Revisión sistemática de literatura: MOOC K-12 y STEAM

Systematic literature review: MOOC K-12 and STEAM

Ginna Viviana Leytón Yela **Víctor Andrés Bucheli Guerrero**

Universidad del Valle, Colombia

Hugo Armando Ordoñez Erazo
Universidad del Cauca, Colombia

Recibido: 22/09/2021
Aceptado: 22/10/2021
Publicado: 13/12/2021

Correspondencia de autores:
hugoordonez@unicauca.edu.co

Resumen

Objetivo: Identificar los trabajos de investigación centrados en los MOOC K-12 o STEAM, enfocados en el potencial de los entornos de educación, remodelando el marco educativo actual en la educación secundaria. **Metodología:** Se definieron ecuaciones para realizar las búsquedas de articulos publicados en conferencias y revistas mediante bases de datos bibliográficas, q ue p ermitieron sustraer los articulos para construir el corpus de referencias. Posteriormente, se analizan y sintetizan las investigaciones más significativas para el desarrollo de la revisión sistemática. **Resultados:** Esta revisión permitió identificar las tendencias, desafíos y oportunidades sobre MOOC K-12 o STEAM, identificando las implementaciones de cursos de ciencias de computación o pensamiento computacional en educación secundaria. Este tipo de iniciativas han obtenido resultados positivos, puesto que se han incrementado el número de estudiantes que seleccionan la programación informática como eje fundamental en el examen nacional para educación superior. **Conclusiones:** Las investigaciones se han centrado en su mayoría en la incorporación de la programación con cursos complementarios ofertados de forma opcional en educación secundaria, con un enfoque combinado, es decir, se llevan a cabo clases tradicionales con el apoyo de los MOOC. La gran mayoría de los MOOC para educación secundaria se han implementado específicamente para satisfacer las necesidades de los docentes y estudiantes de educación secundaria, con el fin de mejorar y apoyar el plan de estudios, y motivar a los estudiantes como futuros miembros activos en la creación de tecnología.

Palabras clave: MOOC, cursos online masivos y abiertos, K-12, STEM, STEAM, educación secundaria, pensamiento computacional.

Abstract

Objective: To identify research works focused on MOOCs K-12 or STEAM, focused on the potential of education environments, reshaping the current educational framework in secondary education. **Methodology:** Equations were defined to search for articles published in conferences and journals using bibliographic databases, which allowed subtracting the articles to build the corpus of references. Subsequently, the most significant investigations for the development of the systematic review are analyzed and synthesized. **Results:** This review will identify trends, challenges, and opportunities on MOOC K-12 or STEAM, identifying the implementations of courses in computer science or computational thinking in education. This type of initiative has obtained favorable results since the number of students who select computer programming as a fundamental axis in the national exam for higher education has increased. **Conclusions:** Most of the research has focused on incorporating programming with complementary courses offered optionally in secondary education, with a combined approach, that is, traditional classes are carried out with the support of MOOCs. Most MOOCs for secondary education have explicitly been implemented to meet the needs of secondary education teachers and students to enhance and support the curriculum and motivate students as future active members in creating technology.

Keywords: MOOC, massive open online courses, K-12, STEM, STEAM.

Como citar (IEEE): G. Leytón-Yela., V. Bucheli-Guerrero., y H. Ordoñez-Erazo. "Revisión sistemática de literatura: MOOC K-12 y STEAM". Investigación e Innovación en Ingenierías, vol. 9, n°3, 57-81, 2021. DOI: https://doi.org/10.17081/invinno.9.3.5546

Introduction

For some time now, there has been a great debate on incorporating computer science or programming at the primary and secondary school levels for students to be active members of society, not only as consumers of technology but also as creators of it in the future.

Indeed, countries with more excellent technological experience have begun incorporating the K-12 curriculum in school education, starting in the early grades, and continuing through grade 12 [1]. Students acquire a knowledge base in problem-solving and computational thinking.

Likewise, STEAM training presents strategies such as creating clubs, communities, programming, robotics tournaments, and even the teaching of critical thinking and programming languages implemented in their curricula. In the case of Europe, this is done through face-to-face, blended or virtual classes with the support of virtual platforms, code platforms, or massive open online courses [2].

As mentioned above, several computational tools support teaching, among them the open education system that has evolved in various forms. This open trend has developed with MOOCs, which aim at sharing and open access to knowledge. MOOCs are accessible to anyone, regardless of age, educational experience, or location [3]. There has been a growing interest in promoting and teaching programming to the global audience through MOOCs, such as Platzi, edX, Coursera, Udemy, and Udacity. This ever-increasing trend began to focus on lower educational levels, not only higher education.

Since 2013, incorporating these types of courses as standalone initiatives for youth and K-12 educators has begun. A small but growing group of researchers [4, 5, 6, 7] have identified that MOOCs in K-12 education can help solve gaps in problem solving of all kinds by using valuable content and different learning strategies. In addition to the importance of equipping the new generations with skills in programming and computational thinking [8].

There is a great deal of academic research related to implementing K-12 or STEAM through platforms such as MOOCs. Still, it is necessary to establish which platforms successfully implemented globally, what beneficial results it has brought in the educational field, and what artificial intelligence (AI) techniques support the development of courses through MOOC-like platforms.

The realization of this systematic literature review on the implementation of K-12, STEAM, or computer science or computational thinking courses in secondary education seeks to contribute relevant findings that have transformed education. The paper is organized as follows: the methodological process to select the relevant scientific articles on the subject and those that answer the open research questions defined in the Methodology are presented; then, the analysis of the results obtained in the systematic review classified in the three aspects mentioned above; finally, we showed the discussion of the results of the literature review.

Methodology

To develop this study, we implemented the systematic literature review methodology to identify evidence-based research, subtract relevant information for future studies, and create knowledge from publications directly related to programming MOOCs in secondary education. We included:

- The bibliographic databases.
- The identification criteria with the search equations.
- The selection criteria allowed obtaining the corpus of references.

The stages of the applied methodology are; inquiry, identification, selection, analysis, and synthesis of the selected articles [9].

Table 1. Questions and motivations.

Research questions	Motivation
RQ.1. ¿What is the status of relevant studies on K-12 or STEAM MOOCs published since 2015?	This question is intended to elicit specific information about programming MOOCs in secondary education, as a starting point in the recognition of basic concepts such as information about the origins.
RQ.2. ¿What K-12 or STEAM MOOC platforms have been successfully implemented in secondary education?	This question is intended to identify the main types of platforms that have achieved good results in secondary education.
RQ.3. ¿How have K-12 MOOC or STEAM platforms transformed education?	This question is intended to provide a description of the positive and negative transformation of MOOCs in secondary education.
RQ.4. ¿What AI techniques have been implemented in K-12 or STEAM MOOC platforms?	This question is intended to delve into the IA techniques implemented in MOOCs for high school, as a support in the teaching-learning process.

Source: Own elaboration.

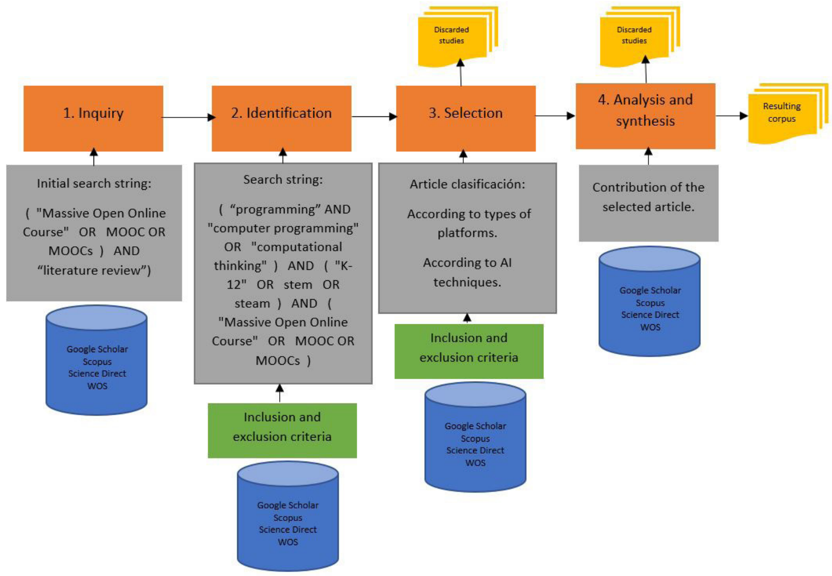
Research questions and motivation for the study

We determined a set of research questions and motivation for the study through this process, as shown in table 1.

