REVISTA
CÁTEDRA

Uso de SCRATCH en el aprendizaje de Programación en Educación Superior

Use of SCRATCH in Learning Programming in Higher Education

Hamilton Pérez-Narvaéz
Universidad Central del Ecuador, Quito, Ecuador
hperez@uce.edu.ec
https://orcid.org/0000-0002-4496-2549
Rosabel Roig-Vila
Universidad de Alicante, Alicante, España
rosabel.roig@ua.es
https://orcid.org/0000-0002-7466-7364
Lilian Jaramillo–Naranjo
Universidad Central del Ecuador, Quito, Ecuador
lmjaramillon@uce.edu.ec
http://orcid.org/0000-0002-0586-4292

(Recibido: 19/12/2019; Aceptado: 21/12/2019; Versión final recibida: 13/01/2020)

Cita del artículo: Pérez-Narváez, H. y Roig-Vila, R. (2020). Use of SCRATCH in Learning Programming in Higher Education. *Revista Cátedra, 3*(1), 27-45.

Resumen

En los últimos años, el término pensamiento computacional y el lenguaje de programación Scratch han irrumpido en el escenario de la enseñanza, primordialmente a nivel de educación básica y bachillerato. Sin embargo, no se evidencian estudios que fundamenten el uso del lenguaje Scratch a nivel superior, por lo que, se constituyó en objetivo de la presente investigación, la verificación del desarrollo del pensamiento computacional entre los estudiantes de Primer Semestre de la Carrera de Informática de la Universidad Central del Ecuador con base a las ocho dimensiones del pensamiento computacional que son: identificación de patrones, uso de instrucciones, variables, secuencia, operadores, reuso, abstracción y funcionamiento y detección de errores. Con esta finalidad se aplicó un diseño cuasiexperimental con preprueba-postprueba y grupos intactos, con enfoque cuantitativo, de campo con apoyo documental y de nivel explicativo. Posteriormente, para los análisis estadísticos se utilizó la herramienta SPSS, tanto para el cálculo de las medidas de tendencia central y la desviación estándar, así como en la comprobación de las hipótesis mediante pruebas T y en la elaboración de figuras estadísticas. Los resultados obtenidos evidenciaron un mayor desarrollo en los niveles del pensamiento computacional con los estudiantes de Primer Semestre de la Carrera de Informática de la Universidad Central del Ecuador que emplearon Scratch.

Palabras clave
Computación, enseñanza, lenguaje, pensamiento, programación, Scratch.

Abstract
In recent years, the term computational thinking and Scratch programming language have broken into the teaching stage, primarily at the level of elementary and high school education. However, there are no studies that support the use of Scratch language at a higher level, thus the aim of this research was the verification of the development of computational thinking among the students of the First Semester of the Computer Science Degree of the Central University of Ecuador based on the eight dimensions of computational thinking that are: identification of patterns, use of instructions, variables, sequence, operators, reuse, abstraction, operation and error detection. For this purpose, a quasi-experimental design with pre-test-post-test and intact groups was applied, with a quantitative, field approach with documentary support and explanatory level. Subsequently, the SPSS tool was used for the statistical analyzes, both for the calculation of the measures of central tendency and the standard deviation, as well as in the verification of the hypotheses by means of T tests and in the elaboration of statistical figures. The results obtained evidenced a greater development in the levels of computational thinking with the students of the First Semester of the Computer Science Career of the Central University of Ecuador who used Scratch.

Keywords
Computing, teaching, language, thinking, programming, Scratch.

1. Introduction
In recent years, questions such as the following have been discussed by organizations related to the educational field: what are the capacities, abilities, skills, competencies that students must achieve through education in the 21st century? How important is it for the development of people in contemporary society? To answer these questions, several scenarios and multiple proposals on the subject have been proposed; However, there is one that shows a lot of interest for its repercussions, not only in the educational aspect, but in all areas of life called computational thinking.

Computational thinking is a term that Zapotecalt (2015) describes as “a process of problem solving” (p. 12) that involves the development of skills such as: the approach or formulation of problems, using computers for analysis, modeling and resolution. He proposes to organize data in a logical way and then represent them by means of abstractions, arriving at his modeling, algorithmization and finally “generalizing and transferring this problem-solving process to a great diversity of these” (Zapotecalt, 2015, p. 13).

This criterion on the unlimited potential of computing for problem solving led the International Society for Technolgy In Education (ISTE) and the Computer Science Teachers Association (CSTA) to operationalize the concept of computational thinking. In this way, educators of different levels understand what the subject is about and could develop it with their students, expanding their abilities, their creativity and taking advantage of the mainstreaming of information technology in all areas of knowledge.

