Informatics in Education, 2023, Vol. 00, No. 00, –

DOI: 10.15388/infedu.2023.21
ACCEPTED VERSION (WITHOUT THE JOURNAL LAYOUT).

Investigating block programming tools in high school to support Education 4.0: A Systematic Mapping Study

Ana Paula Juliana Perin[1], Deivid Eive dos S. Silva[1], Natasha M. C. Valentim[1]

[1] *Department of Informatics, 383-391 Evaristo F. Ferreira da Costa St, Federal University of Paraná, Brazil*

e-mail: apjperin@inf.ufpr.br, dessilva@inf.ufpr.br, natasha@inf.ufpr.br

Abstract. In Education 4.0, a personalized learning process is expected, and that students are the protagonist. In this new education format, it is necessary to prepare students with the skills and competencies of the 21st-Century, such as teamwork, creativity, and autonomy. One of the ways to develop skills and competencies in students can be through block programming, which can be used with emerging technologies such as robotics and IoT and in an interdisciplinary way. Thus, block programming in High School is important because it is possible to work on aspects such as problem-solving, algorithmic thinking, among other skills (Perin et al., 2021), which are necessary in the contemporary world. Thus, our Systematic Mapping Study (SMS) aims to identify which block programming tools support of Education 4.0 in High School. Overall, 46 papers were selected, and data were extracted. Based on the results, a total of 24 identified block programming tools that can be used in high school collaboratively and playfully and with an interdisciplinary methodology. Moreover, it was possible to see that most studies address block programming with high school students, demonstrating a lack of studies that address block programming with teachers. This SMS contributed to identifying block programming tools, emerging technologies, audience (teacher or student), and learning spaces where block programming is being worked on.

Key words: Block Programming Tools, Education 4.0, Computational Thinking.

1. Introduction

Currently, the need for education more in line with the contemporary world can be seen in order to prepare students for the challenges of the 21st-Century, such as dealing with emerging technological resources and processes Silva et al. (2020) . This need led to a new learning format known as Educação 4.0 (de Oliveira, 2015). In Education 4.0, a personalized learning process is expected in which the student is the protagonist, which makes it possible to prepare students with 21st-Century skills and competencies. In the literature, preparing a way for students with Education 4.0 is through block programming (Da Silva, 2018; da Silva, 2020).

In this context, block programming tools can help teachers customize curriculum content through multimedia resources such as images, graphics, texts, videos, and audio (da Silva, 2020). Moreover, the use of block programming tools can encourage students to be protagonists in their learning. Block programming allows students to create simulations of real situations, develop digital games, contextualize the contents of the basic curriculum in a practical way, among other activities.

In the context of Education 4.0, block programming has been used as a visual-friendly methodology that can help arouse students' curiosity to learn Computer Science concepts related to programming, computational thinking, software development, and logic in a way general without worrying about syntax problems in programming (Fronza et al., 2015). Block programming environments have specific characteristics and colors related to their functions (commands and values). These characteristics and colors make it possible to form the structures of the programs. On the other hand, commands and values are related to docking blocks, allowing the act of programming to be less complex (de Souza Rios et al., 2019). Moreover, the introduction of block programming can prepare students, enabling young people to be able to make decisions and solve problems in a creative, collaborative way, and with a dynamic mindset to learn new knowledge (Lima et al., 2017).

Coding is part of logical thinking, considered one of the important skills of the 21st-Century (Papadakis and Orfanakis, 2018). Furthermore, it was noticed that block programming could be worked with emerging technologies such as Robotics and the Internet of Things (IoT), among others. Thus, it is believed that the combination of block programming and emerging technologies can bring benefits to the classroom, such as a) providing curriculum adaptation, making it more flexible; b) providing interactivity and collaboration in the classroom; and c) allowing experiences in practical activities for student engagement, relating theoretical content to practice.

Therefore, this article presents the results of a Systematic Mapping Study (SMS), which sought to answer the main research question: "What block programming tools have been used in High School and that can support in the context of Education 4.0?". For this SMS, High School was chosen, as Brezolin and Silveira (2021) and Szabo et al. (2019) carried out systematic studies on block programming with students, and these authors concluded that most publications are aimed at Elementary School. Furthermore, in the literature, most block programming initiatives in high school are related to developed countries (WEIRD - Western educated, industrialized, rich, and democratic) (Cobo et al., 2021). This demonstrates that block programming in high school is not a reality in developing countries. In this sense, we sought to investigate characteristics that can improve the use of blocks programming in developing countries, as this approach can help to adapt the teaching of Computer Science concepts (Flórez et al., 2017) at this level of education, encourage students to enter higher courses at areas of computing; and, preparing students for life in the 21st-Century, as well as developing skills such as computational thinking. Therefore, it is believed that the use of block programming tools and emerging technologies can help High School students' learning (Weintrop and Wilensky, 2015).

Overall, this SMS contributes:

- To identify which block programming tools, emerging technologies, audience, and learning spaces where block programming is being worked on.
- To perceive the need for initiatives that support the teacher in the use of block programming and/or development of computational thinking in High School.

- To perceive that most block programming teaching initiatives take place at school, i.e., few studies are carried out outside the High School environment.
- To perceive that there is still a need for methodologies that can work block programming in an interdisciplinary way in High School.
- To perceive that most studies do not mention the use of emerging technologies in High School, which is an important aspect in the context of Education 4.0.

Moreover, this SMS can contribute with high school educationalists, because they can have access to emerging technologies and block programming tools identified in the context of Education 4.0. In addition, high school educationalists can use the learning spaces also identified in this SMS to work on block programming in an interdisciplinary and playful way, promoting the development of students' 21st Century skills and competencies.

For this purpose, automatic searches were carried out in this SMS in digital libraries, according to the guidelines proposed by Kitchenham and Charters (2007). The main results identified were: 1) a total of 24 block programming tools that can be used in High School collaboratively and playfully, and with an interdisciplinary methodology; 2) few studies mention the use of block programming in disciplines from other non-exact areas, such as Arts in an interdisciplinary way; 3) most studies were conducted with students, demonstrating a lack of studies conducted with teachers, since teacher training initiatives can strengthen the use of block programming in the classroom; and 4) Educational Robotics, Digital Games and IoT were the main emerging technologies identified and worked on teaching block programming in High School.

This article is organized into 6 more Sections. Section 2 presents the evolution of education from Education 1.0 to Education 4.0. Section 3 presents the related work. Section 4 presents the methodology used to plan, conduct and analyze the results of SMS. Section 5 presents the threats to validity. Section 6 addresses the discussions. Finally, Section 7 presents the final considerations and future perspectives.

2. Background

This section presents the concepts necessary to understand the scope of SMS, such as Education 4.0 and block programming.

2.1. *Evolution of Education*

Education has been adapted over time in order to support the modes of production caused by the industrial revolutions (Puncreobutr, 2016). Thus, it is characterized in four phases: Education 1.0, 2.0, 3.0, and 4.0.

Education 1.0, related to the 1st Industrial Revolution, is known for the teacher-student relationship, in which there was no personalization of the curriculum and the goal of teaching was directed to the ability to read, write and calculate Silva et al. (2020); Hartono et al. (2018); Koul and Nayar (2021); Puncreobutr (2016). In Education 1.0, the teacher kept all the information and students didn't have active role in the learning process, they only received the information (Puncreobutr, 2016; Montoya et al., 2021). Thus, this education is characterized as traditional education.

Education 2.0, related to the 2nd Industrial Revolution, demanded reading and writing skills from students, generated by the need for improvement/training of people to work in the production line Silva et al. (2020); Führ (2018). Teaching was based on memorizing of information to operate machines and work tools (de Oliveira, 2019; Puncreobutr, 2016). In Education 2.0, textbooks were included for the transmission of knowledge (Hartono et al., 2018) and the classroom was seen in a homogeneous way and with a standardized teaching and learning methodology (Führ, 2018).

Education 3.0, related to the 3rd Industrial Revolution, is characterized by the arrival of computers. In Education 3.0, there was an exchange of information between students and teachers. Thus, the student could create knowledge and not just consume it (de Oliveira, 2019; Puncreobutr, 2016; Silva et al., 2020) . Education starts using technologies to support teaching and learning (Puncreobutr, 2016; Hartono et al., 2018; Koul and Nayar, 2021). There was the insertion of the digital whiteboard in schools (Hartono et al., 2018). Digital media and social media are seen as the basis for interactive learning (Puncreobutr, 2016). Thus, it is understood that technology allows students unlimited access to information sources and promotes support for autonomous learning. However, the teaching and learning processes have not changed as the way of teaching is still aligned with the Education 2.0 teaching model Silva et al. (2020); Koul and Nayar (2021).

Education 4.0, the focus of this SMS, will be detailed in the following subsection.

2.2. *Education 4.0*

Education 4.0, related to the 4th Industrial Revolution, demands an education more in line with the contemporary world. Students need to be prepared for the challenges of the 21st-Century and be protagonists in their learning Silva et al. (2020); Koul and Nayar (2021). The term Education 4.0 had its first mention in 2015 (Ciolacu et al., 2017). Education 4.0 refers to a new way of learning to prepare students to develop knowledge, skills, and experiences in all aspects of life (Puncreobutr, 2016; de Oliveira, 2019). Thus, the classroom can be designed based on some characteristics, such as: a) Personalization of teaching material to meet different types of learning; b) Gamification with playful elements that help in the motivation and engagement of the student; c) Activities adapted according to the student's knowledge and learning; and d) Analysis methods to identify students with difficulties, among others (Ciolacu et al., 2017).

Thus, students need to be capable and ethical in using new technological tools. In addition, students need to be critical, creative, reflective, and leading citizens (Alda et al., 2020). Additionally, to reinforce Education 4.0, an adequate methodology is important so that students can understand the materials used for the teaching and learning processes (Hartono et al., 2018). Thus, planning and preparation of teachers for the use of new pedagogical methodologies are necessary to allow the protagonism of the student (Santos et al., 2019). It is also necessary for the teacher to (1) encourage collaborative learning through social experiences and interactions; (2) work on autonomy, decision-making, and critical thinking, enabling the student to choose what to learn, when, how, where and why; (3) enable the student to learn by doing, using technology as a support/means; and (4) allow experiences beyond the classroom (Hartono et al., 2018).

2.3. *Block programming and computational thinking in the context of Education 4.0*

In the context of Education 4.0, block programming encourages digital culture and the digital world at school (dos Santos Silva et al., 2019; Aono et al., 2017). Furthermore, it is believed that block programming environments can make programming, which is naturally related to the concepts of Mathematics and Logic, more attractive through a more intuitive and visual experience for students. An important skill that can be related to block programming is computational thinking, characterized by a set of skills to solve problems, design systems, and understand human behavior, being supported by concepts from Computer Science (Wing, 2006). The teaching of computational thinking can be inserted in High School through the introduction of block programming languages (Vinayakumar et al., 2018b). Block programming can help the student to be the author of new technologies, not just a consumer (Souza Rios et al., 2019). The importance of teaching programming in High School is recognized and that this level of education needs to be accompanied by a methodology that enables the engagement and motivation of students, so that the difficulties encountered can be overcome and students remain interested in learning (Scaico et al., 2013).

One way to keep students engaged can be through block programming allied to emerging technologies, such as a) Robotics through *Lego*; b) IoT mediated by *Scratch for Arduino (S4A)*; and, c) 3D modeling and simulation and/or digital games using the *ENGAGE* tool, among others. Thus, by making use of block programming and emerging technologies in the classroom, teachers will also benefit from the teaching process by learning about the features of the block programming tool they want to work with, learn through the exchange from experiences with other teachers, and deepens their understanding of their own experiences and the experiences of their students with programming (Haduong and Brennan, 2019). In addition, the teacher will be able to encourage the development of skills in students based on practical activities that bring the most diverse resources of the digital era to the classroom (Führ, 2018).

