

Towards the Development of Computational Thinking and Mathematical Logic through Scratch

Benjamín Maraza-Quispe¹, Ashtin Maurice Sotelo-Jump²

Olga Melina Alejandro-Oviedo³, Lita Marianela Quispe-Flores⁴

Lenin Henry Cari-Mogrovejo⁵, Walter Cornelio Fernandez-Gambarini⁶, Luis Ernesto Cuadros-Paz⁷

Facultad de Ciencias de la Educación, Universidad Nacional de San Agustín de Arequipa, Arequipa-Perú^{1,3,5,6,7}

Facultad de Bioingeniería Universidad de Ingeniería y Tecnología Lima-Perú²

Facultad de Psicología, RR. II. y Cs. de la Comunicación, Universidad Nacional de San Agustín de Arequipa, Arequipa-Perú⁴

Abstract—Currently the need to provide quality education to future generations has led to the development of new teaching methodologies, within this fact the tools provided by information technologies have been positioned as the future of learning, in this sense, the learning to program is no longer considered a selective skill in the field of computing, being today a necessity for any student who wants to be competent in this globalized and dynamic world. Within this context, the present research aims to analyze to what extent the use of the Scratch programming language allows the development of computational thinking skills and mathematical logic. The methodology consisted of the application of programming fundamentals through Scratch 3.0 to an experimental group composed of 25 students who were randomly selected from a population of 100 students, the data collection was carried out through a test of logical reasoning standardized by Acevedo and Oliva and a test of levels of computational thinking standardized by González. According to the results, a significant difference is postulated in the performance of the students in both tests, having a more considerable improvement in the criteria: Loops, Control of Variables (CV), Probability (PB) and Combinatorial Operations (CB). Therefore, it is concluded by highlighting the importance of teaching basic concepts of Computer Science such as computational thinking and mathematical logic, since it contributes to the internalization of concepts when developing algorithms in problem-solving.

Keywords—Scratch; computational thinking; logic reasoning; teaching

I. INTRODUCTION

Nowadays, within the curricular networks of different Educational Institutions of Regular Basic Education (EBR), the implementation of IT tools in education is inappropriate due to a lack of competence of teachers in their use and also in part due to a lack of knowledge of the virtues that bring these strategies into the student's cognitive development, thus creating a perception of complexity towards programming for both students and teachers [1].

However, the virtue of the combination of conventional methodologies with IT tools are decisive, thus establishing young people who use computational thinking and mathematical-logical reasoning in their daily lives. According to [2] computational thinking can be defined as the ability to solve problems through capabilities such as algorithms and

computational methods, this thinking is divided into four main processes: decomposition, abstraction, pattern recognition and algorithms. The aforementioned processes will have a fundamental role in the analysis of the benefits it brings to students, being these evaluated through the test designed by [3].

Similarly, the reasoning is made up of various capacities associated with mathematics, such as Pythagorean arithmetic or Euclidean geometry. On the other hand, beyond traditional conceptions, at present mathematical logic is considered as the reasoning that causes science through the validation of knowledge by the scientific method [4]. The importance of logical reasoning has always been valued in primary and secondary education. However, in the context presented, the following question arises: to what extent does the teaching of programming language through Scratch 3.0 develop Logical Reasoning and Computational Thinking in students of Regular Basic Education?

According to the existing bibliography, there seems to be a tentative answer that affirms said development of skills, but even more, there is an improvement in the acceptance of the error, since the success of this type of software lies in the development of the expected skills, thanks to the learning programming by goals [5].

II. THEORETICAL FRAMEWORK

A. The Programming Language

In the world of computing, any technology or object that is called or has a computer or processor inside, works with a single language called binary code, that is, basically for a computer there are only ones and zeros.

Therefore, when it comes to the programming language, it is subdivided into two: high-level and low-level language; In this case, in the present work the first one will be taken as the central point, since the programs that will be exposed later use this type of programming language. These are closer to mathematical and natural language [4].

