

Ensino de Programação utilizando Computação Física: uma Revisão Sistemática da Literatura

Humberto A. P. Zanetti, Marcos A. F. Borges

Faculdade de Tecnologia
Universidade de Campinas, Limeira – SP, Brasil
h016304@dac.unicamp.br, marcosborges@ft.unicamp.br

Resumo: Esta pesquisa tem como objetivo discutir sobre a teoria, as aplicações e as principais estratégias didáticas para ensino de Programação utilizando recursos de Computação Física. Pesquisas sobre esse tema são recentes. No meio acadêmico, ainda há questões a serem discutidas sobre o alinhamento entre as atuais pesquisas. Essas questões podem causar má interpretação ou dúvidas ao se escolher ou adotar práticas sobre o tema. Nesta Revisão Sistemática da Literatura, foram avaliados 15 artigos, com o objetivo de apresentar uma visão crítica sobre as pesquisas mais recentes e suas aplicações.

Palavras-chave: *Ensino de programação; Computação física; Revisão da literatura.*

1. Introdução

Atualmente, a tecnologia faz parte da rotina diária de todos e é muito acessível. Jovens estudantes dominam boa parte dos recursos tecnológicos e computacionais, devido à sua vivência com tecnologia desde a infância. É cada vez mais esperado que a escola forneça um ambiente em que a tecnologia esteja presente, não só como uma ferramenta de ensino, mas em dinâmicas nas quais esses jovens deixem de ser meros “consumidores” de tecnologia e se tornem “criadores”. Nesse contexto, o papel fundamental da escola e dos educadores é orientar e conduzir práticas que catalisem a motivação e a criação com esses jovens.

A programação de computadores tem potencial para ser uma atividade educativa que pode auxiliar de maneira efetiva a aquisição de habilidades cognitivas fundamentais para diversas outras tarefas, como a atitude crítica-reflexiva e a resolução de problemas [13]. Também auxilia o aluno a explicitar um modelo mental de solução, podendo expressar de maneira quase fiel a sequência de resolução e verificar a causa-e-efeito [14].

Computação Física (CF) é a integração da computação com o mundo físico, através de sensores e atuadores, amparados por sistemas embarcados, comunicando-se via redes de computadores e provendo ambientes automatizados [22]. A CF tem como objetivo conectar o mundo virtual com o real, com a criação de novas interfaces intuitivas entre objetos e seres humanos. Utilizar da CF em práticas de ensino pode oferecer recursos didáticos que ajudem a promover a criatividade e a compreensão da aplicação de conceitos [15].

Atividades envolvendo componentes eletrônicos e robótica, alinhados com o ensino de programação, podem diminuir as dificuldades de alunos iniciantes no processo do aprendizado de programação. Por exemplo, utilizando um robô móvel e uma arena com obstáculos, para demonstrar conceitos como estruturas condicionais (*if-else*) e de repetição (*for* e *while*) [23].

2. Metodologia

Para este trabalho, foi realizada uma Revisão Sistemática da Literatura (RSL), que permitiu

identificar e analisar principais aspectos da área de pesquisa que envolve CF e ensino de Programação.

2.1. Questões de Pesquisa

O principal objetivo desta RSL é avaliar estudos realizados no período de 2010 a 2019, que apresentem contribuições ao tema de uso da CF no ensino de programação. A principal questão a ser respondida com este presente trabalho é: “*Quais são as abordagens didáticas aplicadas para o ensino de programação utilizando ferramentas e artefatos de Computação Física?*”. Com base na questão principal apresentada, outras 2 questões de pesquisa (QP) mais específicas foram definidas:

QP1: *Quais as motivações para o uso de CF no ensino de programação?*

QP2: *Quais são as principais plataformas e ferramentas de CF utilizadas nas práticas?*

2.2. Métodos Aplicados na Revisão da Literatura

A presente RSL foi realizada seguindo a metodologia PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) [3]. As bases de dados utilizadas para a busca de artigo foram ACM, Google Scholar, IEEE, Science Direct, Scopus e Springer, em trabalhos publicados entre os anos de 2010 e 2019. A chave (*string*) de pesquisa utilizava termos em português e inglês, como “computação física” (“*physical computing*”), ensino ou aprendizagem (“*teach*”, “*teaching*”, “*learning*”) e programação (“*programming*”, “*coding*”)

Os critérios de inclusão dos trabalhos selecionados foram definidos a partir da capacidade de atender pelo menos uma das questões de pesquisa e, após isso, seguir os outros critérios de inclusão, mostrados no Quadro 1. Foi feita a aplicação dos critérios de inclusão e exclusão através de uma leitura rápida (*scanning*), dando ênfase nos resumos, seções de resultados e/ou conclusões. Maiores detalhes da metodologia adotada estão disponíveis no site do projeto.¹

¹ Endereço do site: <https://tinyurl.com/projetoCFAS>

Quadro 1. Critérios de inclusão e exclusão.