3. Related work

A Systematic Literature Review (SLR) was carried out by Szabo et al. (2019) to explore studies on the teaching of introductory programming in primary and secondary education during the period 2003 to 2017. The authors considered block programming, textual programming, and tangible programming. The selected studies address computational thinking. The SLR had two research questions, being: RQ1 "What developments were made in introductory education in K-12 programming between 2003 and 2017?" and RQ2 "What evidence has been reported for different aspects of introductory education in K-12 programming?". The libraries used for SLR were ACM, IEEE Xplore, ScienceDirect, SpringerLink, and Scopus. The search for articles was carried out on May 27 2018, and 5056 articles were returned, where 108 articles were considered. In this SLR, it was noticed that most articles address studies in Elementary School and the beginning of High School (aged approximately 10-15 years). The authors note that there is a consensus in studies that the *Scratch* is a block programming tool more attractive to elementary school students but less attractive to high school students, who tend to prefer environments that allow more advanced programming, such as the MIT AppInventor for Android. It was also noticed that students demonstrated more engagement with Lego NXT and similar. However, graphical alternatives to block programming tools can be more valuable to students as they focus on programming. In addition, the authors identified a need to support teachers in teaching the programming curriculum and in overcoming other barriers, teacher workload, and lack of confidence in using the tools. Also, the authors do not provide evidence of systematic or generalized initiatives to prepare and support teachers with technological and pedagogical skills useful for teaching programming in Elementary and High School.

Another SLR was performed by Souza et al. (2018), where it was identified that it is possible to apply educational robotics in different contexts. This SLR aims to gather information about educational environments for the practice of robotic programming using LEGO. The research questions were: "RQ1: What programming environments and languages have been used to teach through LEGO Robotics”; "RQ2: How has LEGO Robotics been used in education?”; and "RQ3: What has been the target audience for using LEGO Robotics?”. The search for articles was carried out from October 2017 to April 2018, in the ACM, IEEEXplore, Science Direct, and Scopus libraries. The selected studies considered students at different levels of education. Regarding basic level students, most were related to Elementary and High School, followed by undergraduate and graduate education. The main findings of SLR are related to skills in using educational robotics as teamwork and problem solving, involving interdisciplinary content in Elementary and High School, through block programming. Additionally, it was identified that the use of LEGO with augmented reality can facilitate the teaching of programming and contents in an interdisciplinary way through STEM (acronym for Science, Technology, Engineering, and Mathematics).

The SLR performed by Morales et al. (2019) sought to analyze the use of the Alice block programming tool in Elementary and High School for teaching object-oriented programming. The libraries used to search for articles were ACM, IEEEXplore, Science Direct, Google Scholar, Microsoft Academic, ERIC, SpringerLink, DOAJ, SciELO, and Redalyc. The authors highlighted that most of the identified studies sought to determine the effect of using the Alice tool as a resource for introductory programming learning. Most of the initiatives to experiment with Alice in introductory programming learning show the effectiveness of this tool as a support resource in this complex process. In addition, the experiences reported in cases of inclusion are female students, people with dyslexia, among other groups.

Brezolin and Silveira (2021) carried out an SMS on technological tools used to promote computational thinking and programming teaching. The authors performed their searches in the Brazilian databases such as (1) Brazilian Symposium on Informatics in

Education, (2) Workshop on the Teaching of Computational Thinking, Algorithms and Programming, (3) Brazilian Journal of Informatics in Education, (4) Workshop on Computer Education, (5) Workshop on Informatics in the School, and (6) Symposium on Human Factors in Computer Systems. The research questions were: "QP1 - What is the target audience of these studies?", "QP2 - What is the main objective of these studies?" and "QP3 - What tools have been used?". The authors identified that most of the articles are directed to Elementary School, followed by Higher and High School. Finally, the educational levels with fewer studies are the Technical, kindergarten, and those directed to the general public. In this SMS, most studies mention Block Programming Languages, followed by digital games, textual programming languages, Mindstorms, and Arduino Robotics kits. The studies that mention Block Programming Languages are independent of a variety of tools. The most cited tool was Scratch, followed by the Code.org platform and MIT App Inventor.

Lin and Weintrop (2021) carried out an SLR to answer two research questions: (RQ1) What is the current state of the design of block-based programming environments?; and, (RQ2) What design approaches are currently being used to support learners in transitioning from block-based to text-based programming?. To answer the research questions, Lin and Weintrop (2021) sought to: (1) identify a significant number of tools that rely on the block-based programming approach, (2) analyze and categorize environments, and (3) understand each tool in relation to block and text-based programming. In total, 101 tools were identified, of which 46 allowed a deep analysis with the classification for which the environment was designed, resources, and the relationship with textual programming. From the analysis, it was possible to identify four approaches that support the transition from programming in blocks to textual programming: blocks only, dual mode, unidirectional, and hybrid transition. Regarding the limitations of this SRL, the authors considered only the technical characteristics of the tools, such as environment, domain, and the relationship with textual programming. In addition, in this SLR, the authors did not focus on tools that were used in High School, and perhaps because of this, they identified a lot of programming tools in blocks.

Weintrop et al. (2019) present a study carried out with the TEC Rubric tool, developed to assist in decision making in Computer Science (CS) curricula in grades K-12 (Elementary and High School). Weintrop et al. (2019) realized the need to support teachers to make CS content more accessible. In this sense, the authors carried out an SLR that served as the basis for composing the TEC Rubric, focusing on: (a) Teacher Accessibility, (b) Equity (creation of effective and accessible learning opportunities), and (c) Content, which is intended for use by two profiles: for teachers, it covers educational decisions; and, for designers, it involves creating effective, accessible and equitable learning opportunities. Regarding teacher accessibility (a), studies have shown that teachers play an important role in supporting their students' intellectual growth and promoting a positive and equitable learning space. One of the challenges for teachers to teach SC in classrooms is related to teacher training so that they can act autonomously. Regarding equity (b), it was found that the probability of a student successfully taking CS courses is related to their level of access to these courses, qualified teachers, and resources available in the classroom. Finally, regarding the contents (c), it was noticed that, although the SC area is not new, the contents to be included in SC education are still discussed. This is because CS standards are not static and need to be reviewed and updated on a regular basis, not be considered complete and finalized. In addition, CS content standards in the K-12 series should be informed and updated by research. Weintrop et al. (2019) identified a document (K-12 Computer Science Framework (K12CS, 2016)) that was considered useful in presenting a broader view of CS contents distributed between Elementary and High Schools.

In general, the studies identified focus on teaching programming in blocks and/or computational thinking, or are directed to tools that help in the teaching and learning process of programming in Elementary and High School. Most studies point to the greater use of block programming tools in Elementary School, focusing on teaching the Computer Science curriculum. An SMS was not identified that investigates, in a general way, which programming tools in blocks are used in the teaching and learning processes of programming in blocks and/or computational thinking related to contents of subjects of High School, that is, of the common curriculum, and that involves the context of Education 4.0. Differently from the studies presented in this section, the SMS presented in this article sought to verify if programming in blocks was taught in the context of Education 4.0, and if the teaching and learning process of programming in blocks was carried out together with subjects of the common curriculum, such as Physics and Biology, and using emerging technologies such as Robotics and Digital Games; what emerging technologies were used; whether there was training in the use of block programming tools and who received training, teacher or student; who was responsible for teaching block programming; among other features. The motivation of the SMS was to understand how the investigated characteristics contribute to the development of skills and abilities with a focus on students, so necessary for life in the 21st century, in order to promote education more aligned to Education 4.0.

4. Systematic Mapping Study

This section aims to present the strategies of the Systematic Mapping Study (SMS) carried out in this research. According to Kitchenham and Charters (2007), the SMS is characterized by the classification and grouping of primary studies, being an important process to identify gaps in research and get an overview of a research topic.

4.1. *SMS goal*

The goal of this SMS was structured and defined according to the GQM paradigm *(Goal-Question-Metric)*, proposed by Basili and Rombach (1988) and presented in Table 1.

4.2. *Research Question and Sub-questions*

The main research question to be investigated is: “What block programming tools have been used in High School that can support in the context of Education 4.0?”. Research subquestions were also defined, described in Table 2 , to obtain information that helps to answer the main question.

Table 1
SMS goal according to GQM paradigm

Analyze	scientific publications
For the purpose of	to characterize
With respect to	Block programming tools applied in High School to support Education 4.0
From the point of view of	researchers of Informatics in Education and Computer Science Education
In the context of	primary sources available on SCOPUS, ACM, IEEExplore, and SpringerLink

4.3. *Search strategy*

The search strategy was organized according to Kitchenham and Charters (2007), allowing to safeguard the research integrity and reduce bias. Thus, (1) the scope of the research was defined, which addresses the methods used to search for articles; (2) the languages chosen for the selection of articles; (3) the search string used in SMS; (4) the inclusion and exclusion criteria of articles; (5) the articles selection process; and (6) the data extraction strategy.

4.3.1. *Research scope*

In this SMS, automatic search methods were used in digital libraries and using keywords that compose the search *string*. The digital libraries chosen for the automatic search were ACM[1], IEEExplore[2], Springer Link[3] and Scopus[4], as these databases return studies published in the most diverse areas of knowledge. The ACM and IEEExplore libraries were chosen as a reference for studies published in the field of Computing. Springer Link for being a publication reference for studies published in the field of Education. And Scopus, for being one of the largest indexing bases.

4.3.2. *Language of articles*

The languages chosen for the selection of articles were English and Portuguese. English, as it is the language adopted by the vast majority of international conferences and journals related to the research topic. And Portuguese for being the native language of the researchers of this article.

4.3.3. *Search string*

In order to improve and structure the searches in the selected digital libraries, the PICOC *(Population, Intervention, Comparison, Outcome, and Context)* was used, based on Kitchenham and Charters (2007). *Population* refers to where the research topic was observed, *Intervention* is related to what was investigated, *Comparison* concerns what was

[1]https://dl.acm.org/
[2]https://ieeexplore.ieee.org/Xplore/home.js
[3]https://www.scopus.com/search/
[4]https: //link.springer.com/

Table 2
SMS Subquestions

Research Subquestions	Objective and possible answers
SQ1. Was block programming used at school?	Investigate whether block programming was used at school. The use of block programming at school can be classified in one of the answers: **a) Yes**; **b) No**; **c) Not mentioned in the study**.
SQ1.1. In which learning space is block programming being used to support Education 4.0?	Identify the learning spaces where block programming activities are cultivated. The spaces used can be: **a) Computer Laboratory:** f the teaching of block programming was carried out in the context of a computer laboratory; **b) Extension activities:** if the teaching of block programming was carried out in the context of an extension action/activity; **c) In the classroom:** if the teaching of block programming was carried out in the context of a classroom equipped with portable computers; **d) Others**; **e) Not mentioned in the study**.
SQ2. Who is responsible for teaching block programming?	Investigate whether block programming was taught by the teacher or by a computer instructor. Those responsible for teaching block programming can be: **a) Teacher**; **b) Computer instructor**; **c) Researchers**; **d) Others**.
SQ3. Was there training in using the block programming tool? If yes, who received training?	Identify if there was training in using the tool and who received training. Possible answers could be: **a) Yes, Teacher**; **b) Yes, Student**; **c) Not mentioned in the study**.
SQ4. To which target audience and high school grade are block programming tools being used?	Investigate which target audience and which high school grade the block programming tools are using. The target audience that used block programming can be classified into: **a) Student**; **b) Teacher**; The high school grades where block programming was applied can be: **a) 1st grade of High School /K-10 (15 years):** if block programming using computational resources was addressed in the 1st grade of High School in Brazil or K-10 in the United States of America (USA); **b) 2nd grade of High School/K-11 (16 years old):** if block programming using computational resources was addressed in the 2nd grade of High School in Brazil or K-11 in USA; **c) 3rd grade of High School/K-12 (17 and 18 years old):** if block programming using computational resources was addressed in the 3rd grade of High School in Brazil or K-12 in USA; **d) Not mentioned in the study**.
SQ5. In which class were block programming activities carried out?	Investigate which high school class block programming tools were used. The classes can be: **a) Math**; **b) Physics**; **c) Chemistry**; **d) Biology**; **e) Others**. **f) Not mentioned in the study**.

compared, *Outcome* refers to the contribution of the studies that was carried out, and *Context* covers the context for comparison. In this SMS, the PICOC was applied as follows: ***Population*** **(P)**: High School. ***Intervention*** **(I)**: Block programming tools used in High School. ***Comparison*** **(C)**: Not applicable, as the purpose of the research is not to compare block programming tools. ***Result*** **(O)**: Use of block programming tools to support Education 4.0. ***Context*** **(C)**: Not applicable as there is no comparison.