There are many types of programming languages below, in Table I which places Scratch within the types of programming languages:

TABLE I. TYPES OF PROGRAMMING LANGUAGE

Language type	Description	Software
C++	Intended for the development of programs or packages. Arduino Cross-platform Python programming language, ideal for beginners.	Eve online, Panda3D
Visual	It is a programming language in a growing state, mainly used to teach basic knowledge of the programming language.	Scratch, CODE
JavaScript	Designed to create programs that will be stored in web pages, it is also ideal for creating and implementing effects and actions.	Android Studio

B. Scratch 3.0 Software

Scratch defines itself as a program that allows you to create stories, games or animations; which can then be shared with the community, which provides positive feedback for users. In addition, it helps improve creativity and systematic thinking in young people [6].

In addition, among the numerous visual programming software, Scratch is the educational program par excellence, which provides a solution to the common abandonment of programming courses, caused by a perceived high difficulty that this activity entails [7].

Likewise, Scratch has a constructive and above all active teaching process, which generates in students a better experience when learning to program, and as Paper defended: programming languages must have a “low floor” and a “high ceiling” In other words, it should not be a challenge to understand how to start programming, however the possibilities must be gigantic [7].

C. Computational thinking

In this modern and changing world, technology is an important part of the development of new methods to improve the productivity of certain products [8].

Computational thinking is that way of solving problems, using computational methods or methods normally used by technology such as algorithms [2].

This process consists of several parts to follow, which are developed through a specific reasoning called computational logic.

Computational thinking is of great importance for current demands, these being characterized by the constant problems that employees are subjected to, for example. In short, computational thinking is imperative if you want to “survive” in this globalized world. Because the teaching of this system provides the population with a tool so that, as mentioned above, it improves production and efficiency in any type of work environment and in daily life.

In this case, greater importance will be given to decomposition as a fundamental part of computational logic and therefore of computational thinking.

D. Logical Reasoning

Logical reasoning is composed of various capacities associated with mathematics, such as Pythagorean arithmetic or Euclidean geometry. On the other hand, beyond traditional conceptions, at present mathematical logic is considered as the causal reasoning of science through the validation of knowledge by the scientific method [9].

It is extremely important, the importance of logical reasoning has always been valued in primary and secondary education. However, in recent years, the development of these mathematical skills is the same, the methods can be varied, in this case we will study the impact of IT on the learning of mathematical logic.

E. Teaching Scratch 3.0 to EBR Students

Reference [7], dealing specifically with Scratch, as this is a type of block-oriented visual programming language, it has a constructive teaching process, ideal to initiate students to programming. This gradual process accompanied by the development of expected skills through a programming system for individual goals makes Scratch 3.0 the programming software par excellence for young people.

On the other hand, it is worth emphasizing what the literature in recent years says about the development of both thoughts and their relationship with Scratch: first, logical reasoning is composed of various capacities associated with mathematics, such as Pythagorean arithmetic or Euclidean geometry. On the other hand, at present, mathematical logic is considered as the reasoning that gives rise to science through data validation [4] that is why authors such as [10] present a curriculum for students that are oriented to their true needs; therefore, globalization is considered as the first axis.

As for how Scratch manages to develop the aforementioned capacities, everything lies in the principles with which it was devised, below in Table II, the three fundamental principles are presented according to [7].

TABLE II. SCRATCH 3.0 PRINCIPLES [7]

Scratch Principles	Description
The programming language must be playful	The ease with which you can try different options to complete a certain action is essential to improve the experience of what it is to program.
The programming language must be meaningful	When a person wants to learn something new, one of the best ways to do it is if the activity is meaningful to the person, that is, it has a certain degree of relevance and authenticity for each user. Scratch is designed precisely to meet this requirement; it is diverse and personal at the same time.
The programming language must be social	Scratch is closely linked to its website, since in this way a good community has been consolidated around the MIT platform so that each user, regardless of age, can share your work with the whole world with just one click; being able to receive the necessary improvements for their animation, comic strip, game or project in general. In short, thanks to the Scratch website, you have personal feedback for each user.

The model proposed by [7] is achieved thanks to the Scratch interface. The same one that had an original design motivated to meet the learning needs and, besides, create interest in children and young people. Thus, in the first instance, Scratch began to be used in places outside the classroom, although it would inevitably reach the curricula of thousands of schools around the world, due to this invitation to exploration and exchange with peers [11].

