Journal of Positive Psychology & Wellbeing
2022, Vol. 6, No. 2, 90 – 107
http://journalppw.com
ISSN 2587-0130

Design Of Algorithms For Developing Computational Thinking In Elementary And Middle School Students.

Cesar Augusto Hernández-Suárez[1], Audin Aloiso Gamboa-Suárez[2], William Rodrigo Castro-Avendaño[3]

[1]Universidad Francisco de Paula Santander
https://orcid.org/0000-0002-7974-5560
cesaraugusto@ufps.edu.co
[2]Universidad Francisco de Paula Santander
https://orcid.org/0000-0001-9755-6408
audingamboa@ufps.edu.co
[3]Universidad Francisco de Paula Santander
https://orcid.org/0000-0002-7510-8222
williamavendano@ufps.edu.co

Abstract

The research is oriented towards the development of Computational Thinking through the design of algorithms in technology with Scratch, with students of basic secondary education. The objective of the study was to develop computational thinking through the design and implementation of algorithms to solve problems. The study was framed in the qualitative paradigm through the action research method, oriented to transform the current pedagogical practices regarding the solution of problems that students have to solve with technology. The study context was composed of sixth grade students with intentional sampling. The technique and instrument were observation and a Likert scale questionnaire. The results show that students perform some daily tasks whose sequences allow them to be initiated in the basic structures of sequential programming, which enhances processes towards problem solving. It is concluded that problem-solving skills can be fostered with strategies that use technological tools such as Scratch, since students are motivated by the use of ICT.

Keywords: Computational thinking, technology, algorithms, Scratch, Moodle platform.

Resumen

La investigación se orienta hacia el desarrollo del Pensamiento Computacional a través del diseño de algoritmos en tecnología con Scratch, con estudiantes de educación básica secundaria. El objetivo del estudio fue desarrollar el pensamiento computacional a través del diseño e implementación de algoritmos para resolver problemas. El estudio se enmarcó en el paradigma cualitativo mediante el método de investigación-acción, él se orientó a transformar las actuales prácticas pedagógicas respecto a la solución de problemas que tienen los estudiantes para resolver problemas con tecnología. El contexto de estudio estuvo compuesto por estudiantes del grado sexto con muestreo intencional. La técnica y el instrumento fueron la observación y un cuestionario tipo Likert. Los resultados evidencian que los estudiantes realizan algunas tareas diarias cuyas secuencias permiten iniciarse en las estructuras básicas de la programación secuencial, lo que potencia procesos hacia la

resolución de problemas. Se concluye que la habilidad para la resolución de problemas se puede fomentar con estrategias que utilicen herramientas tecnológicas como Scratch, debido a que los estudiantes se motivan con el uso de las TIC.

Palabras clave: Pensamiento computacional, tecnología, algoritmos, Scratch, plataforma moodle.

INTRODUCTION

Education in Colombia as a constitutional right, with purposes and mechanisms for a permanent, personal, cultural and social process, aims at forming students integrally (Aguilar-Barreto et al., 2017), with dignity, respectful of their rights and duties, where the curriculum, the pedagogical, didactic praxis, as well as the new learning scenarios, are components that have been facing education at all levels as highlighted in Chaparro et al. (2018), where ICT is included as a transversal axis and with it computational thinking (Parada & Avendaño, 2013; Avendaño & Parada, 2011).

In this context, educational institutions also face the same challenges in the area of technology and informatics, defined as a fundamental area by Law 115 (1994, art. 23). This area uses computer science concepts to formulate and solve problems in order to provide a quality education that forms better citizens by generating opportunities for progress and prosperity for all. Thus, several studies provide a look at the design of algorithms in technology in the school environment and its impact on mental development, which has become a new paradigm since Dr. Wing (Carnegie Mellon University, USA) used the term Computational Thinking (hereinafter PC), to describe how a computer scientist thinks and the benefits that this way of thinking could have on everyone. It is an approach to solving problems, designing systems and understanding human behavior that is based on fundamental concepts of computation (Wing, 2006).

Under these guidelines, the International Society for Technology in Education (ISTE), together with the Computer Science Teachers Association (CSTA), addressed the K-12 Computer Science framework in the USA (CSTA & ISTE, 2011). In Latin America, the proposal of Gurises Unidos (Uruguay) and Fundación Telefónica-Movistar (2017), on the topic: Computational Thinking. A contribution for today's education, whose purpose is to promote actions aimed at transforming the education of children and adolescents.

In the Colombian context, Giraldo (2014) pointed out that "...the technology and computer science area plans do not contemplate in their contents, related concepts... for the solution of problems, they only look for the student to handle the tool. In addition, there is no training in programming..." (p. 14) for the development of the PC, a situation similar to that found in the educational institution under study. Therefore, it is deduced that students advance in their studies lacking skills to find solutions to problems and meet needs, which results in poor cognitive, creative, critical-valuative and transformative processes (Ministry of National Education [Mineducación], 2008); hence, it is necessary that educational institutions not only teach the use and management of digital tools, but also teach students to "be competent in technology" (Mineducación 2006; 2008).

In Contreras-Colmenares and Jimenez-Villamarín (2020), it is highlighted that another cause that affects the acquisition of skills for problem solving with technology is that teachers use strategies for the development of their classes on meaningless content and that are not consistent to develop critical and computational thinking, out of context of the guidelines of the Mineducación (2006; 2008), as mechanisms for innovation and changes to improve existing processes that have a significant impact on learning processes and that are transversal, with tools that facilitate and optimize the capacity for abstraction of the real world in students (Penagos et al., 2017).