In Table 3, are the keywords and *s*earch string, which are divided into three parts. The first part represents the population, which are works aimed at high school; the second

Research Subquestions	Objective and possible answers
SQ6. What tools support Education 4.0 in High School?	Investigate which tools support Education 4.0 in High School. The tools can be: **a)** ***Scratch***; **b)** ***Mit App Inventor***; **c)** ***Scratch* for Arduino**; **d) LEGO Mindstorms**; **e) Blockys**; **f) DuinoBlock**; **g) Others**.
SQ6.1. What are the emerging technologies can be used to support Education 4.0?	Investigate which emerging technologies can be used to support Education 4.0. Emerging technologies can be: **a) Educational Robotics Costa et al. (2017):** if block programming was used in the teaching of Robotics. An example is a possibility of programming a robot that calculates the distance of an obstacle and deviates, which simulates an autonomous car; **b) IoT Souza (2016):** if block programming was used in teaching IoT. An example is a possibility of programming an LED to turn on or off by means of a command, which can simulate home automation (home automation); **c) Augmented Reality Vidotto et al. (2018):** if block programming was used in teaching Augmented Reality, for example, augmented reality glasses for viewing an animated and interactive video; **d) Others**; **e) Not mentioned in the study**.

represents the intervention, what is intended to be found, as technologies related to block programming; and the third represents the outcome, related to learning and/or thinking through block programming in the context of Education 4.0.

Table 3
Keywords and search string.

	Search string	
Population	("high school" OR "senior high" OR "K-10" OR "K-11" OR "K-12")	AND
Intervention	("block programming" OR "block-based programming" OR "block-based coding" OR "block interface" OR "block-based tool" OR "block-based platform" OR "block-based language" OR "block-based approach" OR "block-based methodology" OR "block-based process" OR "visual block programming")	AND
Outcome	("e-learning" OR "active learning" OR "Education 4.0" OR "blended learning" OR "computational thinking")	

4.3.4. *Article selection criteria*

Each article selected in this SMS was independently evaluated by three researchers, who decided whether or not this publication should be included considering a set of criteria divided into (1) Inclusion Criteria (IC) and Exclusion Criteria (EC).

Inclusion criteria of articles:

- **IC1.** Publications that present initiatives that support block programming and computational thinking in High School through block programming tools in the context of Education 4.0.

- **IC1.2.** Publications describing studies on the use of technologies that help block programming and computational thinking through block programming in High School in the context of Education 4.0.

Exclusion criteria of articles:

- **EC1.** Publications that did not attend inclusion criteria were not selected.
- **EC2.** Publications that have a different language of English or Portuguese.
- **EC3.** Publications that do not have content available for reading and analyzing data (especially in cases where papers are paid or not made available by search repository).
- **EC4.** Publications that have already been added to another search engine defined in our SMS (duplicate).
- **EC5.** Publications that were not peer-reviewed (gray literature).

4.3.5. *Article selection process*

In this SMS, the selection of primary studies was performed in two stages (first filter and second filter). In the first filter, the title and abstract were read by three researchers. A justification was provided for each article excluded in this step. In the second filter, the articles that passed in the first filter were fully read. The two stages went through the same processes: 1) The researchers analyzed the inclusion and exclusion criteria (defined in Subsection 4.3.4) and registered the results obtained; 2) The researchers reached a consensus when there was no unanimity in the inclusion or exclusion of a publication; 3) In the consensus step, in case of disagreement about the inclusion of a publication, the study was included for the next step.

4.3.6. *Data extraction strategy*

The data extraction strategy adopted in this SMS provided a set of possible answers for each research subquestion defined above. The data obtained were registered in a document for further analysis and a summary of the results. The possible answers for each research subquestion are explained in more detail in Table 2.

4.4. *Quantitative Results*

The search *string* was applied on March 5, 2020, in the four selected digital libraries. Figure 1 presents an overview of the number of studies returned and selected in the first and second filters. Overall, 507 studies were returned, of which 155 were from ACM, 11 from IEEExplore, 99 from SpringerLink, and 242 from Scopus. In the first filter, 239 studies were selected by reading the title and *abstract*. Of the 239 studies, 83 are from ACM, 11 from IEEExplore, 37 from SpringerLink, and 108 from Scopus. In the second filter, 46 studies were selected. The list containing selected studies is available in a technical report [5] (Appendix A - Table 5) . At this stage, the articles were fully read. Thus, of the 46 selected studies, 17 are from ACM, one from IEEExplore, seven from SpringerLink and 21 from Scopus.

[5] https://figshare.com/s/b78cc4879a2d5bd904b1

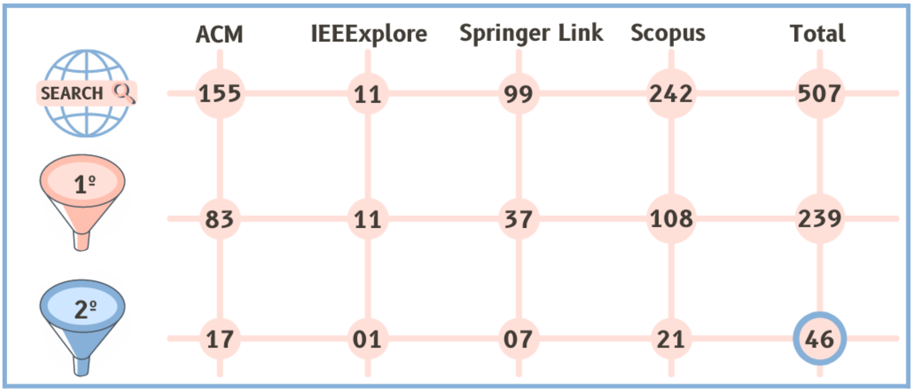

Fig. 1. Articles returned and selected in the 1st and 2nd filter.

Table 4 present an overview of the results of the quantitative analysis based on the number of studies related to each answer to the subquestions. In addition, in subquestion SQ6, some information was omitted and placed in the category “Others” because there were many answers. In some subquestions, the sum of the percentages does not result in 100%, as the studies could be classified in more than one of the subquestion’s options. The results of this subquestion are presented in detail in Subsection 4.5.

Table 4
Overview of quantitative analysis.

Research sub-questions	Possible answers	Paper	Percentage (%)
SQ1. Was block programming used at school?	No	9	19.57
	Not mentioned in the study	9	19.57
	Yes	28	60.87
SQ1.1 In which learning space is block programming being used to support Education 4.0?	Maker Laboratory	1	2.17
	Extension activities of universities in partnership with schools	2	4.35
	Virtual Learning Environment (VLE)	2	4.35
	Computer Laboratory	4	8.70
	Summer Camp/ Workshops/ Events	6	13.04
	Classroom	9	19.57
	The article does not specify the learning space	9	19.57
	At school, but not specified learning space	13	28.26
SQ2. Who is responsible for teaching block programming?	Computer Instructor	5	10.87
	Teacher	8	17.39
	Researchers	36	78.26
SQ3. Was there training in using the block programming tool? If yes, who received training?	Teacher	4	8.70
	Student	19	41.30
	Not mentioned in the study	24	52.17

4.4.1. *Publication year and venue*

Figure 2 presents an overview of the year of publication of the studies selected in this SMS. The term Education 4.0 had its first mention in 2015 (Scheer, 2015 *apud* Ciolacu et al., 2017). For this reason, the studies returned in this SMS are from that year on. Between 2016 and 2018, the number of articles on block programming and/or computational thinking in High School in the context of Education 4.0 increased, reaching its peak in 2019. There were not many studies returned in 2020 because the data collection period for this SMS was March 2020.

The places where the studies were published were also considered, being: *Journal*, *Conference Workshop*, and *Symposium*. Regarding the studies published in Journals, three studies were published in *Education and Information Technologies* (EIT), Two studies were published in *Computational Applications in Engineering Education* (CAEE); *Educational Technology Research and Development* (ETRD), *IEEE Access* (IA), *International Journal of Technology Enhanced Learning* (IJTEL), *Journal of Science Education and Technology* (JSET)) and *Transactions on Computing Education* (TOCE). Finally, one study was published in *Advanced Scientific Engineering Information Technol-*

Research sub-questions	**Possible answers**	**Paper**	**Percentage (%)**
SQ4. To which target audience and high school grade are block programming tools being used?	Teacher	8	17.39
	Student	38	82.61
	3rd grade of High School/K-12 (17 and 18 years old)	9	19.57
	2nd grade of High School/K-11 (16 years old)	11	23.91
	1st grade of High School /K-10 (15 years old)	14	30.43
	Not mentioned in the study	31	67.39
SQ5.In which class were block programming activities carried out?	Biology	1	2.17
	English	1	2.17
	Programming Logic	1	2.17
	Multidisciplinary	1	2.17
	Social or Human Studies	2	4.35
	Physics	3	6.52
	Computer Science	6	13.04
	Science	8	17.39
	Mathematics	8	17.39
	Not mentioned in the study	24	52.17
SQ6. What tools support Education 4.0 in High School?	Engage	2	4.35
	Micro:Bit	2	4.35
	LEGO Mindstorms	2	4.35
	Blocky	3	6.52
	PencilCode	3	6.52
	NetsBlox	3	6.52
	Alice	3	6.52
	Snap!	5	10.87
	MIT App Inventor	6	13.04
	Scratch	13	28.26
	Others	14	30.38
SQ6.1 What are the emerging technologies can be used to support Education 4.0?	Augmented Reality	1	2.17
	3D Modeling	1	2.17
	3D printing	1	2.17
	Virtual Simulation or 3D Simulation	3	6.52
	Cloud Computing	4	8.70
	IoT	8	17.39
	Digital Games	8	17.39
	Educational Robotics	11	23.91
	Not mentioned in the study	22	47.83

ogy (ASEIT), *Computers in Human Behavior* (CHB), *Computer Systems for Future Generation* (FGCS), *Journal of Autism and Developmental Disorders* (JADD), *Journal of Computing Sciences in Colleges* (JCSC), *Journal of Parallel and Distributed Comput-*

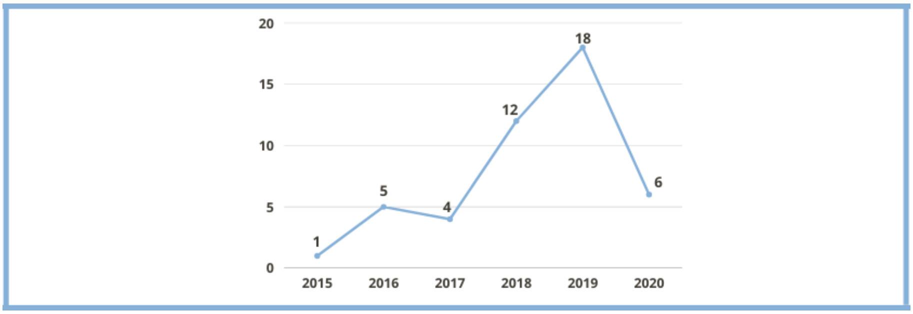

Fig. 2. Year of publication of the studies selected in this SMS.

ing (JPDC), *Technology, Knowledge and Learning* (TKL) and *Transactions on Emerging Topics in Computing* (TETC).

Regarding *Conferences*, three studies were published in *Interaction Design and Children* (IDC). Moreover, one study was published in the *Conference on Human Factors in Computing Systems* (CHI), *Computer Supported Collaborative Learning* (CSCL), *International Conference on Computer Supported Education* (CSEDU), *Engineering Education Conference* (EDUCON), *International Conference on Education Technology and Computers* (ICETC), *International Conference on the Learning Sciences* (ICLS), *Innovation and Technology in Computer Science Education* (ITiCSE), *ACM Technical Symposium on Computer Science Education* (SIGITE) and *Technological Ecosystems for Enhancing Multiculturality* (TEEM).

