

HELPBLOCK: uma ferramenta web baseada na biblioteca Blockly para apoio ao ensino de algoritmos

Maria Claudete Schorr Wildner¹

Evandro Franzen²

Eduardo Rodrigues Gomes³

RESUMO

Com o avanço da tecnologia novas profissões foram surgindo, principalmente na área da computação. Neste contexto, a necessidade de novos profissionais para ocuparem essas vagas vem crescendo consideravelmente. O problema está na elevada taxa de desistências e reprovações nas disciplinas iniciais dos cursos da área da computação, o que em parte se deve à falta de preparo dos estudantes na Educação Básica. Para Brackmann (2017) o raciocínio lógico e técnicas de programação geralmente não são assuntos trabalhados na Educação Básica, o que segundo Giraffa e Mora (2015) contribui com a evasão e reprovação dos estudantes nas disciplinas iniciais dos cursos. Pensando em diminuir o índice de desistências ou minimizar as dificuldades, foram desenvolvidas técnicas de programação visual possibilitando um maior exercício do raciocínio lógico, deixando a parte de códigos e sintaxe para um segundo momento. Este artigo tem como objetivo apresentar uma ferramenta web denominada HelpBlock que utiliza a biblioteca Blockly possibilitando aos estudantes a resolução de exercícios e criação de algoritmos através de blocos lógicos conectáveis. A mesma foi testada em uma turma de algoritmos composta por 23 estudantes, onde foi possível constatar um aumento no interesse destes em relação aos exercícios propostos, assim como uma boa aceitação na forma alternativa de aprender programação. Com os resultados da intervenção foi possível perceber uma melhora significativa na aprendizagem de algoritmos e programação.

Palavras-chave: Blockly. Ensino de algoritmos. Ferramenta web.

1. Introdução

¹ Doutorando em Informática na Educação – PPGIE–UFGRS

² Doutorando em Informática na Educação – PPGIE –UFGRS

³ Graduado em Sistemas de Informação – Univates

Os cursos de computação têm como conteúdo inicial o ensino de algoritmos que é realizado na disciplina de Algoritmos e Programação. Porém seu conteúdo em alguns momentos tende a intimidar os estudantes, causando evasão ou reprovação. Segundo as Diretrizes Nacionais Curriculares (DCN) e os referenciais de formação (SBC, 2017) para cursos da área de computação (MEC, 2016), todos os cursos devem contemplar uma base sólida de conteúdos relacionados ao raciocínio lógico e à programação de computadores.

Estas disciplinas estão entre as que apresentam as maiores taxas de evasão. De acordo com Watson e Li (2014), a média de reprovação é superior a 30%, ou seja, aproximadamente um terço dos estudantes reprovam. A pesquisa foi realizada em 15 países e 51 instituições diferentes e foram analisados também os fatores que influenciam nestas taxas. O Brasil aparece como terceiro país com maior índice de reprovação, com percentual superior a 50% de discentes não aprovados (Ramos et al., 2015).

Conforme Giraffa e Mora (2015), a evasão nas disciplinas dos níveis iniciais dos cursos da área da computação não é um fato recente. Esta situação já vem causando preocupação a mais tempo nas IES (Instituições de Ensino Superior), bem como na sociedade em geral. Iepsen (2013) enfatiza que diante da percepção das dificuldades existentes, as formas de ensinar algoritmos se tornaram alvo de estudos que tem como objetivo diminuir as adversidades dos estudantes.

Assim como Iepsen (2013) e Giraffa e Mora(2015) demonstraram preocupação com o ensino de algoritmos, outros autores também tiveram, desenvolvendo ferramentas como Scratch, Portugol, Visualg, Blockly tendo como objetivo auxiliar na compreensão e aprendizagem de algoritmos. Com base nisso e nos altos índices de evasão e reprovação, foi desenvolvida uma ferramenta web utilizando a biblioteca Blockly, possibilitando que os estudantes resolvessem os algoritmos utilizando uma plataforma de programação visual. A proposta foi dividida em 5 etapas, sendo elas o desenvolvimento da ferramenta, o pré-teste, a utilização da ferramenta pelos estudantes, o pós-teste e o questionário de satisfação. A intervenção foi realizada com uma turma de 23 estudantes de algoritmos e programação de uma IES do Rio Grande do Sul.