Critérios de Inclusão	Critérios de Exclusão
<p>I1. Publicados após o ano de 2010, e até 2019.</p> <p>I2. Artigos com práticas pedagógicas e/ou proponha alguma metodologia ou ferramenta didática.</p> <p>I3. Abordar práticas de programação e CF</p>	<p>E1. Artigos que apresentem apenas revisão da literatura.</p> <p>E2. Não abordar ensino de programação ou CF.</p> <p>E3. Trabalhos que não contemplem nenhuma das questões de pesquisa</p>

Com isso, foram selecionados 15 artigos que atendem o propósito desta RSL, que foram publicados entre 2013 e 2018, embora tenham sido buscadas pesquisas sobre o tema desde 2010, só surgiram trabalhos relevantes no período indicado. Após essa seleção, foi realizado um fichamento através de um formulário de extração com o propósito de identificar as contribuições de cada artigo.

3. Resultados e Discussão

Esta seção apresenta uma visão geral dos artigos selecionados na Seção 3 e as discussões das questões apresentadas na Seção 2.

3.1. Discussão sobre a QP1

Foi analisado se há relatos ou descrições nos artigos que apontem quais foram as motivações que levaram a adoção da CF nas práticas descritas ou no desenvolvimento de alguma plataforma ou ferramenta. Dentre as motivações descritas, foi possível categorizar 5 áreas distintas, que conseguem trazer um consenso e agrupamentos coerentes entre os trabalhos.

A categorização utilizada nesta RSL foi feita baseando-se nos relatos e descrições mais comuns nos artigos recuperados pelas buscas. Essa categorização tem como objetivo prover uma referência para o mapeamento de futuros pesquisadores interessados em CF e ensino de programação. As áreas definidas foram: “Ambiente dinâmico e/ou motivador” (ADM); “Compreensão de Conceitos ou Paradigmas de Programação” (CCPP); “Engajamento e Criação” (EC); “Resolução de Problemas” (RP); e “Visualização e Identificação de Erros” (VIE). O Quadro 2 traz uma breve descrição de cada categoria e a relação de artigos.

Os artigos que se enquadram na categoria ADM foram a maioria, pois a busca por um ambiente mais dinâmico e motivacional é algo muito buscado em práticas didáticas que envolvam programação. O ambiente tradicional de programação pode ser pouco estimulante, potencializando possíveis problemas e dúvidas que os alunos tenham [8]. Para esses trabalhos, a adoção de recursos da CF pode ser um aliado para esse propósito, devido a uma maior ludicidade das atividades.

A segunda categoria com mais artigos, a CCPP, aborda um problema bastante comum entre os iniciantes em programação, a pouca capacidade de abstração. A aprendizagem de programação demanda habilidades como abstração, capacidade de identificar generalização

e pensamento crítico [4]. Meios que ajudem a visualização, de maneira mais concreta, fora do computador, dos resultados de certos conceitos de programação podem ser bastante positivos.

Quadro 2. Descrição das categorias e relação dos artigos.

Id.	Descrição	Artigos
ADM	Relatos que indicavam que a presença de CF tinha o objetivo de criar um ambiente mais dinâmico e motivacional, diferente do tradicional, despertando maior interesse pelos alunos.	[1], [2], [6], [9], [10], [11], [12], [16], [17], [21]
CCPP	Artigos que procuram meios de trazer instrumentos didáticos para ajudar os alunos na abstração e visualização de conceitos abstratos relacionados à programação.	[1], [5], [6], [7], [10], [12], [18], [19]
EC	Participação ativa dos alunos e estímulo à criação de novos artefatos e soluções usando CF, estimulando a cultura <i>maker</i>	[5], [20]
RP	Trabalhos que abordam a habilidade de tomada de decisão e resolução de problemas, estimulando a participação do aluno na construção de soluções.	[10], [21]
VIE	Estudo que utilizam a CF para proporcionar um ambiente que possa trazer <i>feedback</i> que sejam melhores interpretados e os erros visualizados de maneira mais tangível.	[10], [16]

Há um alinhamento com os artigos da categoria VIE, pois a busca por instrumento que tornem os conceitos de programação mais tangíveis auxilia na interpretação e na percepção de erros que estão ocorrendo, e consequentemente, a busca por soluções.