Of the studies published in Workshops, two were published in the *Blocks and Beyond Workshop* (B&B), followed by *CEUR Workshop* (CEUR) and *Workshop in Primary and Secondary Computing Education* (WiPSCE) with one study published in each.

Finally, about the studies published in Symposiums, six studies were published in the *ACM Technical Symposium on Computer Science Education* (SIGCSE) and one in the *International Symposium on End User Development* (ISEUD) were identified.

4.5. *Results of the subquestions*

4.5.1. *Using block programming at school (SQ1)*

The results related to the use of block programming at school (SQ1) are represented per study in Appendix B (Table 6, available in a technical report[6]). Of the studies selected in this SMS, 60.87% (N = 28) were carried out in the school environment. The use of programming in a school environment can be done in different ways. One of the ways to insert it in school is through teaching programming, including software development, the development of computational thinking, and programming logic. Another way is through the practice of programming in an interdisciplinary way, covering the relationship between theory and practice of contents of the common high school curriculum in the context of block programming as mentioned by Oro et al. (2015). Cardoso and Antonello (2015)

[6]https://figshare.com/s/b78cc4879a2d5bd904b1

also state that it is possible to work with several emerging technologies, for example, with Robotics or Digital Games.

The authors Buffum et al. (2016) conducted a study using the *ENGAGE* [7] software in a game-based learning environment about Computer Science. Participated 18 students and 04 professors, divided into three steps: 1) Construction of the curriculum together with the teachers, where a character was created in the role of a student with the idea that she is a computer scientist responsible for solving a certain socially relevant mystery; 2) Teacher training, where later they could teach the curriculum developed in the first stage, integrating it into their classes; and 3) Implementation of the study with students. In relation to the activities carried out by the teachers, they demonstrated that they were satisfied with the training they received and appreciated how the game-based learning environment supported instruction. Regarding students, most responded positively to the experience, and classroom observations showed student engagement. In addition, the authors reported that preliminary studies revealed significant student learning gains.

Furthermore, 19.57% (N = 9) of the studies were not carried out in a school environment. For example, Gonçalves et al. (2019) described a study that took place in a summer camp format and involved 16 students. This study had a partnership between the school and the academy. The activities were carried out at the Polytechnic Institute of Bragança (IPB) in the Control, Automation and Robotics Laboratory of the School of Technology and Management, and covered the teaching of Robotics and IoT. These authors highlighted the importance of students knowing how to deal with block programming tools and the need to work with them at school. Finally, 19.57% (N = 9) of the studies do not mention the location where block programming took place.

The result of SQ1 shows that few studies are being carried out outside the school environment. One of the reasons for this may be the lack of partnerships between Universities and Schools. A possible solution to this situation is: a) to conduct actions that discuss the role of the University in the school community, b) the University to design strategies to support the school in supporting the teaching and learning processes, c) enable the teacher to use new strategies for teaching, for example, technical visits, activities in the *maker* laboratory, participation in University projects, and d) providing students with knowledge of other environments and new experiences through these initiatives and partnerships. Furthermore, conducting activities in an external environment allows students to be the author of their own knowledge based on new experiences, diversifying their learning routine, and encourages group experience and discoveries of their own skills.

4.5.2. *Learning spaces (SQ1.1)*

The learning spaces (SQ1.1) were classified into: a) *Maker* Laboratory, b) University extension activities in partnership with schools, c) Activities on online platforms (when via *e-mail* or Moodle), d) Computer Lab, e) Summer camp/events/ workshops, f) Classroom, g) learning space not specified (when the activity does not mention the environment or the location), and h) at school, but the learning space is not specified. These categories

[7] http://projects.intellimedia.ncsu.edu/engage/

were originated from the studies selected in this SMS. This result can be found in detail per study in Appendix B (Table 6 available in a technical report[8]).

Thus, a study carried out in the *Maker* laboratory by Lee and Malyn-Smith (2020) was noticed. This environment can be a way to develop learning through "learning by doing". In addition, these environments can provide interdisciplinary learning, as they are usually equipped with computers, drones, 3D printers and other technologies. This type of environment allows students to exercise their creativity and be a shared environment that allows them to work and seek solutions collaboratively. However, even with the benefits, the lack of studies in this environment is perceived.

In addition, two studies related to the university's extension activities in partnership with high schools were noticed. This type of partnership between Higher Education and Basic Education institutions is important and can provide learning for both involved. The university can disseminate the knowledge gained from the results of their studies to those involved and/or provide assistance to the school community as well as learn from the needs of these schools. On the other hand, schools can enjoy new pedagogical approaches in the most diverse areas of teaching, providing students with new learning experiences, which allows the exploration of new knowledge and contact with the university environment. Even if the benefits are two-way, there is a need to further encourage building these partnerships. An example of block programming activity in which there was this partnership between university and school was reported in Eguiluz et al. (2020). In this study, the extension activities aimed to understand how programming beginners use computational thinking to solve a set of challenges through block programming. To solve the challenges, the platform *Kodetu* [9] was used. Participants needed to define astronaut's movements to bypass existing blocks. Regarding the influence of the characteristics of the participants, it was noticed that girls tend to abandon challenges before boys, presenting greater difficulty. Regarding the authors' perception, it was noticed that the extension activity helped to better understand the first steps taken by beginning programmers and helped define strategies to improve student learning.

On the other hand, Virtual Learning Environments (VLE) can support teachers in their training, facilitating their personal development through non-presential training, and teaching and learning processes of programming blocks for students. Two studies working with learning environments were identified in this SMS. An example that can be cited is the use of the Moodle platform for the training of teachers, which was carried out by Haduong and Brennan (2019), who worked daily for 21 days programming in blocks and sharing solutions. The activities consisted of creating a project on *Scratch* [10], sharing their work with the *online* community (Twitter and *Scratch* community) and reflecting about your learning. Thus, in the development part, the participants could create solutions in the most diverse ways. In the reflection part, participants posted their contributions, reporting on what was challenging, what strategy was used, and what was most satisfying.

[8] https://figshare.com/s/b78cc4879a2d5bd904b1
[9] http://kodetu.org/
[10] https://scratch.mit.edu

Regarding the space used as the computer lab, four studies were observed. In the laboratory, students can have contact with computers and other digital equipment and peripherals, which allow working on digital culture and inclusion, and can help in the teaching and learning processes of block programming. In this sense, Mørch et al. (2019) carried out a study in which 19 students participated in a computer lab. In this study, qualitative analysis methods were applied to teach and engage students in the Science discipline. Every two weeks, students attended a school close to the one they were studying regularly to participate in the course. The course covered block programming with the tools *Blockuino* [11] and *Micro:Bit* [12]. Students showed interest in block programming and commented that it would be good to relate the subject to extracurricular activities. In addition, students perceived an interdisciplinary relationship between block programming and common course curriculum content and commented that they could learn about velocity and acceleration and other similar subjects, such as calculating values and other formulas. In addition, the authors found evidence that block programming can aid learning and that students prefer to learn by contextualizing content with real-life situations.

Of the works carried out in the form of summer camps, events or *workshops*, six studies were perceived. These types of activities can allow the first contact with block programming, Educational Robotics, and other activities that facilitate learning. In this sense, Glenn et al. (2020) carried out a study with 14 students in a workshop format, where Educational Robotics, IoT and Augmented Reality (AR) activities were developed collaboratively using block programming. Students used *StoryMakAR*, an AR-IoT storytelling toolkit that combines physical construction, electronics, and an AR environment by blending physical and virtual content. The authors mentioned that the students had fun while building the devices and that they enjoyed creating physical-virtual interactions.

About the classroom, nine studies were perceived. The use of block programming in the traditional learning space can encourage students to learn and develop other programming activities combined with Robotics, IoT, through content from subjects such as Mathematics, Physics and others. In this sense, Fronza et al. (2019) developed an activity with the participation of 30 students. The block programming software used were *Scratch* and *MIT App Inventor* [13]. The goal was to develop problem-solving skills in the student, from computational thinking strategies with activities related to Software Engineering, where students needed to develop software for mobile devices in life cycle (Analysis, Implementation and Testing). Thus, fundamental concepts of algorithms and programming were taught and, as the students developed, the activities became more complex. The activities were related to the disciplines of Mathematics and Science. There was the development of *software* and *hardware* activities, simulating a semaphore using *Scratch*, integrating the *Raspberry Pi* as *hardware* target.

Finally, it can be noted that nine studies do not mention the space outside the school environment and thirteen do not describe the school space where block programming took place. The results of SQ1.1 demonstrate that few studies report activities in other spaces

[11] https://blockuino.skaperiet.no/
[12] https://microbit.org/
[13] http://ai2.appinventor.mit.edu

than the common classroom. It is worth encouraging the use of other learning spaces, such as the *maker* laboratory, which can enhance and enrich student learning, making them technology producers and not just consumers. In this sense, public investment is needed, also demanding training of users of these spaces (teachers, students and other interested parties) in the use of technologies made available by the *maker* laboratory. Extension activities, on the other hand, provide students with new learning experiences and work with the student's protagonism from their active participation with new pedagogical approaches in the most diverse areas of teaching. VLEs enable learning and the development of extracurricular activities. Summer camp activities, events and *workshops* also provide students with active learning and contact with other students and knowledge, providing the exchange of knowledge and collaboration. The computer lab provides activities to learn concepts related to disciplines through Digital Games and other types of educational software or learning objects. Thus, it is believed that activities that explore diverse environments provide students with new experiences, stimulate creativity and the search for knowledge, and develop new skills.

4.5.3. *Responsible for teaching block programming (SQ2)*

Those responsible for teaching block programming identified in this SMS were: a) researchers; b) teachers accompanied by researchers; and c) computer instructor, also accompanied by researchers. The result of this subquestion can be found in detail per study in Appendix B (Table 6[14]).

It can be seen that the researchers conducted 78.26% (N = 36) of the studies. It is believed that this result occurs because researchers are interested in teaching concepts of Computer Science such as Programming Logic, Logical Reasoning and Computational Thinking, Educational Robotics, IoT and others. In addition, researchers need to validate and/or experience their research in real-world environments such as the classroom. An example of a study where researchers conducted the teaching process was carried out by Weintrop and Wilensky (2015). The study aimed to compare three programming classes: 1) read-only (read encoding in *JavaScript* programmed in blocks); 2) read and write (read the program converted to *JavaScript* and create new command blocks); and 3) graphical (block programming only). The study was conducted with students from grades K-10 (1st grade of high school), K-11 (2nd grade of high school) and K-12 (3rd grade of high school), where the classes were separated by groups and type of teaching programming. Each grade was assigned a programming modality (only reading, reading and writing and graphical) and followed the same curriculum. The researchers mention that the study's findings can aid in the design of text-based programming tools, as students often cited the navigability of the block-based environment as a feature that makes it easy to use and prepares them for the transition of text-based tools.

In addition, 17.39% (N = 8) of studies were performed by teachers accompanied by researchers. The inclusion of block programming in teacher-mediated high school should be taken into account in the application of didactic sequences that encourage student development in all areas of education and also stimulate computational thinking, which is

[14] https://figshare.com/s/b78cc4879a2d5bd904b1

a 21st-Century skill as described by Vinayakumar et al. (2018b). An example of a study where the teacher had an active role in the teaching and learning process of programming in blocks and followed up by researchers was conducted by Seralidou and Douligeris (2019). In this study, teachers passed on content that included the design and programming of Android apps for mobile devices. Students worked in groups of two or three people on each computer. Teachers gave students activity sheets that mentioned the time available to carry out the activity and then observed the students during each class. The researchers followed the high school teachers to assess the methodology used in the classroom, from both the teacher's and the student's point of view, and it was noticed that students and teachers have a positive attitude about the implementation and use of the *s*oftware MIT App Inventor. In addition, teachers noticed that students were engaged during activities using *MIT App Inventor*.

