Research Article

Received: 21 January 2022 / Accepted: 2 April 2022 / Published: 5 May 2022

Documentary Review about Computer Programming for the Development of Computer Thinking, Logical-Mathematical Skills, and Scientific Skills

Daniel F. Hillon-Salgado[1]

Wilmer A. Gómez Fierro[2]

Johan J. Molina-Mosquera[1]

[1]Electronic Engineering Department,
Faculty of Engineering, Surcolombiana University,
Neiva, Huila, Colombia
[2]Department of Degree in Natural Sciences,
Faculty of Education, Surcolombiana University,
Neiva, Huila, Colombia

DOI: https://doi.org/10.36941/ajis-2022-0064

Abstract

Programming in school spaces has been essential because the main objectives are the training and development of skills of those who use it. This way, it contributes to the student's activity from the professional field and in practical life. Therefore, in this article, a documentary review is carried out to inquire about the implementation of programming in school to contribute to the development of computational thinking, logical-mathematical, and scientific skills. The examination revealed essential research findings that consider programming an indispensable tool for the axes proposed in this work.

Keywords: *Programming, education, skills, thinking*

1. Introduction

In recent years, the need has emerged on the educational institutions, to incorporate information and Communication Technologies (ICT) into their professorships to allow students to develop skills and abilities in this competency. That enables them to learn a topic and take it in context to their social reality (Tejera et al., 2020).

The need for changes in the educational paradigm is a result of the desire to integrate schools into the current technological context, which requires new analysis schemes that capture the emerging trends (Tezanos, 2001). This new paradigm caused a new way of teaching and learning, which intervened in the roles of teachers and students (Berrocoso et al., 2015).

Therefore, new tools, such as computer programming, are being taught and learned within school contexts in a significant way. However, according to Monjelat et al. (2018), the idea of linking programming in schools is an idea that dates to the 20th century.

However, it declined because it was complex in the languages used and students perceived it as uninteresting or relevant to educational and social processes.

Today, the world of technology has permeated all sectors and has transformed people's reality. This phenomenon is a significant impetus in young people who perceive a necessary digital culture (Castellanos et al., 2017; Kirschner and De Bruyckere, 2017). Therefore, educational institutions have established that technology is integrated into the curriculum with programming, supported simultaneously with the other areas for transversality of knowledge and comprehensive training for students (Llorens, 2015; Rincón and Ávila, 2016).

In the same way, when computer programming is correlated in school curricula, it is sought to deepen the digital field and favor an active methodology. Which encourages the generation of new mental structures that help understand theories already conceived and new learning, promoting the analysis and solution of errors (Llorens, 2015; Olabe et al., 2015).

Due to all the above, in the following work, we have joined forces to carry out a bibliographic review to inquire about the implementation of programming in school to develop computational thinking, logical-mathematical skills, and scientific skills.

2. Methodological Aspects

The present document review research is a mix, of transversal, and retrospective types. Where the document review technique proposed by Flick (2004) is used. In this sense, a review of 36 articles from scientific journals between 2000 and 2021 is established and compiled in the databases Scielo, Redalyc, Web of Science, ERIC, Dialnet, as well as in other journals related to education.

For each reference, an Educational Analytical Summary (EAS) was constructed to extract the primary information related to the study, thus showing the synthesis, proposals, approaches, and highlighting its essential elements. Next, table 1 an example of EAS worked on in this investigation.

Table 1: Educational Analytical Summary (EAS) model used for the documentary analysis

Code EAS	A.1
Document type	Magazine article.
Magazine section	N.A.
Print Type	Digital.
Circulation level	Free access.
Access to the document	Mathematics education.
Qualification	Learning mathematics through the programming language R in Secondary Education.
Author	Briz Redón, Á., y Serrano Aroca, Á.
Place of work and position	Department of Applied and Technological Sciences, Faculty of Veterinary and Experimental Sciences, Catholic University de Valencia San Vicente Mártir, Valencia, Spain,
Publication	Briz Redón, Á., And Serrano Aroca, Á. (2018). Learning mathematics through the programming language R in Secondary Education. Mathematics Education, 30 (1), 133-162.
Keywords	Mathematics education; Computational thinking; Education. secondary; Programming language.
Synthesis	The article proposes that programming through computers will allow a significant advantage at an educational level. It can give students a greater capacity for logical reasoning, structured thinking, and imagination. Therefore, they propose a documentary review about mathematics teaching through programming in the first place. Followed by this, they suggest using the programming language R for students to develop polynomial equations. It was reported that the students reached the proposed objectives. However, some had complications due to the inherent complexity of the programming.
Sources	29 bibliographic references.

