

APLICAÇÃO DO SCRATCH NO ENSINO DE PROGRAMAÇÃO PARA INICIANTES

Natália Rodrigues de Melo*, Rayane Aparecida Guimarães**,
Niltom Vieira Júnior***

RESUMO

O aprendizado de programação permite que os alunos, além de criarem programas de computadores, desenvolvam habilidades que são importantes para qualquer área. Muitos docentes ainda enfrentam dificuldades no processo de ensino-aprendizagem de disciplinas que, assim como a programação, são práticas e exigem grande dedicação dos alunos. As ferramentas computacionais podem auxiliar neste processo, tornando as aulas mais dinâmicas e despertando o interesse dos alunos. Neste contexto, destaca-se o *Scratch*, uma plataforma desenvolvida para ensinar lógica de programação para crianças e adolescentes. Este trabalho tem como objetivo a criação de roteiros práticos que introduzem conceitos de programação que podem ser abordados por meio do *Scratch*. Nos roteiros é apresentada a teoria relacionada aos conteúdos, assim como a aplicação destes conceitos por meio da resolução de problemas no *Scratch*. Tais roteiros foram avaliados por docentes da área da computação. Estes os consideraram efetivos para o ensino de programação.

Palavras-chave: programação; *Scratch*; roteiros práticos.

TEACHING PROGRAMMING FOR BEGINNERS USING SCRATCH

ABSTRACT

Learning computer programming allows students to develop skills that are important in all areas besides the creation of computer programs. Many teachers still face challenges in the teaching-learning process of classes such as computer pro-

* Mestra em Eletromagnetismo Aplicado pela Universidade Federal São João del-Rei (UFSJ). Atualmente é Analista de Produto na empresa ACT Consultoria em Tecnologia Ltda. ORCID: 0000-0002-3091-9339. Correio eletrônico: natalia.rdsmelo@gmail.com

** Mestra em Engenharia de Sistemas e Automação pela Universidade Federal de Lavras (UFLA). Atualmente é Professora Visitante no Instituto Federal de Minas Gerais (IFMG), Campus Formiga. ORCID: 0000-0001-6585-5390. Correio eletrônico: rayane.ap@outlook.com

*** Doutor em Engenharia Elétrica pela Universidade Estadual Paulista (UNESP). Professor no Instituto Federal de Minas Gerais (IFMG), Campus Arcos. ORCID: 0000-0003-1077-8302. Correio eletrônico: niltom.vieira@ifmg.edu.br

programming that are practical and require much of students' dedication. Computational tools may be helpful in this process since they make classes more dynamic and arouse students' interest. In this context is Scratch, a platform developed to teach programming logic for children and teenagers. The objective of this study is to create scripts to introduce programming concepts that may be demonstrated by using Scratch. The scripts present programming concepts and problems resolution as well using Scratch. The scripts' evaluations were taken from computer programming teachers and they considered them to be effective for teaching programming.

Keywords: *programming; Scratch; scripts.*

APLICACIÓN SCRATCH EN LA ENSEÑANZA DE PROGRAMACIÓN PARA PRINCIPIANTES

RESUMEN

Aprender a programar permite a los estudiantes, además de crear programas de computadora, desarrollar habilidades que son importantes para cualquier área. Muchos docentes aún enfrentan dificultades en el proceso de enseñanza-aprendizaje de asignaturas que, como la programación, son prácticas y requieren una gran dedicación por parte de los estudiantes. Las herramientas computacionales pueden ayudar en este proceso, dinamizando las clases y despertando el interés de los estudiantes. En este contexto, se destaca Scratch, una plataforma desarrollada para enseñar lógica de programación a niños y adolescentes. Este trabajo tiene como objetivo crear guiones prácticos que introduzcan conceptos de programación que se puedan abordar a través de Scratch. En los guiones se presenta la teoría relacionada con los contenidos, así como la aplicación de estos conceptos a través de la resolución de problemas en Scratch. Dichos guiones fueron evaluados por profesores del área de computación. Estos los consideraron efectivos para la enseñanza de la programación.