Finally, 10.87% (N = 5) of the studies were carried out by a computer instructor, also accompanied by researchers. The researcher can assist in developing the computer instructor in relation to their training, which, in turn, can help teachers from other disciplines provide a better use of block programming software in the teaching and learning processes. In this way, they can significantly enhance the construction of interdisciplinary knowledge, computational thinking and programming logic. An example of a study where the person responsible for teaching block programming was a Computer Instructor is presented by A-Ghamdi et al. (2016). In this study, the Computer Instructor had the role of teaching programming concepts in blocks, namely: variables and methods, lists, *if* declaration, *loops* and databases. The activities were carried out over six days and the tool used was the *MIT App Inventor 2* (AI2). On the first day of *workshop*, the AI2 interface was presented, used throughout the event. On the second day, the concepts of variables and methods were explained through active learning techniques. On the third day, lists were introduced by applying the activity "Searching for the treasure". On the fourth day, *if* and *loops* repetition concepts were presented. Two activities were carried out on the fifth day, a database and a guessing game. On the sixth and last day, the students dedicated themselves to their final projects. In this study (A-Ghamdi et al., 2016), from observations, the authors provided *insights* on AI2, and improvements and limitations over the previous version, and on how to build a better curriculum for the teaching block programming.

The results of SQ2 demonstrate that at least one researcher is involved in all studies. It is believed that this result occurs because researchers are mostly interested in publishing the results of studies involving block programming. Some of the researchers worked actively in the process of teaching block programming, assisting the teacher or computer instructor, or just accompanying these professionals to support the application of a teaching methodology.

4.5.4. *Block programming training (SQ3)*

In this SMS it was noticed that 41.30% (N = 19) of the studies mention that students received training in some block programming tool. In this sense, introductory activities, through videos and/or tutorials, allowed the student to become familiar with the block programming tools and understand concepts such as programming logic. An example of a study that had introductory training in block programming was carried out by Eglash et al. (2019). 48 students participated in the activities, and the activities consisted of building twelve-step *Anishinaabe* (indigenous tribe) arches in *CSnap* [15]. Familiarization with the tool allowed students to watch an animation with the steps to build the program and then move on to developing open design activities. They explore the software's limitations and features and gradually modifying the original algorithm by experimenting with trial and error to perform simulations for the physical rendering of arcs *Anishinaabe*. Afterward, students were divided into four groups and challenged to share what they learned about historical concepts and cultural background about tribes.

Only 8.70% (N = 4) of the studies mentioned that teachers received training in block programming. It is believed that this result is related to a large part of the studies being conducted by the researchers themselves and the learning being centered on the student. One way to support the training process of these teachers can be through learning objects that show how block programming can be used in the classroom. A study that had this concern in relation to teacher education was carried out by Jocius et al. (2020). 116 teachers from the Humanities, Sciences and Mathematics participated in this study, and used the platform *Snap!*[16]. Teacher training began with the presentation of computational thinking (PRADA – *Pattern Recognition, Abstraction, Decomposition, and Algorithms*), followed by code infusion sessions, in which the learning structure *Use-Modify-Create* was used, allowing the use, modification and creation of new codes during the learning process. Teachers also performed a collaborative activity to map and describe the patterns of the PRADA elements, create a lesson plan for their subject, and suggest activities that could be implemented in the classroom. Finally, teachers created complementary teaching materials, such as slides, links and handouts, to present what they learned to other participants, guests and school administrators.

Finally, 52.17% (N = 24) of the studies do not mention whether there was training before using the block programming tool. The result of this subquestion can be found in detail per study in Appendix B (Table 6[17]). The results of SQ3 show that few studies mention the training of high school teachers in block programming. In undergraduate degree courses in the Exact area, there are Computer Science disciplines that encourage tools such as text editors and electronic spreadsheets. Other courses in the Engineering area have disciplines more focused on the Computing area, using programming. There is an opening for encouragement and insertion of block-based programming teaching in the latter case. In other undergraduate courses, it is not common to have disciplines that encourage programming in teaching and learning processes. Thus, it is necessary to allow teachers to know the tools such as block programming. These features can encourage computational thinking and facilitate the learning of content practically for students. Therefore, it may be possible to associate this professor's course content with block programming. This initiative may help make the curriculum more flexible, allowing the teacher to follow the changes that block programming tools bring to society and education.

[15] https://csdt.org/applications/38/run

[16] https://snap.berkeley.edu/

[17] https://figshare.com/s/b78cc4879a2d5bd904b1

4.5.5. *Target Audience and High School Grade (SQ4)*

This subsection presents an overview of the target audience and high school grades that participated in the teaching and learning processes of block programming in the studies selected in this SMS. This result can be found in detail per study in Appendix B (Table 6[18]). In relation to the target audience, of the 46 studies, 38 were carried out with students and 8 were carried out with professors. In addition, 1 study mentions that the target audience was high school, without specifying whether it was with a teacher and/or student. Of the studies that involve the teacher, all are related to their training in block programming. Thus, it is possible to reaffirm that teacher education initiatives are few compared to student education. An example of a study with teacher and student was carried out by Buffum et al. (2016), in which researchers trained teachers to use game-based block programming tools that can be used for various contexts, and later these teachers taught what they learned from their students.

In relation to the High School grades, 14 studies were conducted in the 1st grade of High School (K-10), 11 studies in the 2nd grade of High School (K-11), 09 studies in the 3rd high school grade (K-12), and 31 studies did not mention which high school grade block programming was worked on. It is believed that teaching block programming in high school can improve the development of the learning process in general, stimulate mathematical learning, develop logical reasoning, interpretation, problem-solving and critical thinking, encourage creativity and teamwork, and allow to work on contents in an interdisciplinary way. In this sense, a study carried out with the three grades of High School, integrating the teaching of computational thinking and STEAM (*Sciences, Technology, Engineering, Arts and Mathematics*) was carried out by Lee and Malyn-Smith (2020). The activities took place in a school's *MakerSpace* and used the *Scratch* tool. Students combined design and engineering representation activities and drew a circuit diagram for the indicator. Regarding the algorithms, the scalability was worked on where students developed parallel processing algorithms to mass-produce a construction of LEGO Mindstorms bricks [19]. On the software development performed through block programming, the activity involved designing and constructing an automated device to count paper clips. In the context of Science, the collection and analysis of DNA substrand data, sensor data, and historical texts were worked on, and also involved block programming.

The results of SQ4 show that most studies were carried out with students and a large part does not mention which high school grade the programming in blocks was carried out. Of the studies that mentioned the high school grade, most were carried out in the 1st year (K-10), followed by the 2nd year (K-11) and, finally, in the 3rd year (K-12). The few studies identified in the 3rd grade of high school might be because this grade is more directed towards preparing the entrance exams of university. It was also noticed that few studies were conducted with teachers. Teacher training initiatives can strengthen the use of block programming, as they promote the teaching of computational concepts, and the development of skills such as creativity, autonomy, logical reasoning, computational thinking, problem-solving, teamwork and the protagonism of both the student and the teacher. All these skills are necessary for the 21st-Century.

[18] https://figshare.com/s/b78cc4879a2d5bd904b1

[19] https://education.lego.com/

4.5.6. *Classes where block programming was worked on (SQ5)*

The results of this subquestion can be found in detail per study in Appendix B (Table 7[20]). The subjects that had the content related to block programming were: Mathematics (8 studies), Science (8 studies), Computer Science (6 studies), Physics (3 studies), Social and/or Human Studies (2 studies), Programming Logic (1 study), English (1 study), Biology (1 study) and 1 Multidisciplinary study. The classification of classes is based on the nomenclature used by the authors of the selected articles in this SMS.

An example of a study carried out in the Mathematics class was carried out by Zainal et al. (2018). The activity was developed through a module and based on the four stages of the experiential learning cycle of Kolb's theory (Concrete Experience, Reflective Observation, Abstract Conceptualization and Active Experimentation). These stages were composed of five main activities, namely: 1) Watching videos (Concrete Experience), where videos are shown to students to help them understand the main idea of the activity and the mathematical concepts; 2) Read guide modules (Reflective Observation), where after students have a new experience with the video, they gain a deeper understanding of the experience and further improvement through the concepts described in the Guidebook Module. Also, to test the students' understanding, they used their newly acquired knowledge to do an angle exercise. In this way, students were able to observe how the mathematical concept is used to determine direction through the guide module; 3) Assemble robotic components (Abstract Conceptualization), where students need to think about how to transform the new knowledge acquired into logic, organizing visual blocks in the block programming software; 4) Programming via blocks (Active Experimentation), where students performed some experiments and played with logic, changing angles and degrees; and 5) Play a robotic game. This module developed by the authors incorporated concepts such as angles (degrees), robots, visual programming and a game. The results showed that the methodology used can efficiently increase students' interest.

Hutchins et al. (2020) used *Design-Based Research* (DBR) to develop, evaluate and refine the *Collaborative Computational* STEM (C2STEM) environment through block programming in Physics class. An evidence-centric design was used to decompose the overall curriculum into four modules: (1) 1D land transport involving constant acceleration and deceleration, (2) 2D constant velocity motion for transport across a river, and (3) motion accelerated 2D (with gravity as a factor) for package delivery to a remote area using a flying drone; and (4) 1D and 2D motion with forces (including static and dynamic friction). The tool used was *Netsblox*[21]. The results obtained from this experimentation were the perception that students who worked with C2STEM developed a better understanding of Physics and Computational Thinking concepts and practices than students who learned through a traditional curriculum. This implies that activities involving programming in blocks together with a discipline can be more attractive in the teaching and learning processes and provide a more playful environment.

An example of a multidisciplinary initiative was carried out by Dong et al. (2019), in which they conducted activities to train teachers in block programming. Teacher training

---

[20]https://figshare.com/s/b78cc4879a2d5bd904b1

[21]https://netsblox.org/

was developed mainly for Mathematics, Science, and Humanities. Some projects were developed for all these classes. The activities were carried out using the PRADA model and the *Snap!* tool. Moreover, it is noticed that 24 studies do not mention the teaching and learning of block programming in a specific class in High School. These studies cite the teaching of computational concepts such as computational thinking using block programming.

In the results of SQ5, most studies with block programming involved class from the Exact area (18 studies, see Table 4), showing a gap in the use of block programming in class from other areas, such as Arts and Portuguese in an interdisciplinary way. Thus, it is interesting to contextualize the difference between multidisciplinarity and interdisciplinarity and how they can be worked on in the classroom. According to Nogueira (2002), in the multidisciplinary approach, each class contributes to the student's learning in a fragmented way, with its contents pertinent to their field of knowledge individually and without the concern of interconnecting the subjects with each other. On the other hand, interdisciplinarity aims to promote the integration of content from different disciplines so that the areas of knowledge complement each other and serve as support for learning from each other. Thus, there is still a need to use methodologies that allow working with block programming in an interdisciplinary way.

4.5.7. *Tools that support block programming teaching and learning processes (SQ6)*

A total of 24 tools were identified, namely: 1) *Scratch*, 2) *MIT App Inventor*, 3) *Snap!*, 4) *Alice*, 5) *NetsBlox*, 6) *PencilCode*, 7) *Blocky*, 8) *LEGO Mindstorms*, 9) *Micro:Bit*, 10) *Engage*, 11) *Scratch4Arduino* (S4A), 12) *ChoiCo*, 13) *Code.org*, 14) *mblock*, 15) *Blockuino*, 16) *AgentsCubes*, 17) *DeviceMakar*, 18) *Inteliblox*, 19) *Csnap*, 20) *NetTango*, 21) *Kodetu*, 22) *OzoBlocky*, 23) *Roblock*, and 24) *Tickle*. The results of this subquestion can be found in detail per study in Appendix B (Table 8[22]).