Objective	propose the use of one of the most popular programming languages of the moment, R, as a tool to deal with the content of the subject of Mathematics.
Trouble	Programming can be a tool that develops skills for understanding the content in mathematics, specifically the polynomial equations of Baldor's algebra.
Methodology	Worked with two groups, one of 15 students and the other of 18, for a total of 33 students. four sessions of Fifty minutes to each of the two groups, making eight sessions. The last two were dedicated to putting into practice, using this same language, the resolution of polynomial equations of integer coefficients. Finally, the students were presented with a survey with various questions concerning the sessions received.
Population	Thirty-three students between 14 and 15 years of age from an educational institution in Spain.
Conclusions	It is established that programming as a didactic tool allowed students to reinforce their knowledge in mathematics. However, the implementation of this methodology was not efficient in a certain percentage of the students due to difficulties typically associated with the initial learning of a programming language. It related, above all, with the use of its syntax.
Type of job	Research - Methodological innovation
Author of the EAS	

3. Development

3.1 Programming and Computational Thinking

Developing computational thinking in students is a trend that has become popular in recent years and is related to other skills such as logic, mathematics, criticism, among others (Narváez and Roig, 2015). One of the first authors to cite a concept referring to computational thinking was Jeannette Wing, who mentioned "computational thinking, is a set of skills that are useful for people in their social environment, and not only of a computational nature" (Wing, 2006).

However, this concept was improved and provided an approach. Wing (2009), mentions that:

> *"Computational thinking is a process that involves posing problem situations and their solution methods." Like this concept, González (2019), says that computing thinking must "formulate problems in a way that allows the use of a computer to solve them, organizing and logically analyzing data, as well as representing data through abstractions, automating solutions through algorithms."*

Along the same lines, some authors are more focused on academia, mentioning that computational thinking will be related to a methodology that focuses on implementing the conceptions of computer science to solve everyday problems, design systems, and perform routine tasks (Basogain et al., 2015). Similarly, in this article, Basogain and collaborators present concepts given by programming companies such as Scratch from MIT. Where it is defined:

> *"Programming allows students to begin to develop as computational thinkers: they learn basic concepts of computation and mathematics, and at the same time they also learn design strategies, problem-solving, and other forms of collaboration" (ScratchEd Team, 2015).*

In the same way, other authors propose computational thinking to an applicative context, such as Zhou (2006), who mentions that computational thinking emphasizes using the conceptions of computing to solve problems, such as those related to human behavior. In addition, he says that computational thinking allows us to foresee those mathematical calculations are not enough in the computer, but that it needs understanding in the brain. Therefore, this thinking is abstract and automatic (Liu et al., 2015).

As mentioned by the different authors, it is agreed that computational thinking is related to the abilities that a person can develop for solving problems according to their skills in handling computers. It is beyond a mechanistic ability and becoming a more dynamic process where it uses the abstraction of a programming language, which also increases the probability of solving social problems.

In this sense, computational thinking in school will allow the development of educational skills in students, as summarized in Table 2.

Table 2: Implementation trend of programming for the development of computational thinking

Trend	Authors
Use of video games for teaching	González, 2019; Kazimoglu et al., 2012; Lee y Ko, 201; González, 2019.
Creation of projects with a fixed theme	Taborda y Medina, 2014; Calao et al., 2015; Sáez y Cózar, 2017; Zapata et al., 2018; Alsina y Acosta, 2018; Martínez et al., 2016; Sáez y Cózar, 2017
Creation of environments to generate situations	Abril, 2016; Barr y Stephenson, 2011; García et al., 2016; Larkins y Harvey, 2010
Tool as troubleshooting	Román et al., 2015; Delgado et al., 2013

Abril (2016) establishes that "computational thinking will enable problem-solving and learn essential skills for all human beings, such as teamwork." Therefore, this thought can be applied to any reasoning, contributing to any area of knowledge (Barr and Stephenson, 2011; García et al., 2016). Learning programming at school enables educational innovation to develop problem-solving skills, a key characteristic of computational thinking (Román et al., 2015). In addition, this allows students to apply that knowledge and skills to a practical value in their daily lives (Delgado et al., 2013). This thinking is directed towards science students and all areas (Larkins and Harvey, 2010).