F. The Literature on Scratch and its Impact on Teaching

Especially in the last decade, studies have been conducted at schools about student acquisition in programming and computational thinking skills. Scratch is regarded as a useful tool in teaching programming or ensuring that students acquire computational thinking skills [12-13].

Scores obtained from assessing Scratch projects via Scratch web tool and students' Computational Thinking Levels do not differ based on gender. In other words, gender is not influential on students' project assessment scores. While literature presents extensive proof for the impact of gender on student's characteristics related to computers or programming [14-15]. There is a significantly high relationship between students' Scratch skills and their computational thinking skills. In other words, development in students' programming skills in Scratch will cause similar increases in their computational thinking skills or improvements in their Computational Thinking Levels will generate increases in their Scratch skills. Literature provides extensive proof that the process of programming is not mechanical, but a thinking discipline [16].

According to [17], in an investigation developed which aimed to compare the scores of fifth-grade students obtained from Scratch projects, the scores obtained of the Scale of Levels of Computational Thinking and Examine this comparison in terms of different variables. A correlational research model was used in the study in which 31 students participated. Students were taught basic programming using Scratch for 6 weeks' period. At the end of the training, the students' scheduling skills were measured through the Dr Scratch web tool. Computational thinking skills were measured using Scale of levels of computational thinking that includes 5 factors: creativity, problem-solving, algorithmic thinking, collaboration and critical thinking. The data were analyzed for internal reliability to calculate the reliability of the scale. Cronbach's alpha reliability coefficient was found to be 0.809. It was found that the scores obtained by the students by use of any of the measurement tools did not differ depending on gender or period of computer use, however, a significant high-level relationship was observed between students' programming skills with Scratch and your computational thinking skills.

III. METHODOLOGY

La metodología utilizada está enfocada a un estudio cuantitativo para determinar puntos concluyentes sustentados en datos numéricos sometidos a tratamiento estadístico para corroborar su validez, de esta manera es que se opta por este tipo de metodología en lugar de un estudio cualitativo, con el fin de buscar la mayor certeza de que el tratamiento es el principal motivo de los resultados y no factores externos,

además de ser más relevante al momento de replicarlo en diferentes contextos mundiales.

A. Objective

Analyze the development of computational thinking and mathematical logic through the visual programming language Scratch.

B. Population and Sample

The total population is made up of 100 students in the third grade of Regular Basic Education, of which 25 students were selected for the experimental group through a simple random sampling. With the sample described above, the constant evaluations to which they will be subjected will be important, through tests designed to quantify the use of computational thinking in solving problems.

C. Process

Learning sessions were designed consisting of a minimum of four lessons, each lasting sixty minutes and divided into four parts: purpose, where the goal of the class is expressed; development, where we proceed with the explanation of the blocks and the structure of visual programming; evaluation, a moment in which an exercise with the name of "Challenge" is proposed and finally exit, the final section where all the concepts are recovered and conclusions are drawn.

Finally, the sample was evaluated with a pre-test, before applying the programming lessons and at the end of the lessons, it was completed with a post-test; both evaluations as a data collection instrument.

D. Data Collection Instrument

Two collection instruments were used due to the two fields of study: mathematical logic and computational thinking. In the case of the first, what was proposed by Tobin and Copie (1981), and their study "Test of Logical Thinking" (TOLT), and the subsequent conversion and validation to the Spanish language, carried out by [18], the same that assesses through open and closed questions five criteria on logical reasoning: proportionality (PP), control of variables (CV), probability (PB), correlation (CR) and combinatorial operations (CB).

In the case of computational thinking, it was based on Gonzales' study, in which, through a test of 28 multiple-choice items, it is aimed at the standard quantification of the levels of computational thinking in the subjects, in solving problems by helping each other. With computational concepts: Basic Directions, Loops (repeat times), Loops (repeat until), Simple Conditional, Compound Conditional, While (while) and Simple Functions.

The test lasts 45 minutes and as an objective population to students from twelve years to fifteen years [3].