We identified scientific documents such as journals and conferences in Science Direct, Scopus, and Web of Science for the systematic literature review. In the searches, we found 110 papers, of which three were duplicated. Finally, we selected 100 scientific documents to form the final corpus. The following selection criteria made the selection.

- Articles published from 2015 onwards.
- Scientific publications in English and Spanish.
- If there are articles in different bibliographic repositories, compare if they are the same and leave the most recent one.
- If there are short and full versions of the same study, the latter is included.
- Articles directly related to the subject matter (MOOC K-12 or STEAM).
- Articles that present studies related to secondary education.

Fig. 1 Stages of systematic review.



Source: Own elaboration.

Inquiry

Stage 1, corresponding to the Inquiry, includes:

- Verification of the existence of systematic literature review publications on MOOCs in the subareas of Engineering or Computer Science.
- A general search is performed in the selected databases to verify the size of the reference corpus, using the keywords and the initial search string located in Table 2.
- Identify the keywords needed in the search equation and the subject areas to consider in the search, selection, and analysis.

Identification

Stage 2 for identification includes:

- With the definition of the research questions and the motivations of the study, we determined and selected the search equation, the relevant bibliographic databases, and the inclusion and exclusion criteria. For the definition of the search equations, we considered the keywords identified in the research topic as mentioned above.
- A search is carried out to compile the reference corpus. Table 2 shows the search equations, the results obtained, and the results saved.

Table 2. Search equations, data source and number of items retrieved for each search.

No.	Search string	Source	Recovered
1	TITLE-ABS-KEY ((("Massive Open Online Course" OR mooc OR moocs) AND "literature review")) AND (EXCLUDE (PUBYEAR , 2014) OR EXCLUDE (PUBYEAR, 2013) OR EXCLUDE (PUBYEAR , 2012))	Scopus, Science Direct	51
2	("programming" AND "computer programming" OR "computational thinking") AND ("K-12" OR stem OR steam) AND ("Massive Open Online Course" OR MOOC OR MOOCs) AND (LIMIT-TO (SUBJAREA, "COMP") OR LIMIT-TO (SUBJAREA , "ENGI")) AND (EXCLUDE (EXACTKEYWORD , "Higher Education"))	Scopus, Science Direct, Web of Science	71
3	((programming OR "computer programming" OR "computational thinking") AND ("massive open online courses" OR "MOOC") AND ("K-12" OR "STEM" OR "STEAM") AND "artificial intelligence") AND (LIMIT- TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA, "ENGI"))	Scopus, Science Direct	32

Source: Own elaboration.

Selection

Stage 3 for Selection includes:

- According to the inclusion and selection criteria, we analyzed the titles, abstracts, and keywords of the studies obtained in the searches.
- The quality and relevance of the studies are evaluated by applying a Likert scale to determine whether the articles are transcendental to the study.
- We only performed papers with a minimum score of 3 are selected, according to the rankings determined in the study.

Analysis and synthesis.

Stage 4 for Analysis and synthesis includes:

- Obtain and analyze the content of selected studies in databases.
- Using reference management software such as Zotero and Mendeley for complete document extraction and detailed bibliographic organization.
- Conduct literature analysis within the study and across studies in the final corpus.
- Perform a descriptive synthesis to identify the categories of Analysis, for this case, years of publication, authors, types of MOOC platform.
- Answering the research questions defined in the methodology. The most notable features of the corpus are extracted. For this purpose, we used tools such as SARS and VOSViewer to identify other keywords and authors relevant to the research, to include them in the search equations.

Results

This section aims to present the most significant results of the systematic literature review carried out between February 2021 and June 2021.

When identifying the articles in the selection stage, the strategies implemented in secondary education for learning programming through MOOC K-12 or STEAM type platforms are analyzed. In the same way, the benefits or advantages of implementing this type of course in secondary education and the IA techniques implemented in this type, of course, are identified. In this way, we answered each of the research questions defined in the methodology.

Stage 1: Inquiry, previous literature review.

Initially, prior consultation of systematic literature reviews related to MOOCs, published between 2015 and 2019, is carried out. This initial exploration allows identifying existing reviews about MOOCs that could support the search process through the methodologies applied in such reviews, the topics addressed, the issues resolved, and keywords established, in addition to being a starting point for the preliminary search processes in this research. We selected seven reviews of the literature on MOOCs.

In 2018, there are systematic literature review studies [10, 11, 12]. The first one allows identifying the accessibility requirements of MOOCs around MOOC platforms and content providers, considered accessibility needs for students with disabilities, elderly students, and foreigners.

The second examines MOOC implementations and evaluations, typically done by rating successes and challenges in MOOC revenue earned, acceptance, and completion rates. And the third focuses on understanding the implications of taking a MOOC for students, examining subjective considerations inherent in learning such as motivation, emotional and intellectual engagement.

In 2020, there will be studies related to MOOC [13] about the origins and the evolution of MOOC concepts from 2012 to 2019. In addition, the article shows the MOOC-type platforms that have given language courses, their strengths, and limitations. In the same way, special mention is made of the specialized courses and didactic applications present in MOOCs.

Another work [14] addresses the recommendation systems in MOOCs, emphasizing the need for this type of system, the proposed systems, and their implementation through videos, books, and the application of artificial intelligence. Finally, this year's study [15] focuses on the K-12 MOOC model, in which consistency is sought with pedagogical methods that best suit how these types of students learn. A blended MOOC model can enhance learning and ensure teaching efficiency in introductory courses, which could help students prepare for college or as capstone courses for underachieving students or advanced courses that offer new teaching subjects.

In 2021, the study [16] presents the current support for self-regulated learning in MOOCs, using technologies based on psychological models between 2010 and 2020. This study was working on student abandonment that mainly occurs in MOOCs due to the lack of methodologies supported by ICT, designed to help students self-regulate their learning.

After this initial search, we carry out specific investigations to access the articles that can answer the research questions defined in the methodology.

Table 3. Selected literature, MOOC reviews.

Year	Reference
2018	Research challenges in accessible MOOCs: a systematic literature review 2008–2016 [10]. A thematic literature review of the implementation of MOOCs - 2008 to 2018 [11]. Survey on understanding the implications of MOOCs in engineering education [12].
2020	Recommender Systems for MOOCs: A Systematic Literature Survey (January 1, 2012 - July 12, 2019) [13]. MOOCs: Origins, Concept and Didactic Applications: A Systematic Review of the Literature (2012–2019) [14]. Research trends in K-12 MOOCs: A review of the published literature [15].
2021	Self-Regulated Learning in Massive Online Open Courses: A State-of-the- Art Review [16].

Source: Own elaboration.

Stage 2: Identification.

At this stage, we carried out the searches defined in the methodology in Table 2. A total of 111 studies were identified, removing duplicate studies. Table 4 shows the distribution of the studies identified by type and year of publication, including the number in numerical value and percentage.

Table 4. Distribution by type and year of the studies identified in the stage 2.

	2015	2016	2017	2018	2019	2020	2021	
Non-indexed journal articles	2	1	2	1	0	1	0	7 (6%)
Indexed articles	8	7	13	24	19	29	4	104 (94%)
Total studies	10 (9%)	8 (7%)	15 (14%)	25 (23%)	19 (17%)	30 (27%)	4 (4%)	111

Source: Own elaboration.

Stage 3: Selection.

In this stage, we filtered the number of studies selected, where we identified 60 out of the initial 111 in the previous step. We discarded studies that did not present specific topics or keywords concerning the research. The distribution of the selected studies is in Table 5.

Table 5. Distribution by type and year of the studies identified in the stage 3.