Also, as an important antecedent for the formulation of computational thinking, Pérez (2017) points to the Third International Conference of Informatics in secondary schools, held in 2008 in the city of Turon, Poland:

> Where several experts analyzed the advantages and difficulties of starting programming learning at an early age with the intention of promoting the development of skills related to computational thinking in students to develop in an environment where the presence of computer technology is increasingly common and important (p. 23).

The need for a second education in computing is mentioned, where the teachers who master this area of knowledge are the bridge that relates other people to the understanding of the principles of computer science. This intention is evidenced in the formulation of a curriculum that exceeds the limitations of the current ones, in which only the management of ICT is taught.

Mittermeir and Syslo (2008) summarize the generality of the conferences by pointing out “several authors in these participations respond to these problems by addressing the question of what computer education has to offer young people to go beyond the skills of how to they use computers. ”(p. 3). Facing this situation, there is a need for research that establishes the advantages of using symbolically mediated tools for programming learning as considered in the Scratch programming language. It is also necessary to determine Scratch's contribution to the development of skills such as modeling, abstraction, pattern recognition, as well as improving values such as cooperative work, tenacity and perseverance in the teaching-learning process.

When doing research based on scientific principles, a critical position is assumed, which Martínez (2001) claims when he proposes not to accept technology as an obligation or simply to oppose it as considered dehumanizing, but that “the incorporation of technology must go preceded by a critical analysis of the needs to be covered with it and the implications that its use has, being willing to accept its consequences ”(p. 197). Taking into account that recommendation and knowing that learning is made up of several curricular elements such as content, methodologies, resources among others; It is precisely in this last component that the research aims to validate the use of Scratch 2.0 as a tool that contributes to the formation of future IT teachers in the development of the cognitive, procedural and emotional fields in an encompassing way.

In this sense, the article is structured so that, initially, related studies and educational experiences about computational thinking are presented, after that, concepts and definitions regarding the elements of computational thinking are presented. Next, the methodology used for the design of the research that was carried out with a quantitative field approach with documentary and explanatory level support is presented, followed by the results obtained, and finally the conclusions obtained are presented.

The objective of this research is to verify the significant difference in the development of computational thinking in students who use Scratch for learning Programming compared to those who do not use it, that is, it is to analyze a before and after the application of a intervention program.

2. Related Concepts

2.1 Scratch

As indicated by Pascual (2015) Scratch “is a programming language developed by a research group of the Massachusetts Institute of Technology (MIT), its main feature is its visual and multimedia programming environment that allows you to create programs easily and intuitively through a graphical interface ”(p. 1).

López (2015) indicates that the Scratch tool “makes programming more fun for anyone who faces learning to program for the first time” (p. 11), and considers that this tool was designed to express ideas creatively with the objective of encouraging the development of logical and computational thinking skills.

Scratch is a new programming environment and as Alba (2008) mentions, “it uses the metaphor of embedded pieces to animate objects that are on the screen and shows all the necessary elements of use such as the stage, the objects and the elements of the language ”(p. 1).

Isuri Sormenezko Zerbitzuak in Scratch Teaching Guide for Teachers (2010) mentions: “Scratch helps to easily understand mathematical and computer concepts that are very well integrated into the program, such as: interactive processes (loops), conditional criteria (yes, then , yes-no), the coordinates in a plane, the variables, etc. ”(p. 4-5), in this way these concepts can be learned within a meaningful and motivating context, since it will not be the same to understand the meaning of the use of the variables in the traditional programming, that when they are being used for the control of visualization of some animation that is being created at that moment.

The Scratch tool works through the creation of projects, based on a main idea, along with modeling and experimentation, to achieve a final product, these project design processes as mentioned in Scratch Teaching Guide for Teachers (2010) , “Develop the necessary skills to get to have: a creative thought, a logical thought, a development of ideas, from its initial conception to the finished project, a clear communication, a systematic analysis, capacity for collaboration, an interactive reflection., etc. ”(p. 6-7).

Scratch allows you to learn to program through experimentation creatively, which helps the development of logical thinking and better understand computational functioning. In addition, it allows the development of mental skills, favoring the understanding of the fundamentals of programming.

2.2 Computational Thinking

Computational thinking is the ability that individuals have to solve problems through the use of technology, so one of the promoters of computational thinking, Wing (2006) points out that “computational thinking involves solving problems, designing systems and understanding human behavior, making use of the fundamental concepts of computer science ”(p. 33), that is, the essence of computational thinking is to think like a computer scientist when it comes to solving a problem. Wing (2006) says that computational thinking is a way of thinking and "that these are useful skills for everyone, not just for computer scientists." (P. 33).

The definition of computational thinking that is considered the most appropriate, is that given by Jeannette Wing (2006), Corporate Vice President of Microsoft Research and Professor of Computer Science Department Carnegie Mellon University, who popularized the term in his Computational Thinking article, stating that “the computational thinking represents a universally applicable attitude and a set of skills that everyone, not just computer scientists, would be eager to learn and use. ”(p. 33).