The most used tools for teaching block programming were *Scratch* (13 studies), *MIT App Inventor* (6 studies) and *Snap!* (5 studies). *Scratch* stands out for being one of the pioneering tools for encouraging block programming. Scratch has sound and image items that allow you to create interactive stories, animations and games that can contribute to the development of creativity, logical reasoning and computational thinking (Vinayakumar et al., 2018a). The versions work on *desktop* and *web* and have some hardware-based extensions like the *Micro:Bit*[23] board and the LEGO Mindstorms EV3[24]. Thus, this tool makes it possible to work with emerging technologies such as Robotics and IoT. The *MIT App Inventor* is an environment that, in addition to enabling the learning of programming and the development of computational thinking, is specific for the development of apps for *smartphones* and *tablets*. In addition, it was also developed in the *MIT Media Lab* research lab of the *Massachusetts Institute of Technology*. *Snap!* is based on *Scratch* and was developed at the University of California at Berkeley. It also has extensions for technologies

[22] https://figshare.com/s/b78cc4879a2d5bd904b1

[23] https://scratch.mit.edu/microbit

[24] Developer's page: https://en.scratch-wiki.info/wiki/LEGO_MINDSTORMS_EV3_Extension

like Lego and Arduino[25]. In addition, this platform allows the creation of several educational projects related to Mathematics and Natural Sciences, simulations of experiments and recording of content with animated presentations.

An example of a study using the *Scratch* was carried out by Lazarinis et al. (2019). The study was performed in *online* course format in Moodle platform and aimed at teachers. The methodology used was instructional, and learning was based on the example and problem-solving. Teachers had to complete all assignments at the end of each unit to complete the course. Teachers' progress was monitored through Moodle reports. Teachers needed to achieve a passing score of greater than or equal to sixty to proceed to the next unit. In addition, they had to submit an assignment at the end of the course, which the researchers manually assessed. During the course, teachers took their questions through forums.

Scaradozzi et al. (2019) performed a study using *MIT App Inventor*. The study aimed to identify and analyze the main activities of students. The activities took place over a 12-week period, with the first eight weeks of the curriculum being learning about the *MIT App Inventor* developing initial apps as an introduction to the system. And over the past four weeks, students have worked in pairs or triads to design and build their own apps. In this way, the students created apps for mobile devices to help to raise awareness about the pollution of a river that runs through the city. According to the authors, it was noticed that students tried to discover the codes on their own or through discussion with their group members and hardly reviewed the codes they had built previously to support the creation of new applications.

Regarding the *Snap!*, Kahn et al. (2018) carried out a study that presents a learning process for students about the concepts of Artificial Intelligence (AI) through block programming. The study aimed to analyze the feasibility of this learning process, divided into 4 phases: 1) Preparation of the learning process: to prepare the computer so that each student can connect to the internet, define the position that the student can work individually without limiting their access to colleagues so that they can discuss the topic; 2) Introduction: concepts of *Snap!* should be addressed and students can discuss aspects of AI; 3) Learning process: the student will learn how to use computer speech using speech synthesis. This phase refers to the student making a machine or computer act like a human; and 4) Observation and measurement: refer to the assessment of the learning process, which can be performed in two ways: an assessment and observation test. According to the authors, the students enjoyed the learning process. Additionally, block programming has helped to increase student confidence in AI programming.

The results of SQ6 demonstrated the number of tools identified in this SMS. All these tools have their particularities and can be used with emerging technologies such as Robotics, IoT, Digital Games development, Augmented Reality, and others. Furthermore, these block programming tools can be used in high school collaboratively and playfully to develop computational thinking or other 21st-Century skills.

[25] https://snap.berkeley.edu/extensions

4.5.8. *Emerging Technologies (SQ6.1)*

The emerging technologies identified in this SMS were used with the tools cited in the previous subsection. These technologies were worked on in block programming teaching and learning processes. Technologies are categorized into Educational Robotics, Digital Games, IoT, Cloud Computing, Virtual Simulation or 3D Simulation, 3D Modeling, 3D Printing and Augmented Reality. The results of SQ6.1 are in Appendix B (Table 7[26]).

Educational Robotics was identified in eleven studies among these emerging technologies. This indicate the increase in initiatives that use robotics in schools, enabling the student to build a robot with the most diverse interactions, such as carrying objects, communicating by light signals, and emitting sounds. In this sense, Orlando et al. (2020) perform a robot programming work with students using *Blocky*[27] and EV3 LEGO Mindstorms. The activity consisted of dropping the ball on an inclined plane or making a car travel a distance in uniformly accelerated motion. Students tested the programmed robot and reflected on the physics concepts involved in the experiment.

Eight studies were identified regarding the use of block programming to develop Digital Games. Digital Games appeared as a didactic resource, containing characteristics that enable students to learn in a playful, collaborative way, both inside and outside the classroom. In Filvà et al. (2019), students formed groups of two or three people, received theoretical training on the block programming platform *Scratch*, and performed a practical activity, which was to create a *Scratchroom*. Participants created scenarios, players, objects and sounds in *sprites* format, developing the computational logic behind each action in a playable and fun way.

Regarding studies involving block programming with IoT, eight studies were identified. IoT also enriches the student's protagonism, as it allows a better understanding of the contents learned at school through its application in contexts and elements of the real world; for example, calculate how long each LED of an Arduino should stay on. An example of a study using block programming and IoT with Robotics was carried out by Papavlasopoulou et al. (2019). The study was based on DBR and coding experiments based on constructionism. An extension of *Scratch* called *Scratch* for Arduino (S4A) was used, with extra blocks to control the robots, in which the Arduino was attached to connect them to the computer. As a result, students were more engaged, were able to adopt deliberative thinking, understand and imitate mechanical thinking during coding.

Regarding block programming and Cloud Computing, four studies were identified. Cloud Computing supports some of the ideas and practices of computational thinking that are emphasized in the Computer Science curriculum, such as communication, collaboration, and cybersecurity, and supports collaborative editing of multiple computers, allowing students to work together on shared projects using their own computers. An example of a study involving Cloud Computing was carried out by Broll et al. (2018) using the *NetsBlox* tool. This tool allows users to create distributed applications. The study aimed to teach networking and cryptography concepts to high school students. Two studies

[26] https://figshare.com/s/b78cc4879a2d5bd904b1

[27] https://developers.google.com/blockly

were conducted for teaching computer programming, covering concepts of networks, remote procedure calls, messages, and a free theme for students to work on a project of their choice. The authors mentioned that overall, the study results showed promise for introducing distributed computing. Despite the short duration of the studies, the students showed a significant improvement in computational thinking and computer network assessments.

In general, the results of SQ6.1 present what are and how emerging technologies are being worked on teaching block programming in high school. The use of these block programming tools, computational thinking and/or programming logic combined with emerging technologies work the confidence, collaboration and creativity of students, allowing them to develop, simulate and build real solutions through experimentation, promoting protagonism and learning autonomy of the students. However, there is still a gap in the use of actions that encourage these emerging technologies by integrating block programming. Forums, free video channels, support materials can be means of exchanging information and experiences in the use of new technologies and for the development of programming activities.

5. Threats to validity

As with all SMS, there are risks that can affect the SMS results. Therefore, we sought to minimize bias through a strict protocol for data extraction. In addition, the researcher selected the articles and carried out the peer review, through two steps: 1) three researchers read the title and abstract of all articles (first filter). In case of divergences in the inclusion or exclusion decisions, an attempt was made to resolve it by reaching a consensus. If they did not reach a joint conclusion, the article was included for a next step, the second filter. For both stages, it was necessary to justify if an article was excluded; 2) in the second filter, the first researcher read all the articles and extracted the data. The other researchers verified the excluded articles with their justifications, as well as the included articles and their extractions. A threat to validity may be related to articles that did not specify the high school grade that block programming was working. In this case, we considered the age of students, when the article contained this information.

Another limitation of our SMS is not having searched in other libraries such as Web of Science and ERIC. Thus, the results found can be considered limited because they do not include publications of interdisciplinary research in the field of social sciences. Moreover, we do not include the term “block- programming”, which could influence the number of studies returned. In future work, we will seek to extend this SMS in these other digital libraries and to include the term "block-programming".

6. Discussions

The contribution of this SMS is the identification and categorization of blocks programming tools used in high school teaching and learning processes in the context of Education 4.0. Throughout the article, we analyze and discuss the use of block programming tools with teachers and/or students, the training of teachers to use the tools, and the use of block programming tools combined with emerging technologies, among other characteristics. The motivation for this SMS was to understand how these investigated characteristics contribute to developing skills and competencies, necessary for life in the 21st Century, to promote a teaching and learning process more aligned with Education 4.0.

Our SMS differs from others found in the literature. For example, Lin and Weintrop (2021) searched for tools that allow teaching programming in a hybrid way (textual and in blocks). These authors sought to analyze the tools identified in a technical way, such as the platform (web, desktop or mobile) that the tool works on, and the operating system, among others. They categorized many tools that rely on the block-based programming approach, comparing each tool against block-based and text-based programming. In our SMS, we investigated aspects related only to block programming, even if the tool also allowed textual programming. In addition, this SMS focused on verifying how block programming was taught in the context of Education 4.0. In other words, we sought to investigate whether block programming tool was combined with the use of emerging technologies, such as Robotics and Digital Games. Additionally, it was verified whether the teaching-learning processes of block programming were carried out in an interdisciplinary way, whether there was training in the use of block programming tools and who received the training; who was responsible for teaching block programming; among other features.

Our SMS also differs from the work of Weintrop et al. (2019). The authors carried out an SLR, whose objective was to analyze CS contents with a focus on: (a) Teacher Accessibility, (b) Equity (creation of effective and accessible learning opportunities) and (c) Content, which is intended for two profiles: for teachers, covers educational decisions; and, for designers, it involves creating effective, accessible and equitable learning opportunities. Weintrop et al. (2019) also mention that, in addition to access to content, there is a need to train and support teachers in the use of these tools, which is in line with our findings. In this sense, our SMS complements Weintrop et al. (2019)'s perception of the teacher's need to play an active role in the teaching and learning processes of block programming, mainly related to their training.

Another finding from our SMS was that most of the studies (N = 36) were conducted by the researchers. Furthermore, in all studies, at least one researcher was involved in the teaching and learning processes using block programming in the context of Education 4.0. According to Führ (2018), the teacher can facilitate the development of skills and competencies in students, combining pedagogical knowledge with practical use in order to bring the most diverse resources of the digital age closer to the classroom. In this sense, supporting and encouraging the teacher to use block programming in an interdisciplinary way can promote the integration of contents from different disciplines so that the areas of knowledge complement each other and support each other's learning. Thus, the importance of training and encouraging high school teachers to use block programming tools in their classes in an interdisciplinary way is highlighted.

No study was identified in the literature investigating the environments and spaces in which block programming worked. In our SMS, it was identified that few studies are being carried out outside the school environment, and few report activities in different spaces of the common classroom. In this sense, carrying out activities in an external environment allows the student to be the author of his own knowledge based on new experiences, diversify his learning routine, and encourage group experience and discoveries of his own skills. This is in line with what Hartono et al. (2018) states, that encouraging the use of other learning spaces, such as the maker lab, can enhance and enrich student learning, making them technology producers and not just consumers.

This SMS also identified that most of the studies were carried out with students and few were conducted with teachers. Therefore, teacher training initiatives are necessary and can strengthen block programming in the classroom and in an interdisciplinary way. According to Santos et al. (2019), this type of initiative promotes the teaching of computational concepts, and the development of skills such as creativity, autonomy, logical reasoning, computational thinking, problem-solving, thus allowing a teaching and learning process in the context of Education 4.0.

Our findings also show that most of the studies with block programming in the context of Education 4.0 involve disciplines in the Exact Sciences area (18 studies), showing a gap in the use of block programming in disciplines from other areas, such as Arts. Thus, there is still a need to use methodologies that allow working with blocks programming in an interdisciplinary way and involving other areas. Garcia et al. (2017) and Saraiva et al. (2020) mention that students who enroll in CS courses have a deficit in text interpretation skills since high school. These difficulties impact the understanding and resolution of algorithmic problems in CS courses.

This SMS also presented how emerging technologies are being worked on in teaching block programming in high school. According to Cardoso and Antonello (2015) and Souza Rios et al. (2019), the use of these block programming tools, computational thinking and/or programming logic, combined with emerging technologies, works on students' confidence, collaboration and creativity, allowing them to develop, simulate and build real solutions through experimentation. In addition, it promotes the protagonism and autonomous learning of students. Our SMS identified a gap in the use of actions that encourage using these emerging technologies integrated with block programming.