	2015	2016	2017	2018	2019	2020	2021	
Non-indexed journal articles	2	0	2	1	0	1	0	6 (10%)
Indexed articles	6	5	11	12	9	10	1	54 (90%)
Total studies	8 (13%)	5 (8%)	13 (22%)	13 (22%)	9 (15%)	11 (18%)	1 (2%)	60

Source: Own elaboration.

Stage 4: Analysis and synthesis.

The selected and analyzed researchers answer the research questions defined in Table 1 of the methodology section.

The following table (Table 5) shows detailed information about each one of the items in chronological order by year of publication, title, type of publication, source (Scholar Google GS, Scopus SCO, Science Direct SCD, Web of Science WOS), research dimension as well as the area of interest defined in [14] and implemented in this research. For being of great importance and relevance to this:

- Origins, definitions, and concepts related to MOOCs.
- MOOC platforms were implemented successfully.
- MOOCs or specialized courses that have transformed secondary education.
- IA techniques implemented in MOOCs).

Table 6. Detailed information on the selected studies.

Year	Title	Document type	Source	Investigation dimension	Interest area
2015	Computational thinking and the new learning ecologies [2]	Article	GS	Implementation of computational thinking in teaching/programming, school learning.	4
2015	Developing a Computer Programming MOOC [8]	Article	SCD	Methodology analysis for the creation of MOOCs.	1
2015	Computer programming and robotics in basic education [17]	Article	GS	Implementation of computational thinking in teaching/programming, school learning.	2, 3
2015	A Purposeful MOOC to Alleviate Insufficient CS Education in Finnish Schools [18]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3
2015	Learning outside the classroom through MOOCs [19]	Article	SCD	Analysis of MOOC implementation in schools.	1, 2, 3
2015	A Global Snapshot of Computer Science Education in K-12 Schools [20]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	2,3
2015	Designing for deeper learning in a blended computer science course for middle school students [21]	Article	WOS	Analysis of MOOC implementation in schools.	2, 3
2015	Programming web-course analysis: How to introduce computer programming? [22]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2
2016	Code Yourself" and "A Programar": A bilingual MOOC for teaching computer science to teenagers [23]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3

2016	PC-01: Introduction to computational thinking: Educational technology in primary and secondary education [24]	Article	GS	Implementation of computational thinking in teaching/programming, school learning.	1, 2, 3
2016	A Review of Models for Introducing Computational Thinking, Computer Science and Computing in K–12 Education [25]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	2
2016	The impact of STEM experiences on student self-efficacy in computational thinking [26]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	1
2016	Towards an Innovative Computer Science & Technology Curriculum in UAE Public Schools System [27]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	1
2017	Research challenges in accessible MOOCs: a systematic literature review 2008–2016 [9]	State of the art	SCO	State of the art of MOOC.	1
2017	Can MOOCs Support Secondary Education in Computer Science? [28]	Article	GS	Analysis of MOOC implementation in schools.	1, 2, 3
2017	Computational thinking in Colombian schools: international collaboration on innovation in education[29]	Article	GS	Implementation of computational thinking in teaching/programming, school learning.	1
2017	The teaching of computer programming in Primary Education: current situation, analysis and presentation of classroom experiences in Spain [30]	Thesis	GS	Implementation of computational thinking in teaching/programming, school learning.	2
2017	Adoption of Computer Programming Exercises for Automatic Assessment [31]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	3
2017	Teaching Software Engineering Principles to K-12 Students: A MOOC on Scratch [32]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3
2017	Teaching CS to CS teachers: Addressing the need for advanced content in K-12 professional development [33]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	3
2017	Computational thinking as an emergent learning trajectory of mathematics [34]	Article	SCO	Analysis of MOOC implementation in schools.	2, 3

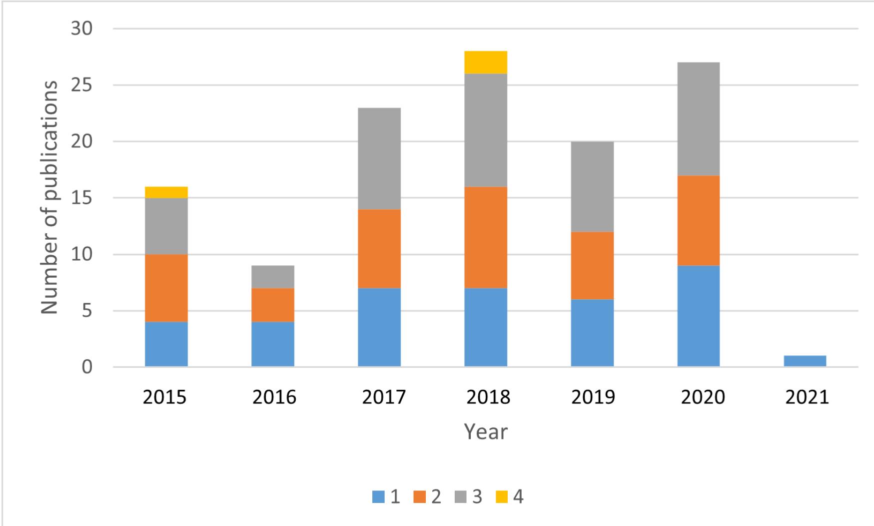
2017	Educating Computer Science Educators Online - A Racket MOOC for Elementary Math Teachers of Finland [35]	Article	SCO	Analysis of MOOC implementation in schools.	2, 3
2017	A Tool for Introducing Computer Science with Automatic Formative Assessment [36]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	3
2017	Computational Thinking as Springboard for Learning Object-Oriented Programming in an Interactive MOOC [37]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3
2017	Different underlying motivations and abilities predict student versus teacher persistence in an online course [38]	Article	WOS	Implementation of computational thinking in teaching/programming, school learning.	1, 2, 3
2017	MOOC architecture model for computer programming courses [39]	State of the art	SCO	State of the art on MOOC.	1
2018	Survey on understanding the implications of MOOCs in engineering education [12]	Article	SCO	Evaluation of motivation in MOOC.	1, 2
2018	A thematic literature review of the implementation of MOOCS - 2008 to 2018 [11]	State of the art	SCO	State of the art on MOOC.	1
2018	Troubleshooters for tasks of introductory programming MOOCs [7]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3, 4
2018	¿Can programming really be for everyone? Analysis of the experience within a MOOC. [40]	Article	GS	Analysis of MOOC implementation in schools.	1, 2
2018	Incorporating Computational Thinking in the Classrooms of Puerto Rico: How a MOOC Served as an Outreach and Recruitment Tool for Computer Science Education [41]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3
2018	Teaching Software Engineering in K-12 Education: A Systematic Mapping Study [42]	State of the art	SCO	State of the art on MOOC.	1, 2, 3
2018	Supporting Computational Thinking Development in K-6 [43]	Article	SCO	Analysis of MOOC implementation in schools.	2, 3
2018	Improving Assessment of Computational Thinking Through a Comprehensive Framework [44]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	3

2018	Code ABC MOOC for math teachers [45]	Article	SCO	Analysis of MOOC implementation in schools.	2, 3
2018	Computational Thinking and Online Learning: A Systematic Literature Review [46]	State of the art	SCO	State of the art on MOOC.	3
2018	A Free-Choice Social Learning Network for Computational Thinking [47]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	3, 4
2018	Computational Thinking in pre-university Blended Learning classrooms [48]	Article	SCD	Implementation of computational thinking in teaching/programming, school learning.	2, 3
2018	MOOC: Computational thinking (applied) for teachers [49]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3
2019	A blended learning course for playfully teaching programming concepts to school teachers [50]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	1, 2, 3
2019	What motivates enrolment in programming MOOCs? [51]	Article	WOS	Evaluation of motivation in MOOC.	1
2019	Evaluating Digital Worksheets with Interactive Programming Exercises for K-12 Education [52]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3
2019	Integrating Professional Tools in Programming Education with MOOCs [53]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3
2019	Preparing K-12 Teachers to Inspire Future Coders: It Doesn't Have to be Complex [54]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	2, 3
2019	Exploring the Role of Facebook as Collaboration Platform in a K-12 MOOC [55]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3
2019	MOOCs in Secondary Education - Experiments and Observations from German Classrooms [56]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3
2019	Computational thinking for preservice teachers in Thailand: A confirmatory factor analysis [57]	Article	GS	Implementation of computational thinking in teaching/programming, school learning.	3
2019	Early Programming Education and Career Orientation: The Effects of Gender, Self-Efficacy, Motivation and Stereotypes [58]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	3