A concept that comes from the computational world is the one pointed out by Raja (2014) that indicates “the computational approach is based on seeing the world as a series of puzzles, which can be broken into smaller pieces and gradually solved through of logic and deductive reasoning ”(p. 1), so, the computational approach is an intuitive way to address several of the existing methods of learning psychology.

Thus, it is possible to point to the eight dimensions that define computational thinking that are:

1. Identification of patterns. - It consists of extracting information from objects that allow to establish properties between sets of said objects.
2. Use of instructions. - Set of data inserted in a structured sequence for the processor to interpret and execute.
3. Variables. - Corresponds to the area reserved in the main memory of a computer.
4. Sequence. - It is a series or sequences of elements.
5. Operators. - They are symbols that indicate how operands should be manipulated.
6. Reuse. - Corresponds to reuse.
7. Abstraction. - It consists of isolating an element from the rest of the elements that accompany it.
8. Operation and error detection. - It consists of detecting and controlling errors.

2.3 Ubiquituous Technology

The idea of ubiquitous technologies was introduced by Weiser (1991), who described ubiquitous technologies as "environments surrounded by computers, and communication networks, in conjunction with the interaction of human beings." (P. 1-10). Pérez and Addati (2013) notes that “ubiquitous technologies give us a new vision of society, seen through the improvements that occur in the quality of life of citizens. It can be said, then, that the ubiquity of technologies is given by the availability of services, processes and information linked to them anywhere and at all times ”(p. 2), so this type of technology helps to improve the quality of life of people through the use of tools that allow managing information in an easier and more efficient way, and also that they can be used at any time and place.

The ubiquitous technology will allow the emergence of new applications where all objects are functioning integrated, which will lead to a huge commercial, economic and social opening.

3. Related Studies

Among the important educational experiences in Latin America about computational thinking is Mexico, with the National Institute of Astrophysics, Optics and Electronics, based in Puebla. It is one of the references in the work on the topic of development of online courses, texts, collection of experiences such as simulations, projects, events among others, which have been proposed as an objective to modify the reality of the country in the use of Information Technology.

In 2015, an investigation carried out by Hitschfeld, Pérez and Simmonds documents an experience in Chile carried out with school-level students, and expresses a concern when referring to the need for an educational leap in students, from consumer of technology to generator of technology, the authors point to programming as a source of knowledge that makes it possible to generate these changes, mainly if working with the little ones without waiting for them to arrive at the university (Hitschfeld, Pérez and Simmonds, 2015). The authors believe that most people can use ICT in everyday situations, but if the problem is different from the usual, their response capacity decreases; because the continuous work in usual tasks or mechanized actions does not contribute to the development of skills such as abstraction, logical organization of data, pattern recognition, algorithmization, modeling and generalization, not to mention the low importance acquired by the practice of values. In the words of, Hitschfel, Pérez and Simmonds (2015) the "confidence in the management of complexity, persistence in working with difficult problems, tolerance to ambiguity, ability to deal with open problems, and ability to communicate and work with others to achieve a common goal or solution. "(p. 31-32).

In Ecuador, educational experiences in computational thinking are scarce and isolated, mainly related to the use of Scratch, language and tool that Pérez and Roig (2015) describe for their interaction with the user as "symbolically mediated programming environments" (p 6), due to its graphical interface characteristics, object-oriented programming using pre-designed blocks and intuitive environment designed by the Massachusetts Institute of Technology (MIT) and which has now reached a high level worldwide.

The Polytechnic School of the Coast (ESPOL) and Yachay University, each on their own, have developed projects related to the use of Scratch. Both the projects developed by ESPOL and those of Yachay are fundamentally oriented towards children learning Scratch as a programming language to elaborate projects, but they do not clearly demonstrate the intention to develop computational thinking, which leads them to think that they continue to pay more attention to the teaching of the computer tool that to the possibilities considered within the computational thinking for the student, that is, they do not reach the integral proposal of the ICT in the education of which Cabero et. al (2003), which exposes the importance of observing the educational process from all its components and relationships, for the author the problem lies in:

> the lack of teacher training for their educational incorporation to grant more significance than those they have as curricular instruments, to reproduce with them traditional educational models and not intended for the communicative possibilities of ICTs, and to use their own organizational principles on them an analog and non-digital school (p. 18).

The Computer Science Department of the Faculty of Philosophy of the Central University of Ecuador, trains professionals to be teachers in this area of knowledge since 1996, in recent times it presents difficulties in the development of skills related to computational thinking with its students . Currently, the career continues using traditional tools for learning programming and algorithm design being the most applied and current, so to speak, Data Flow Diagram (DFD) and PSeint with the difficulties it presents in its use. In addition, errors of symbologies are observed, since they are oriented to a paradigm little used in programming.