Finally, through this SMS it was noticed that, due to the number of block programming tools available, the teacher may be confused in choosing the most appropriate tool for his/her context of use. Therefore, it is necessary to investigate and develop strategies to support the teacher in choosing block programming tools, as well as to support the training of this teacher.

7. Conclusions and future work

This article presented the results obtained in an SMS to identify which block programming tools can be used in High School in the context of Education 4.0. Searches for studies were performed automatically in four digital libraries. Of the 507 articles returned, 46 met the inclusion criteria and were extracted. The protocol for executing this SMS was defined according to the Kitchenham and Charters (2007), and we tried to answer 8 research sub-questions. The answers achieved in each sub-question provided an overview of the use of block programming in the context of Education 4.0 in High School.

The data show that: **(SQ1)** most of the block programming teaching initiatives take place at school. Few studies are carried out outside the school environment. This demonstrates that initiatives in this context need to be discussed, as it is believed that the environment outside the school can allow the student to build knowledge and develop skills based on new experiences; **(SQ1.1)** few studies are performed in *maker* laboratory, computer laboratory, and others. This demonstrates that students have little access to these spaces that can enable new experiences, develop skills such as creativity, critical thinking, and computational thinking. It is believed that the little use of some spaces is due to the lack of them at school; **(SQ2)** in all studies at least one researcher is involved in the use of programming tools in high school and can contribute to the advancement of Education 4.0. Some of the researchers actively worked in the use of block programming tools, others help the computer teacher or instructor, or just accompany these professionals in conducting classes with block programming; **(SQ3)** few studies mention the training of high school teachers in block programming. In this sense, initiatives need to be encouraged and shared in scientific articles in order to disseminate experiences and results of the teaching processes used; **(SQ4)** most of the studies were carried out with students and most of them do not mention which high school grade the programming in blocks was carried out. It was also noticed that few studies were conducted with teachers. This demonstrates a need for initiatives that support this teacher in the use of block programming, especially in an interdisciplinary way. Block programming can help students develop 21st-Century skills, such as logical reasoning, computational thinking, teamwork, autonomy, creativity, and others; **(SQ5)** there is still a need to use methodologies that allow working with block programming in an interdisciplinary way. This may be a consequence of the lack of incentive regarding the possibilities of applying block programming in classes. Thus, they end up not using methodologies that support the contextualization of their classes through block programming; **(SQ6)** of the block programming tools identified in the SMS, the largest use was *Scratch*, *Mit AppInventor* and *Snap!*. Moreover, some of the identified tools can be used with emerging technologies to support student learning, such as Robotics, Digital Games, IoT, and others; **(SQ6.1)** most studies do not mention the use of emerging technologies, which demonstrates a gap in the use of this type of technology, which is important for the Education 4.0 context. Emerging technologies can enable the student to work with confidence, solve problems, think critically to create innovative solutions and processes. For this reason, they are considered important as they help prepare students for the challenges of the 21st-Century, such as dealing with Industry 4.0.

As future work, we intend to carry out a benchmark in order to identify other block programming tools that can be used in High School. The motivation for the benchmark is to characterize these tools. In this SMS a diversity of block programming environments and their functionalities was perceived, even with the relevance of the studies related to block programming tools, some of these are comparative evaluation to verify their feasibility to a given context based on pre-defined criteria for the teaching and learning processes. A comparative study of block programming tools was not identified in the literature, taking into account characteristics such as which operating systems and platforms the block programming tools work; which materials exist and are available for both the teacher and the student, emerging technologies that can be worked together with these tools, among other characteristics. Therefore, a benchmark will be performed in order to identify as many block programming tools as possible and compare them from the point of view of some of their characteristics. Subsequently, we intend to analyze which classes and emerging technologies these tools are actually used, making a relationship with the data identified in this SMS. Finally, after carrying out the benchmark, a tool will be developed. This tool could help high school teachers of subjects of the common curriculum (Mathematics, Portuguese, History, among others) in the decision making of which one is best suited to their context of use, based on the characteristics investigated.

References

Sharefah A-Ghamdi, Noha Al-Rajhi, Nouf Al-Onaizy, and Hend Al-Khalifa. 2016. *IEEE Global Engineering Education Conference, EDUCON* 10-13-April-2016 (2016), 383–388. https://doi.org/10.1109/EDUCON.2016.7474582 cited By 4.

Rivika Alda, Helen Boholano, and Filomena Dayagbil. 2020. Teacher Education Institutions in the Philippines towards Education 4.0. *International Journal of Learning, Teaching and Educational Research* 19, 8 (2020), 137–154.

Alexandre Hild Aono, Hugo Vianna Silva Rody, Daniela Leal Musa, Vanessa Andrade Pereira, and Jurandy Almeida. 2017. A utilização do scratch como ferramenta no ensino de pensamento computacional para crianças. In *Anais do XXV Workshop sobre Educação em Computação*. SBC.

Victor Basili and H Dieter Rombach. 1988. Towards a comprehensive framework for reuse: A reuse-enabling software evolution environment. (1988).

Carmen Vera Scorsatto Brezolin and Milene Selbach Silveira. 2021. Panorama Brasileiro de Uso de Ferramentas para Desenvolvimento do Pensamento Computacional e Ensino de Programação. In *Anais do XXIX Workshop sobre Educação em Computação*. SBC, 398–407.

Brian Broll, Ákos Lédeczi, Hamid Zare, Dung Nguyen Do, János Sallai, Péter Völgyesi, Miklós Maróti, Lesa Brown, and Chris Vanags. 2018. A visual programming environment for introducing distributed computing to secondary education. *J. Parallel and Distrib. Comput.* 118 (2018), 189–200. https://doi.org/10.1016/j.jpdc.2018.02.021 cited By 5.

Philip Sheridan Buffum, Megan Hardy Frankosky, Kristy Elizabeth Boyer, Eric N. Wiebe, Bradford W. Mott, and James C. Lester. 2016. Empowering All Students: Closing the CS Confidence Gap with an In-School Initiative for Middle School Students. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. Association for Computing Machinery, Memphis, Tennessee, USA, 382–387. https://doi.org/10.1145/2839509.2844595

Rogério Cardoso and Sérgio Antonello. 2015. Interdisciplinaridade, programação visual e robótica educacional: relato de experiência sobre o ensino inicial de programação. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, Vol. 4. 1255.

Monica Ciolacu, Ali Fallah Tehrani, Rick Beer, and Heribert Popp. 2017. Education 4.0—Fostering student's performance with machine learning methods. In *2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME)*. IEEE, 438–443.

Cárdenas Jesennia Cobo, Amilkar Puris, Pavel Novoa-Hernández, Águeda Parra-Jiménez, Jesús Moreno-León, and David Benavides. 2021. Using Scratch to Improve Learning Programming in College Students: A Positive Experience from a Non-WEIRD Country. *Electronics* 10, 10 (2021). https://doi.org/10.3390/electronics10101180

Thaise Costa, Fábio Cristiano, Patrícia da Rocha, and Danielle Danielle. 2017. O Ensino de Linguagem de Programação na Educação Básica Através da Robótica Educacional: Práticas e a Interdisciplinaridade.. In *Anais do Workshop de Informática na Escola*, Vol. 23. 687.

Edson Vieira Da Silva. 2018. Educação a distância. *CIET: EnPED* (2018).

Leo Victorino da Silva. 2020. Tecnologias digitais de informação e comunicação na educação: três perspectivas possíveis. *Revista de Estudos Universitários-REU* 46, 1 (2020), 143–159.

Cláudio de Oliveira. 2015. TIC'S na educação: a utilização das tecnologias da informação e comunicação na aprendizagem do aluno. *Pedagogia em Ação* 7, 1 (2015).

Enoque Fôro de Oliveira. 2019. Ensino de Geografia e Educação 4.0: Caminhos e Desafios na Era da Inovação. *Revista Amazônica sobre Ensino de Geografia* 1, 01 (2019).

Luiz Kevin de Souza Rios, Almir de OLiveira Costa Junior, João Paulo Felizardo Lima, Elloa Barreto Guedes, et al. 2019. Uma Análise Comparativa entre Ambientes de Programação em Blocos para a Interação com o Arduino. *Anais do Simpósio Ibero-Americano de Tecnologias Educacionais* (2019).

Yihuan Dong, Veronica Catete, Robin Jocius, Nicholas Lytle, Tiffany Barnes, Jennifer Albert, Deepti Joshi, Richard Robinson, and Ashley Andrews. 2019. PRADA: A Practical Model for Integrating Computational Thinking in K-12 Education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. Association for Computing Machinery, Minneapolis, MN, USA, 906–912. https://doi.org/10.1145/3287324.3287431

Deivid Eive dos Santos Silva, Marialina Corrêa Sobrinho, and Natasha Valentim. 2019. Criação de Jogos Educacionais para apoiar o Ensino da Matemática: um Estudo de Caso no Contexto da Educação 4.0. In *Anais do Workshop de Informática na Escola*, Vol. 25. 1179.

Ron Eglash, Michael Lachney, William Babbitt, Audrey Bennett, Martin Reinhardt, and James Davis. 2019. Decolonizing education with Anishinaabe arcs: generative STEM as a path to indigenous futurity. *Educational Technology Research and Development* (Dec. 2019). https://doi.org/10.1007/s11423-019-09728-6

Andoni Eguiluz, Mariluz Guenaga, Pablo Garaizar, and Cristian Olivares-Rodriguez. 2020. Exploring the Progression of Early Programmers in a Set of Computational Thinking Challenges via Clickstream Analysis. *IEEE Transactions on Emerging Topics in Computing* 8, 1 (2020), 256–261. https://doi.org/10.1109/TETC.2017.2768550 cited By 3.

Daniel Amo Filvà, Marc Alier Forment, Francisco José García-Peñalvo, David Fonseca Escudero, and María José Casañ. 2019. Clickstream for learning analytics to assess students' behavior with Scratch. *Future Generation Computer Systems* 93 (2019), 673–686. https://doi.org/10.1016/j.future.2018.10.057 cited By 21.

Casallas Rubby Flórez, Francisco Buitrago and, Marcela Hernández, Alejandro Reyes, Silvia Restrepo, and Giovanna Danies. 2017. Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research* 87, 4 (2017), 834–860.

Ilenia Fronza, Luis Corral, and Claus Pahl. 2019. Combining Block-Based Programming and Hardware Prototyping to Foster Computational Thinking. In *Proceedings of the 20th Annual SIG Conference on Information Technology Education (SIGITE '19)*. Association for Computing Machinery, Tacoma, WA, USA, 55–60. https://doi.org/10.1145/3349266.3351410

Ilenia Fronza, Nabil El Ioini, and Luis Corral. 2015. Students want to create apps: leveraging computational thinking to teach mobile software development. In *Proceedings of the 16th annual conference on information technology education*. 21–26.

Regina Candida Führ. 2018. O Dilúvio Digital e seus Impactos na Educação 4.0 e na Indústria 4.0. *Investigação em Governança Universitária: Memórias* 188 (2018), 37–54.

Léo Garcia, Daiany Lara, Franciano Antunes, Cristiano Miranda, and Claudia Peres. 2017. Análise da Evasão no Ensino Superior e suas Motivações: Um Estudo de Caso em um Curso de Sistemas de Informação. In *Anais do XIII Simpósio Brasileiro de Sistemas de Informação* (Lavras). SBC, Porto Alegre, RS, Brasil, 527–534. https://doi.org/10.5753/sbsi.2017.6084

Terrell Glenn, Ananya Ipsita, Caleb Carithers, Kylie Peppler, and Karthik Ramani. 2020. StoryMakAR: Bringing Stories to Life With An Augmented Reality & Physical Prototyping Toolkit for Youth. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, Honolulu, HI, USA, 1–14. https://doi.org/10.1145/3313831.3376790

José Gonçalves, José Lima, Thadeu Brito, Laiany Brancalião, Caio Camargo, Vitor Oliveira, and Miguel Á. Conde. 2019. Educational Robotics Summer Camp at IPB: A Challenge based learning case study. In *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'19)*. Association for Computing Machinery, León, Spain, 36–43. https://doi.org/10.1145/3362789.3362910

Paulina Haduong and Karen Brennan. 2019. Helping K–12 Teachers Get Unstuck with Scratch: The Design of an Online Professional Learning Experience. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. Association for Computing Machinery, Minneapolis, MN, USA, 1095–1101. https://doi.org/10.1145/3287324.3287479

Sugiarto Hartono, Raymond Kosala, Suhono Harso Supangkat, and Benny Ranti. 2018. Smart Hybrid Learning Framework Based on Three-Layer Architecture to Bolster Up Education 4.0. In *2018 International Conference on ICT for Smart Society (ICISS)*. IEEE, 1–5.