2020	MOOCs: Origins, Concept and Didactic Applications: A Systematic Review of the Literature (2012–2019) [14]	State of the art	GS	State of the art on MOOC.	1, 2, 3
2020	Research trends in K-12 MOOCs: A review of the published literature [15]	State of the art	SCO	State of the art on MOOC.	1, 2, 3
2020	Recommender Systems for MOOCs: A Systematic Literature Survey (January 1, 2012 - July 12, 2019) [13]	State of the art	GS	State of the art on MOOC.	1, 3
2020	A computer programming hybrid MOOC for Greek secondary education [5]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3
2020	Modelo y lecciones aprendidas del proceso de creación de MOOCs para enseñar a programar [6]	Article	GS	Analysis of MOOC implementation in schools.	1, 2, 3
2020	Building a Community of STEM Educators in Nigeria Using the TeachAKid2Code Program [59]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	1, 2, 3
2020	Computational thinking and assignment resubmission predict persistence in a computer science MOOC [60]	Article	SCO	Analysis of MOOC implementation in schools.	1, 2, 3
2020	EarSketch: An authentic, STEAM-based approach to computing education [61]	Article	SCO	Analysis of MOOC implementation in schools.	2, 3
2020	Education in programming and mathematical learning: Functionality of a programming language in educational processes [62]	Article	SCO	Implementation of computational thinking in teaching/programming, school learning.	3
2020	Programming MOOCs–different learners and different motivation [63]	Article	SCO	Evaluation of motivation in MOOC.	1, 2
2020	Analyzing learners' engagement and behavior in MOOCs on programming with the Codeboard IDE [64]	Article	SCO	Programming MOOC with the use of external tools.	1, 3
2021	Self-Regulated Learning in Massive Online Open Courses: A State-of-the-Art Review [16]	State of the art	GS	State of the art on MOOC.	1

Source: Own elaboration.

Fig. 2 Distribution over time in the areas of interest in MOOC K-12 or STEAM defined in this research.



Source: Own elaboration.

Figure 2 shows the trend in the areas of interest researched over time. The year 2018 shows an increase in research corresponding to K-12 or STEAM MOOCs, focusing on MOOC-type courses or specialized courses in secondary education computer science and specifying the successfully implemented platforms. Furthermore, it is among the few years that present evidence of AI implementation in MOOCs for secondary education.

Discussion

The results obtained in this systematic review have shown that the number of studies related between MOOCs and K-12, STEM, or STEAM is significant. Some of them have found relevant information on MOOCs and K-12 education, STEAM, and computational thinking aimed at teachers and students. This section provides answers to each of the questions defined in the methodology, based on the main findings.

Status of relevant research related to MOOC K-12 or STEAM.

This section establishes the answer to the first research question, where we identified K-12 or STEAM courses implemented in secondary education through MOOC-type platforms. We considered the conceptual information corresponding to MOOCs, history, types of platforms cited in some studies.

Of the 60 studies selected, 38 provide conceptual information on MOOCs, i.e., 63% of the studies show a general introduction to the type of platforms, basic concepts, history, and complementary information. It is relevant to mention the most significant findings in this field. Typically, studies start with a brief introduction to open learning as a critical factor to address the multiple challenges continuously presented worldwide, where information technologies have transformed education, provoking new ways of learning and new ways of offering education [8]. In this context, MOOCs have emerged as free online courses, easily accessible, to teach a global audience [3], in the case of this research to the early teaching of programming.

The definition of the MOOC concept is coined by [3] as follows:

“A MOOC is an online course with the option of open registration, a publicly shared curriculum, and expected outcomes. MOOCs integrate social networking, accessible online resources, and are facilitated by professional leaders in the field of study” [65].

The MOOC concept was known since 2008 and addressed by [14] through the University of Manitoba in Canada, which introduced the origin of the MOOC concept, alluding to "Connectivism and Connective Knowledge."

Subsequently, the idea became relevant in 2011 with another Stanford University course on Artificial Intelligence, creating several MOOCs, mainly focused on educational technologies [64].

This includes various universities, organizations, and entrepreneurship companies that offer MOOCs, including leading universities such as the Massachusetts Institute of Technology (MIT), Harvard University, Stanford University, University of Edinburgh, and platforms designed as Coursera, edX, FutureLearn, Udemy, among others. The first, associated with Stanford, Princeton, and other universities, the second, founded by MIT and Harvard, the third was based by The Open University and, Udemy, which has been targeted for adult professionals and businesses [3].

Regarding the most significant characteristics of MOOCs, such as their open access, offered through the Internet and freely available on a massive scale, they also have a series of resources, start and end dates, and evaluation methods [3, 6, 14].

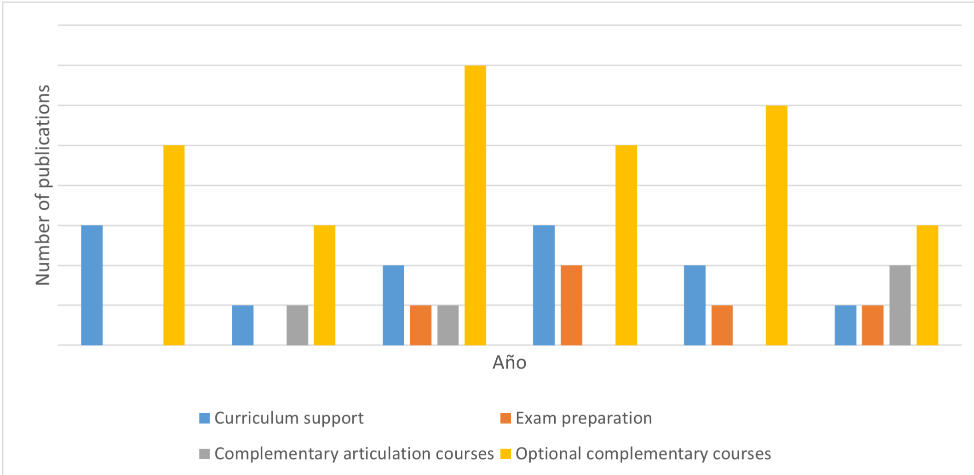
Regarding the characteristics of MOOCs aimed at K-12 teachers and students, studies have appeared in the literature. Since 2013 [32, 42, 52, 54], there have been studies on interaction in K-12 MOOCs and their benefits, as to the great potential in the enrichment of learning opportunities and the adaptation of pedagogical models according to this population [15].

On the other hand, we mentioned different forms of implementation, (I1) Support to curriculum that includes computational thinking, (I2) Preparation for tertiary education exams, (I3) Complementary courses for articulation process, and (I4) Complementary approaches offered optionally in secondary education. Table 7 specifies the jobs related to each of the K-12 or STEAM MOOC implementation forms.

The 48% of the studies correspond to the implementation of K-12, STEAM, computational thinking in primary and secondary through an education curriculum. There are curriculum regulations or subjects with logical development and programming for problem-solving in various countries and institutions.

In the second place, with 20%, studies related to preparation for tertiary education, where MOOCs in K-12 education are cited, have had a positive result in the professional development of students and teachers and preparation for national exams for access to higher education.

Fig. 3 Distribution of the different types of implementations between 2015 and 2021.



Source: Own elaboration.

Table 7. Selected studies with implementation of K-12 or STEAM MOOCs.

Area of interest	2015	2016	2017	2018	2019	2020
Curriculum support that includes computational thinking (I1)	[2, 17, 19]	[25]	[28, 30]	[43, 45, 49]	[67, 53]	[60]
Tertiary education exam preparation (I2)	NA	NA	[37]	[7, 48]	[54]	[5]
Complementary courses for articulation processes (I3)	NA	[24]	[29]	NA	NA	[58, 63]
Complementary courses offered as an option in secondary education (I4)	[8, 18, 20, 21, 22]	[3, 6, 26]	[31, 32, 33, 34, 35, 36, 38]	[40, 41, 44, 46, 47]	[50, 51, 52, 55, 56, 57]	[59, 61, 62]

Source: Own elaboration.