2. Embasamento Teórico

A disciplina de Algoritmos é a base dos cursos de graduação na área de computação, e é considerada imprescindível e obrigatória. Para PEREIRA (2010), algoritmo nada mais é do que uma receita que mostra passo a passo os procedimentos necessários para a resolução de um problema. Assim como os estudantes passam por dificuldades para resolver problemas, os professores acabam tendo dificuldades de ensinar o conteúdo. Para (NOBRE, 2002), algumas dificuldades pelas quais os professores do ensino de programação passam são: reconhecer as habilidades inatas de seus estudantes; propor técnicas de soluções de problemas; aproveitar a capacidade que o estudante tem em abstrair, tanto na escolha de estruturas de dados a serem utilizadas quanto na busca de possíveis soluções; despertar o sentimento de cooperação e colaboração entre os estudantes.

A dificuldade de compreensão e aplicação da lógica de programação pela qual os acadêmicos passam é uma das maiores causas de reprovação e desistência no decorrer do curso, isto se deve pelo fato da maioria ser egresso do ensino médio regular, onde não tiveram base técnica de lógica e algoritmos (JUNIOR, 2015). Quando essa dificuldade não é solucionada a tempo, acaba resultando em prejuízos enormes aos discentes e docentes da instituição, tornando difícil e lenta a formação acadêmica do estudante e, muitas vezes, incompatível com a dificuldade das atividades práticas que lhe serão exigidas durante o curso.

Noschang et. al (2014) cita que o ensino da programação inicial dos cursos deve focar no desenvolvimento da lógica de programação dos estudantes e a capacidade de resolver problemas, deixando a aprendizagem de uma linguagem de programação para um segundo momento. Também argumenta que as IDEs (Integral Development Environment) profissionais disponíveis (Netbeans, Eclipse, entre outras) estão longe de ser a melhor opção, pois na sua maioria apresentam suas opções e mensagens no idioma inglês, o que acaba sendo um obstáculo para a aprendizagem dos estudantes.

Uma das maneiras encontradas para melhorar o desempenho dos discentes nas disciplinas de programação foi o desenvolvimento de ferramentas para auxiliar na programação. Essas ferramentas têm como objetivo apresentar ambientes de boa usabilidade

aos estudantes, proporcionando uma melhor experiência de cada um deles com a programação. Com estas características podemos citar: Scratch e Blockly.

O Scratch Ambiente foi produzido pelo Lifelong Kindergarten Group do Massachusetts Institute of Technology/MIT Media Lab, está disponível para download desde Maio de 2007 no site www.scratch.mit.edu. Segundo Alencar et. al (2014), o Scratch é uma linguagem gráfica de programação desenvolvida com o objetivo de ser de fácil manuseio e bastante intuitiva. Por possuir uma IDE, pode ser considerada uma linguagem de programação visual. Permite trabalhar com conceitos como: criação de interfaces, números randômicos, sequência, iteração, condições variáveis, sincronia, execução paralela, interação em tempo real, lógica booleana, tratamento de evento entre outros (VENTORINI, 2014).

No Scratch (Figura 1) é possível trabalhar com imagens, sons, fotos, criar desenhos, alterar aparências, dar movimento aos objetos, fazer com que haja interação entre as coisas. Tudo é feito através da conexão entre blocos, possibilitando inclusive que crianças tenham condições de usar o software para desenvolver pequenos jogos e animações.

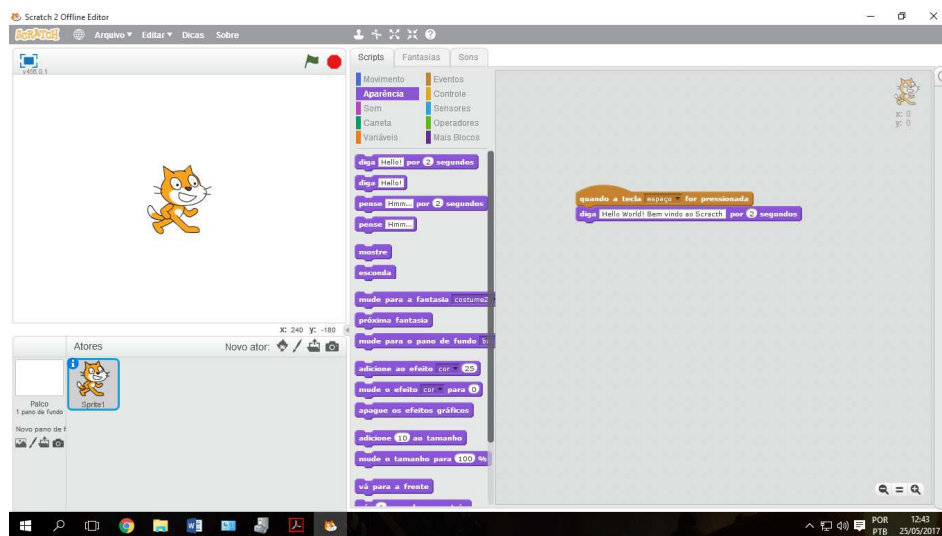


Figura 1: Tela de principal do Scratch
Fonte: Dos autores, 2018.