4. Methodology

The research design corresponds to quasi-experimental with pre-test-post-test and intact groups, with quantitative, field approach with documentary support and explanatory level.

The population selected were the students of the First Semester of the Computer Science Degree of the semester April-September 2015, consisting of two courses, the one corresponding to the morning session with the denomination of First Semester A and the second of the evening session called First Semester B The groups were randomly selected, Parallel A was chosen as an experimental group while Parallel B became a control group, both composed of a total of 40 students.

Next, table 1 is shown where the general population total is summarized and classified by gender, establishing the number and percentage of male and female participants.

		Frecuency	Percentage	Valid Percentage	Acumulated Percentage
Valid	male	64	80.0	80.0	80.0
	female	16	20.0	20.0	100.0
	Total	80	100.0	100.0	

Table 1. Population

In relation to the ages of the participants in the experimental group, it is established as shown in Table 2 that the average is 19.9 being the most frequent age among participants 19 and 20 years.

		Frecuency	Percentage	Valid Percentage	Acumulated Percentage
Valid	18	7	17.5	17.5	17.5
	19	12	30.0	30.0	47.5
	20	12	30.0	30.0	77.5
	21	3	7.5	7.5	85.0
	22	3	7.5	7.5	92.5
	23	1	2.5	2.5	95.0
	24	1	2.5	2.5	97.5
	26	1	2.5	2.5	100.0
	Total	40	100.0	100.0	

Table 2. Ages of the experimental group population

According to Table 2, the ages of the participants of the control group average 20.28 and the most frequent age among the group is 19 years.

		Frecuency	Percentage	Valid percentage	Acumulated percentage
Valid	18	10	25.0	25.0	25.0
	19	11	27.5	27.5	52.5
	20	3	7.5	7.5	60.0
	21	5	12.5	12.5	72.5
	22	5	12.5	12.5	85.0
	23	4	10.0	10.0	95.0
	25	1	2.5	2.5	97.5
	30	1	2.5	2.5	100.0
	Total	40	100.0	100.0	

Table 3. Ages of the population of the control group

In relation to the objective, regarding the verification of the development of computational thinking between the control and experimental group, the hypothesis was established: There are differences in the development of the computational thinking skills of the First Semester students of the Computer Science Career of the Central University of Ecuador.

4.1. Instruments for information gathering

Three instruments were designed to collect information with different intentions: a satisfaction scale questionnaire about the skills of computational thinking that students consider to possess, and two questionnaires of problems related to computational thinking. These instruments served to establish the advances in the aforementioned skills, and were applied at different times in the research process.

The pretest and posttest tests were organized by the structured base test model with questions or reagents of direct questioning, each questionnaire consists of sixteen problems. Andrade (2013) mentions that the difference between objective and structured-based tests is that the second "privileges cognitive processes of a higher level than simple knowledge by memorization" (p. 4), which allows to develop knowledge and improve skills to assimilate and process information.

The first instrument was applied at the beginning of the investigation and another of similar characteristics was applied at the end, in order to compare the results with respect to the control and experimental group.

The criteria that guided the structure for the elaboration of the problems in both instruments were the eight dimensions of computational thinking that are: identification of patterns, use of instructions, variables, sequence, operators, reuse, abstraction and operation and detection of errors on the that two problems were formulated for each dimension selected. It is necessary to point out that, the level of difficulty of the problems is basic or initial, since they are first semester students very few of them know about programming, so applying an instrument of greater difficulty in the problems would not have been consistent with the degree of knowledge of the participants and their understanding, this in no case affected the scientific character and its validity.

For the statistical analyzes, the SPSS 20 tool was used both for the calculation of the measures of central tendency and the standard deviation, as well as in the verification of the hypotheses and the elaboration of statistical figures where the comparison of means by means of T tests was applied.

Once the information from the survey, the pretest and posttest questionnaire was collected, they were admitted to three different matrices, assigning each student a numerical code that identifies them and another one for gender, so that it is possible to group them later based on said variable.

In the implementation stage the following activities are carried out:

1. Temporalization of the activity in relation to the planning of the subject during the semester. Specifically, Programming I contains a unit called Problem Solving Technique, where the work with algorithms is approached, an appropriate scenario for the investigation of the computational thinking and Scratch variables, which has a duration of approximately one month and a half, time with which Research to carry out the experience with the students.
2. Design of the microcurricular planning for each session applying the constructivist methodology known as the learning cycle or ERCA.
3. Selection of activities and problems to work in each session.
4. Realization of planned activities with the intervention of the Scratch tool, took place during seven sessions that began on April 27, 2015 and ended on June 8, 2015. Each session lasted three contact hours and hours of autonomous work that students did when solving problems proposed both individually and collaboratively.