Hence, the need to enrich pedagogical practices mediated by ICT to promote teaching, and likewise, to train students anytime and anywhere, allowing the obliquity of learning in content on the PC, such as: Abstraction, algorithmic thinking, analysis, creativity, among others, which according to the CSTA & ISTE (2011) includes how, when and where technology can help us solve problems.

Consequently, the problem reality indicated, as well as its justification, allows to have the objective of describing the development of thinking in the solution of problems with technology achieved by students through the design and implementation of algorithms with the Scratch tool.

DEVELOPMENT OF COMPUTATIONAL THINKING THROUGH THE DESIGN OF ALGORITHMS WITH SCRATCH TECHNOLOGY

CP is a fundamental skill used by everyone in the world (Wing, 2012). It is defined by Wing (2006, cited in Zapotecatl, 2018) as "the thought processes involved in formulating problems and representing their solutions, so that those solutions can be effectively executed by an information processing agent, as human, computer, or combinations of humans and computers" (p. 3).

On the other hand, the ISTE & CSTA (2011) point out that PC is a problem-solving process that involves the mastery of skills such as: Formulating problems to solve them; organizing data logically and analyzing them; representing data through abstractions; automating solutions through algorithmic thinking; identifying, analyzing and implementing possible solutions in order to find the most efficient and effective combination of steps and resources; finally, generalizing and transferring that problem-solving process to a great diversity of these, which coincides with what is mentioned in Prada-Núñez et al. (2020), where the importance of mathematical processes for the development of competencies necessary for life is highlighted. Likewise, according to Gurises Unidos and Fundación Telefónica - Movistar (2017), PC involves applying core components such as:

Decomposition. Which is a process of deconstruction, which develops the ability to break down complex problems into smaller, more manageable parts, making even the most complicated task or problem easier to understand and solve.

Pattern recognition. Skill that involves mapping similarities and differences or patterns between small (decomposed) problems, to help solve complex problems. The goal is to find patterns that help simplify tasks.

Abstraction. It involves filtering out (or ignoring) unimportant details, which make a problem easier to understand and solve. This allows developing models, equations, an

Algorithm design. It consists of determining the appropriate steps to take and organizing them into a series of instructions (a plan) to solve a problem or complete a task correctly.

From there, it is accurate that the PC is a process, which can be part of the classroom, including primary grades, where students can cognitively develop "the ability to execute, evaluate, understand and create computational procedures" (Guerrero & García, 2016, p. 165), and teachers ensure that these learn to think in a way that allows them to access and understand the digital world, together and with the other, in order to prepare them for future success.

Zapotecatl (2018) argues that "for computational thinking, algorithms are considered a key practice to be able to formulate problems and apply solutions in a methodical and orderly manner" (p. 99). He also argues that algorithms can be expressed in a variety of ways, including natural language, pseudocode, flowcharts, among others, and avoid the ambiguities that often occur in natural language because they are more structured representations.

In summary, it follows that exploring the nature of PC and its cognitive and educational implications as elements of education in the curriculum (Zapata-Ros, 2015) is related to other thoughts such as logical, analytical, algorithmic, abstract, divergent and critical... because each of these encompasses a branch in which computational thinking is involved (Barrera & Montaño, 2016). Thus, every human being endowed with cognitive capacity, acts as a subject that establishes both general and particular explanations about the

image and/or simulations to represent only the important variables.

objects, facts or phenomena that constitute reality.

According to Piaget (1952), "cognitive development" involves processes in mental operations, which occur in four distinct stages in individuals: sensorimotor, preoperational, concrete operations and formal operations (the latter from 11 to 12 years of age onwards). Therefore, under a constructivist view, "it arises from a process of organization of the interactions between subjects..." (García, 2000, p.61). Therefore, learning as an active and constructive process, links new information with existing information and is contextualized, instead of being acquired from its own interpretation of reality (Novak, 1988). Hence, the development of the PC can be approached from a cognitive and social constructivism, with the use of technologies such as Scratch for the design of the algorithm.

On the other hand, for Piaget (as cited in Arancibia et al., 1999, p. 76), intelligence is the capacity to keep adapting the subject's schemas or representations to the world that surrounds the subject, constructed by the subject and in which he/she develops. Adaptation is the process that explains development and learning through assimilation and accommodation. Assimilation incorporates information into a pre-existing schema, suitable for integration (understanding), in accordance with "the acquisition of new information depends to a high degree on the relevant ideas that already exist in the cognitive structure" (Ausubel, cited in Ontoria et al., 2006, p. 22), while accommodation produces essential changes in the schema.

Thus, the Piagetian theory of intellectual development assumes the individual construction of certain psychological structures, which allows adaptation to reality, through the evolutionary development whose direction is endogenous. Although this point of view is contrary to the position of Vygotsky (1978, p. 94), who proposes an exogenous development process, where what the student can do today with the help of another student or teacher, tomorrow he/she will be able to do by him/herself (González et al., 2011). That is, students experience a process of "social construction" of cognitive development, through a Zone of Proximal Development (ZDP), which according to Vygotsky (1980, cited in Vallejo et al., 1999), as the distance between the child's actual level of development as it can be determined from independent problem solving and the highest level of potential development and as it is determined by problem solving under adult guidance or in collaboration with more capable peers (Medina Romero et al, 2021).