Palabras clave: *programación; Scratch; guiones prácticos.*

1 INTRODUÇÃO

A programação é uma disciplina da área da Computação que capacita os estudantes para a criação de programas de *software*. Segundo Scaico *et al.* (2013), é de extrema importância aprender a programar, pois os conceitos relacionados a algoritmos são úteis para todas as áreas, não se restringindo à Computação. Por meio da programação, o aluno consegue desenvolver diversas capacidades, dentre elas o raciocínio lógico e o pensamento computacional, os quais são essenciais para resolução de problemas.

O ensino de programação é considerado um processo complexo e que requer várias habilidades. Pesquisas apontam que, nos últimos anos, a disciplina de pro-

gramação tem apresentado altos índices de reprovação e evasão. Alguns dos motivos que fundamentam tais indicadores negativos são a desmotivação dos estudantes e a aplicação constante do modelo de “ensino tradicional” (BORGES, 2000; RODRIGUES, 2004).

Dentre as soluções levantadas para estes problemas, destaca-se a importância da utilização de ferramentas computacionais que auxiliem o aluno no processo de ensino-aprendizagem. Tal metodologia favorece a ministração de aulas mais dinâmicas, em que a teoria é demonstrada com maior clareza, além de permitir que os alunos tenham mais autonomia para aprender nos momentos em que não contam com a supervisão do professor (SILVA; SILVA; SANTOS, 2009).

Neste contexto, destaca-se a importância do *Scratch*, uma ferramenta que possibilita a criação de histórias interativas ou jogos por meio da combinação de recursos visuais e sonoros, como imagens, gráficos e músicas. Diversas pesquisas mostram que o *Scratch* favorece o ensino de programação para iniciantes (RESNICK *et al.*, 2009). Por intermédio do *Scratch*, os alunos são capazes de criar programas de *software*, pensando de forma criativa e sistêmica na resolução de problemas.

Este artigo tem como objetivo utilizar uma metodologia de ensino alternativa para introdução dos conceitos de programação utilizando o *Scratch*. O ensino é promovido por intermédio de roteiros práticos e é voltado para alunos de cursos técnicos relacionados às Ciências Exatas. Observou-se que muitos destes cursos possuem, na sua grade curricular, disciplinas que exigem o conhecimento dos conceitos de programação, mas não incluem esta disciplina na sua grade curricular. Portanto, pretende-se facilitar o aprendizado destes alunos mediante conteúdos que já aplicam conceitos de programação, assim como incentivá-los a continuar nesta área.

2 PENSAMENTO COMPUTACIONAL E LÓGICA DE PROGRAMAÇÃO

Wing (2006) define o termo pensamento computacional (*Computational Thinking*) como a capacidade de resolver problemas, projetar sistemas e compreender o comportamento humano, utilizando os conceitos fundamentais da Ciência da Computação. O desenvolvimento do pensamento computacional não só permite que os estudantes entendam como o mundo funciona, mas também fornece habilidades consideradas essenciais para a resolução de problemas complexos (JOHNSON *et al.*, 2014).

O conhecimento em programação serve de suporte às atividades cognitivas associadas ao pensamento computacional (GROVER; PEA, 2013). Além disso, contribui para o desenvolvimento de habilidades dos alunos, como raciocínio lógico e resolução de problemas utilizando algoritmos (KAFAI; BURKE, 2014).

Muitos estudantes consideram a disciplina de programação complexa e apresentam dificuldade para compreender os conceitos associados à lógica computacional (DENNER; WERNER; ORTIZ, 2012). Essas limitações fazem com que a motivação e a participação dos alunos no curso sejam reduzidas. Forte e Guzdial (2014) afirmam que o fracasso da disciplina de programação deve-se, principalmente, à falta de interesse dos alunos nas aulas. Tais problemas identificados no processo de ensino-aprendizagem levantam questionamentos sobre a eficácia das ferramentas de programação utilizadas.

Margulieux, Morrison e Decker (2012) destacam que as dificuldades associadas ao ensino de programação podem ser superadas se essa experiência for transformada em algo mais agradável e lúdico para os estudantes. Neste contexto, a utilização de diferentes ferramentas no ensino de programação motiva os iniciantes a aprenderem programação e contribuem para que estes se sintam interessados em continuar seus estudos na área (RESNICK *et al.*, 2009).