Despite this, computer programming by itself does not develop computational thinking skills. Instead, it should be complemented by learning tools. Such as computer games, which serve as models to introduce all computer programming concepts. As a result of the entertaining the student while developing computational thinking skills (Lee and Ko, 2011; Kazimoglu et al., 2012; González, 2019). Additionally, block application programming can be approached early since programming concepts help improve their skills (Taborda and Medina, 2014; Calao et al., 2015; Sáez and Cózar, 2017). Finally, using educational robots that playfully enable programming knowledge promotes the development of computational thinking (Martínez et al., 2016; Alsina and Acosta, 2018; Zapata et al., 2018).

Furthermore, these investigations, like the one conducted by González (2019), show that to develop an inclusive educational method for teaching programming and computational thinking in elementary school, the students need to build a classroom experience. Using programming, the game's construction allowed us to analyze behaviors and social differences.

In the field of block programming, Sáez and Cózar (2017), show in their research the impact of developing computational thinking through the creation of musical content by elementary school students through visual block programming, Being this methodology of significant effect for the understanding of computer science concepts, artistic development, creativity, and computational skills.

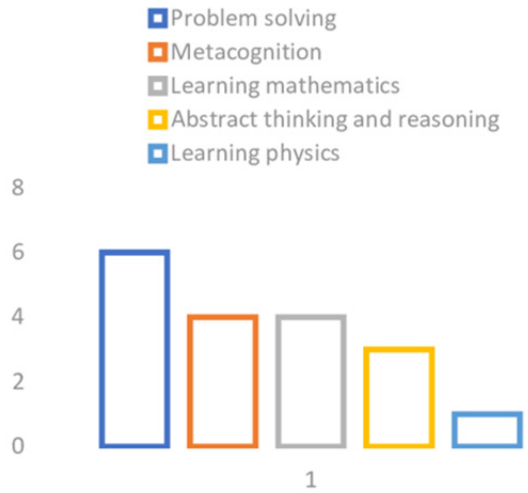
Finally, with an applicative approach to robotics, Zapata et al. (2018), developed an investigation using a robotic platform to acquire computational and mathematical skills in elementary school students and demonstrating in this way that robotics favors learning by discovery, offering the possibility of reinforcing theoretical concepts through experimentation and implementation from computing.

3.2 Logical-Math Skills

The ability of students to use numbers reflectively is related to expressions of mathematical intelligence. This type of intelligence shows positive reasoning, abstraction, and analysis of schematic sequences (García, 2010). However, some authors mention that programming constitutes a mathematical ability process, such as Saeli et al. (2011). Consequently, mathematics has become fundamental tools for solving problems and for the demonstration of daily events, so that, from the school environment, teachers must feel the need to teach it through technological resources, since these are a cultural part of the present generations (Rozo and Pérez, 2014; Williner, 2014).

Programming and its relationship to pedagogical processes, such as mathematics teaching, have been proposed since the 21st century. The articles found, such as those by Papert (1972), establish a broader vision of a system educational in which technology is used to make individuals think. Among these, it proposes orientation axes to use programming as an instrument to teach mathematical equations.

This educational vision of programming has allowed the authors to consider it a teaching tool in various curricular contexts, thus allowing the development of a series of logical-mathematical skills that are evidenced in graph 1.



Graph 1: Benefits of the implementation of programming for the development of logical-mathematical skills.

As evidenced in graph 1, it is established that some authors consider that logical-mathematical skills contribute to problem-solving, logical analysis, metacognition, and to contrast their thoughts (Woods, 2000; Grover, 2009; Herrera, 2015). From this perspective, Palma, and Sarmiento (2015) show in their review 18 publications ranging from 2004 to 2013, where they compare experiences in the teaching of mathematics using programming as a base tool.

In this sense, we highlight contemporary research such as Briz and Serrano (2018), which establish the use of R programming languages as a tool to teach topics in the area of mathematics, allowing in this way, a transversal learning with other contents of the site, which led to a greater interest on the part of the student. Similarly, Oro et al. (2015), designed a Hackers School project with high school students between 16 and 14. Weekly guided workshops were held to bring the language of the narratives to a programming language through problems of implicit mathematical content. As mentioned by Barcelos and Silveira (2012), programming serves as an intermediate tool between the narrative problem and mathematical strength.