The contents and problems proposed in each session are shown in Table 4

Date	CONTENT	PROPOSED PROBLEMS
27-04-2015	- Problem solving process. - Definition of algorithms. - Characteristics - Phases of the Polya Method. - Entry and exit of information.	Three algorithms with the model of data entry, processes and outputs.
04-05-2015	- Mathematical operations. - Graphing shapes with scrolling Scratch objects.	Five problems that include graphing, movement and mathematical operations

DOI: *https://doi.org/10.29166/10.29166/catedra.v3i1.2006*

11-05-2015	- Simple conditions - Composite conditions - Nested conditions.	Four problems that with decision making, mathematical calculations and screen movements with changing scenarios and objects.
18-05-2015	- -Composite conditions - -Nested conditions.	Six problems that include decision making, mathematical calculations, simulations, changes of scenarios and objects.
25-05-2015	- -Composite conditions - -Nested conditions.	Five problems that include decision making, mathematical calculations, simulations.
01-06-2015	Repetition Loops Defined	Four problems that include repetition structures, mathematical operations and objects displacement.
08-06-2015	Defined and undefined repeat ties	Six problems with indefinite repetition ties that simulate real situations and games.

Table 4. Organization of the intervention process

5. Results

In search of similarities or differences between the control and experimental group, a comparison of the results by dimensions of computational thinking was made using measures of central tendency such as mean and standard deviation, assuming similar when there is a difference of ± 0.1. This is how Table 5 is shown, shown below:

Informe									
GROUP		IDENT PATTERNS	USE OF INSTRUCT.	VARIABLES	SEQUENCE	OPERATORS	REUSE	ERROR DETECTION	ABSTRACTION
CONTROL	Media	.43	.66	.76	.64	.60	.45	.67	.59
	N	80	80	80	80	80	80	80	80
	Desv. típ.	.497	.476	.428	.484	.493	.501	.471	.495
EXPERIMENTAL	Media	.75	.67	.79	.61	.60	.53	.55	.64
	N	80	80	80	80	80	80	80	80
	Desv. típ.	.436	.471	.412	.490	.493	.503	.501	.484
Total	Media	.59	.67	.78	.63	.60	.49	.61	.61
	N	160	160	160	160	160	160	160	160
	Desv. típ.	.494	.472	.419	.486	.491	.501	.489	.489

Table 5. Comparison of average control and experimental groups

Table 5 shows the means of each dimension of computational thinking considered in the research for the control and experimental group obtained at the end of the investigation, according to the following detail:

- In the pattern identification dimension, the mean of the control group corresponds to .43 ± .497 and of the experimental group corresponds to .75 ± .436, the results show that the experimental group reaches an average higher than that of the control group .
- In the instruction use dimension, the mean of the control group corresponds to .66 ± .476 while the average of the experimental group corresponds to .67 ± .471, so that the values allow to consider slightly higher than the experimental group.
- In the variable dimension, the mean of the control group corresponds to .76 ± .428 while in the experimental group it corresponds to .79 ± .412, values that allow it to be considered slightly higher than the experimental group.
- In the sequence dimension, the mean of the control group corresponds to .64 ± .484 while in the experimental group it corresponds to .61 ± .490, values that show a slightly higher level of the experimental group.
- In the operators dimension, the mean of the control group corresponds to .60 ± .493 while in the experimental group it corresponds to .60 ± .493, values that are similar.
- In the reuse dimension, the mean of the control group corresponds to .45 ± .501 while in the experimental group it corresponds to .53 ± .503, a higher score of the experimental group is shown.

- In the error detection dimension, the mean of the control group corresponds to .67 ± .471 while in the experimental group it corresponds to .55 ± .501, a higher score of the control group is shown.
- In the abstraction dimension, the mean of the control group corresponds to .59 ± .495 while in the experimental group it corresponds to .64 ± .484, a higher score of the experimental group is shown.

In summary, the values of the mean of each dimension of computational thinking corresponding to the experimental group are equal to or greater than the means obtained by the control group, except in the cases of the sequence and error detection dimensions, where the control group shows a higher value

Inference statistics were also used to verify results, where the aforementioned hypothesis was worked on

H1: There is a significant difference in the development of computational thinking among students who use Scratch in learning Programming in relation to those who do not use it.

The mathematical expression for the hypothesis is: H1: $\mu e \neq \mu c$

Where μe corresponds to the value of the difference of the experimental group, μc corresponds to the value of the difference of the control group.

Null hypothesis formulated corresponded to:

H0: There is no significant difference in the development of computational thinking among students who use Scratch in learning Programming in relation to those who do not use it.