3 METODOLOGIAS DE ENSINO

A didática tradicional centraliza no professor todo o processo de ensino-aprendizagem, que, muitas vezes, leva o aluno a enxergar o professor como uma autoridade - nele está todo o conhecimento e o poder. A metodologia tradicional de ensino consiste em o professor ser responsável por transmitir os ensinamentos e avaliar os alunos, enquanto cabe ao aluno apenas ouvir, decorar e obedecer (RODRIGUES; MOURA; TESTA, 2011). A abordagem tradicional de ensinar não leva em consideração as opiniões e a diversidade de métodos de aprendizado que os alunos podem ter. Essa situação pode trazer frustrações durante a aprendizagem, bloqueando o interesse do aluno em relação ao conteúdo apresentado.

Dewey se interessou pela pedagogia a partir da observação da escola de seu tempo, focada em valores tradicionais, a qual não teria caminhado junto com o desenvolvimento do tempo, não acompanhando os avanços políticos e sociais. Assim, percebeu que as ideias somente têm importância se forem aplicadas em problemas reais. Para Dewey, os alunos precisam pensar por si mesmos. Para isso, é necessário que os ensinamentos partam de tarefas associadas à vida real, que os conteúdos ensinados sejam vinculados à situação atual e real do aluno. Aborda também a importância de atividades manuais e criativas (MARTINS, 2012).

Com o desenvolvimento avançado da tecnologia, novos conceitos precisam ser abordados e ensinados aos alunos. As Tecnologias de Informação e Comunicação (TIC) estão causando mudanças na rotina escolar, visto que a abordagem das tecnologias atuais pode contribuir para o desenvolvimento de alunos, além de várias habilidades, como o raciocínio lógico e a resolução de problemas (RODRIGUEZ *et al.*, 2015). Tais habilidades são necessárias e precisam ser ensinadas, não somente para serem aplicadas nas áreas de tecnologia, mas também em várias áreas de conhecimento.

O ensino de programação, e por consequência o “pensamento computacional”, introduz inicialmente a estrutura de raciocínio lógico, o desenvolvimento em etapas, normalmente abordado em fluxogramas. Após essa estruturação de problemas e fatos, o professor parte para a programação em si. A metodologia de ensino tradicional de programação consiste em o professor apresentar o conteúdo teórico e, posteriormente, indicar alguns exemplos de aplicação, já propondo alguns exercícios. Leal e Ferreira (2013) afirmam que esse método não consegue motivar os alunos, principalmente aqueles que estão conhecendo a linguagem de programação pela primeira vez, seja no ensino médio, seja no ensino superior. Os alunos, ao se depararem com esse conteúdo novo, não conseguem assimilar sua importância àquela disciplina e a todas as áreas da vida. Além disso, esses alunos enfrentam dificuldades referentes à aprendizagem na qual eles estão envolvidos, levando-os à falta de habilidades para solucionar problemas, à falta de abstração, etc.

Vários autores, como Rodriguez *et al.* (2015), Wangenheim, Nunes e Santos (2014), sugerem que inicialmente a linguagem de programação seja abordada em ambientes gráficos, onde o aluno possa desenvolver e entender o objetivo da linguagem de programação de forma intuitiva e prazerosa. Orienta-se utilizar a programação em blocos, tentando sempre aplicar esse conhecimento em situações cotidianas para estimular nos alunos o desenvolvimento de habilidades cognitivas. Entre as linguagens de programação em blocos mais utilizadas, destaca-se o *Scratch*.

Nos últimos anos, a utilização do *Scratch* tem apresentado contribuições significativas no ensino de programação para crianças e jovens iniciantes. Essa ferramenta utiliza-se de uma metodologia de ensino simples que facilita a compreensão e a aplicação de conceitos de programação, sem precisar lidar com a complexidade de uma linguagem mais avançada. Na seção a seguir, são descritas mais informações sobre o *Scratch*.