Mouhaffel (2018), developed an investigation where he links the aspects of robotics and programming. However, the latter focused on solving problems from the promotion of mathematics and physics, thus demonstrating that ICT's function as a complementary tool to inter-correlate different areas of learning. Benton et al. (2017), designed a ScratchMaths project. They seek to develop curricular materials that contribute to mathematics learning through programming in children between 9 and 11 years old, demonstrating the ease of abstraction of abstract computational thinking for mathematical solving problems.

Psycharis and Kallia (2017), in their purpose, demonstrate the influence of computer programming to develop reasoning, self-efficacy, and problem-solving skills in students. He used two experimental groups, where he conducted math study sessions based on the field of programming with one of them—proving that there were intervals of greater significance in the proposed objectives. Ng and Cui (2020), during a course with five and 6-year-old children, were presented with a series of mathematical problems that were solved in the computer programming environment. This research mentioned that the students used abstract thinking and mathematical models in programming.

Psycharis and Kallia (2017), successfully demonstrated that computer programming significantly improves students' problem-solving skills. However, the difference they obtained with the control group was not statistically significant. Finally, Shahbari et al. (2020), worked with teachers in training from a higher education institute, where they studied metacognitive functions when solving mathematical problems in Scratch programming. Recognizing that programming sequences are essential to developing metacognitive skills, this notion must be accompanied by a good teaching approach to address mathematical issues.

3.3 Scientific Skills

Throughout this work, a series of skills that students have developed from the didactic intervention of computer programming has been addressed. Among these, we can begin computational, logical, and mathematical thinking. In this way, Reyes and García (2014) refer to ability as allowing the student to assimilate the actions and operations addressed within the didactic interventions. Therefore, these skills determine specific processes that integrate a particular mode of student participation (Machado and Montes de Oca Recio, 2009).

The development of didactic sequences based on programming as tools for producing knowledge triggers other types of emerging skills than those mentioned in advance (Bueno, 2018). These are known as higher-order skills. According to Şendağ and Odabaşi (2009), an example of these skills is problem-solving and the capacity for critical thinking and creative thinking. The authors described these types of skills, which carried out their didactic sequences based on programming.

However, programming and the computational environment also involve skills characteristic of scientific literacy, such as: observing facts and phenomena, formulating questions, collecting and analyzing data, measuring, comparing, and evaluating results (Balastegui et al., 2020). Deriving in the scientific attitudes that characterize the objective of what can be considered the purpose of a class, among these attitudes, we find curiosity, honesty, critical assessment, the flexibility of thought, openness of judgments, teamwork, will, and honesty (MEN, 2004; Merino and Torres, 2017).

Table 3 shows some authors who, according to their research, have postulated the scientific skills and attitudes developed under the theme of programming.

Table 3: Scientific skills and attitudes developed during the implementation of the programming

Trend	Scientific skills and attitudes	Authors
Robotics development	Motivation, engagement, measurement, exploring, asking questions, delimiting problems, hypothesizing, planning, controlling variables, using models, classifying, using graphs and tables, recording, using numbers, and operationally defining.	Sáez et al., 2019; Reyes and García, 2014

Trend	Scientific skills and attitudes	Authors
Creation of projects for fixed themes	Problem-solving, logical thinking	Thorat and Kshirsagar (2021)
Teaching tool	satisfaction, commitment, motivation, modeling, prediction	Chis et al., 2018; Domènech, 2020

In the same way, Sáez et al. (2019), developed scientific skills and critical thinking in elementary school students. Using a programming language with blocks for the development of robots in science and content in mathematics, thus highlighting motivation, fun, commitment, and enthusiasm as the most outstanding scientific attitudes in his didactic intervention. Similarly, Thorat and Kshirsagar (2021), highlight scientific attitudes and skills as crucial elements for their programming intervention for problem-solving and logical thinking and the compensation of their difficulties in first-semester students in engineering.

Reyes and García (2014), worked with teachers in training in the branch of science and mathematics to investigate scientific skills in the process of programming with robots. By demonstrating that the intervention could teach students scientific skills, such as measuring, exploring, asking questions, delimiting problems, hypothesizing, planning, controlling variables, using models, classifying, using graphs and tables, recording, using numbers, and operationally defining. On the other hand, Chis et al. (2018), beyond identifying and categorizing scientific skills, was able to find scientific attitudes, such as satisfaction, commitment, and motivation when seeking to demonstrate the effectiveness of the combined teaching approach of a Flipped classroom and problem-based learning in a computer programming module.