The mathematical expression for the hypothesis is: H0: $\mu e = \mu c$

For the statistical check, the normality was first checked by means of the Kolmogorov and Smirnov test, relating the group variables and the difference obtained from the subtraction between the posttest mark and the pretest note, the result of which is shown in the following table:

	GROUP	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
		Estatistical	gl	Sig.	Estatistical	gl	Sig.
DIFFERENCE	EXPERIMENTAL	.154	40	.017	.948	40	.066
	CONTROL	.140	40	.045	.982	40	.757

a. Lilliefors significance correction

Table 6. Normality tests

Table 6 shows the results on normality between the control and statistical groups, obtaining a significance of .017 for the control and .045 for the experimental, which is why the hypothesis that the data comes from groups with different normality.

As a next step prior to the selection of the statistic for hypothesis testing, the test of variances between the groups was performed, the results are shown in the table below:

		Levene's test for equality of variances		T test for equality of means						
		F	Sig.	t	Gl	Sig. (bilateral)	Mean difference	Typ. error of the difference	95% Confidence interval for the difference	
									Inferior	Superior
DIFFERENCE	Equal variances have been assumed	13.855	.000	-1.030	78	.306	-1.0938	1.0618	-3.2077	1.0202
	Equal variances have not been assumed			-1.030	65.227	.307	-1.0938	1.0618	-3.2142	1.0267

Table 7. Independent sample tests

Table 7 shows the results for the Levene variance test that reaches a significance of .000, which is why the hypothesis that both groups have different variances is not ruled out.

With these results it can be established that the groups are not parametric, so, the test of hypothesis test called U of Mann Whitney was selected by the number of students, also because the data were considered independent samples, in addition, the variable organizes the numerical type and corresponded to a longitudinal study.

Based on the SPSS statistical software, the result expressed in Table 8 was obtained for the proposed work hypothesis:

Null hypothesis	Test	Sig.	Decition
The distribution of DIFFERENCE is the same among the GROUP categories	Mann-Whitney U test of independent samples	.728	Retain the null hypothesis

Table 8. Hypothesis testing

According to Table 8, the significance value of .728 is obtained, which is interpreted as not enough evidence to rule out the null hypothesis.

Therefore, it is considered that between the notes of the control group and the notes of the experimental group there is no significant difference in the development of computational thinking using Scratch in Programming learning.

6. Discussions and Conclusions

The comparison of means in the responses of the control and experimental group was presented in Table 5, based on these results, improvements are evidenced in the dimensions identification of patterns, use of instructions, variables, use of operators and error detection in the group that used Scratch in learning Programming. The group that did not use Scratch shows to maintain the same development or improve in the other four dimensions: identification of patterns, use of instructions, variables and error detection so it could be assumed that there is a difference, although not significant between the groups. However, when performing the hypothesis testing test shown in Table 8, it is established that the evidence is scarce to reject the null hypothesis. Therefore, it is assumed that there is not enough evidence to state that Scratch directly helps the development of computational thinking at all levels considered in the investigation.

In comparison to what Sáez, Miyata and Domínguez (2016) report in their research on the creation of multimedia codes through Scratch with students of higher education where they point out "From the results of the Student's t-test, it can be affirmed that there are significant improvements in the results of the administered test, so the program improves students' ability to understand the management of Multimedia Content Programming with Scratch.

”(p. 154), the same results are not observed. Probably some of the causes that originate the different results are due to the origin of the participants, in the case of the research that was mentioned the selected population corresponds to 113 students, among men and women, who voluntarily participate in the EUROMIME program at postgraduate level that strengthens the knowledge in Computer Science of professionals related to the technological branch, while the students who were part of the research at the Central University of Ecuador correspond to the initial level in the Computer Science Career where many of them have not had previous experiences in relation to the creation of programs.

There are not many investigations of the Scratch tool applied at a higher level with which to compare the results, in this regard Taborda and Medina (2012) also agree on this difficulty and express that “nonetheless, and despite the wide acceptance it has had in the community of educators, we have not found published research that has informed about the impact that the use of SCRATCH generates in the classroom. ”(p. 6). Another position regarding the formulation of problems that allow us to better observe other aspects of computational thinking to be evaluated is proposed by Nancovzka, Ternik, Koron and Koron (2017) that express “there are several ways to improve the scaffolding of cognitive development for young people Scratch programmers. One is to introduce exercises focused on parallelism and synchronization, such as asking students to explain parts of the code. Another way is to ask students to find errors in a code ”(p. 10).

As mentioned there are several contributions, but mainly at the level of basic and initial education, others in relation to interest and motivation such as the work of Pérez and Roig (2015) on programming environments not symbolically mediated where the opinion of students is collected of the initial level of the Computer Science Career, about the use of Scratch who consider that "it facilitates the work by its interface, also develops creativity and collaborative work" (p. 16). This leads us to consider that the validity and potential of the Scratch tool is not so evident in all the dimensions of computational thinking that are part of the research, but in certain aspects such as improving motivation, interest, creativity and imagination and facilitating the management of sentences by not concentrating the attention of the students in the syntax as in other programming languages.