In this way, problem solving, when designing algorithms in technology with Scratch, can serve the development of students' thinking, because it establishes a process through which they can recognize signals that identify the presence of a difficulty, anomaly or hindrance of the normal development of a task, collect the necessary information to solve the problems detected and choose and implement the best solution alternatives, either individually or in groups, in order to achieve in them the full potential as constructive and reflective citizens (OECD, 2014, p. 12), through the language based on graphic blocks and with an easy interface for learning, such as the Scratch tool, since, the union of blocks is intended to allow students to program in a simple way, without the need for complex programming, and with an easy interface for learning, such as the Scratch tool, since the union of blocks is intended to allow students to program in a simple way, without the need for complex programming through the language based on graphic blocks and with an easy interface for learning, such as the Scratch tool, since the purpose of the union of blocks is that students can program in a simple way, without the need for complex algorithms. In this regard, Sáez-López and Cózar-Gutiérrez (2015) mention that "the block-based visual programming environment called Scratch, ...poses as a considerable advantage in languages that aim to give novices their first introduction to computing" (p.12). In addition, Sáez-López and Cózar-Gutiérrez (2015) argue that it fosters active learning and active participation when writing by dragging and dropping graphic blocks to compose simple programs that, in turn, allow them to create games, interactive stories or simulations, while learning content and the work fosters an adequate learning process.

THE METHOD

The research is framed within the qualitative approach. The inquiry process was based on the type of action research (Kemmis & McTaggart, 1988; Gamboa, 2020) and on the comprehensive principles (Taylor & Bogdan, 1996). Consequently, the project corresponds to a "process of reflection and continuous transformation of practice, to make it a professional activity guided by appropriate pedagogical knowledge" (Restrepo, 2004, p. 50). In that direction, it focused on a practical action research, because it studies local practices. It involves individual and team inquiry adjusting to the development and learning of the participants and introducing improvements or generating change (Creswell, 2005).

Similarly, a quantitative analysis of the information provided by Scopus under a bibliometric approach on the scientific production related to the Development of Computational Thinking is carried out through the design of algorithms in Scratch technology identified from Latin American institutions in order to know the approach of scientific papers through the use of keywords; by means of this the study intends to know the impact and scope of the research published under the above-mentioned topic thanks to the map of co-occurrence of keywords.

For the proposed biometric analysis, a search is performed in the Scopus database. So, filters are established to identify the publications most related to the topic of study of this document. The selection of articles or research papers is done by establishing the following search criteria:

- Research papers (articles, conference papers, books, book chapters, among others) whose variable of study is Developing Computational Thinking through the design of algorithms in technology with Scratch.
- Research papers published in Latin American countries.

BIBLIOMETRIC ANALYSIS

Keyword Co-occurrence Map

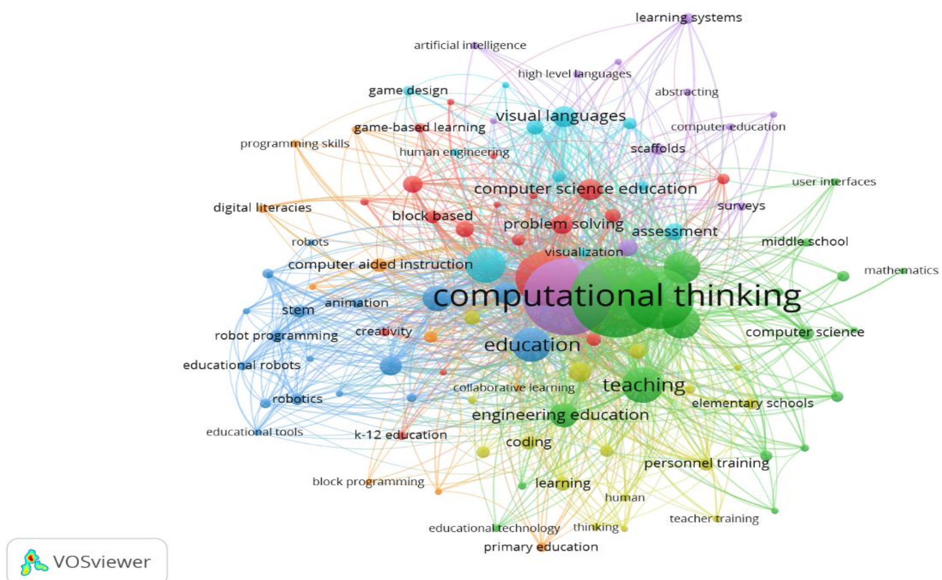

Figure 1. Word co-occurrence
Source: Own elaboration (2021); based on data provided by Scopus.

Computational Thinking was the most frequently used keyword within the total of 361 scientific documents identified through the application of the search filters applied in Scopus, which reached a total of 361 publications related to the study of the topic raised in this research. The main topics addressed in most publications were related to Education, Teaching, Problem Solving, Primary Education, thus confirming the relevance between the set of research found and the development of this article. It is important to know the impact of the publications as in articles such as the one entitled "Effect of different mind mapping approaches on elementary students' computational thinking skills during learning visual programming" (Zhao, Liu, Wang & Su,

2022) whose purpose was to apply mind mapping to programming language instruction in an elementary school and explored the effects of mind mapping on students' Computational Thinking skills when learning to program with Scratch as a programming tool. The findings of this research were able to provide a broader vision among designers and instructors for the development of study methodologies based on problem solving through the didactics offered by this type of virtual tools.

CATEGORIES OF ANALYSIS

The following are the initial categories according to the problem and the research objectives.

Table 1. Categories.

Categories	Definition
Motivation	Seeks to foster students' innovative and reflective capacity.
Computational Thinking	Problem-solving approach that empowers the integration of digital technologies with human ideas (CSTA & ISTE, 2011).
Troubleshooting	Process through which detected problems are recognized and solved and the best solution alternatives are chosen.
Algorithm design in technology	Key practice to be able to formulate problems and apply solutions in a methodical and orderly manner (Zapocatl, 2018).

RESEARCH CONTEXT

The research was carried out in the basic secondary education subsystem, in an educational institution in the city of Cúcuta, Colombia. The students are in the sixth grade and are composed of 113 students and one teacher. Their ages range from 10 to 13 years old, of which 53 are male and 60 are female. The 39 students of course A were selected as study subjects. Because of their voluntary participation.