Outro ambiente de programação visual é o Blockly (Figura 2), este funciona somente na web e tem código aberto, possibilitando o uso em ferramentas próprias de cada desenvolvedor. A biblioteca Blockly serve para adicionar um editor de código visual em aplicações web e Android. O editor Blockly utiliza blocos gráficos conectáveis para

representar os conceitos de código como repetição, condição, *loops* entre outros. Permitindo que os usuários usem princípios de programação sem se preocupar com erros de sintaxe (Blockly API, 2017).

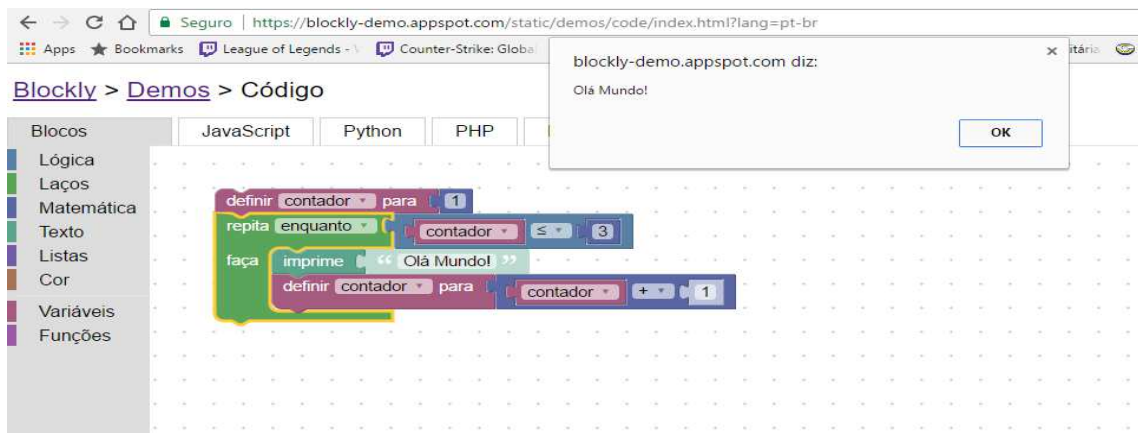


Figura 2: Tela de montagem dos blocos
Fonte: Dos autores, 2018.

3. Metodologia

A pesquisa foi realizada com base na aprendizagem significativa, que segundo Ausubel (1963, p. 58) é o aprendizado de uma nova informação de forma substantiva e não arbitrária. As características básicas da aprendizagem significativa são a não-arbitrariedade e a substantividade.

Não-arbitrariedade significa que um material significativo se relacionado de maneira não arbitrária com um conhecimento já existente no indivíduo. O conhecimento prévio é importante para que haja a compreensão e fixação de novas informações que de certa forma “se ancoram” em conhecimentos relevantes já existentes na estrutura cognitiva da pessoa, Ausubel (1963). Ainda segundo o autor, a substantividade significa que o que o indivíduo aprende é a substância do novo conhecimento, das novas ideias, e não as palavras precisas usadas para expressá-las.

Considerando os conceitos da aprendizagem significativa, a realização do levantamento de dados foi feita através da aplicação de um pré-teste contendo 2 exercícios de programação, seguido de uma intervenção pedagógica onde foi utilizada a ferramenta

HelpBlock, desenvolvida nesta pesquisa. Após isso foi aplicado um pós-teste, que assim como o pré-teste consistiu em 2 exercícios de programação. O pré-teste teve como objetivo medir o conhecimento prévio dos estudantes, e o pós-teste serviu para medir se a ferramenta HelpBlock auxiliou na resolução dos exercícios e conseqüentemente na aprendizagem de algoritmos. Por fim foi solicitado que os estudantes respondessem um questionário de satisfação. A intervenção foi realizada em uma turma de Algoritmos e Programação, contando com 23 estudantes.