4 SCRATCH

O *Scratch* é uma linguagem de programação desenvolvida em um projeto conduzido pelo Instituto de Tecnologia de Massachusetts (MIT), que visa introduzir conceitos de programação para estudantes sem conhecimento prévio na área. O objetivo é incentivar o aprendizado de programação por meio de uma experiência divertida, utilizando recursos visuais e intuitivos (SÁEZ-LÓPEZ; ROMÁN-GONZÁLEZ; VÁZQUEZ-CANO, 2016).

A criação do *Scratch* é uma alternativa para facilitar o desenvolvimento de habilidades fundamentais para que o jovem consiga acompanhar a evolução tecnológica no mundo (MARQUES, 2009). Seus criadores afirmam que essa ferramenta possui potencial para promover o desenvolvimento de competências essenciais à vida no século XXI, como criatividade, raciocínio sistemático e trabalho colaborativo (INSTITUTO DE TECNOLOGIA DE MASSACHUSETTS, 2020a).

Diferentemente das linguagens de programação tradicionais, cuja implementação do código é feita por meio de comandos de texto, o *Scratch* possui blocos que representam tais comandos (GOMES *et al.*, 2014). Os blocos são separados em grupos de acordo com a sua finalidade. Alguns desses grupos representam recursos de programação muito utilizados, como sequência, repetição e estruturas condicionais. Para criar uma rotina utilizando o *Scratch*, basta arrastar e posicionar os blocos de acordo com a ordem de execução. Dessa forma, o aluno consegue criar os programas sem precisar memorizar linguagens e códigos de programação (RODRIGUEZ *et al.*, 2015).

Além de introduzir noções básicas sobre computadores e programação, o *Scratch* também contribui para que o aluno desenvolva competências importantes para resolução de problemas. Essa ferramenta também trabalha conceitos específicos de programação, como estruturas sequenciais e de repetição, lógica booleana, *arrays*, dentre outros (INSTITUTO DE TECNOLOGIA DE MASSACHUSETTS, 2020b). Dessa forma, fica evidente como a aplicação do *Scratch* possui inúmeros benefícios no processo de ensino-aprendizagem de programação, além de contribuir para que o aluno desenvolva capacidades que farão com que ele consiga se destacar em outras disciplinas.

5 METODOLOGIA

Pesquisas recentes comprovam que o emprego de ferramentas computacionais para implementação de animações de algoritmos facilita o aprendizado do aluno e torna o ensino mais colaborativo (SANTOS *et al.*, 2008).

Este trabalho aplica procedimentos metodológicos relacionados à utilização da tecnologia no ensino de programação. O objetivo é desenvolver roteiros práticos como estratégia para o ensino de conceitos de programação para alunos iniciantes, principalmente de cursos técnicos relacionados às Ciências Exatas. Utilizando como referência a grade curricular do curso técnico de nível médio em eletrotécnica do Instituto Federal de Minas Gerais (IFMG), Campus Formiga, observou-se que o este não inclui disciplinas específicas de programação.

Portanto, pretende-se oferecer alternativas para que os alunos adquirissem conhecimentos de computação que pudessem facilitar o aprendizado em disciplinas mais avançadas, assim como motivá-los a seguir uma carreira na área das Exatas. O *Scratch* foi escolhido para desenvolvimento dos roteiros e como proposta para resolução destes. A escolha da plataforma foi fundamentada nas seguintes questões:

- a) quais conceitos de programação podem ser aprendidos pelos alunos por meio do *Scratch*?
- b) o *Scratch* é capaz de atingir os objetivos de aprendizagem definidos?
- c) o *Scratch* facilita o ensino/aprendizagem de programação?
- d) o *Scratch* motiva os alunos a aprenderem?
- e) o *Scratch* é capaz de prender a atenção dos alunos e proporcionar uma experiência divertida?

Com base nestas questões, foram elaborados seis roteiros que tiveram uma sequência lógica definida de forma a facilitar e estimular o aprendizado do aluno, conforme mostrado no Quadro 1.