IV. ANALYSIS AND RESULTS

A. Data Collection, Classification and Analysis

According to the results obtained, the performance of the sample between criteria was considerably varied, so it was decided to carry out a detailed study by the criterion of each collection instrument: in the case of mathematical logical reasoning, the marks were classified into five criteria that are

those presented by the author of the test and in the case of computational thinking, the same was done but in seven evaluation criteria. In this way, we seek to obtain more accurate and specific data on in which fields there is a significant improvement after treatment (Scratch 3.0 teaching).

The final classification table for the Logical Reasoning Test (TRL) is shown below. With a maximum score of 50 per criterion.

From the data processed in Table III, not all the criteria evaluated by the Acevedo and Oliva test had the same change. The second and third criteria are specifically rescued, which explain: use of variables and proportionality where there is a more significant change concerning the others. This is due to the same process of using Scratch [7].

Regarding the second field of study, as mentioned above, the same principle applies with the difference that each criterion of the Computational Thinking Test has a maximum score of 100.

As analysed from Table IV, specifically, in the second and third criteria, there is a more significant improvement. These criteria measure the ability of students to use loops both in numerical repetitions (repeat how many) and in repetitions with conditionals at the end (repeat until). This concept is related to what [19]: (...) When writing code, students learn how to organize a process, recognize routines or repetitions and discover errors in their computational thinking when their program does not work according to the idea or expectation with which it was conceived. All of them are key features of computational thinking. (...) Which would explain the best performance in the criteria most closely related to the processes that define computational logic and problem solving, fundamental pillars in computational thinking.

TABLE III. EVALUATION CRITERIA IN THE LOGICAL REASONING TEST

Criteria	Total of the results obtained	
	Pre-test	Post-test
Proportionality	48	50
Variables Control	26	37
Probability	27	36
Correlationt	27	29
Combinatorial Operations	33	33

TABLE IV. EVALUATION CRITERIA IN THE COMPUTATIONAL THINKING TEST

Criteria	Total of the results obtained	
	Pre-test	Post-test
Basic Directions	99	100
Loops (repeat times)	41	71
Loops (repeat until)	39	69
Simple conditional	42	50
Compound Conditional	33	48
While	40	55
Simple Functions	32	42

B. Validation of the Proposal

To validate the presented proposal, the use of analysis of variance for paired samples was used, through this statistical treaty it is sought to conclude whether the treatment carried out had a significant impact or otherwise the results are not good enough to affirm that teaching Scratch improves logical reasoning and computational thinking.

According to Fig. 1, the results show a greater significant effect between the pre and post-test.

According to Fig. 2, the results show a greater significant effect between the pre and post-test, based on the postulate of [18] where there is evidence of a greater development by criteria in the post-test.

On the other hand, although there is a considerable improvement in the criteria explained, analyzing the means of both groups of data, the difference of is 1.16. Using the studied authors and analyzing Fig. 1, it can be explained: that a possible cause may be the orientation of the methodology in the treatment, due to the educator’s determining degree in logical reasoning, as concluded by [20]: Development of Logical Reasoning: (...) The teacher must provide his students with the necessary tools to learn, thus mediating their learning. It is an urgent need to promote the development of capacities and values in the classroom. (...)

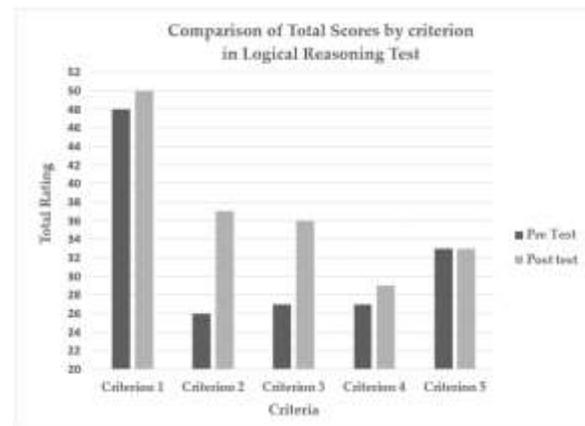


Fig. 1. Comparison per the Criterion of Results in the Logical Reasoning Test.