Finally, in small percentages of 8% and 7%, there are studies related to complementary courses for articulation processes and complementary courses offered optionally in secondary education, respectively. Of the 60 articles selected, 10 (17%) do not specify the type of implementation since they show the systematic literature review of MOOCs, MOOC K-12 or STEAM, methodologies for MOOC development, among others.

K-12 or STEAM MOOC platforms successfully implemented in secondary education

Many studies provide information on the success of K-12 or STEAM MOOCs; 39 (65%) of the 60 studies selected delved into the success or limitations that have arisen in the implementation in secondary education.

Among the results found, it is paramount to include computational thinking in primary and secondary education, as is the case of the K-12 curriculum [28], which incorporates the fundamental concepts and programming languages in education. Or also implemented individually as PC-01 as in studies [24, 29, 48]; where students successfully achieved the academic objectives established in the project, expressing a feeling of satisfaction with participation in the course with motivating activities and creative interests, establishing these implementations both in high schools in the Dominican Republic and Colombia.

For the methodology of this type of courses, video lessons, interactive tests with Scratch, evaluate, be evaluated, and participate in forums were used; this was a project supported by Renata and the University of the Basque Country. Another successful platform is the Core.org MOOC, which is part of an American nongovernmental organization with the same name, which has more significant influence worldwide with the promotion of programming in school, has the initiative that all students should learn to program, applying K-12 [1, 30].

Within its curriculum they include, formulating problems to be solved with the computer, organizing and analyzing logically, representing data with models, automating solutions with algorithmic thinking, identifying, analyzing, and implementing possible solutions, generalizing and transferring the solution. This knowledge is transversal in the educational systems, and the premise is that all students should learn to program. Both the K-12 curriculum and the Q2L curriculum [2] have been implemented in this same country. They design games and thinking systems that require computational thinking, using the Gamestar Mechanic software to develop video games.

Given the deficit of engineers in Europe, a program called" Opening up Education" was created to teach programming to children and young people and thus raise students' interest in engineering. From this project, others emerged, such as the Autonomous Community of Madrid that incorporated the subject" Technology, programming and robotics" in high school, established by a decree in 2015 [2]. It is essential to train the teachers who implement the courses; this training is carried out with the support of Telefonica through a MOOC. Likewise, the Universidad Oberta de Catalunya [40] promotes activities to develop computational thinking, with programming to society, through a MOOC of introduction to programming with a learning-by-doing methodology for participants of all ages, this methodology emphasizes practice over theory. More than 2,000 people have participated in classroom training in the clubs and online courses. By converting thought into objects through algorithms, data structure, and procedures, personal knowledge is shared with others, making computational thinking participatory.

On the other hand, in the United Kingdom, the educational reform law was created, which incorporates three core subjects: mathematics, science, and English, and additionally, the subject Computing, in which Computer Science is recognized as a rigorous academic discipline for the careers of the future, it is compulsory in primary and secondary schools [2]. It covers algorithms, data structures, and computer programming. The teaching is divided into stages KS1 to KS4 (from 5 to 16 years old). There are some criticisms from teachers for not carrying out prior training to implement this type of project. Therefore, recommended for learning assessment technologies such as online courses, for example, the MOOC entitled Cambridge GCSE Computing Online, designed to support the teaching and learning of Computing in schools.

Another pioneer country in implementing computer programming and robotics in basic education curricula is Estonia [30]. They included the introduction of programming in its schools, starting in primary education, through the ProgeTiger program financed by the government and created in 1997, enabling schools to have a broadband connection, providing teacher training in the use of TIC.

They also allow virtual learning environments to develop didactic material. Since 2012, pilot projects have been initiated at different educational levels. The teachers select the programs to be used in the subjects; they can also incorporate in their curriculum subjects from other technological areas, the objective being that the skills provided to the students in TIC support the digital progress of Estonia and are a source of prosperity for the country [17]. One of the problems that have arisen is the availability of programming teachers; for this reason, the most advanced students have been asked to teach MOOC-type courses titled to instruct thousands of young people in secondary education.

Similarly, in Finland, the University of Helsinki prepared a CS1 MOOC for high school students as a first pilot plan. They approach computer-related topics to motivate participation in more courses and even careers [64]. As mentioned above, one of the major problems is the preparation of teachers in programming, for this reason; in Finnish schools, mathematics teachers were prepared through a programming MOOC to support Finnish education; through workshops, the ABC MOOC, and the Racket MOOC [18, 28, 35, 45]. This type of strategy has been very successful in regulating programming in the subject of mathematics in Finland.

In the case of Greece [4], they decided to create a programming MOOC through the blended learning approach with example and problem-solving. Also, they made PROG15 [5, 54], focusing on how this kind can help prepare for the national tertiary education exams.

The students enrolled belong to different Greek vocational schools, and several teachers also enrolled in the course. The course was a great success, as 2,382 students took part in the Greek national computer programming exams, i.e., 12.2% of the population of secondary school students.

Similarly, Uruguay and Edinburgh created a MOOC [3, 6], included in the primary and secondary school curricula on the formal study of computer skills and competencies. In addition, the article presents guidelines for developing a MOOC. This massive open online course, with access through the internet for free to many people, consists of videos (explanations or tutorials), texts, images, forums, and evaluations.

The University of Edinburgh offers a development model, specifically from Coursera, aimed at young people aged 12 to 17 without programming skills for five weeks, obtained first with national participation. Subsequently, they participated in 117 countries; for this reason, the programming MOOC was developed in collaboration by the teams of the universities mentioned, resulting in a design with bilingual delivery.

In Germany, implemented the first MOOC with more than 7000 high school participants was successful in learning the Python programming language. The success and performance rate of the MOOC MINTEC has been positive, especially for participants with previous skills or talents, the aptitude for teamwork, and how the students perceive the peer review evaluation method as a suitable method of assessment. Likewise, there are different European and intercontinental strategies for learning to program in children and young people [30], as described in table 8.

In Spain there are several initiatives such as "Programamos Community, Young Programmers Club in Valladolid", Citilab in Barcelona and Complubot in Madrid.

This review provides an answer to the second research question: Which MOOC K-12 or STEAM platforms were successfully implemented in secondary education.

Other countries implemented a great variety of strategies such as MOOCs and how these benefit children and young people with the learning of programming. Thus, more curriculums include teaching programming in primary and secondary education, with formal studies of computational skills and competencies.

Transformation of education with the implementation of MOOC K-12 or STEAM platforms

This section answered the third research question raised to determine which case studies have positively transformed secondary education. It identifies the potential, positive experiences, and benefits of MOOCs for students and teachers. Most of the reflections on K-12 or STEAM MOOCs emphasize the implementation of computational thinking and programming and show the potential of MOOCs in learning experiences for teachers and students. Similarly, some studies present the difficulties in implementing such MOOCs in a classroom as a support strategy, preparation for tertiary education, or modality of articulation with higher education. Given that only 20% of related studies have directly incorporated computer science curriculum or subject matter in K-12 education by educational regulation in some countries, it is necessary to study the research done as supplemental courses offered as electives or as course work in secondary education.

Table 8. Strategies or projects for teaching programming to children and young people.

Name of the strategy	Type	Description	Platform type
CoderDojo	Community	For kids with Scratch, learning programming languages and robotics with 675 clubs in 54 countries.	MOOC.
Code's Cool	Community	It is an open and informal peer-to-peer programming community. It has an event organized by the European Commission every year, called CodeWeek.it. The two organizations collaborate with the creation of a MOOC.	Collaboration in the creation of a MOOC.
CS Unplugged	Didactic material	It is a collection that aims to teach computer science without the need for a computer.	Didactic material incorporated in MOOCs.
Code.org	Non-profit organization	With the Hour of Code project, they seek to increase interest and participation in computer science in schools. They advise the U.S. government in the design of new educational policies and have recognized companies such as Apple and Microsoft that have joined the Code.org movement.	MOOC.
CS First	Syllabus	It is an initiative of Google and Microsoft, Google, and Intellect, which collaborate with the Computer Arts Society (CAS) belonging to the British Computer Society.	MOOC.
Codeacademy	Interactive Platform	It is a free online platform that offers free programming courses in different programming languages that offers badges or medals for completing the exercises.	Interactive platform created by a MOOC company.
Code school	Platform	It is a platform that offers programming courses by managing practical lessons in which they manage incentives through prize draws. It is now called Pluralsight.	MOOC.
Scratch	Application	It was developed by MIT and allows learning programming in a graphical and very intuitive way.	Collaboration in the creation of MOOCs.