Os trabalhos da categoria EC dão ênfase na relação de engajamento do aluno junto a prática e ao incentivo de utilizar a tecnologia não apenas como um consumidor, mas sim como um criador.

Os trabalhos que se enquadram na categoria RP tinham como objetivo proporcionar um cenário que explorasse a habilidade de tomada de decisão e resolução de problemas, que são essenciais para a programação.

É importante notar que alguns artigos aparecem em mais de uma categoria, denotando algumas intersecções. Os artigos [1], [6], [10] e [12] aparecem em ADM e CCPP, mostrando que trabalhos buscam ambientes motivadores com o objetivo de trazer maneiras de ajudar na abstração de conceitos complexos. O trabalho A3 aparece nas categorias CCP e EC, por tratar sobre a criação de artefatos a partir elementos da CF e programação, para promover a visualização de conceitos abstratos. Por fim, uma análise possível de intersecção de categorias é o artigo [10], presente em 4 categorias distintas (ADM, CCPP, RP e VIE), tratando-se de uma pesquisa que apresenta práticas detalhadas, lúdicas, motivacionais e focadas em trazer conceitos abstratos de maneira mais tangível através de puzzles.

3.2. Discussão sobre a QP2

O objetivo principal desta questão foi fazer um levantamento de quais são as principais plataformas ou ferramentas utilizadas, podendo se identificar, inclusive,

se há novas ferramentas sendo desenvolvidas. Buscou-se identificar o cenário atual e quais tecnologias são mais adotadas nas pesquisas e as novas demandas.

As tecnologias utilizadas ou desenvolvidas nos trabalhos selecionados foram: a plataforma de prototipagem Arduino, o kit de robótica *Lego® Mindstorms*, a plataforma de tecnologias vestíveis *Lilypad* Arduino e as placas de eletrônica *Makey Makey* e *micro:bit*. Também houve 3 trabalhos que desenvolveram suas próprias tecnologias. A Figura 1 mostra um gráfico com relação total dos artigos e seus identificadores, além dos números de artigos.

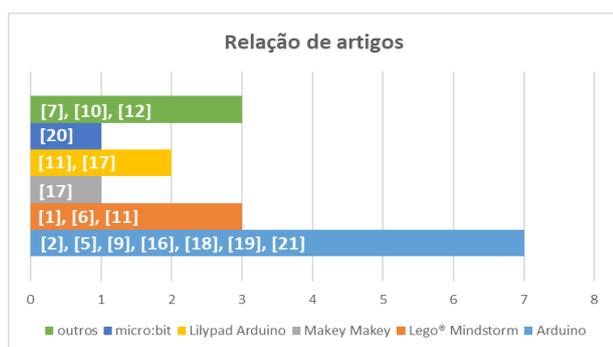


Figura 1. Relação dos artigos por ferramenta/plataforma.

A presença do Arduino na maioria dos trabalhos era esperada, por se tratar de uma plataforma de CF acessível, tanto por disponibilidade no mercado, quanto devido ao seu caráter de projeto aberto de hardware. Também existe muito material e literatura voltado à plataforma. A versatilidade do Arduino é outro ponto positivo na sua adoção, pois pode ser incluído em projetos simples e complexos, envolvendo práticas de robótica e Internet das Coisas.

O baixo custo e fácil manutenção também são atrativos para instituições de ensino. Para esse estudo, a plataforma *Lilypad* (baseado em Arduino) foi separada das demais, devido à sua aplicação específica em tecnologias vestíveis. Os dois estudos que usam *Lilypad* possuem práticas com um foco estritamente em desenvolvimento de projetos objetivando a criação de dispositivos vestíveis (*wearable*) e ambos usam outras tecnologias conjuntamente: um usa também *Makey Makey* e outro *Lego® Mindstorms*.

O kit didático *Lego® Mindstorms* aparece em 3 artigos, sendo utilizado em práticas de robótica e programação, que é justamente o objetivo principal dessa ferramenta. Mesmo com seu alto custo no mercado, seus benefícios são notáveis nos estudos, como a versatilidade de construções de modelos automatizados e a facilidade de programação. As demais plataformas didáticas, *Makey Makey* e *micro:bit*, que foram desenvolvidas para criar pequenos projetos e ter uma interface de programação intuitiva, aparecem em apenas 1 artigo para cada ferramenta. Ambas são ferramentas com recursos limitados e custo relativamente alto, se comparadas com as plataformas citadas anteriormente.