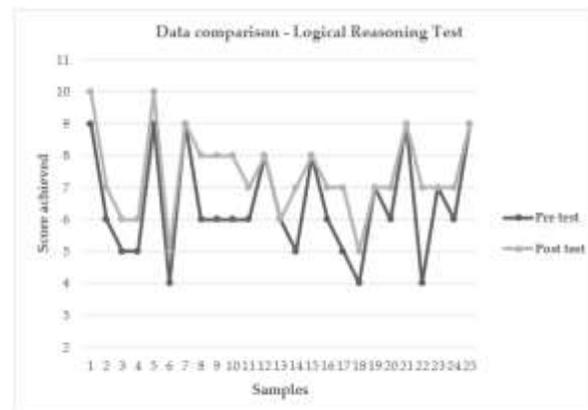


Fig. 2. Comparison of General Results in the Logical Reasoning Test.

Likewise, developing in the case of the first and fifth criteria, when dealing with Propositions and Combinatorial Operations, the characteristics of the sample and the treatment time may have influenced the results, because they are more complex concepts than when comparing them with age. The sample mean, is not yet fully internalized.

A considerable difference between the pre and post-test is observed in Fig. 3 and Fig. 4, which promises good results in the treatment. However, the improvement has not been uniformed in all the criteria, see Table III, in which criteria such as basic directions and simple functions have fairly close results.

In conclusion, the treatment in the selected sample has had the expected impact, significantly increasing performance within the TPC, based on [3] study, thus validating the contribution of various authors in the development of Computational Thinking through the teaching of Scratch.

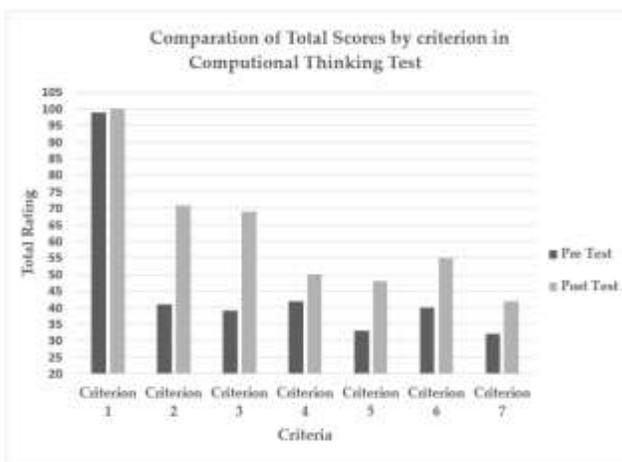


Fig. 3. Comparison per Criterion of Results in the Computational Thinking Test.



Fig. 4. Comparison of General Results in the Computational Thinking Test.

C. T-student for Two Related Samples

The statistical treatise used consists of the formulation of two hypotheses: the null hypothesis and the alternative, depending on the results of the treaty, one of the hypotheses is approved or refuted.

1) Analysis by criterion within the logical reasoning test:

Ho = There is no significant difference in the means of the results in the TRL before and after the treatment.

Ha = There is a significant difference in the means of the results in the TRL before and after the treatment.

$\alpha = 5\%$

According to the data presented in Table V, given that the resulting P value is less than 5% or 0.05, then the null hypothesis is rejected and the alternative hypothesis is accepted, in other words, if there is a significant difference after treatment.

2) Analysis by criterion within the logical reasoning test:

Once the data were validated within the TRL, the significance was then studied by criterion. In this way, more specific conclusions were obtained in singular fields and the impact of the treatment (teaching of Scratch 3.0 in third-year high school students) in the sample.

According to the data presented in Table VI we have

TABLE V. T-TEST FOR TWO PAIRED SAMPLES IN THE LOGICAL REASONING TEST

	Variable 1	Variable 2
Average	6.44	7.4
Variance	2.76	1.45
Observations	25	25
Pearson's correlation coefficient	0.86	7.4
Hypothetical difference of means	0	
Degrees of freedom	24	
T statistic	5.70	
P (T <= t) one tail	3.49	
Critical value of t (one-tailed)	1.71	
P (T <= t) two tails	0.0000070	
Critical value of t (two-tailed)	2.063	

TABLE VI. T-TEST BY CRITERION IN THE LOGICAL REASONING TEST

Criteria	P-Value	alpha	Significant difference
Proportionality	48	50	NO
Variables Control	26	37	SI
Probability	27	36	SI
Correlation	27	29	NO
Combinatorial Operations	33	33	SI

3) General test of the computational thinking test:

Ho = There is no significant difference in the means of the results in the TPC before and after the treatment.