Bibliography

Alba, R. (2008). Iniciándose en la programación con Scratch. Observatorio Tecnológico del Ministerio de Educación, Cultura y Deporte de España. Retrieved from http://recursostic.educacion.es/observatorio/web/fr/software/programacion/619-iniciandose-en-la-programacion-con-scratch?format=pdf

Andrade, X. (2013). Guía para la elaboración de pruebas de base estructurada. Quito: Universidad Central del Ecuador. Retrieved from http://es.slideshare.net/italito/guia-para-la-elaboracin-de-pruebas-de-base-estructurada

Cabero, J., Castaño, C., Cebreiro, B., Gisbert, M., Martínez, F., Morales, J: y Prendes, M. (2003). Las nuevas tecnologías en la actividad universitaria. Pixel Bit. Revista de Medios Y Educación, 20, 81–100. Retrieved from http://acdc.sav.us.es/ojs/index.php/pixelbit/issue/view/37

Hitschfeld, N., Pérez, J. y Simmonds, J. (2015). Pensamiento computacional y programación a nivel escolar en Chile: el valor de formar a los innovadores tecnológicos del futuro. Bits de Ciencia, 12, 28–33.

Isuri, S. (2010). Scratch Guía Didáctica para Profesores. Isuri Sormenezko Zerbitzuak Servicios Creativos. 4-7. Retrieved from https://bit.ly/2S3whsK

López, J. (2015). Programación con scratch cuaderno de trabajo para estudiantes. Fundación Gabriel Piedrahita Uribe. Cuarta edición. 11. Retrieved from http://eduteka.icesi.edu.co/pdfdir/AlgoritmosProgramacionCuaderno1.pdf

Martínez, F. (2001). El profesorado ante las nuevas tecnologías. In C. y T. Junta de Extremadura. Consejería de Educación (Ed.), Sociedad de la información y Educación (pp. 195–218). Mérida. Retrieved from https://bit.ly/2s1fjk5

Mittermeir, R. y Syslo, M. (2008). Informatics Education-supporting Computational Thinking. (M. Mittermeir, R y Syslo, Ed.) (1st ed.). Berlín: Springer. Retrieved from https://bit.ly/2Zawcoz

Nančovska, I., ŠerbecTernik, Z., Koron, A. y Koron, T. (2017). Learning Programming Concepts through Maze Game in Scratch. Retrieved from https://www.researchgate.net/publication/320170373_Learning_Programming_Concepts_through_Maze_Game_in_Scratch

Pascual, J. (2015). Scratch, programación sencilla y gratis para niños y mayores. Retrieved from https://computerhoy.com/noticias/software/scratch-programacion-sencilla-gratis-ninos-mayores-37925

Pérez, G. y Addati, G. (2013). Documentos de trabajo: Tecnologías Ubicuas. Editor Jorge M. Streb. Nro. 531. 1-2. Retrieved from https://ucema.edu.ar/publicaciones/download/documentos/531.pdf

Pérez, H. y Roig, R. (2015). Entornos de programación no mediados simbólicamente para el desarrollo del pensamiento computacional. Una experiencia en la formación de profesores de Informática de la Universidad Central del Ecuador. RED. Revista de Educación a Distancia, 14(46), 1–22. Retrieved from http://www.um.es/ead/red/46/rosabel.pdf

Pérez, H. (2017). Uso de Scratch como herramienta para el desarrollo del pensamiento computacional en programación I de la carrera de informática. Universidad Central del Ecuador. (23). Retrieved from https://rua.ua.es/dspace/bitstream/10045/82731/1/tesis_hamilton_omar_perez_narvaez.pdf

Raja, T. (2014). We can code it!. Retrieved from http://www.motherjones.com/media/2014/06/computer-science-programming-code-diversity-sexism-education

Sáez, J., Miyata, Y., y Domínguez, M. (2016). Creative Coding and Intercultural Projects in Higher Education: a Case Study in Three Universities / Codificación creativa y proyectos interculturales en Educación Superior: Un estudio de caso en tres universidades. RIED. Revista Iberoamericana de Educación a Distancia, 19(2), 145–165.

Taborda, H. y Medina, D. (2012). Programación de computadores y desarrollo de habilidades de pensamiento en niños escolares: fase exploratoria. Cali: Universidad ICESI.