Houve 3 trabalhos que desenvolveram tecnologias próprias para serem utilizadas em práticas com CF: *Talkoo kit* no artigo [7], *Bots & (Main)Frames* no artigo [10] e *i*CaTch* no artigo [12]. A *Talkoo kit* tem como objetivo criar um ambiente de ensino de conceitos de computação, com facilidade de montagem de projeto através de módulos *plug-and-play* e com ambiente de programação visual. O projeto *Bots & (Main)Frames* propõe um ambiente de desafios (*puzzle*) para o ensino de programação, utilizando blocos físicos baseados em tecnologia tangível. Por fim, a plataforma *i*CaTch* apresenta um kit didático usando CF e tecnologia vestíveis, para estimular o ensino de programação e criação de projetos de dispositivos vestíveis.

4. Conclusões

Este estudo teve como principal objetivo fazer o levantamento e análise de pesquisas que abordam recursos da CF no ensino e aprendizagem de programação. Para alcançar esse objetivo, foi realizada uma RSL buscando por trabalhos nas principais bases de trabalhos acadêmicos publicados a partir de 2010.

Foi observada uma maior concentração de artigos de pesquisas que buscam a CF como um recurso didático com propósito de criar um ambiente motivador, que promova maior engajamento por parte do aluno e que apresente elementos que auxilia no aprendizado de conceitos de programação, muitas vezes abstratos. Esses aspectos são recorrentes em diversos trabalhos que objetivam novas estratégias ou recursos para o ensino de programação. Também foi identificado que há poucos trabalhos focados em desenvolvimento de habilidades necessárias ao programador, como resolução de problemas ou identificação e correção de erros.

A plataforma mais utilizada nas pesquisas foi o Arduino, seguida pelo kit didático *Lego® Mindstorms*, ferramentas essas presentes há mais de uma década no mercado. A plataforma *Lilypad*, baseada em Arduino, apresenta uma solução para tecnologias vestíveis, podendo trazer soluções criativas e acessíveis. As ferramentas *Makey Makey* e *micro:bit*, mais recentes, também foram citadas.

Bibliografia

- [1] Anfurrutia, F. I. et al. (2016) Incorporating educational robots and visual programming environments in introductory programming courses. In 2016 International Symposium on Computers in Education (SIIE) (pp. 1-4). IEEE. DOI: [10.1109/SIIE.2016.7751835](https://doi.org/10.1109/SIIE.2016.7751835).
- [2] Arakliotis, S.; Nikolos, D. G.; Kalligeros, E. (2016) LAWRI: A rule-based arduino programming system for young students. In 2016 5th International Conference on Modern Circuits and Systems Technologies (MOCASST) (pp. 1-4). IEEE. DOI: [10.1109/MOCASST.2016.7495150](https://doi.org/10.1109/MOCASST.2016.7495150).
- [3] Galvão, T. F.; Pansani, T. D. S. A.; Harrad, D. (2015) Principais itens para relatar Revisões sistemáticas e Meta-análises: A recomendação PRISMA. *Epidemiologia e Serviços de Saúde* 24: 335-342.