Na próxima seção será apresentada a ferramenta, sua utilização e os resultados obtidos com a utilização da mesma.

4. Análise e Discussão dos Dados

A ferramenta HelpBlock foi desenvolvida com o intuito de ser um ambiente intuitivo e de fácil compreensão por parte dos usuários. Esta contém um sistema de cadastros de exercícios classificados por tipo e nível de dificuldade, além disso, disponibiliza um local de programação visual, onde o estudante pode criar seu programa utilizando blocos conectáveis. Estes blocos foram desenvolvidos com base na biblioteca Blockly. Para testar a ferramenta foram disponibilizados na ferramenta exercícios que normalmente são utilizados nas aulas de Algoritmos e Programação, estes foram recomendados por professores da disciplina. O estudante pode escolher na lista o exercício que deseja resolver, em seguida é encaminhado para a página de resolução de exercícios, que é mostrada nas figuras 3 e 4. A figura 3 apresenta a parte superior da página, que contém ao lado direito a área de desenvolvimento, onde os estudantes terão de arrastar os blocos para criar os algoritmos e no lado esquerdo os blocos de programação divididos nos seguintes grupos:

- **Logic** – Contém os blocos lógicos de condição como por exemplo o if else, número maior ou menor, verdadeiro ou falso.
- **Loops** – Contém os blocos de repetição como por exemplo o while e o for.
- **Math** – Contém os blocos de matemática, que são blocos que trazem o cálculo da raiz quadrada por exemplo ou até blocos que verificam que determinado número é par ou ímpar, restos da divisão entre outros.

- **Text** – Contém os blocos de texto como por exemplo o imprime, converte para maiúscula, bloco criador de texto, e também um bloco de entrada que solicita ao usuário alguma informação.
- **Variables** – Neste grupo é possível criar as variáveis, assim como utilizar blocos específicos para definir valores a cada variável criada.



Figura 3: Tela de resolução de exercícios, parte superior

Fonte: Dos autores, 2018.

Já a figura 4 traz a parte inferior da tela de resolução, onde existe um local de apresentação do código já com um algoritmo em formato de código, assim como os botões mostrar código, executar, limpar, salvar, carregar resposta e entregar, que realizam as seguintes funções:

- **Mostrar código** – Converte em javascript os blocos que estão na área de desenvolvimento e apresenta na área de apresentação de código.
- **Limpar** – Limpa a área de apresentação do código.
- **Executar** – Pega os blocos que estão na área de desenvolvimento e executa o algoritmo formado.
- **Salvar** – Pega todos os blocos que estão na área de desenvolvimento, converte em texto e salva no banco, porém é importante salientar que cada estudante pode salvar apenas uma resposta por exercício, caso exista uma resposta já salva de determinado usuário para determinado exercício a próxima vez que o usuário clicar em salvar essas respostas irão se sobrepor mantendo apenas o último registro salvo.
- **Carregar resposta** – Busca o que está salvo no banco e monta na área de desenvolvimento.

- **Entregar** – Envia para o professor os blocos que estão na área de desenvolvimento, juntamente com um comentário falando sobre como foi a resolução do exercício e se teve alguma dificuldade durante. Cada estudante pode entregar apenas uma resposta por exercício.



Figura 4: Tela de resolução de exercícios, parte inferior
Fonte: autoria própria

Os testes da ferramenta foram realizados em três momentos, sendo eles o pré-teste, a intervenção pedagógica e o pós-teste. Primeiramente foi apresentada a proposta da pesquisa e feito alguns comentários sobre os motivos de ter sido escolhida a programação visual para tentar melhorar o ensino nas cadeiras iniciais dos cursos de informática.

Em seguida foi solicitado que os acadêmicos resolvessem duas atividades utilizando a ferramenta com a qual estavam trabalhando na disciplina, neste caso o Dr. Java. Estas atividades foram utilizadas como pré-teste para identificar os conhecimentos prévios dos estudantes. O primeiro exercício proposto a grande maioria conseguiu realizar, apenas um ou dois estudantes tiveram algum erro de sintaxe. No exercício dois houve algumas confusões na hora de interpretar o enunciado, provavelmente por envolver porcentagem, assim alguns estudantes calcularam cada uma das porcentagens separadas sobre o valor inicial e outros calcularam primeiro uma das porcentagens sobre o valor inicial e a segunda porcentagem calcularam sobre o valor gerado pelo primeiro cálculo. Mas a maioria foi bem e conseguiu resolver o problema, contudo ainda apareceram alguns erros clássicos de sintaxe.

