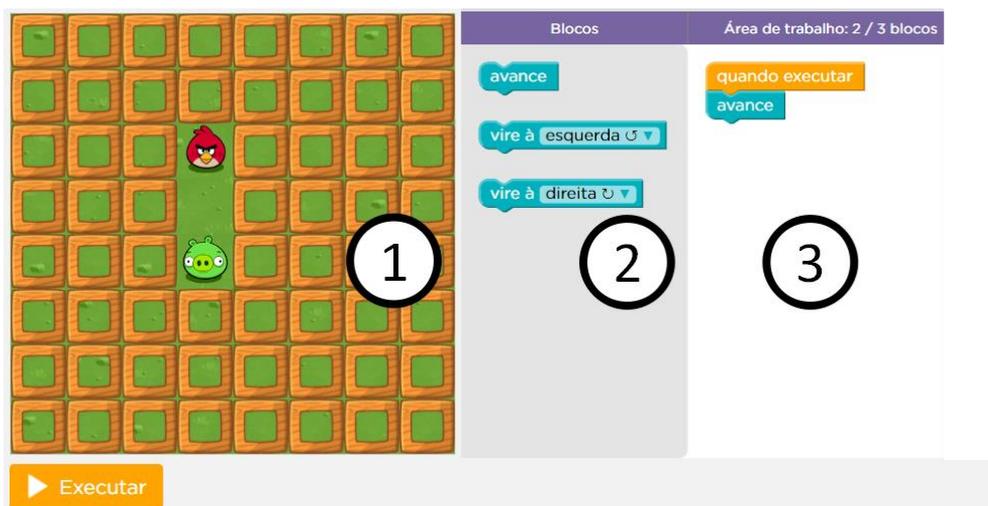
Fonte: Adaptada de Scratch [MIT 2019]

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).

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.

Figura 3. Problema no Code.org



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

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. (2018) investigaram os aspectos motivacionais no processo de ensino-aprendizado em programação em um estudo de caso com estudantes do curso de BTI da UFERSA, campus Pau dos Ferros. 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, falta de interesse por aulas teóricas e erros contínuos no código.

Por sua vez, Moreira et al. (2018) realizaram 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 está em desenvolver a lógica de programação. Além desta, os alunos apontaram outras dificuldades, como, por exemplo, o entendimento da sintaxe, a falta de tempo para se dedicar a disciplina e a interpretação dos problemas propostos.

No cenário internacional, 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 habilidade, esforço, dificuldade na tarefa e sorte.

### **4 RESULTADOS E DISCUSSÕES**

Para avaliar o projeto, aplicou-se Pré-Algoritmos em dois semestres seguidos, 2018.1 e 2018.2. Para identificar a eficiência do curso, comparou-se o desempenho dos alunos que foram aprovados em Pré-Algoritmos (e também matriculados em 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%. Dessa forma, os alunos como 50% ou mais de frequência foram considerados aprovados em Pré-Algoritmos.

**Tabela 2. Resultado do projeto Pré-Algoritmos**

<b>Descrição</b>	<b>Valores</b>
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 %
<b>Porcentagem de Aprovados em Pré-Algoritmos e Algoritmos 2018.1 e 2018.2</b>	76 %
<b>Porcentagem de Aprovados em Algoritmos 2018.1 e 2018.2</b>	50 %
<b>Porcentagem de Alunos Tradicionais Aprovados em Algoritmos 2018.1 e 2018.2</b>	36 %

**Fonte:** Autoria própria

Na Tabela 2, podem ser vistos os dados referentes aos dois semestres de execução do projeto para ações de apoio à melhoria do ensino de graduação. Em ambos os semestres, foram aprovados 42 alunos (22 no período 2018.1 e 20 no período 2018.2). Nos dois períodos participaram de Algoritmos em torno de 200 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.

Portanto, o projeto Pré-Algoritmos demonstrou eficiência na redução da taxa de insucessos na disciplina de Algoritmos e mostrou potencial para melhorar o desempenho dos alunos no ensino introdutório de programação, considerando o âmbito de estudo como sendo o curso de BTI da UFERSA, campus Pau dos Ferros. Não somente melhorar o desempenho, como também reduzir taxas de evasão e falta de motivação para os estudantes continuarem no referido curso de graduação do ensino superior.

#### 4.1 AMEAÇAS À VALIDADE E LIMITAÇÕES

O trabalho apresenta ameaças com respeito à 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 esta ameaça, realizou-se uma chamada pública, em que todos os alunos possuíam a mesma oportunidade de fazer parte do curso, o que pode incluir tantos alunos motivados para o aprendizado quando alunos que apresentam dificuldades.

Como ameaça à validade externa, o curso foi aplicado apenas na UFERSA e poderia não funcionar para outras instituições. Para mitigar esta 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, incluindo professores e bolsista.

### 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, campus Pau dos Ferros, vinha mantendo um alto índice de insucessos na disciplina de Algoritmos a cada semestre letivo, incluindo reprovações, evasões e desmotivações.

Para amenizar este problema, este trabalho propôs o projeto para ações de apoio à melhoria do ensino de graduação intitulado de “Pré-Algoritmos”. Este 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 disto, apenas 36% dos alunos tradicionais foram aprovados em Algoritmos.

Desta forma, conclui-se que este projeto auxiliou na redução do número de insucessos na disciplina Algoritmos e possui potencial para melhorar as taxas de insucessos na disciplina de Algoritmos.

Como trabalhos futuros, 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.

## REFERÊNCIAS

- [BEZ e TONIN 2014] BEZ, J. L. e TONIN, N. A. (2014). URI *online judge* e a internacionalização da universidade. Disponível em <https://scratch.mit.edu/>. Acessado em 27 de fevereiro de 2020.
- [Code.org 2019] Code.org (2019). Code.org. Disponível em <https://code.org/>. Acessado em 27 de fevereiro de 2020.
- [DEITEL e DEITEL 2011] DEITEL, Harvey, DEITEL, Paul. (2011). C - Como Programar. Pearson Universidades. 6 Ed.
- [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. Disponível em <https://scratch.mit.edu/>. Acessado em 27 de fevereiro de 2020.
- [Moreira et al. 2018] Moreira, G. L., Holandal, W., da S. Coutinho, J. C., and Chagas, F. S. (2018). Desafios na aprendizagem de programação introdutória em cursos de ti da UFERSA, campus Pau dos Ferros: um estudo exploratório. *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 programação na perspectiva de alunos iniciantes em programação da universidade federal rural do Semi-Árido 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. Disponível em <http://visualg3.com.br/>. Acessado em 27 de fevereiro de 2020.