- [4] Gomes, A.; Mendes, A. J. (2007) Learning to program-difficulties and solutions. In: International Conference on Engineering Education–ICEE, v. 2007.
- [5] Jang, Y.; Lee, W.; Kim, J. (2015) Assessing the usefulness of object-based programming education using arduino. *Indian Journal of Science and Technology* 8(S1): 89-96. DOI: [10.17485/ijst/2015/v8iS1/57701](https://doi.org/10.17485/ijst/2015/v8iS1/57701).
- [6] Jin, K. H.; Haynie, K.; Kearns, G. (2016) Teaching elementary students programming in a physical computing classroom. In: Proc. 17th annual conference on information technology education, p. 85-90. DOI: [10.1145/2978192.2978238](https://doi.org/10.1145/2978192.2978238).
- [7] Katterfeldt, E. S. et al. (2018) Physical computing with plug-and-play toolkits: Key recommendations for collaborative learning implementations. *International Journal of Child-Computer Interaction* 17: 72-82. DOI: [10.1016/j.ijcci.2018.03.002](https://doi.org/10.1016/j.ijcci.2018.03.002)
- [8] Lahtinen, E.; Ala-Mutka, K.; Järvinen, H. M. (2005) A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin* 37(3): 14-18. DOI: [10.1145/1067445.1067453](https://doi.org/10.1145/1067445.1067453).
- [9] Martin, C.; Hughes, J.; Richards, J. (2017) Learning Experiences in Programming: The Motivating Effect of a Physical Interface. In: Proc. 9th International Conference on Computer Supported Education (CSEDU) p. 162–172. DOI: [10.5220/0006375801620172](https://doi.org/10.5220/0006375801620172).
- [10] Melcer, E. F. (2017) Exploring the Effects of Physical Embodiment in a Puzzle-Based Educational Programming Game. In: Proc. 2017 ACM SIGCHI Conference on Creativity and Cognition, p. 532-538) DOI: [10.1145/3059454.3078704](https://doi.org/10.1145/3059454.3078704).
- [11] Merkouris, A.; Chorianopoulos, K.; Kameas, A. (2017) Teaching programming in secondary education through embodied computing platforms: Robotics and wearables. *ACM Transactions on Computing Education (TOCE)* 17(2): 1-22. DOI: [10.1145/3025013](https://doi.org/10.1145/3025013).
- [12] Ngai, G. et al. (2013) Designing i* CATch: A multipurpose, education-friendly construction kit for physical and wearable computing. *ACM Transactions on Computing Education (TOCE)* 13(2): 1-30. DOI: [10.1145/2483710.2483712](https://doi.org/10.1145/2483710.2483712).
- [13] Papert, S. (1994) A máquina das crianças. Porto Alegre: Artmed.
- [14] Pea, R. D.; Kurland, D. M. (1984) On the cognitive effects of learning computer programming. *New Ideas in Psychology* 2(2): 137-168. DOI: [10.1016/0732-118X\(84\)90018-7](https://doi.org/10.1016/0732-118X(84)90018-7).
- [15] Przybylla, M.; Henning, F.; Schreiber, C.; Romeike, R. (2017) Teachers' Expectations and Experience in Physical Computing. In: Int. Conf. on Informatics in Schools: Situation, Evolution, and Perspectives, p. 49-61. Springer. DOI: [10.1007/978-3-319-71483-7_5](https://doi.org/10.1007/978-3-319-71483-7_5).
- [16] Przybylla, M.; Romeike, R. (2014) Physical Computing and Its Scope--Towards a Constructionist Computer Science Curriculum with Physical Computing. *Informatics in Education* 13(2): 241-254. DOI: [10.15388/infedu.2014.05](https://doi.org/10.15388/infedu.2014.05).
- [17] Richard, G. T.; Kafai, Y. B. (2015) Making physical and digital games with e-textiles: a workshop for youth making responsive wearable games and controllers. In: Proc. 14th Int. Conf. on Interaction Design and Children, p. 399-402. DOI: [0.1145/2771839.2771926](https://doi.org/10.1145/2771839.2771926).
- [18] Rubio, M. A. et al. (2014) Enhancing an introductory programming course with physical computing modules. In: Proc. 2014 IEEE Frontiers in Education Conference (FIE), p.1-8. IEEE. DOI: [10.1109/FIE.2014.7044153](https://doi.org/10.1109/FIE.2014.7044153).
- [19] Rubio, M. A.; Hierro, C. M.; Pablo, A. P. D. M. (2013) Using arduino to enhance computer programming courses in science and engineering. In: Proc. EDULEARN13 Conference (1-3). IATED. Barcelona, Spain.
- [20] Sentance, S. et al. (2017) Teaching with physical computing devices: the BBC micro: bit initiative. In: Proc. 12th Workshop on Primary and Secondary Computing Education, p. 87-96. DOI: [10.1145/3137065.3137083](https://doi.org/10.1145/3137065.3137083).
- [21] Sohn, W. (2014) Design and evaluation of computer programming education strategy using Arduino. *Advanced Science and Technology Letters* 66(1): 73-77. DOI: [10.14257/astl.2014.66.18](https://doi.org/10.14257/astl.2014.66.18).
- [22] Stankovic, J. A. et al. (2005) Opportunities and obligations for physical computing systems. *Computer* 38(11): 23-31. DOI: [10.1109/MC.2005.386](https://doi.org/10.1109/MC.2005.386).
- [23] Zanetti, H., e Oliveira, C. (2015). Práticas de ensino de Programação de Computadores com Robótica Pedagógica e aplicação de Pensamento Computacional. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação, 4(1): 1236.