Revista Tecnologias na Educação – Ano 10 – Número/Vol.25 –Julho 2018

tecnologiasnaeducacao.pro - tecedu.pro.br

Com base nestes resultados foi possível iniciar a segunda etapa da proposta, onde foi apresentada a ferramenta HelpBlock. Algumas explicações sobre o funcionamento da ferramenta foram realizadas. Os estudantes puderam interagir com a ferramenta, criar exemplos, tirar dúvidas. Após terminar as explicações e demonstrações seguiu-se para a terceira parte, onde foi aplicado o pós-teste solicitando que resolvessem mais dois exercícios semelhantes aos dois exercícios do pré-teste.

Durante a resolução dos exercícios utilizando o HelpBlock, alguns estudantes pediram ajuda, pois tinham dúvidas de qual bloco deveriam utilizar para realizar determinadas tarefas, mas assim que mostrado e explicado uma vez, o estudante já conseguia dar sequência na programação. Outro ponto a destacar é a parte de tirar dúvidas dos estudantes que ficou mais fácil, pois é possível mostrar onde cada parte do algoritmo deve se encaixar para funcionar e o motivo de ter que ser daquela forma.

Após finalizar o pré-teste, intervenção com a utilização do HelpBlock e pós-teste, pediu-se para que os estudantes respondessem a um questionário de satisfação que tinha como objetivo obter a opinião dos estudantes em relação a prática de programação visual e também sobre a experiência que tiveram com o HelpBlock. Analisando as respostas obtidas no questionário foi possível perceber que a maioria dos estudantes percebeu uma melhora na resolução e compreensão dos algoritmos com a utilização da ferramenta. Isso é perceptível nas respostas (Quadro 1) referente a primeira pergunta do questionário, onde foi questionado se a ferramenta com programação visual auxiliou na resolução dos exercícios.

Sim, porque é possível entender melhor o que está acontecendo.

Sim, pois não precisa digitar e assim consegue manter o pensamento.

Facilitou demais, pois simplificou o entendimento.

Sim, pois trata apenas lógica, deixando as formalidades por conta do próprio programa.

Quadro 1: Resposta dos estudantes referente a pergunta número 1 do questionário de satisfação
Fonte: Dos autores, 2018.

Na questão de número 4 (Gráfico 1), foi solicitado ao estudante deixar uma nota para a

ferramenta utilizada. Dois estudantes deram nota 7, cinco deram nota 8, dez deram nota 9 e três deram nota 10. Com isso foi possível concluir que a ferramenta foi bem aceita e que despertou interesse na maioria.

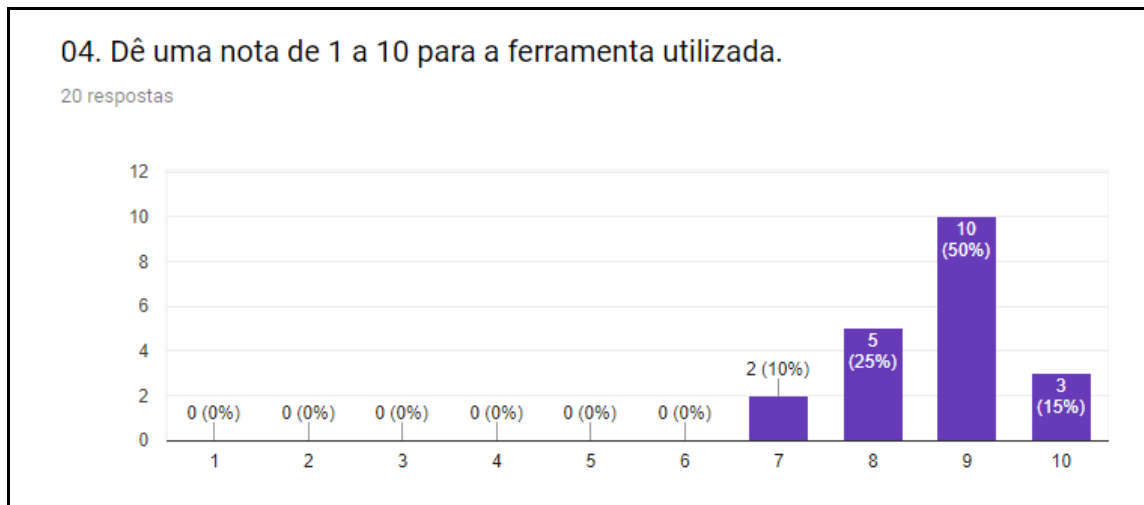


Gráfico 1: Pergunta número 4 do questionário de satisfação
Fonte: Dos autores, 2018.