PROCEDURE

The procedure proposed by Kemmis (1989, cited in Latorre, 2003) was used, where questions of improvement and social change are considered. It involves planning an action, putting it into practice, observing it and, finally, reflecting on what happened. Thus, a series of phases described below was implemented:

Planning Phase

Strategies and didactic guides on the design of algorithms were designed in a face-to-face way in classes, as well as online with the Scratch tool. For the contents and standards of problem solving with technology for sixth grade, the ISTE Standards (2016) for students were considered. Within the planning process, the following work method was carried out:

a) A sequence of sessions was designed as a strategy to be developed with the participants: 1. Algorithms and pseudocode; 2. Getting to know Scratch; 3. Use of simple variables.
b) In each didactic sequence, several cooperative and collaborative activities were developed. Likewise, a learning environment was installed in Moodle. Therefore, face-to-face and online activities are arranged to design the algorithm with Scratch to motivate the participants.

Action and Observation Phase

This phase corresponds to the implementation of the plan developed, as well as observation. Each student records his or her work, achievements and goals in his or her practices. At this point, Creswell (2012) points out that this type of research can collect "...quantitative, qualitative or both types of data" (p. 577). From there, data collection instruments were applied, as well as, observation, field diary and questionnaire. Also, a Computational Thinking Test (CTT) proposed by Román (2015) to make a diagnosis in students of the lack of problem-solving skills with technology and PC.

Regarding the virtual learning environment, an implemented pedagogical proposal was presented, both with didactic and technological aspects. The pedagogical proposal was based on the design of didactic sequences with guides in the use of Scratch to design algorithms and improve problem-solving skills in technology. Meanwhile, the technological component was designed taking into account the ADDIE model for the instructional design of the virtual learning environment (Albarracín-Villamizar et al., 2020).

Reflection Phase

An attempt was made to describe in the participants of course A, the strategies supported by technology for the development of PC. Therefore, the entry test was applied in three phases. For moments one and two the test was done collaboratively and the in third moment it was done individually. The purpose was to observe and analyze the results found with respect to the opinions of the participants in the different moments in which the diagnosis was applied with respect to the PC, to do problem solving in technology and know how to do problem solving in technology, where the processes of sequencing, debugging and completion can be ascertained through the logical use of syntax of computer programming languages: Basic sequences, loops, iterations, conditionals, functions and variables (Román-González et al., 2015).

Regarding problem solving, it was observed that some students present partial problems in comprehension where loops are involved. There are perceived inconveniences with previous notions when decomposing a task into steps, which makes it difficult to recognize some higher order patterns. It was proposed to the students to communicate orally the difficulties encountered in solving the proposed problems. In this way, the mental processes and procedures students used to reach the solution is recorded from the students' own words, and at the same time, to help those who have greater difficulties.

In relation to the CTT, PC was only measured at its lowest levels of cognitive complexity ('recognize' and 'understand'). Also, in its higher levels of complexity ('apply' and 'assimilate').

ANALYSIS OF RESULTS

ANALYSIS OF QUANTITATIVE RESULTS

After applying the survey that revolved around the PC, the findings are presented.

Table 2. Decomposition.

Items	Always	Almost always	Sometimes	Almost never	Never

	Fs	%	Fs	%	Fs	%	Fs	%	Fs	%
1	8	20.5%	16	41 %	6	15.4%	4	10.3%	5	12.8%
2	8	20.5%	19	48.7%	9	23.1%	2	5.1%	1	2.6%
3	17	43.6%	8	20.5%	11	28.2%	2	5.1%	1	2.6%
4	16	41%	11	28.2%	11	28.2%	1	2.6%	0	0%
5	26	66.7%	5	12.8%	5	12.8%	1	2.6%	2	5.1%
6	12	30.8 %	13	33.3%	11	28.2%	2	5.1%	1	2.6%

Source: Own elaboration.

Table 2 refers to decomposition. For item 1, 20.5% of the students surveyed stated that they always analyze a subtraction problem in parts to find the solution, while 41% almost always do so, while 15.4% said that they sometimes do so. In the case of item 2, 48.7% of the students expressed that they almost always read a paragraph, analyzing the text as a whole. 20.5% always do such analysis, while 23.1% revealed that they do it sometimes. With respect to item 3, only 43.6% always examine what the main idea in the text is, while 20.5% almost always identify it. Meanwhile, 28.2% do it sometimes.

In relation to item 4, 41% stated that they always identify the main character first when they read, 28.2% said almost always and another 28.2% said that they sometimes identify the main character first. On the other hand, in item 5, 66% expressed that they first identify the elements they need when they are going to bake a cake. 12.8% expressed that they almost always and identify the elements they need. Meanwhile, 12.8% (sometimes). When examining item 6, it was found that 30% of the respondents always determine what they want to obtain, before doing a task. Likewise, 33.3% almost always establish it, while in 28.2% (sometimes).

When observing the results, which pay attention to the decomposition when doing some daily tasks by students such as cooking, doing homework, reading or doing mathematical operations, whose results coincide with those issued by CSTA & ISTE (cited in Barrera & Montaño, 2015) who argue that PC reinforces educational standards in all subjects to enhance the student's ability to solve problems and thus develop higher order thinking. Finally, a representative group of students expressed that they almost always divide problems into small parts for their solutions. This is consistent with the OECD's statement (OECD, 2014) that identifying and analyzing problem situations leads to constructive and reflective citizens.

Table 3. Pattern recognition.