Ha = There is a significant difference in the means of the results in the TPC before and after the treatment.

$\alpha = 5\%$

Table VII shows that, as in the case of the logical reasoning test, the value of P is less than alpha, which is why it is stated that there is a significant difference.

As shown in Table VIII, there is a significant difference in the means of the Computational Thinking Test results. Therefore, it is concluded that the teaching of Scratch does have significant effects on the development of Computational Thinking.

These indicators of learning behaviors in virtual learning environments are very important for self-regulation and reflection of students and teachers within their teaching and learning context. Likewise, teachers could provide very effective feedback by knowing the indicators of learning behavior in which they have weaknesses. That is why teachers considered that these data could help in the redesign of their courses [21].

TABLE VII. T-TEST FOR TWO PAIRED SAMPLES IN THE COMPUTATIONAL THINKING TEST

	Variable 1	Variable 2
Average	13.04	17.4
Variance	19.12	22.42
Observations	25	25
Pearson's correlation coefficient	0.28	-
Hypothetical difference of means	0	
Degrees of freedom	24	
T statistic	-3.98	
P (T <= t) one tail	0.00	
Critical value of t (one-tailed)	1.71	
P (T <= t) two tails	0.000553121	
Critical value of t (two-tailed)	2.06	

TABLE VIII. T-TEST BY CRITERION IN THE COMPUTATIONAL THINKING TEST

Criteria	P-Value	alpha	Significant difference
Basic Directions	0.16	0.05	NO
Loops (repeat times)	0.00	0.05	SI
Loops (repeat until)	0.00	0.05	SI
Simple conditional	0.31	0.05	NO
Compound Conditional	0.07	0.05	NO
While	0.08	0.05	NO
Simple Functions	0.06	0.05	NO

V. DISCUSSION AND CONCLUSIONS

The present investigation concluded that the treatment used in the sample has managed to have a significant impact on the results of both fields of study: computational thinking and logical reasoning.

In the first place, in the case of Computational Thinking, the benefits of programming teaching in students, with abilities related to problem-solving, as well as of all ages, are affirmed, without making any type of distinction because the activity it adapts to the individual capacities of each person. Leaving aside the development of the capacities evaluated during the present work, the improvements also focus on soft skills within the students, bringing them closer to a modern and dynamic environment, typical of the world of work.

Secondly, the learning of mathematical logic has been confirmed by statistical analysis, which affirms the advantages of the implementation of IT in apparently foreign areas such as school mathematics, all this achieved through the application of the TRL a standardized test. In this way, the teaching of Scratch is recommended more specifically to improve the performance of students in the subject of control of variables and probability.

Third, the teaching environment that Scratch creates is worth emphasizing, especially with a view to the future implementation of virtual environments in thousands of schools around the world. This software allows students to work as a team in addition to allowing a sociocultural exchange that enriches their perspective of the world.

In this way, to a large extent, the teaching of Scratch 3.0 has allowed the development of computational thinking and logical reasoning in high school students, which augurs the good relationship between IT and Education, innovating and giving away new learning tools by which, the new generations are formed with modern and fundamental capacities in today's world.

VI. RECOMMENDATIONS AND FUTURE WORK

To obtain a higher percentage of reliability in the results, it is recommended to obtain the data through a qualitative approach to observe qualities and behaviors and work with students from a more personalized perspective using some Artificial Intelligence techniques [22].