Finally, Domènech (2020), in his research, developed a transversal intervention in the field of programming, thus relating programming in the design of an ecosystem simulator with different mathematical functions, creating in this way scientific skills, such as modeling from programming and prediction concerning the structure of ecosystems in science.

4. Final Considerations

Based on an extensive review of the research of programming in the didactic intervention of the school. It is established that it has contributed not only from the technological and digital environment but also in a significant way to develop computational thinking, logical skills - Mathematics and scientific skills for the training of students in primary, high school, and university levels.

Concerning the contexts used in the teaching of programming for the development of computational thinking, there are a set of methodologies showing a range from interaction with programming platforms to the design of robots. It seeks to link science computational with the creation of strategy, problem-solving, and skills of practical value such as teamwork and recognition of social dynamics.

On the other hand, programming to develop logical-mathematical skills stood out for the transversality of the proposed methodologies. Logical analysis, problem-solving and metacognitive analyses were developed from narrations, workshops, study sessions, and robot design, all mediated from programming.

Similarly, the scientific skills developed from programming were modeling, exploration, planning, and using variables, always accompanied by scientific attitudes, where satisfaction, commitment, and motivation stood out. In addition, they were developed from didactic sequences typical of programming platforms and were also taken into areas such as robotics and modeling in environments such as ecosystems.

In conclusion, it is essential to highlight programming as a tool beyond education. It is focused on a theme and enables the student to develop transversal learning by involving a series of skills such as computational, logical-mathematical, and scientific, as well as the fundamental scientific attitudes and skills not only for approaches to the area but also to understand social realities and the student's context.

References

Abril Piedra, H. (2016). Aplicando lenguajes de programación scratch en la enseñanza de matemática a los estudiantes de nivelación de la Universidad Nacional de Educación. Revista de divulgación de experiencias pedagógicas MAMAKUNA, 1, 87-95.

Alsina, Á., & Acosta, Y. (2018). Iniciación al álgebra en Educación Infantil a través del pensamiento computacional: una experiencia sobre patrones con robots educativos programables. UNIÓN. Revista Iberoamericana de Educación Matemática, 52, 218-235.

Balastegui Tomàs, M., Palomar, R., & Solbes Matarredona, J. (2020). ¿En qué aspectos es más deficiente la alfabetización científica del alumnado de Bachillerato?. Revista Eureka sobre Enseñanza y Divulgación de las Ciencias, 2020, 17(3), 3302.

Barcelos, T. S., & Silveira, I. F. (2012). Pensamento computacional e educação matemática: Relações para o ensino de computação na educação básica. In XX Workshop sobre Educação em Computação, Curitiba. Anais do XXXII CSBC, 2, 23.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is Involved and What is the role of the computer science education community? ACM Inroads, 2(1), 48-54.

Basogain, X., Olabe M. A. & Olabe J. C. (2015). Pensamiento computacional a través de la programación: paradigma de aprendizaje. Revista de Educación a Distancia, 46(6). https://doi.org/10.6018/red/46/6

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. Digital Experiences in Mathematics Education, 3(2), 115-138.

Berrocoso, J. V., Sánchez, M. R. F., & Arroyo, M. D. C. G. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. Revista de Educación a Distancia (RED), 46 (3), 1-18.

Briz Redón, Á., & Serrano Aroca, Á. (2018). Aprendizaje de las matemáticas a través del lenguaje de programación R en Educación Secundaria. Educación matemática, 30(1), 133-162.

Bueno, P. M. (2018). Aprendizaje basado en problemas (ABP) y habilidades de pensamiento crítico ¿una relación vinculante?. Revista Electrónica Interuniversitaria de formación del profesorado, 21(2), 91-108.

Calao, L. A., Moreno León, J., Correa, H. E., & Robles, G. (2015). Developing Mathematical Thinking with Scratch. Lecture Notes in Computer Science, 17–27.

Castellanos, A., Sánchez, C. & Calderero, J. F. (2017). Nuevos modelos tecnopedagógicos. Competencia digital de los alumnos universitarios. Revista Electrónica de Investigación Educativa, 19(1), 1-9.