Items	Always	Almost always	Sometimes	Almost never	Never

	Fs	%	Fs	%	Fs	%	Fs	%	Fs	%
7	8	20.5%	12	30.8%	10	25.6%	6	15.4%	3	7.7%
8	11	28.2%	5	12.8%	11	28.2%	7	17.9 %	5	12.8%
9	26	66.67%	8	20.51%	3	7.69%	2	5.13%	0	0%
10	10	25.64%	15	38.56%	10	25.64%	4	10.26%	0	0%
11	23	58.97%	10	25.64%	4	10.26%	2	5.13%	1	2.56%
12	6	15.38%	15	38.46%	13	33.33%	5	12.82%	0	0%

Source: Own elaboration.

Item 7 of Table 3 showed that 30.7% of the respondents almost always classify each animal with its species in a logical way when they read a story about the animal world, while 20.5% said that they always specify it. Meanwhile, 25.6% expressed that they sometimes do such classification. In the case of item 8, 28.2% always analyze the veracity of the response, while 28.2% said that sometimes they do so as well. Others said they sometimes (12.8%) consider truthfulness. Regarding item 9, 66.67% of the students were of the opinion that, in order to solve an equation, they always order in an appropriate way to give it a solution, while 20.51% indicated that almost always. Regarding item 10, it was verified that 38.56% of the respondents said that they almost always establish a relationship between the data of the problem, while in the alternatives they always and sometimes with equal percentage (25.64%) said that they establish such relationship.

Regarding item 11, 58.97% of the students considered that they always organize the data of the problem in order to find the solution. Meanwhile, 25.64% indicated that they almost always do it and another group mentioned that they sometimes do it (10.26%). Likewise, in item 12, 38.46% almost always classify a figure according to its characteristics. On the other hand, 33.33% indicated that they sometimes do such classification, while 15.38% said that they always do it.

So that the results found, evidence that most students perform simple sequencing processes (decomposition) and sometimes seek comparisons, differences or patterns between difficulties (decomposed), which help to solve problems, which is in accordance with the approach of Gurises Unidos and Fundación Telefónica - Movistar (2017) for the recognition of patterns in the problems presented that help to simplify the tasks to do problem solving in technology.

Table 4. Abstraction.

Items	**Always**		**Almost always**		**Sometimes**		**Almost never**		**Never**	
	Fs	**%**	**Fs**	**%**	**Fs**	**%**	**Fs**	**%**	**Fs**	**%**

13	5	12.82%	10	2564%	13	33.33%	7	17.95%	4	10.26%
14	14	35.9%	15	38.46%	6	15.38%	1	2.56%	3	7.69%
15	17	43.59%	13	33.33%	9	23.08%	0	0 %	0	0 %
16	11	28.21%	10	25.64%	11	28.21%	3	7.69%	4	10.26%
17	8	20.51%	13	33.33%	10	25.64%	3	7.69%	5	12.82%
18	22	56.41%	12	30.77%	5	12.82%	0	0 %	0	0 %

Source: Own elaboration.

In Table 4, item 13 established that 33.33% of the respondents sometimes represent the solution symbolically when they have a reasoning problem. While 25.64% said that they almost always symbolize it, although 17.95% almost never. Regarding item 14, 38.46% almost always imagine the different characteristics to solve when they draw a car. Another 35.9% said they always guess the various characteristics, while 15.38% said they sometimes consider it.

On the other hand, item 15 showed that 43.59% of the students said that they always conceptualize before solving the problem. A 33.33% indicated that they almost always do so. Another group of respondents said “sometimes” (23.08%). Regarding item 16, it was shown that 28.21% “always” and “sometimes”, respondents said that to solve geometry exercises they use figures to solve the exercise. On the other hand, 25.64% indicated that they almost always do so. Regarding item 17, 33.337% of the students considered that they almost always locate an address using sketches or street guide. Some 25.64% said sometimes, while other respondents said they always (20.51%) use it. Similarly, in item 18, 56.41% indicated that they always require instructions to guide the preparation and preparation of a cake. Others revealed that 30.77% (almost always) and 12.82% (sometimes) require instructions to guide them.

Thus, the results obtained show that students filter data without hierarchy which could mean a difficulty when abstracting models, images or simulations to represent the important variables of the problems in the technological era. With which abstraction could create solutions for problems in the real world of the 21st century (Rincón & Ávila, 2016).

Table 5. Algorithm design.

Items	**Always**		**Almost always**		**Sometimes**		**Almost never**		**Never**	
	Fs	**%**	**Fs**	**%**	**Fs**	**%**	**Fs**	**%**	**Fs**	**%**
19	6	15.38%	25	64.10%	5	12.82%	2	5.13%	1	2.56%
20	15	38.46%	9	23.08%	12	30.77%	5	12.82%	1	2.56%
21	13	33.33%	13	33.33%	5	12.82%	3	7.69%	5	12.82%
22	9	23.08%	4	10.26%	15	38.46%	4	10.26%	7	17.95%
23	7	17.95%	8	20.51%	14	35.90%	5	12.82%	5	12.82%

24	9	23.08%	4	10.26%	12	30.77%	6	15.38%	8	20.51%

Source: Own elaboration.

Table 5 shows the results of the algorithm design on the PC. Revealing for item 19, that 64.10% of the students, almost always look for several solution alternatives of an arithmetic problem, while 15.38% always perform such action. Another 12.82% said that they sometimes look for options.

In the case of item 20, 38.46% of the students said that always the subtraction exercises have only one solution procedure. Some 30.77% expressed that sometimes they have only one way. 23.08% revealed that they almost always think that there is only one way to solve them. As for item 21, the respondents indicated that always (33.33%), almost always (33.33%), sometimes (12.82%) mathematics exercises have unique solutions to solve them.