Quadro 1 - Temas dos roteiros

Roteiros	Temas
1	Pensamentos computacionais
2	Definição de variáveis
3	Estrutura sequencial e operadores
4	Condicionais (IF)
5	Condicionais (IF/ELSE IF)
6	Estruturas de repetição

Fonte: elaborado pelos autores.

Os roteiros foram submetidos à avaliação de professores do IFMG, Campus Formiga, sendo que todos ministram ou já ministraram disciplinas relacionadas à programação. O objetivo era obter um retorno baseado na experiência dos docentes e verificar se os roteiros poderiam contribuir para o ensino de programação e qual a significatividade destes.

6 RESULTADOS E DISCUSSÕES

Ao final deste trabalho, foram desenvolvidos seis roteiros práticos que abordam os principais conceitos de introdução à programação. Cada roteiro aborda o conceito de forma clara, objetiva e, principalmente, de forma didática, para que alunos principiantes, que nunca tiveram uma experiência com situações de lógica ou programação, possam acompanhar tal ensinamento de forma prazerosa. Em todos os roteiros, são abordadas situações cotidianas e rotineiras, tornando assim o ambiente mais conhecido para o aluno.

No primeiro roteiro, é introduzido o Pensamento Computacional, abordando a execução estruturada e o sequenciamento de muitas atividades rotineiras. Para ensinar o Pensamento Computacional, de forma a associá-lo a uma situação real, foi utilizada uma receita de bolo de cenoura. Nesse exemplo, foi apresentado o modo de preparo do bolo, para que os alunos pudessem observar a sequência das atividades, assim como a associação de mais de uma atividade, apresentando-lhes as funções lógicas **E** e **OU**. Por fim, são propostas atividades para serem elaboradas pelos alunos, seguindo os conceitos aprendidos no roteiro.

A partir do segundo roteiro, a plataforma do *Scratch* é utilizada para implementar os conceitos introdutórios de programação. Assim, em cada roteiro, é inserido o conhecimento base e a implementação no *Scratch*, havendo uma implementação exemplo e outra para ser realizada pelos alunos. Em todas essas atividades, o aluno consegue visualizar as ações implementadas por eles mesmos no *Scratch*, a partir de animações referentes à rotina.

No segundo roteiro, foi introduzido o conhecimento sobre Variáveis. Nele, apresenta-se, de forma clara e simples, o que é uma variável em programação e como ela é implementada no *Scratch*. Além disso, são apresentados os blocos de variáveis existentes no *Scratch* e suas funcionalidades.

Depois de introduzidas as variáveis no conhecimento dos alunos, o terceiro roteiro é preparado para o ensino dos operadores existentes em programação. É importante que esses operadores sejam os primeiros a serem introduzidos, visto que são eles que definirão as “atitudes” a serem tomadas no programa. Desse modo, cada operador existente no *Scratch*, operadores matemáticos, de comparação e lógicos, são explicados e definidos, sempre associados ao cotidiano. A atividade de exemplo apresentada neste roteiro é uma atividade simples para aquisição de notas. Nela tais notas devem ser somadas, e o aluno, informado se foi ou não aprovado. Nessa atividade, o aluno, além de utilizar os conceitos de variáveis aprendidos no segundo roteiro, utiliza os operadores matemáticos e de comparação.

Nos roteiros quatro e cinco, são abordadas as estruturas condicionais **IF** e **ELSE IF**, de forma didática e com vários exemplos de situações cotidianas, principalmente aquelas em que é preciso tomar decisões levando em consideração alguma condição. Assim como nos roteiros anteriores, são apresentados os blocos de condições no *Scratch*, e as atividades utilizadas sempre relacionadas aos roteiros anteriores, fixando assim o conhecimento adquirido em aulas passadas.

Já no sexto roteiro, são abordados os laços de repetições para finalizar o conhecimento introdutório à programação. Neste roteiro são abordados os laços controlados por contadores, controlados logicamente e o que executa infinitamente. Para cada tipo são apresentados exemplos de aplicação, para que o aluno

consiga visualizar a diferença entre os tipos de laços. Mesmo apresentando tais exemplos, existe a atividade exemplo; o professor pode implementá-la juntamente com os alunos, de forma didática.