Source: Own elaboration.

In general, many of the studies cited [15, 19, 28] show that MOOCs have the potential to be part of the positive influence in secondary education, presenting a reduction in dropout and support students in the preparation of their university studies. Among these is the preparation for future learning assessments with the MOOC FACT [21] for secondary education, hosted on OpenEdX; in which the outcome of the test is evidenced as promising, as well as positive indicators in the computer science curriculum, which triggered the curiosity to learn more in students.

Similarly, different strategies have been implemented in Greece to transform secondary education through programming MOOCs called PROG15 and PROG16 with assistance from social networks such as Facebook [5, 54]. In the first version of the course, a reception of 12.2% of participants was achieved, according to the total number of students belonging to the different Greek vocational schools, thus achieving that more students take the computer programming exams for the national tertiary education exams. On the other hand, it is interesting how using social networks, the number of students increased, more students enrolled and participated more frequently through publications and participation in Facebook. Is this a more attractive place for interaction and collaboration of the course? In this sense, the Facebook groups presented a benefit, transformation, and positive comments complementing MOOC PROG15 and PROG16. In Germany, created systems called MOOCs MintEC [56] to support secondary education classrooms. Demonstrating that it is possible to incorporate the MOOC in the school where teachers have successfully used these courses, both in school, home, and extracurricular environments, as a key and enriching element in education.

In the study [28], many of the teachers are skeptical about the integration of digital media, which is why other research seeks to train not only students but also teachers in computational thinking as in the study [18], where teachers in Finland challenged with the regulation of programming in the subject of mathematics. Many of the teachers do not know about organizing and incorporating into their school classes around programming tasks. Still, a programming MOOC supports the Finnish education system and teacher training with up-to-date teaching materials. This programming MOOC has provided opportunities for schools that lack strategies for implementing programming in the curriculum in Finland.

In the researches [34, 35, 45], the MOOC ABC was implemented for students and to support mathematics teachers in preparing the programming in their classes. The result is encouraging since this type of resource has allowed the feasible preparation of teachers in teaching programming to take it to the classroom subsequently.

Another paper [43] discusses the CSER digital technologies course curriculum in MOOC for the K-6 context, aligning key concepts from formal curriculum statements, teacher evaluations, discussion of key challenges and concerns. A community was created to share ideas of lessons learned and discuss critical challenges identified by teachers within the first-course cohort. So, by using programming MOOCs for teachers as well, favorable results are obtained. Teachers feel better prepared when planning upcoming lessons and better understand what high school students require [52].

Another vital transformation in secondary education occurred in studies [3, 6, 49]. Students from the first MOOC cohort in programming with better academic performance support new students through forums, providing students with a solid foundation in computer science that inspires them to develop their programming skills further. ”The MOOCs developed aim to contribute to logical and computational skills in young people, skills in demand today” [6].

Not only are programming MOOCs are found in the literature, and MOOCs include other essential competencies such as Software Engineering [32, 42]. The results demonstrate the feasibility and positive contribution of having Software Engineering education in K-12; it was also possible to take the course at home with a guardian or parent, being a transforming factor of education by involving parents in this experiment. The results were positive; obtaining high grades.

One of the main tasks challenging to control and transform in secondary education is the self-regulation of learning since most young people do not have habits for time management in online education as in the case of MOOCs. Therefore, a literature review on the current state of self-regulated learning support in MOOCs is cited [16], using technologies based on psychological models for 2010 and 2020. It is also essential since it allows working on student dropout that mainly occurs in MOOCs due to the lack of self-regulation in learning.

Therefore, K-12 MOOCs have transformed secondary education through a blended classroom model, where the traditional classroom is worked with the support of MOOC-type tools or platforms to motivate students, as is the case of studies [41, 62, 67].

Artificial intelligence in K-12 or STEAM MOOC platforms for secondary education

This section answers the fourth research question to determine which case studies have implemented artificial intelligence in programming MOOCs. Only 3 (5%) of the 60 projects present the incorporation of artificial intelligence in programming MOOCs for secondary education.

In the project [7], implemented decision trees are used in education, and features are visualized to be easily understandable to those unfamiliar with programming in machine learning. The preprocessed data is used to train a decision tree model to visualize the skill structures that affect the learner in program development.

On the Open School platform [47], online learning resources and programming exercises can compile and execute through intelligent tutoring and assessment, including automatically generating feedback to help students improve their learning outcomes.

Another study is Alibi [36]. It is a Chatbot designed to support massive open online courses, including automatic formative assessment capabilities with immediate feedback at the homework level. The platform introduces high school students to computer science concepts innovatively, using state-of-the-art natural language processing techniques.

As can be observed, there is no evidence in the selected studies of the explicit incorporation of IL techniques in programming MOOCs for secondary education. Very few specify the method they have implemented in summative and formative evaluation or assistance through Chatbot. In future work, it would be essential to develop tools that support the development of various tasks in MOOC-type courses, and even more so for high school students, who need more assistance as they are first-time learners of programming.

Conclusions

Advances in the development of open and online resources have become a transformative trend in traditional education, as incorporated into university curricula and the potential of K-12 MOOCs in secondary education. This systematic literature review article on the implementation of K-12, STEAM, or programming in secondary education through MOOCs for teachers and students, of which 60 relevant studies published between 2015 and 2021 are related.

This research reveals how they have revolutionized and transformed secondary education for both students and teachers. Most MOOCs for secondary education have been implemented specifically to meet the needs of secondary education teachers and students to enhance and support the curriculum by incorporating computational thinking and programming. Therefore, the existence of a mixed approach is notable with the implementation of the MOOC.

On the other hand, studies such as [4, 5, 54] have shown how the implementation of programming MOOCs in Greece has increased the number of students who select computer programming as a fundamental axis in the national examination for higher education. In addition to using social networks to motivate students in the development of the course, however, there are still difficulties in increasing the completion rate of this type of course and the preparation of teachers who support them.

Regarding teacher preparation [34, 35, 45], countries such as Finland have obtained good results since before implementing programming in the subject of mathematics in secondary education; teachers were prepared with the necessary knowledge to teach classes using MOOC ABC. This type of initiative obtained favorable results implemented in other countries.

The review also identified few studies related to certain aspects of MOOCs, such as the automation or implementation of IA; only 3 of the selected studies addressed the implemented technique, which would allow automating some processes charged to the teacher, as they are massive courses. In the future, it is necessary to expand the research that implements this type of techniques.