REFERENCES

- [1] Maraza Quispe, B. (2011). Influencia de un entorno multimedia de simulación por computadora en el aprendizaje por investigación de la Física. Nuevas Ideas en Informática Educativa, TISE.
- [2] Moreno León, J. (23 de marzo de 2014). Programamos. Obtenido de Videojuegos y "apps": <https://programamos.es/que-es-el-pensamiento-computacional/>.
- [3] Román Gonzalez, M., Pérez Gonzalez, J. C., & Jiménez Fernández, C. (2015). Test de Pensamiento Computacional: diseño y psicometría general. Congreso Internacional sobre Aprendizaje, Innovación y Competitividad.
- [4] Hernandez, L. (2004). Fundamentos de la programación. Madrid, España.
- [5] Sánchez Rey, A. (2016). UNIVERSIDAD DEL PAÍS VASCO. Obtenido de Archivo Digital Docencia Investigación: <https://addi.ehu.es/bitstream/handle/10810/20672/TFG%20Ane%20Sanchez.pdf;jsessionid=B9E714318BADDCE1081E77929F588C86?sequence=2>.

- [6] Scratch-MIT. (s.f.). Scratch. Obtenido de Acerca de Scratch: <https://scratch.mit.edu/about>.
- [7] López-Escribano, C., & Sánchez-Montoya, R. (2012). *Revistas UM*. Obtenido de <https://revistas.um.es/red/article/view/233521/179471>.
- [8] Colegio de Bachilleres. (2008). *Lógica Computacional y Programación*. Mexico.
- [9] Sangüillo Fernandez-Vega, J. (2008). *El pensamiento lógico matemático*. Madrid: Ediciones Akal, S.
- [10] Chamorro, C. (2011). *Revistas UM*. Obtenido de <https://revistas.um.es/educatio/article/view/132961/122661>.
- [11] Maloney, J., Resnick, M., Rusk, N., Silverman, R., & Eastmond, E. (Noviembre de 2010). *The Scratch Programming Language*. *ACM Trans*, 15.
- [12] S. Catlak, M. Tekdal. and F. Baz. The Status of Teaching Programming with Scratch: A Document Review Work. *Journal of Instructional Technologies & Teacher Education*, 2015, 4(3), 13-25.
- [13] O. Ozyurt. and H. Ozyurt. A Study for Determining Computer Programming Students' Attitudes Towards Programming and Their Programming Self - Efficacy. *Journal of Theory and Practice in Education*, 2015, 11(1), 51-67.
- [14] S. Toker. An Assessment of Pre-Service Teacher Education Program in Relation to Technology Training for Future Practice: A Case of Primary School Teacher Education Program, Burdur.
- [15] (Unpublished master's thesis), 2014, Graduate School of Education, METU.
- [16] A.P. Ambrosio, F. M. Costa, L. Almeida, A. Franco, and J. Macedo. Identifying cognitive abilities to improve CS1 outcome. Paper presented at the *Frontiers in Education Conference (FIE)*, 2016.
- [17] Ali OLUK, Özgen KORKMAZ, "Comparing Students' Scratch Skills with Their Computational Thinking Skills in Terms of Different Variables", *International Journal of Modern Education and Computer Science(IJMECS)*, Vol.8, No.11, pp.1-7, 2016.DOI: 10.5815/ijmeecs.2016.11.01.
- [18] Acevedo, J. A., & Oliva, J. (1995). Validación y aplicaciones de un test de razonamiento lógico. *Revista de psicología general y aplicada*, 339-351.
- [19] Valverde Berrocoso, J., Fernández Sánchez, M. R., & Garrido Arroyo, M. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia*(46), 18.
- [20] Cerillo MM. (2002). *Enseñar a Pensar: Desarrollo de Razonamiento Lógico*. Aulo Abierta.
- [21] Maraza-Quispe, B., Alejandro-Oviedo, O., Choquehuanca-Quispe, W. (2020). Towards a Standardization of Learning Behavior Indicators in Virtual Environments. *International Journal of Advanced Computer Science and Applications*, Vol. 11, No. 11. From https://thesai.org/Downloads/Volume11No11/Paper_19-Towards_a_Standardization_of_Learning_Behavior.pdf.
- [22] Maraza, B. (2016). hacia un Aprendizaje Personalizado en Ambientes Virtuales. *Campus Virtuales*, Vol. 5, num. 1, pp. 20-29. en www.revistacampusvirtuales.es.