Chis, A. E., Moldovan, A. N., Murphy, L., Pathak, P., & Muntean, C. H. (2018). Investigating flipped classroom and problem-based learning in a programming module for a computing conversion course. Journal of Educational Technology & Society, 21(4), 232-247.

Delgado, J., Güell, J., García, J., Conde, M. & Casado, V. (2013). "Aprendizaje de la programación en el Citilab", Revista Iberoamericana de Ciencia, Tecnología y Sociedad, 8(23), 123-133.

Domènech Casal, J. (2020). Diseñando un simulador de ecosistemas. Una experiencia STEM de enseñanza de dinámica de los ecosistemas, funciones matemáticas y programación. Revista Eureka sobre enseñanza y divulgación de las ciencias, 17(3), 320201-320217.

Flick, U. (2004). Triangulación en la investigación cualitativa. Un compañero de la investigación cualitativa , 3 , 178-183.

García, A. V. (2010). La didáctica de la enseñanza de la matemática: ¿es posible aplicarla a la enseñanza de la programación?. Revista académica e institucional de la UCPR, (88), 5.

García Peñalvo, F. J., Reimann, D., Tuul, M., Rees, A., & Jormanainen, I. (2016). An overview of the most relevant literature on coding and computational thinking emphasizes the relevant issues for teachers. Belgium: TACCLE3 Consortium. doi:10.5281/zenodo.165123.

González, C. S. G. (2019). Estrategias para la enseñanza del pensamiento computacional y uso efectivo de tecnologías en educación infantil: una propuesta inclusiva. Revista Interuniversitaria de Investigación en Tecnología Educativa, 7, 85-97.

González González, C. S. (2019). Estado del arte en la enseñanza del pensamiento computacional y la programación en la etapa infantil. Education in the Knowledge Society, 2(1), 17-35.

Grover, S. (2009). Computer science is not just for big kids. Learning & Leading with Technology, 37 (3), 27-29.

Herrera, P. C. (2015). Matemáticas y computación: Uso de programación visual para el desarrollo de material didáctico en un entorno educativo. Blucher Design Proceedings, 2(3), 581-588.