Weiser, M. (1991). The Computer for the 21st Century. Scientific American Ubicomp Paper. Retrieved from https://www3.nd.edu/~cpoellab/teaching/cse40463/weiser.pdf

Wing, J. M. (2006). Computational Thinking. Communications of the acm /Vol. 49, No. 3. Retrieved from https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf

Zapotecalt, J. (2015). Pensamiento computacional. Curso Pensamiento Computacional. Puebla: Instituto Nacional de Astrofísica, Óptica y Electrónica de Puebla. Retrieved from http://www.pensamientocomputacional.org/index.php/home/menu-definicion

Authors

HAMILTON PEREZ- NARVÁEZ obtained his Bachelor of Science in Education specialization in Computer Science, Specialist in Virtual Environments OEI. Master in Higher Education and Doctor in Educational Research at the University of Alicante. He has written several articles related to Educational Technology and computational thinking. Professor of Technical Bachelor of Informatics, Professor of the Central University of Ecuador in the chairs of: Programming, Didactics of Informatics and Pedagogy, Professor of the Pontifical Catholic University of Ecuador in ICT courses.

He is currently director of the Pedagogy of Experimental Sciences in Informatics and coordinator of the Ecuadorian Network of technology and educational innovation. He has been tutor of thesis of graduate and postgraduate in the Faculty of Philosophy, coordinator of Master in Education mention Management of ICT-mediated learning.

ROSABEL ROIG -VILA Rosabel Roig Vila is a Doctor of Pedagogy (extraordinary prize) and Professor of the University of Alicante [UA] in the Department of General Didactics and Specific Didactics.

She is currently director of the Institute of Education Sciences (ICE) of the University of Alicante. She has been Dean of the Faculty of Education of the University of Alicante (2005-2009). It has recognized two research sections and 7 teaching trienniums. She is editor of the scientific electronic journal Journal of New Approaches in Educational Research (NAER) (http://www.naerjournal.ua.es), indexed in SCOPUS and ESCI. For years, he has directed the EDUTIC-ADEI research group of the University of Alicante (VIGROB-039) with the commitment to investigate the teaching-learning processes as a means to expand scientific knowledge and educational innovation. He has coordinated R&D research projects in Spain and the European Union. His line of research focuses on the field of Information and Communication Technologies applied to education. Since 2003 he directs the Master's Degree in the UA: "Education and Information and Communication Technologies" and participates in other interuniversity Masters (UOC, University of Huelva, University of Cádiz and University of the Balearic Islands). He has taught, by invitation, in the Dottorato di recerca in Pedagogy at the Università degli Studi Suor Orsola Benincasa. She is the coordinator of several organized training courses for university teachers. He has taught courses and presentations by invitation at the Polytechnic University of Catalonia, University of Bilbao, University of Valencia, etc. She is coordinator of the distribution lists EDUTIC and ARTICUA (RedIRIS-CSIC). She has participated in more than 30 Doctoral Thesis courts as a member, secretary and president. He has written more than 100 publications, such as: Evaluation of the pedagogical quality of MOOCs, Assessment of Web Content Accessibility Levels in Spanish Official Online Education Environments, Investigating Curriculum Change in the European Higher Education Area, Articulation Technologies Information and Communication in education, The use of ICT in the Special Education classroom: teacher perception. He has participated in various international scientific committees such as Comitato Scientifico Internazionale della Collana publisher Pedagogie e didattiche (Casa Editrice Pensa Multimedia), Advisory Board 2012 Global Education Conference (Global Education Conference Network) and E-Learning Conference Committee (IADIS-International Association for Development of the Information Society is a non-profit association).

LILIAN JARAMILLO-NARANJO obtained her Master's Degree in Technologies for Management and Teaching Practice from the Faculty of Education of the Pontifical Catholic University of Ecuador (Ecuador) in 2015. She obtained the title of Master in Education and Social Development at the Faculty of Communication Arts and Humanities of the UTE University (Ecuador) in 2008. She obtained a Bachelor's Degree in Biology from the Faculty of Philosophy, Letters and Educational Sciences of the Central University of Ecuador in 2007. She obtained the title of Doctor of Science of the Education.

She is currently an assistant professor at the Faculty of Philosophy, Letters and Education Sciences of the Central University of Ecuador in the Computer Pedagogy Career. She was a full time professor at the UTE University of Ecuador. She is a columnist and member of the International Review Board of Sophia Magazine of the Salesian Polytechnic University (Ecuador). She is also associate editor of the Chair Magazine of the Faculty of Philosophy, Letters and Education Sciences of the Central University of Ecuador. Her main research topics include education, ICT, biology, natural sciences, natural science teaching, curriculum, computer pedagogy, development of metacognitive strategies for teaching practice from ICT. She has participated in the Graduate Institute as a tutor and in international presentations organized by the Technical University of the North. She is the author of book chapters and articles published in high impact journals (Emerging Source Citation Index, Latindex, Redalcy, Scielo). She is the author of the National Reality Book edited by the UTE University 2019.