In relation to item 22, 38.46% stated that they sometimes know solutions to problems using algorithms and 23.08% said they always know solutions to a problem using algorithms. On the other hand, item 23 verified that 35.90% of the respondents can sometimes represent a possible solution to perform a subtraction using algorithms, 20.51% said almost always, while 17.95% indicated that they always perform the action. On the other hand, when examining item 24, it was found that 30.77% of the respondents sometimes give a solution to a problem using algorithms. Likewise, 23.08% always provide the solution using algorithms, although 20.51% never establish steps to provide a solution with algorithms in a task.

Observing the results, it is evident that students perform some daily tasks whose sequences allow them to be initiated in the basic structures of sequential programming, enhancing processes and steps towards problem solving. Thus, students partially describe the ordered sequence of steps leading to the solution of a given problem (Vasquez, 2012).

ANALYSIS OF QUALITATIVE RESULTS

The analysis of the categories according to direct observation is presented below.

Table 6. Direct observation in class.

Categories	Direct observation
Motivation	Questions about algorithms and Scratch are generated among students. During the diagnostic and work sessions, the students showed great interest in the exercises they were performing. Most of the students were motivated to use the computers and Scratch. Impatience and eagerness to finish the guide cause them to make mistakes during the process. This evidences difficulty in working with the tools in Scratch. Some students proposed solutions to the problems, while those who had difficulties were eventually helped by their peers who had already finished.

Computational Thinking	At the beginning, the students sought the teacher's approval in the actions undertaken to decompose some sequences. Students visualize the problem from different perspectives to be decomposed and propose solution models from the decomposition of steps. In the first sessions it was observed that students made little use of the PC, however, after understanding what decomposition and pattern recognition is, they improved and others with outstanding activities in abstraction and algorithm design with block technology in Scratch.
Troubleshooting	It was observed that some students perform very fast analysis and launch very accelerated responses. Students are challenged by searching for the answer to each of the questions. Spaces for discussion and analysis were generated. In pseudocode, there was some alarm due to the new vocabulary; some did not understand these concepts. They helped each other to understand them. Problem-solving skills improved through the application of all sessions.
Development of Activities	The constructivist approach allowed the active and collaborative learning of the students in the development of the activities, who were able to establish the steps to follow to reach an objective and carry out the activities. Therefore, teamwork was fundamental to learn about the Scratch program, since it generated some uneasiness about its usefulness. At the beginning the guide was handed out for individual work, but they looked for a partner for better understanding. They were allowed to work in groups. Excellent collaborative work is demonstrated in the implementation of the program.
Algorithm design in technology	When starting to work out algorithms individually, students skip processes thinking that they are obvious and do not need to be written down. A pedagogical silence is created for the analysis of each of the questions. Few students see the process of creating an algorithm as easy, however, as they progressed through the work sessions and interacting with the work tool, it was observed that a large group of students proposed solution sequences based on the algorithm design, which allowed them to develop skills and thinking in decomposition and pattern recognition.

What was stated in the previous sections, accounts for the PC category. In addition, the taste for the use of Scratch was verified and the transfer of knowledge of the theoretical principles, PC development, to form critical, constructive and reflective citizens was achieved (OECD, 2014) and likewise, the interest in block programming was boosted. Which agrees with Sáez-López and Cózar-Gutiérrez (2015) that block programming favors the active participation of students while they learn contents set out in the study plan (Zapata-Ros, 2015). This provides an answer to the third object of the research.

Thus, it is confirmed that with Scratch, factors consisting of the grouping of events, sequences and parallelisms are formed (Sanchez, 2016) for the development of PC. These components can be grouped by the difficulty involved for students in formulating problems and representing their solutions, so that these solutions can be effectively executed by a computer (Wing, 2006, as cited in Zapotecatl, 2018), a situation that was demonstrated in the developed sessions. Thus, Scratch as an ideal tool for the development of visual skills, allows organizing data logically and analyzing them; representing data through abstractions; automating solutions through algorithmic thinking; (ISTE & CSTA, 2011), as well as the combination of steps when using block programming with and visual images in a methodical and orderly manner (Zapotecatl, 2018). In addition, the role of the students was active and collaborative as they participated and questioned using critical thinking. Which coincides with Sáez-López and Cózar-Gutiérrez (2015) who argue that in visual programming when designing simple algorithms and programs, active learning is encouraged. Thus, it can be seen that the development of CP in the classroom involves the ability to elaborate, evaluate, understand and create computational procedures (Guerrero and García, 2016).

DISCUSSION AND CONCLUSIONS

The design and application of the learning environment required a planned methodology with strategies according to the digital didactic resources to be worked. In this sense, the didactic guides were the cognitive plans that guide the didactic strategy for students to develop the PC through the design of algorithms in technology with Scratch, which responds to the general objective. This is in accordance with Schunk (2012) because strategies were created (didactic guides in different sessions). In this way, problem solving skills can be fostered with the application of strategies that use technological tools such as Scratch, which motivate students through interactive activities and effective tools (Diago & Arnau, 2017) to strengthen logical processes that allow correct modeling, in addition to fostering creativity with interactive actions and effective tools can help strengthen logical processes (Barrera & Montaño, 2015). Hence, motivation should be an incentive within the strategy for PC development, which coincides with Gurises Unidos (Uruguay) and Fundación Telefónica - Movistar (2017) who argue that technologies, as an element of motivation for learning, favoring actions aimed at transforming the education of children and adolescents. Therefore, motivation is an essential category in teaching and learning under a constructivist vision with technology. Because a motivated student interacts with the rest of the students. As demonstrated in the results found. Since, motivation allows adaptation to the environment and the development of intelligence, as a constant capacity for adaptation (Piaget, 1952). This adaptation is the process that produces the assimilation and accommodation of new knowledge, such as the notions of algorithm design with Scratch technology for the development of the PC.