5. Conclusões e/ou Propostas

Ao finalizar a ferramenta HelpBlock e a intervenção com estudantes de algoritmos e programação percebeu-se que a ferramenta, desenvolvida por disponibilizar recursos bastante satisfatórios aos estudantes, pode ser utilizada na aprendizagem de algoritmos. A quantidade de blocos com variadas funções foi importante para criar a solução algorítmica, bem como, a facilidade de interação. A ferramenta além de auxiliar na aprendizagem do estudante, facilitou na correção dos exercícios pelo professor, pois o estudante por meio da ferramenta envia a resposta para o professor. Outra facilidade é a conversão de blocos em código, permitindo que o estudante compare sua solução em blocos com a solução em linhas de código.

Ao concluir este trabalho confirma-se a hipótese de que oferecer novas maneiras de ensino aos estudantes desperta o interesse e a curiosidade destes, fazendo com que eles busquem o conhecimento. Para que haja aprendizagem não basta apenas que os professores se empenham em trazer conteúdo, mas sim, que exista um comprometimento dos estudantes em aprender, isso ocorre quando há interesse e motivação por parte do estudante. Com a utilização da ferramenta HelpBlock foi possível perceber os estudantes motivados,

interessados em resolver as atividades propostas, logo, o objetivo do desenvolvimento e utilização da ferramenta foi alcançado.

Para trabalhos futuros, pretende-se integrar esta ferramenta em um sistema maior, disponibilizando esta maneira alternativa de programar com blocos para um número maior de estudantes.

6. Referências Bibliográficas

ALENCAR, Gersica A.; FREITAS Ana K; PESSOA, Maércio dos S.; MARTINS Danielle J. S. **Utilizando o SCRATCH nas aulas de Lógica de Programação do Proeja: Um relato de experiência.** TISE 2014. Disponível em: <http://www.tise.cl/volumen10/TISE2014/tise2014_submission_110.pdf> Acesso em: 25 mai. 2017

AUSUBEL, D.P. (1963). *The psychology of meaningful verbal learning*. New York, Grune and Stratton.

Blockly API. Disponível em: <<https://developers.google.com/blockly/guides/overview>> Acesso em: 25 mai 2017.

BRACKMANN, Christian Puhlmann. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica.** Tese de Doutorado, PPGIE, 2017.

GIRAFFA, Lucia Maria Martins; MORA, Michael da Costa. **Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno.** III Conferencia Latinoamericana sobre el Abandono en la Educación Superior (III CLABES), 2015.

JUNIOR, José A.T. de Lima **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.** 27 abr. 2015. Disponível em: <<http://web.unifoa.edu.br/cadernos/edicao/27/5-15.pdf>> Acesso em: 20 abr. 2018.

NOSCHANG, Luiz F. ; PELZ, Fillipi; JESUS, Elieser A. ; RAABE, André L. A. **Portugol Studio: Uma IDE para Iniciantes em Programação.** XXXIV Congresso da Sociedade Brasileira de Computação - CSBC 2014. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2014/001.pdf>> Acesso em 24 mai 2017.

PEREIRA, Ana Paula. **O que é algoritmo?.** 12 mai. 2010 Disponível em<<https://www.tecmundo.com.br/programacao/2082-o-que-e-algoritmo-.htm>> Acesso em:19 abr 2018.

Revista Tecnologias na Educação – Ano 10 – Número/Vol.25 –Julho 2018

tecnologiasnaeducacao.pro - tecedu.pro.br

Scratch. Disponível em: <https://scratch.mit.edu/> Acesso em 23 mai 2017.

VENTORINI, André Eduardo; FIOREZE, Leandra Anversa. **O Software Scratch: Uma contribuição para o ensino e a aprendizagem da matemática.** 08 ago. 2014. Disponível em: <http://w3.ufsm.br/ceem/eiemat/Anais/arquivos/ed_4/MC/MC_Venturine_Andre.pdf> Acesso em 25 maio 2017.

Recebido em abril 2018

Aprovado em junho 2018