Bibliographic References

1. Asociación de Maquinaria de Computación, Code.org, Asociación de Profesores de las Ciencias de la Computación, y Centro de Innovación Cibernética y Deepa Muralidhar de la Iniciativa Nacional de Matemáticas y Ciencias, «K – 12 Marco de las Ciencias de la Computación». 2016.
2. J. V. Berrocoso, M. R. F. Sánchez, y M. del C. G. Arroyo, «El pensamiento computacional y las nuevas ecologías del aprendizaje», *Rev. Educ. Distancia RED*, n.º 46, Art. n.º 46, oct. 2015, Accedido: sep. 23, 2020. [En línea]. Disponible en: https://revistas.um.es/red/article/view/240311
3. I. F. de Kereki y A. Manataki, «“Code Yourself” and “A Programar”: A bilingual MOOC for teaching computer science to teenagers», en *2016 IEEE Frontiers in Education Conference (FIE)*, oct. 2016, pp. 1-9. doi: 10.1109/FIE.2016.7757569.
4. F. Lazarinis, C. V. Karachristos, E. C. Stavropoulos, y V. S. Verykios, «A blended learning course for playfully teaching programming concepts to school teachers», *Educ. Inf. Technol.*, vol. 24, n.º 2, pp. 1237-1249, mar. 2019, doi: 10.1007/s10639-018-9823-2.
5. P. Koutsakas, C. Karagiannidis, P. Politis, y I. Karasavvidis, «A computer programming hybrid MOOC for Greek secondary education», *Smart Learn. Environ.*, vol. 7, n.º 1, 2020, doi: 10.1186/s40561-020-0114-1.
6. I. F. D. Kereki y A. Manataki, «Modelo y lecciones aprendidas del proceso de creación de MOOCs para enseñar a programar», en *Sexta Conferencia de Directores de Tecnología de Información, TICAL 2016 Gestión de las TICs para la Investigación y la Colaboración*, sep. 2016, pp. 171-184. Accedido: sep. 23, 2020. [En línea]. Disponible en: https://www.research.ed.ac.uk/portal/en/publications/modelo-y-lecciones-aprendidas-del-proceso-de-creacion-de-moocs-para-ensenar-a-programar(72f2a621-887d-402b-b129-9d651a02446c).html

7. M. Lepp *et al.*, «Troubleshooters for tasks of introductory programming MOOCs», *Int. Rev. Res. Open Distance Learn.*, vol. 19, n.º 4, pp. 56-75, 2018, doi: 10.19173/irrodl.v19i4.3639.

8. «Developing a Computer Programming MOOC», *Procedia Comput. Sci.*, vol. 65, pp. 182-191, ene. 2015, doi: 10.1016/j.procs.2015.09.107.

9. S. Sanchez-Gordon y S. Luján-Mora, «Research challenges in accessible MOOCs: a systematic literature review 2008–2016», *Univers. Access Inf. Soc.*, vol. 17, n.º 4, pp. 775-789, nov. 2018, doi: 10.1007/s10209-017-0531-2.

10. S. Sanchez-Gordon y S. Luján-Mora, «Research challenges in accessible MOOCs: a systematic literature review 2008–2016», *Univers. Access Inf. Soc.*, vol. 17, n.º 4, pp. 775-789, nov. 2018, doi: 10.1007/s10209-017-0531-2.

11. T. Prinsloo y A. Ainslie, «A THEMATIC LITERATURE REVIEW OF THE IMPLEMENTATION OF MOOCS - 2008 TO 2018», *Proc. 2018 AIS SIGED Int. Conf. Inf. Syst. Educ. Res.*, ene. 2018, [En línea]. Disponible en: https://aisel.aisnet.org/siged2018/34

12. J. Shailaja y S. Prathikantham, «Survey on understanding the implications of MOOCs in engineering education», 2018.

13. A. Khalid, K. Lundqvist, y A. Yates, «Recommender Systems for MOOCs: A Systematic Literature Survey (January 1, 2012 – July 12, 2019)», *Int. Rev. Res. Open Distrib. Learn.*, vol. 21, n.º 4, pp. 255-291, jun. 2020, doi: 10.19173/irrodl.v21i4.4643.

14. F. J. Palacios Hidalgo, C. A. Huertas Abril, y M. ª E. Gómez Parra, «MOOCs: Origins, Concept and Didactic Applications: A Systematic Review of the Literature (2012–2019)», *Technol. Knowl. Learn.*, vol. 25, n.º 4, pp. 853-879, dic. 2020, doi: 10.1007/s10758-019-09433-6.

15. P. Koutsakas, G. Chorozidis, A. Karamatsouki, y C. Karagiannidis, «Research trends in K-12 MOOCs: A review of the published literature», *Int. Rev. Res. Open Distance Learn.*, vol. 21, n.º 3, pp. 285-303, 2020.

16. J. Cerón *et al.*, «Self-Regulated Learning in Massive Online Open Courses: A State-of-the-Art Review», *IEEE Access*, vol. 9, pp. 511-528, 2021, doi: 10.1109/ACCESS.2020.3045913.

17. J. M. C. Delgado, «Programación informática y robótica en la enseñanza básica», *Av. En Supervisión Educ.*, n.º 24, Art. n.º 24, dic. 2015, doi: 10.23824/ase.v0i24.17.

18. J. Kurhila y A. Vihavainen, «A purposeful MOOC to alleviate insufficient cs education in Finnish schools», *ACM Trans. Comput. Educ.*, vol. 15, n.º 2, 2015, doi: 10.1145/2716314.

19. T. Brahimi y A. Sarirete, «Learning outside the classroom through MOOCs», *Comput. Hum. Behav.*, vol. 51, pp. 604-609, oct. 2015, doi: 10.1016/j.chb.2015.03.013.

20. P. Hubwieser *et al.*, «A global snapshot of computer science education in K-12 schools», 2015, pp. 65-83. doi: 10.1145/2858796.2858799.

21. S. Grover, R. Pea, y S. Cooper, «Designing for deeper learning in a blended computer science course for middle school students», *Comput. Sci. Educ.*, vol. 25, n.º 2, pp. 199-237, 2015, doi: 10.1080/08993408.2015.1033142.

22. R. Da Silva Ribeiro, L. De Oliveira Brandao, T. V. M. Faria, y A. A. F. Brandao, «Programming web-course analysis: How to introduce computer programming?», 2015, vol. 2015-February, n.º February. doi: 10.1109/FIE.2014.7044140.

23. I. F. De Kereki y A. Manataki, «"Code yourself" and "a programar": A bilingual MOOC for teaching computer science to teenagers», 2016, vol. 2016-November. doi: 10.1109/FIE.2016.7757569.

24. X. Basogain, M. A. Olabe, J. C. Olabe, R. Ramírez, M. Del Rosario, y J. Garcia, «PC-01: Introduction to computational thinking: Educational technology in primary and secondary education», en *2016 International Symposium on Computers in Education (SIIE)*, sep. 2016, pp. 1-5. doi: 10.1109/SIIE.2016.7751816.

25. F. Heintz, L. Mannila, y T. Farnqvist, «A review of models for introducing computational thinking, computer science and computing in K-12 education», 2016, vol. 2016-November. doi: 10.1109/FIE.2016.7757410.

26. J. L. Weese, R. Feldhausen, y N. H. Bean, «The impact of STEM experiences on student self-efficacy in computational thinking», 2016, vol. 2016-June.

27. J. Al-Karaki *et al.*, «Towards an innovative computer science & technology curriculum in UAE public schools system», 2016, vol. 10-13-April-2016, pp. 883-891. doi: 10.1109/EDUCON.2016.7474656.

28. C. T. Grella, T. Staubitz, R. Teusner, y C. Meinel, «Can MOOCs Support Secondary Education in Computer Science?», en *Interactive Collaborative Learning*, Cham, 2017, pp. 478-493. doi: 10.1007/978-3-319-50337-0_45.

29. X. Basogain, Olabe, J. C. Olabe, M. Rico, L. Rodríguez, y M. Amortegui, «Pensamiento computacional en las escuelas de Colombia: colaboración internacional de innovación en la educación», *undefined*, 2017. /paper/Pensamiento-computacional-en-las-escuelas-de-de-en-Basogain-Olabe/92f3ece359fae52e3d220653a017fa438aabe39c (accedido sep. 23, 2020).

30. H. Arranz de la Source, «La enseñanza de la programación informática en Educación Primaria: situación actual, análisis y presentación de experiencias de aula en España», dic. 2017, Accedido: sep. 29, 2020. [En línea]. Disponible en: http://dspace.uib.es/xmlui/handle/11201/3892

31. Y. T. Yu, C. M. Tang, C. K. Poon, y J. W. Keung, «Adoption of computer programming exercises for automatic assessment ’ issues and caution», 2017, pp. 555-564.

32. F. Hermans y E. Aivaloglou, «Teaching software engineering principles to K-12 students: A MOOC on scratch», 2017, pp. 13-22. doi: 10.1109/ICSE-SEET.2017.13.

33. D. Leyzberg y C. Moretti, «Teaching CS to CS teachers: Addressing the need for advanced content in K-12 professional development», 2017, pp. 369-374. doi: 10.1145/3017680.3017798.