Todos os roteiros foram submetidos à avaliação de professores atuantes no IFMG, Campus Formiga, em disciplinas relacionadas à programação, sejam elas para o ensino técnico, sejam para o ensino superior. Nessas avaliações, todos os professores julgaram os roteiros como adequados para a introdução à programação para crianças e adolescentes, alguns ainda sugeriram os mesmos roteiros para o ensino de adultos que nunca tiveram contato com a programação. Todos relataram a importância da introdução à programação de forma mais lúdica, principalmente no ensino para as crianças, pois assim o aprendizado se torna mais fácil e prazeroso.

7 CONSIDERAÇÕES FINAIS

Durante a elaboração dos roteiros, foi possível notar o quanto o uso de plataformas como o *Scratch* pode ajudar no ensino, pois, além de ser uma programação em blocos, os alunos conseguem visualizar as ações realizadas pelo autor (Gato). Por ser uma ferramenta acessível e gratuita, aqueles que se interessarem poderão acessar a qualquer momento e implementar suas próprias rotinas. Além disso, ao atrelar o ensino a situações cotidianas, os alunos reconhecem que a programação e o pensamento lógico estão relacionados às atividades que executam durante todo o dia.

Com a avaliação dos professores atuantes no ensino de programação, pode-se concluir que a utilização desses roteiros poderá ser acrescentada no ensino à introdução à programação, seja de crianças, adolescentes, seja mesmo de adultos que se interessem em aprender sobre esse tema.

Assim, esses roteiros podem ser aplicados, principalmente, em cursos técnicos que possuem disciplinas que exigem o conhecimento em programação e não apresentam uma disciplina específica de programação em sua grade. O professor responsável pela disciplina poderá fazer um ensino orientado de tais roteiros, os quais podem ser abordados de forma simples e rápida. Além disso, a introdução à programação, utilizando uma plataforma como o *Scratch* e conceitos aplicados ao cotidiano, pode aumentar o interesse dos alunos no conhecimento desse tema e incentivá-los a continuar nesta área.

REFERÊNCIAS

BORGES, M. Avaliação de uma metodologia alternativa para a aprendizagem de programação. In: WORKSHOP DE EDUCAÇÃO EM COMPUTAÇÃO, 8., 2000, Curitiba. *Anais* [...]. Curitiba, 2000. Disponível em: <http://200.169.53.89/download/CD%20congressos/2000/SBC%202000/eventos/wei/wei006.ppd>. Acesso em: 12 jan. 2021.

DENNER, J.; WERNER, L.; ORTIZ, E. Computer games created by middle school girls: can they be used to measure understanding of computer science concepts? *Computers & Education*, Scotts Valey, v. 58, n. 1, p. 240-249, 2012.