In addition, the student, being motivated, can carry out processes of socialization of his or her thoughts to solve problems, through the sociocultural environment (Vygotsky, 1978), with the guidance of an adult or in collaboration with more capable peers. Likewise, problem solving with Scratch allows the student to acquire sequencing and pattern recognition skills (Molina et al., 2020), which are fundamental for PC development when learning to use code to create objects. Therefore, it is important to know the literature published to date, related to the scope of the methodologies proposed in the educational field, based on the use of tools such as Scratch for the development of Computational Thinking and thus apply the new knowledge to innovative strategies that are proposed in educational institutions in order to encourage students, the interest in creating skills in problem solving through algorithms and technological resources.

Finally, it is concluded that the didactic guides delivered in the various sessions of the didactic sequence contributed to the development of the students' PC, which allows answering the algorithm design category, because the practices in Scratch were key to formulate problems and apply solutions in a methodical and orderly way. Therefore, the appropriate strategy with Scratch for the design and implementation of algorithms contributes to the development of the PC to achieve the competence of problem solving with technology. Thus, the use of Scratch allows students to be the main actors in the construction of their own learning and knowledge (Sáez-López & Cózar-Gutiérrez, 2015), due to the fact that they work actively and collaboratively. In addition, the role of the teacher as a coach, guide and mediator of the learning processes is fundamental in didactic strategies with technology-supported projects.

REFERENCES

Aguilar-Barreto, A., Velandia–Riaño, Y. R., Aguilar- Barreto, C. P., & Rincón Álvarez, G. (2017). Gestión educativa: Tendencias de las políticas públicas educativas implementadas en Colombia. *Revista Perspectivas*, 2(2), 84–94. https://doi.org/10.22463/25909215.1331

Albarracín-Villamizar, C. Z., Hernández-Suárez, C. A., & Rojas-Suárez, J. P. (2020). Objeto virtual de aprendizaje para desarrollar las habilidades numéricas: una experiencia con estudiantes de educación básica. Panorama, 14(26), 111–133. https://doi.org/10.15765/pnrm.v14i26.1486

Arancibia, V., Herrera, P., & Strasser, K. (1999). Teorías cognitivas del aprendizaje. En *Psicología de la educación.* (pp. 75-96). México: Alfaomega.

Avendaño, W.R. y Parada, A. (2011). Un modelo pedagógico para la reproducción y transformación cultural en las sociedades del conocimiento. *Investigación & Desarrollo*, *19*(2), 398-413. https://dialnet.unirioja.es/servlet/articulo?codigo=6554957

Barrera, R., & Montaño, R. (2015). Desarrollo del Pensamiento Computacional con Scratch. *Nuevas Ideas en Informática Educativa TISE*, *11*, 616-620.

Chaparro Rodríguez, J. C., Jaimes Márquez, M. C., & Prada Núñez, R. (2018). Modernidad y Educación: : Una reflexión a propósito de sus legados y desafíos. *Revista Perspectivas*, 3(2), 121–132. https://doi.org/10.22463/25909215.1672

Congreso de la República de Colombia (1994, 08 de febrero). *Ley 115 de 1994.* https://www.mineducacion.gov.co/1621/articles-85906_archivo_pdf.pdf

Computer Science Teachers Association & International Society for Technology in Education. (2011). *Computational Thinking in K–12 Education leadership toolkit*. https://id.iste.org/docs/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4

Contreras-Colmenares, A. F., & Jiménez-Villamarín, I. (2020). Uso de la tecnología en el desarrollo de competencias de lectura y de escritura. *Revista Perspectivas*, 5(2), 54–71. https://doi.org/10.22463/25909215.2830

Creswell, J. W. (2005). *Educational research: Planning, conducting and evaluating quantitative and qualitative researc*h. Upper Saddle River, New Jersey: Pearson Education.

Creswell, J. W. (2012). *Educational research. Planning, conducting and evaluating quantitative and qualitative research* (4ª. ed.). USA: Pearson.

Diago, P. D., & Arnau, D. (2017). Pensamiento computacional y resolución de problemas en Educación Infantil: Una secuencia de enseñanza con el robot Bee-bot. En FESPM (Ed.), *VIII Congreso Iberoamericano de Educación Matemática (CIBEM)* (pp. 255–263). Zaragoza, España.

García, R. (2000). *El conocimiento en construcción. De las formulaciones de Jean Piaget a la teoría de sistemas complejos*. Barcelona: Gedisa.

Giraldo, L. (2014). *Competencias mínimas en pensamiento computacional que debe tener un estudiante aspirante a la media técnica para mejorar su desempeño en la media técnica de las instituciones educativas de la alianza futuro digital Medellín* (tesis de maestría). Universidad EAFIT, Medellín, Colombia.

González, A., Rodríguez, A., & Hernández D. (2011). El concepto zona de desarrollo próximo y su manifestación en la educación médica superior cubana. *Revista Cubana de Educación Médica Superior*, *25*(4), 531-539.

Guerrero, M., & García, J. (2016). Desarrollo del pensamiento algorítmico con el apoyo de objetos de aprendizaje generativos. *Pixel-Bit*. *Revista de Medios y Educación*. (49), 163-175.

Gamboa, A. (2019). Tradiciones y problemas de la investigación educativa. En: Chévez, C. & Melengue, J. (Eds.), Tendencias y desafios de la investigación educativa: aproximaciones teóricas y metodológicas (25 – 33). El Salvador: INFOD.

Gurises Unidos y Fundación Telefónica Uruguay (2017). *Pensamiento computacional. Un aporte para la educación de hoy*. http://www.eduteka.org/articulos/telefonica-pensamiento-computacional

International Society for Technology in Education. (2016). *ISTE standards for students*. ISTE. https://iste.org/standards

Kemmis, S., & McTaggart, R. (1988). *Cómo planificar la investigación acción*. Barcelona: Laertes.