Kirschner, P. A. & De Bruyckere, P. (2017). The myths of the digital native and the multitasker. Teaching and Teacher Education, 67, 135-142.
Larkins, D. B., & Harvey, W. (2010). Introductory computational science using MATLAB and image processing. Procedia Computer Science, 1(1), 913-919.
Lee, M. J., & Ko, A. J. (2011). Personifying programming tool feedback improves novice programmers' learning. In Proceedings of the seventh international workshop on Computing education research, 109-116.
Llorens, F. (2015). Dicen por ahí... que la nueva alfabetización pasa por la programación. ReVision, 8(2), 11-14
Liu, Y., Ding, H., & He H. C. (2015). language program design. Beijing: Science Press.
Machado Ramírez, E. F., & Montes de Oca Recio, N. (2009). Las habilidades investigativas y la nueva Universidad: Terminus a quo a la polémica y la discusión. Humanidades Médicas, 9(1).
Martínez, N. M. M., Olivencia, J. L., & Meneses, E. L. (2016). Robótica, modelado 3D y realidad aumentada en educación para el desarrollo de las inteligencias múltiples. Aula de Encuentro, 18(2), 158–183.
Merino, E. P., & Torres, C. N. B. (2017). Actitud hacia la ciencia y experiencia investigativa en estudiantes de secundaria. Opción, 33(84), 191-217.
Ministerio de Educación Nacional. (2004). Guía No. 7 La formación en Ciencias: ¡El desafío! Colombia: MEN.
Monjelat, N. G., Cenacchi, M. A., & San Martín, P. S. (2018). ¿Programación para Todos? Herramientas y Accesibilidad: Un Estudio de Caso. Revista latinoamericana de educación inclusiva, 12(1), 213-227.
Mouhaffel, A. G. (2018). Comparación de la enseñanza dos sistemas de programación robótica enfocada a los recursos matemáticos: Arduino+ Scratch y Sistema Lego EV3. International Journal of Innovation and Applied Studies, 25(1), 15-39.
Narváez, H. O. P., & Roig Vila, R. (2015). Entornos de programación no mediados simbólicamente para el desarrollo del pensamiento computacional. Una experiencia en la formación de profesores de Informática de la Universidad Central del Ecuador. Revista de Educación a Distancia (RED), (46).
Ng, O. L., & Cui, Z. (2020). Examining primary students' mathematical problem-solving in a programming context: towards computationally enhanced mathematics education. ZDM, 1-14.
Olabe, X. B., Basogain, M. Á. O., & Basogain, J. C. O. (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. Revista de educación a distancia (RED), (46).
Oro, N. T., Mileidi Pazinato, A., Espíndola Lessa, V., & Forigo, F. (2015). Programação de Computadores e Matemática: potencializando a aprendizagem. In conferência interamericana de educação matemática, 14, 2015, 467-478.
Palma Suárez, C. A., & Sarmiento Porras, R. E. (2015). Estado del arte sobre experiencias de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias matemáticas en primaria. Revista mexicana de investigación educativa, 20(65), 607-641.
Papert, S. (1972). Teaching children thinking. Programmed Learning and Educational Technology, 9(5), 245-255.
Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem-solving. Instructional Science, 45(5), 583-602.
Reyes, D., & García, Y. (2014). Desarrollo de habilidades científicas en la formación inicial de profesores de ciencias y matemáticas. Educación y Educadores, 17(2), 271-285.
Rincón A. I. & Ávila W. D. (2016). Una aproximación desde la lógica de la educación al pensamiento computacional. Sophia, Colección de Filosofía de la Educación, 21, 161-176.
Román Gonzalez, M., Pérez González, J. C., & Jiménez Fernández, C. (2015). Test de Pensamiento Computacional: diseño y psicometría general. In Iii congreso internacional sobre aprendizaje, innovación y competitividad (CINAIC 2015) (pp. 1-6).
Rozo, O. P., & Pérez, V. R. D. (2014). Didáctica de las matemáticas y tecnologías de la información y la comunicación. Educación y desarrollo social, 8(2), 60-81.
Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching programming in Secondary school: A pedagogical content knowledge perspective. Informatics in education, 10(1), 73-88.
Sáez, J. M. & Cózar, R. (2017). Pensamiento computacional y programación visual por bloques en el aula de Primaria. Educar, 53(1), 129–146.
Sáez López, J. M., Sevillano García, M. L., & Vazquez Cano, E. (2019). The effect of programming on primary school students' mathematical and scientific understanding: educational use of mBot. Educational Technology Research and Development, 67(6), 1405-1425.
ScratchEd Team. (2015). Computational Thinking webinars. Obtenido de: http://scratched.gse.harvard.edu/content/1488
Şendağ, S., & Odabaşi, H. F. (2009). Effects of an online problem-based learning course on content knowledge acquisition and critical thinking skills. Computers & Education, 53(1), 132-141.
Shahbari, J. A., Daher, W., Baya'a, N., & Jaber, O. (2020). Prospective Teachers' Development of Meta-Cognitive Functions in Solving Mathematical-Based Programming Problems with Scratch. Symmetry, 12(9), 1569.

Taborda, H., & Medina, D. (2014). Programación de computadores y desarrollo de habilidades de pensamiento en niños escolares: fase exploratoria. Serie documentos de trabajo del CIES, 3, 4-22.

Tejera Martínez, F., Aguilera, D., & Vílchez González, J. M. (2020). Lenguajes de programación y desarrollo de competencias clave. Revisión sistemática. Revista electrónica de investigación educativa, 22 (27), 1-16.

Tezanos, J. F. (2001). La sociedad dividida: estructuras de clases y desigualdades en las sociedades tecnológicas. Madrid: Biblioteca Nueva.

Thorat, S. A., & Kshirsagar, D. P. (2021). Developing Logic Building, Problem Solving, and Debugging Programming Skills Among Students. Journal of Engineering Education Transformations, 34(SP ICTIEE), 402-406.

Williner, B. (2014). Habilidades matemáticas referidas el concepto de Derivada y uso de tecnología. Revista de Didáctica de las Matemáticas, 8(1), 101-124.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Woods, D. R. (2000). An evidence-based strategy for problem-solving. Journal of Engineering Education, 89 (4), 443-459

Zapata, A. S., Costa, D. G., Delgado, P. A. M., & Torres, J. M. (2018). Contribución de la robótica educativa en la adquisición de conocimientos de matemáticas en la Educación Primaria. Magister: Revista miscelánea de investigación, 30(1), 43-54.

Zhou Y. (2007). computational thinking. China Computer Society Newsletter, 3(11): 83-85.