34. P. Niemel, T. Partanen, M. Harsu, L. Leppänen, y P. Ihantola, «Computational thinking as an emergent learning trajectory of mathematics», 2017, pp. 70-79. doi: 10.1145/3141880.3141885.

35. T. Partanen, P. Niemelä, L. Mannila, y T. Poranen, «Educating computer science educators online a racket MOOC for elementary math teachers of Finland», 2017, vol. 1, pp. 47-58. doi: 10.5220/0006257800470058.

36. L. Benotti, M. C. Martínez, y F. Schapachnik, «A Tool for Introducing Computer Science with Automatic Formative Assessment», *IEEE Trans. Learn. Technol.*, vol. 11, n.º 2, pp. 179-192, 2018, doi: 10.1109/TLT.2017.2682084.

37. J. Krugel y P. Hubwieser, «Computational thinking as springboard for learning object-oriented programming in an interactive MOOC», 2017, pp. 1709-1712. doi: 10.1109/EDUCON.2017.7943079.

38. R. M. Higashi, C. D. Schunn, y J. B. Flot, «Different underlying motivations and abilities predict student versus teacher persistence in an online course», *Etrd-Educ. Technol. Res. Dev.*, vol. 65, n.º 6, pp. 1471-1493, dic. 2017, doi: 10.1007/s11423-017-9528-z.

39. B. Yulianto, H. Prabowo, R. Kosala, y M. Hapsara, «MOOC architecture model for computer programming courses», 2017, pp. 35-40. doi: 10.1109/ICIMTech.2016.7930298.

40. D. Bañeres, C. Casado, A. Ornellas, E. Planas, J. Prieto, y M. Serra, «¿Realmente la programación puede ser para todos? Análisis de la experiencia dentro de un MOOC», *Actas Las Jorn. Sobre Enseñ. Univ. Informática*, vol. 3, n.º 0, Art. n.º 0, jun. 2018.
41. P. O. Franco, J. Carroll-Miranda, M. L. Delgado, E. G. López, y G. R. Gómez, «Incorporating Computational Thinking in the classrooms of Puerto Rico: How a MOOC served as an outreach and recruitment tool for Computer Science Education», 2018, vol. 2018-January, pp. 296-301. doi: 10.1145/3159450.3159544.
42. F. C. Pinheiro, C. G. von Wangenheim, y R. M. Filho, «Teaching Software Engineering in K-12 education: A systematic mapping study», *Inform. Educ.*, vol. 17, n.º 2, pp. 167-206, 2018, doi: 10.15388/infedu.2018.10.
43. K. Falkner, R. Vivian, y N. Falkner, «Supporting Computational Thinking Development in K-6», 2018, pp. 126-133. doi: 10.1109/LaTICE.2018.00031.
44. D. Basso, I. Fronza, A. Colombi, y C. Pahl, «Improving assessment of computational thinking through a comprehensive framework», presentado en ACM International Conference Proceeding Series, 2018. doi: 10.1145/3279720.3279735.
45. P. Niemelä, T. Partanen, L. Mannila, T. Poranen, y H.-M. Järvinen, «Code ABC MOOC for math teachers», *Commun. Comput. Inf. Sci.*, vol. 865, pp. 66-96, 2018, doi: 10.1007/978-3-319-94640-5_4.
46. C. Kirwan, E. Costello, y E. Donlon, «Computational thinking and online learning: A systematic literature review», 2018, vol. 2018-November, pp. 650-657.
47. H. Jamil, «A free-Choice social learning network for computational thinking», 2018, pp. 69-71. doi: 10.1109/ICALT.2018.00023.
48. X. Basogain, M. Á. Olabe, J. C. Olabe, y M. J. Rico, «Computational Thinking in pre-university Blended Learning classrooms», *Comput. Hum. Behav.*, vol. 80, pp. 412-419, mar. 2018, doi: 10.1016/j.chb.2017.04.058.
49. I. F. De Kereki, «MOOC: Computational thinking (applied) for teachers», 2018, vol. 2018-July. doi: 10.18687/LACCEI2018.1.1.26.
50. F. Lazarinis, C. V. Karachristos, E. C. Stavropoulos, y V. S. Verykios, «A blended learning course for playfully teaching programming concepts to school teachers», *Educ. Inf. Technol.*, vol. 24, n.º 2, pp. 1237-1249, 2019, doi: 10.1007/s10639-018-9823-2.
51. P. Luik *et al.*, «What motivates enrolment in programming MOOCs?», *Br. J. Educ. Technol.*, vol. 50, n.º 1, pp. 153-165, ene. 2019, doi: 10.1111/bjet.12600.
52. S. Serth, R. Teusner, J. Renz, y M. Uflacker, «Evaluating Digital Worksheets with Interactive Programming Exercises for K-12 Education», 2019, vol. 2019-October. doi: 10.1109/FIE43999.2019.9028680.
53. L. Carter y C. Crockett, «Preparing K-12 teachers to inspire future coders: It doesn't have to be complex», 2019, vol. 2018-October. doi: 10.1109/FIE.2018.8658762.
54. P. Koutsakas, E. Syritzidou, A. Karamatsouki, y C. Karagiannidis, «Exploring the role of facebook as collaboration platform in a K-12 MOOC», *Commun. Comput. Inf. Sci.*, vol. 993, pp. 31-48, 2019, doi: 10.1007/978-3-030-20954-4_3.
55. T. Staubitz, R. Teusner, y C. Meinel, «MOOCs in secondary education - Experiments and observations from German classrooms», 2019, vol. April-2019, pp. 173-182. doi: 10.1109/EDUCON.2019.8725138.
56. Aumgri y S. Petsangsri, «Computational thinking for preservice teachers in Thailand: A confirmatory factor analysis», *Espacios*, vol. 40, p. 12, sep. 2019.
57. E. Aivaloglou y F. Hermans, «Early programming education and career orientation: The effects of gender, self-efficacy, motivation and stereotypes», 2019, pp. 679-685. doi: 10.1145/3287324.3287358.

58. I. C. Emembolu, C. Emembolu, K. Umechukwu, A. Sulaiman, y O. Aderinwale, «Building a Community of STEM Educators in Nigeria Using the TeachAKid2Code Program», *Adv. Intell. Syst. Comput.*, vol. 1135 AISC, pp. 536-545, 2020, doi: 10.1007/978-3-030-40271-6_53.

59. C. Chen, G. Sonnert, P. M. Sadler, y D. J. Malan, «Computational thinking and assignment resubmission predict persistence in a computer science MOOC», *J. Comput. Assist. Learn.*, vol. 36, n.º 5, pp. 581-594, 2020, doi: 10.1111/jcal.12427.

60. R. Moore, D. Edwards, J. Freeman, B. Magerko, T. McKlin, y A. Xambo, «EarSketch: An authentic, STEAM-based approach to computing education», 2016, vol. 2016-June.

61. R. García-Perales y A. Palomares-Ruiz, «Education in programming and mathematical learning: Functionality of a programming language in educational processes», *Sustain. Switz.*, vol. 12, n.º 23, pp. 1-15, 2020, doi: 10.3390/su122310129.

62. P. Luik *et al.*, «Programming MOOCs–different learners and different motivation», *Int. J. Lifelong Educ.*, vol. 39, n.º 3, pp. 305-318, 2020, doi: 10.1080/02601370.2020.1780329.

63. J. M. Gallego-Romero, C. Alario-Hoyos, I. Estévez-Ayres, y C. Delgado Kloos, «Analyzing learners' engagement and behavior in MOOCs on programming with the Codeboard IDE», *Educ. Technol. Res. Dev.*, vol. 68, n.º 5, pp. 2505-2528, 2020, doi: 10.1007/s11423-020-09773-6.

64. J. Kurhila y A. Vihavainen, «A Purposeful MOOC to Alleviate Insufficient CS Education in Finnish Schools», *ACM Trans. Comput. Educ.*, vol. 15, n.º 2, p. 10:1-10:18, abr. 2015, doi: 10.1145/2716314.

65. A. McAuley, B. Stewart, G. Siemens, y D. Cormier, «The MOOC model for digital practice.»