Latorre, A. (2003). *La investigación-acción. Conocer y cambiar la práctica educativa*. Barcelona, España: Ediciones Graó.

Ministerio de Educación Nacional. (2006). *Estándares básicos de competencias en tecnología e informática*. http://www.colegionacionesunidasied.com/pdf/tecno.pdf

Ministerio de Educación Nacional. (2008). *Orientaciones generales para la educación en tecnología. Ser competente en tecnología. ¡Una necesidad para el desarrollo!* Bogotá, Colombia: Mineducación – Ascofade.

https://www.mineducacion.gov.co/1621/articles-160915_archivo_pdf.pdf

Miguel Ángel Medina Romero, Ismael Fernando Valencia Bustamante, Ibeth Catherine Figueroa Sánchez, Paola Elizabeth Pajuelo Garay, Jacksson Yamil Montoya Asprilla, José Manuel Medina. EFFECTIVE COMMUNICATION IN THE CLASSROOM: SPEAKING TECHNIQUES FOR UNIVERSITY TEACHERS. Journal of Positive Psychology & Wellbeing. 2021, Vol. 5, No. 4, 2402 – 2414.

Molina, A., Adamuz, N., & Bracho, R. (2020). La resolución de problemas basada en el método de Polya usando el pensamiento computacional y Scratch con estudiantes de Educación Secundaria. *Aula Abierta*, *49*(1), 83-90.

Novak, J. (1988).*Teoría y Práctica de la educación*. Madrid: Alianza Editorial.

Ontoria, A., Ballesteros, A., Cuevas, C., Giraldo, L., Martín, I., Molina, A., Rodríguez, A., & Vélez, U. (2006). *Mapas conceptuales. Una técnica para aprender*. Madrid: Narcea, S.A.

Organisation for Economic Cooperation and Development (2014). Assessing problem-solving skills. En: PISA *2012. Students' skills in tackling real-lifeproblems* (pp. 25-46). Paris: OECD Publishing. http://doi.org/10.1787/9789264208070-6-en

Parada, A. E., & Avendaño, W. (2013). El currículo en la sociedad del conocimiento. *Educación y Educadores*, *16*(1), 159-174. https://educacionyeducadores.unisabana.edu.co/index.php/eye/article/view/2006

Penagos, M., Mariño, L. F., & Hernández, R. V. (2017). Pensamiento matemático elemental y avanzado como actividad humana en permanente evolución. *Revista Perspectivas*, 2(1), 105–116. https://doi.org/10.22463/25909215.1289

Piaget, J. (1952). *The origins of intelligence in children.* New York: International Universities Press.

Prada-Núñez, R., Hernández-Suárez, C.A., & Fernández-Cézar, R. (2020). Procesos matemáticos en la práctica pedagógica: un comparativo entre Colombia y España. *Aibi Revista De investigación, administración e ingeniería*, 8(1), pp. 29-36. https://doi.org/10.15649/2346030X.629

Restrepo, B. (2004). La investigación-acción educativa y la construcción de saber pedagógico. *Educación y Educadores*, (7), 45-55.

Rincón, A., & Ávila, W. (2016). Una aproximación desde la lógica de la educación al pensamiento computacional. *Sophia: colección de Filosofía de la educación*, (21), 161-176. https://doi.org/10.17163/soph.n21.2016.07

Román, M. (2015). Test de Pensamiento Computacional [Test en línea]. http://goo.gl/IYEKMB

Román-González, M., Pérez-González, J. C. y Jiménez-Fernández, C. (2015). Test de Pensamiento Computacional: diseño y psicometría general. *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC*). Madrid, España. https://doi.org/10.13140/RG.2.1.3056.5521

Sáez-López, J.M. y Cózar-Gutiérrez, R. (2017). Programación visual por bloques en Educación Primaria: Aprendiendo y creando contenidos en Ciencias Sociales. *Revista Complutense de Educación, 28*(2), 409-426.

Schunk, D. H. (2012). *Teorías del aprendizaje. Una perspectiva educativa.* México: Pearson.

Taylor, S. J., & Bogdan, R. (1996). *Introducción a los métodos cualitativos de investigación*. Barcelona: Paidós.

Vallejo, A., García, B., & Pérez, M. (1999). Aplicación de un procedimiento basado en la zona de desarrollo próximo en la evaluación de dos grupos de niños en tareas matemáticas. *Revista de educar*, (9). http://www.quadernsdigitals.net/datos_web/articles/educar/numero9/aplicacion.htm

Vázquez, J. F. (2012). *Análisis y diseño de algoritmos*. México: Red Tercer Milenio S.C.

Vygotski, L. (1978). Interacción entre aprendizaje y desarrollo. En *El desarrollo de los procesos psicológicos superiores* (pp. 123-140). Barcelona: Editorial Crítica.

Vygotski, L. (1980). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University.

Wing, J. (2006). Computational Thinking. View Point. *Comunication of ACM*, *49*(3), 33-35. http://doi.org/10.1145/1118178.1118215

Wing, J. (2012). *Computational Thinking* [Video]. https://www.microsoft.com/en-us/research/video/computational-thinking/

Zhao L., Liu X., Wang C., Su Y.-S. (2022) Effect of different mind mapping approaches on primary school students' computational thinking skills during visual programming learning. *Computers and Education, 181*.

Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. *RED-Revista de Educación a Distancia*, *46*(4), 1-47.

Zapotecatl, J. L. (2018). *Introducción al pensamiento computacional: Conceptos básicos para todos*. México: Academia Mexicana de Computación, A.C.