- FORTE, A.; GUZDIAL, M. Computers for communication, not calculation: media as a motivation and context for learning. *In: PROCEEDINGS OF THE ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES*, 37., 2014, Hawaii. *Anais [...]*. Hawaii, 2014. p. 1-10.
- GOMES, W. F. *et al.* Incentivando meninas do ensino médio à área de ciência da computação usando o scratch como ferramenta. *In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO*, 3., 2014, Porto Alegre. *Anais [...]*. Porto Alegre, 2014. p. 1-12.
- GROVER, S.; PEA, R. Computational thinking in K-12: a review of the state of the field. *Educational Researcher*, Stanford, v. 42, n. 1, p. 38-43, 2013.
- INSTITUTO DE TECNOLOGIA DE MASSACHUSETTS. MIT. *Acerca do scratch*. [S. l.: s. n.], 2020a. Disponível em: <https://scratch.mit.edu/about>. Acesso em: 22 jun. 2020.
- INSTITUTO DE TECNOLOGIA DE MASSACHUSETTS. MIT. *Programming concepts and skills supported in scratch*. [S. l.: s. n.], 2020b. Disponível em: <http://scratched.gse.harvard.edu/sites/default/files/scratchprogrammingconcepts-v14.pdf>. Acesso em: 12 jan. 2021.
- JOHNSON, L. *et al.* *NMC horizon report: 2014 K-12 edition*. Austin, Texas: The New Media Consortium, 2014.
- KAFAI, Y. B.; BURKE, Q. *Connected code: why children need to learn programming*. Massachusetts: MIT Press, 2014.
- LEAL, A. V. de A.; FERREIRA, D. J. Aplicando padrões de seleção no ensino de programação de computadores para estudantes do primeiro ano do ensino médio integrado. *In: ENCONTRO ANUAL DE COMPUTAÇÃO*, 10., 2013, Catalão. *Anais [...]*. Catalão, 2013. p. 268-275.
- MARGULIEUX, L. E.; MORRISON, B. B.; DECKER, A. Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. *International Journal of STEM Education*, v. 7, n. 1, p. 1-16, 2020.
- MARQUES, M. T. P. M. Recuperar o engenho a partir da necessidade, com recurso às tecnologias educativas: contributo do ambiente gráfico de programação scratch em contexto formal de aprendizagem. 2019. Dissertação (Mestrado em Economia Rural) - Universidade de Lisboa, 2009.
- MARTINS, A. R. de Q. *Usando o scratch para potencializar o pensamento em crianças do ensino fundamental*. 2012. Dissertação (Mestrado em Educação) - Universidade de Passo Fundo, Passo Fundo, 2012.
- RESNICK, M. *et al.* Scratch: programming for all. *Communications of the ACM*, New York, v. 52, n. 11, p. 60-67, 2009.
- RODRIGUES, L. P.; MOURA, S.; TESTA, E. O tradicional e o moderno quanto à didática no ensino superior. *Revista Científica do ITPAC*, Araguaína, v. 4, n. 3, p. 34-42, 2011.

- RODRIGUES, M. Experiências positivas para o ensino de algoritmos. *In: ESCOLA REGIONAL DE COMPUTAÇÃO BAHIA-SERGIPE*, 4., 2004, Feira de Santana. *Anais [...]*. Feira de Santana, 2004. Disponível em: <http://www.uefs.br/erbase2004/documentos/weibase/Weibase2004Artigo001.pdf>. Acesso em: 12 jan. 2021.
- RODRIGUEZ, C. L. *et al.* Pensamento computacional: transformando ideias em jogos. *In: WORKSHOP DE INFORMÁTICA NA ESCOLA*, 21., 2015, Maceió. *Anais [...]*. [S. l.], 2015. p. 62-71.
- SÁEZ-LÓPEZ, J.-M.; ROMÁN-GONZÁLEZ,.; VÁZQUEZ-CANO, ?. Visual programming languages integrated across the curriculum in elementary school: a two year case study using “scratch” in five schools. *ELSEVIER - Computers & Education*, v. 97, p. 129-141, 2016.
- SANTOS, R. *et al.* Uma proposta de cenário para ensino de algoritmos e programação com contribuições de cooperação, colaboração e coordenação. *In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO*, 16., 2008, Belém. *Anais [...]*. Belém, 2008. p. 218-227. Disponível em: <http://www.prodepa.gov.br/sbc2008/anais/pdf/arq0197.pdf>. Acesso em: 12 jan. 2021.
- SCAICO, P. D. *et al.* Ensino de programação no ensino médio: uma abordagem orientada ao design com a linguagem scratch. *Revista Brasileira de Informática na Educação*, [S. l.], v. 21, n. 2, p. 92, 2013.
- SILVA, I. F. A.; SILVA, I. M. M; SANTOS, M. S. *Análise de problemas e soluções aplicadas ao ensino de disciplinas introdutórias de programação*. 2009. Dissertação (Mestrado em Economia Rural) - Universidade Federal de Pernambuco, Recife, 2009.
- WANGENHEIM, C. G. V.; NUNES, Vinícius Rodrigues; SANTOS, Giovane Daniel dos. Ensino de computação com SCRATCH no ensino fundamental: um estudo de caso. *Revista Brasileira de Informática na Educação*, [S. l.], v. 22, n. 3, p. 116-124, 2014.
- WING, J. *Computational thinking: communications of the ACM*. New York: Association for Computing Machinery, 2006.

Recebido em: 30 maio 2021.

Aceito em: 28 fev. 2022.