Nicole M Hutchins, Gautam Biswas, Miklós Maróti, Ákos Lédeczi, Shuchi Grover, Rachel Wolf, Kristen Pilner Blair, Doris Chin, Luke Conlin, Satabdi Basu, et al. 2020. C2STEM: A system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology* 29, 1 (2020), 83–100. https://doi.org/10.1007/s10956-019-09804-9 cited By 6.

Robin Jocius, Deepti Joshi, Yihuan Dong, Richard Robinson, Veronica Cateté, Tiffany Barnes, Jennifer Albert, Ashley Andrews, and Nicholas Lytle. 2020. Code, Connect, Create: The 3C Professional Development Model to Support Computational Thinking Infusion. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. Association for Computing Machinery, Portland, OR, USA, 971–977. https://doi.org/10.1145/3328778.3366797

K12CS. 2016. K–12 Computer Science Framework. Available in: <http://www.k12cs.org.>. Accessed in 01 apr. 2022.

Megasari K Kahn, Rani Megasari, Erna Piantari, and Enjun Junaeti. 2018. AI programming by children using snap! Block programming in a developing country. *CEUR Workshop Proceedings* 2193 (2018). https://www.scopus.com/inward/record.uri?eid=2-s2.0-85053707384&partnerID=40&md5=f572adf493a473f9dbe766b2fa5945d9 cited By 2.

Barbara Kitchenham and Stuart Charters. 2007. Guidelines for performing systematic literature reviews in software engineering. (2007).

Surabhi Koul and Burna Nayar. 2021. The holistic learning educational ecosystem: A classroom 4.0 perspective. *Higher Education Quarterly* 75, 1 (2021), 98–112.

Fotis Lazarinis, Christoforos V. Karachristos, Elias C. Stavropoulos, and Vassilios S. Verykios. 2019. A blended learning course for playfully teaching programming concepts to school teachers. *Education and Information Technologies* 24, 2 (March 2019), 1237–1249. https://doi.org/10.1007/s10639-018-9823-2

Irene Lee and Joyce Malyn-Smith. 2020. Computational Thinking Integration Patterns Along the Framework Defining Computational Thinking from a Disciplinary Perspective. *Journal of Science Education and Technology* 29, 1 (Feb. 2020), 9–18. https://doi.org/10.1007/s10956-019-09802-x

Gabriel Loureiro de Lima, Niltom Vieira Junior, José Geraldo Ribeiro Júnior, Elzo Alves Aranha, Jorge Candido, Américo Tristão Bernardes, José Silvério Edmundo Germano, and Adriana Maria Tonini. 2017. Desafios da educação em engenharia: formação acadêmica e atuação profissional, práticas pedagógicas e laboratórios remotos. (2017).

Yuhan Lin and David Weintrop. 2021. The landscape of Block-based programming: Characteristics of block-based environments and how they support the transition to text-based programming. *Journal of Computer Languages* 67 (2021), 101075. https://doi.org/10.1016/j.cola.2021.101075

María Soledad Ramírez Montoya, María Isabel Loaiza-Aguirre, Alexandra Zúñiga-Ojeda, and May Portuguez-Castro. 2021. Characterization of the Teaching Profile within the Framework of Education 4.0. *Future Internet* 13, 4 (2021), 91.

Mariuxi Vinueza Morales, Jorge Córdova Morán, and Jorge Rodas Silva. 2019. Using Alice software as a tool for programming learning: a literature review. In *2019, 17 th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Industry, Innovation, And Infrastructure for Sustainable Cities and Communities"*.

Anders Mørch, Kristina Torine Litherland, and Renate Andersen. 2019. End-User Development Goes to School: Collaborative Learning with Makerspaces in Subject Areas. In *End-User Development (Lecture Notes in Computer Science)*, Alessio Malizia, Stefano Valtolina, Anders Morch, Alan Serrano, and Andrew Stratton (Eds.). Springer International Publishing, Cham, 200–208. https://doi.org/10.1007/978-3-030-24781-2_16

Nilbo Ribeiro Nogueira Nogueira. 2002. Pedagogia de Projetos: uma jornada interdisciplinar rumo ao desenvolvimento das Múltiplas Inteligências. 2002. *São Paulo: Érica* (2002).

Samantha Orlando, Elena Gaudioso, and Felix De La Paz. 2020. Supporting Teachers to Monitor Student's Learning Progress in an Educational Environment with Robotics Activities. *IEEE Access* 8 (2020), 48620–48631. https://doi.org/10.1109/ACCESS.2020.2978979 cited By 1.

Neuza Oro, Ariane Pazinato, Adriano Teixeira, and Ádler Gross. 2015. Olimpíada de Programação de Computadores para Estudantes do Ensino Fundamental: A interdisciplinaridade por meio do Software Scratch. In *Anais do Workshop de Informática na Escola*, Vol. 21. 102.

Stamatios Papadakis and Vasileios Orfanakis. 2018. Comparing novice programing environments for use in secondary education: App Inventor for Android vs. Alice. *International Journal of Technology Enhanced Learning* 10, 1-2 (2018), 44–72. https://doi.org/10.1504/IJTEL.2018.088333 cited By 12.

Sofia Papavlasopoulou, Michail N Giannakos, and Letizia Jaccheri. 2019. Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior* 99 (2019), 415–427. https://doi.org/10.1016/j.chb.2019.01.008 cited By 16.

Ana Perin, Deivid Eive Silva, and Natasha Valentim. 2021. Um benchmark de ferramentas de programação em blocos que podem ser utilizadas nas salas de aula do Ensino Médio. In *Anais do XXXII Simpósio Brasileiro de Informática na Educação* (Online). SBC, Porto Alegre, RS, Brasil, 1162–1173. https://doi.org/10.5753/sbie.2021.217765.

Vichian Puncreobutr. 2016. Education 4.0: New challenge of learning. *St. Theresa Journal of Humanities and Social Sciences* 2, 2 (2016).

Amarildo Enes dos Santos, Carlos Antonio de Oliveira, and Elma Nunes de Carvalho. 2019. Educação 5.0: uma nova abordagem de ensino-aprendizagem no contexto educacional. (2019).

Juliana Saraiva, Amanda Rodrigues, and Vanessa Dantas. 2020. Evasion in the Information System Course from a three-dimensional perspective of factors. *iSys - Brazilian Journal of Information Systems* 13, 3 (Jun. 2020), 05–24. https://doi.org/10.5753/isys.2020.792

Pasqueline Dantas Scaico, Anderson Alves de Lima, Silvia Azevedo, Jefferson Barbosa Belo da Silva, Ewerton Henning Raposo, Yugo Alencar, João Paulo Mendes, Alexandre Scaico, et al. 2013. Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch. *Revista Brasileira de Informática na Educação* 21, 02 (2013), 92.

David Scaradozzi, Laura Screpanti, Lorenzo Cesaretti, M Storti, and Elena Mazzieri. 2019. Implementation and Assessment Methodologies of Teachers' Training Courses for STEM Activities. *Technology, Knowledge and Learning* 24, 2 (2019), 247–268. https://doi.org/10.1007/s10758-018-9356-1 cited By 8.

Eleni Seralidou and Christos Douligeris. 2019. Learning with the AppInventor programming software through the use of structured educational scenarios in secondary education in Greece. *Education and Information Technologies* 24, 4 (July 2019), 2243–2281. https://doi.org/10.1007/s10639-019-09866-7

Deivid Eive Silva, Marialina Correa Sobrinho, and Natasha Malveira Valentim. 2020. Educação 4.0: um Estudo de Caso com Atividades de Computação Desplugada na Amazônia Brasileira. *Anais do Computer on the Beach* 11, 1 (2020), 141–147.

Isabelle ML Souza, Wilkerson L Andrade, Lívia MR Sampaio, and Ana Liz Souto O Araujo. 2018. A Systematic Review on the use of LEGO® Robotics in Education. In *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–9.

Marcelo Varela de Souza. 2016. *Domótica de baixo custo usando princípios de IoT*. Master's thesis. Brasil.

Luiz Kevin de Souza Rios, Almir de OLiveira Costa Junior, João Paulo Felizardo Lima, Elloa Barreto Guedes, et al. 2019. Uma Análise Comparativa entre Ambientes de Programação em Blocos para a Interação com o Arduino. *Anais do Simpósio Ibero-Americano de Tecnologias Educacionais* (2019).

Claudia Szabo, Judy Sheard, Andrew Luxton-Reilly, Brett A Becker, and Linda Ott. 2019. Fifteen years of introductory programming in schools: a global overview of K-12 initiatives. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*. 1–9.

Kajiana Nuernberg Sartor Vidotto, Luana Monique Delgado Lopes, Eliane Pozzebon, and Luciana Bolan Frigo. 2018. Programando jogos com Realidade Aumentada. *Anais do Simpósio Ibero-Americano de Tecnologias Educacionais* (2018), 129–138.

Ravi Vinayakumar, KP Soman, and Pradeep Menon. 2018a. Digital storytelling using scratch: engaging children towards digital storytelling. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 1–6.

Ravi Vinayakumar, KP Soman, and Pradeep Menon. 2018b. Fractal Geometry: Enhancing Computational Thinking with MIT Scratch. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 1–6.

David Weintrop, Merijke Coenraad, Jen Palmer, and Diana Franklin. 2019. The Teacher Accessibility, Equity, and Content (TEC) Rubric for Evaluating Computing Curricula. *ACM Trans. Comput. Educ.* 20, 1, Article 5 (dec 2019), 30 pages. https://doi.org/10.1145/3371155

David Weintrop and Uri Wilensky. 2015. To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th International Conference on Interaction Design* *and Children (IDC '15)*. Association for Computing Machinery, Boston, Massachusetts, 199–208. https://doi.org/10.1145/2771839.2771860

Jeannette Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.

Noor Faridatul Ainun Zainal, Rosseni Din, Nazatul Aini Abd Majid, Mohammad Faidzul Nasrudin, and Abdul Hadi Abd Rahman. 2018. Primary and secondary school students perspective on Kolb-based STEM module and robotic prototype. *International Journal on Advanced Science, Engineering and Information Technology* 8, 4-2 (2018), 1394–1401. https://doi.org/10.18517/ijaseit.8.4-2.6794 cited By 5.

Ana Paula Juliana Perin. Degree in Informatics from the Federal Technological University of Paraná Campus Francisco Beltrão (2017). Specialist in Informatics in Education. Specialist in Systems Engineering by Escola Superior Aberta do Brasil. Master in Informatics from the Federal University of Paraná (UFPR). He has experience in block programming and Educational Robotics with high school teachers and students.

Deivid Eive dos S. Silva is currently a Ph.D student in Informatics at the Federal University of Paraná (UFPR), Brazil. He holds a Master's in Informatics from UFPR. He has experience in the area of Informatics in Education, Software Engineering, and HCI. His research activities focus on the Design and Evaluation of the Learner Experience with Educational Technologies.

Natasha Malveira Costa Valentim is active as professor at the Department of Informatics of the Federal University of Paraná (UFPR). She completed a Ph.D. in Computer Science at the Federal University of Amazonas (UFAM) in 2017. She is currently a member of the Interaction Design for Inclusion and Social Development research group from the UFPR. She researches in the areas of Informatics in Education, Computer Science Education, Software Engineering, and Human-Computer Interaction. She is interested in Education 4.0, Block-programming Teaching and Learning, Usability, User Experience, Accessibility, and Software Quality.