Developing Computational Thinking with a Module of Solved Problems

Yailem Arencibia Rodríguez del Rey

Universidad de Cienfuegos, Cuba

yarencibia@ucf.edu.cu

Isabel Nissandra Cawanga Cambinda

Universidad de Cienfuegos, Cuba

inissandra@ucf.edu.cu

Claudia Deco

Universidad Nacional de Rosario - Pontificia Universidad Católica Argentina, Rosario, Argentina

cdeco@uca.edu.ar

Cristina Bender

Universidad Nacional de Rosario - Pontificia Universidad Católica Argentina, Rosario, Argentina

cbender@uca.edu.ar

Raidell Avello Martínez

Universidad de Cienfuegos, Cuba

ravello@ucf.edu.cu

Klinge Orlando Villalba-Condori

Universidad Nacional de San Agustín. Arequipa, Perú

kvillalbac@unsa.edu.pe

Developing Computational Thinking with a Module of Solved Problems

The objective of this research was to develop a module of solved problems for the development of computational thinking in first-year computer engineering students at the University of Cienfuegos, Cuba.

A contextualized definition of computational thinking is proposed as a cognitive process executed by humans to solve problems using computational concepts involving the components: abstraction, analysis and data representation, problem decomposition, algorithmic thinking, recognition and generalization of patterns, simulation and evaluation. The module consists of 22 solved problems that include issues of Mathematics I, Mathematics II and Discrete Mathematics; and some search problems without heuristics that appear in simple games. The research carried out is an experimental design with pretest and posttest, with a control group and an experimental group to which the intervention was applied, both with 18 students. The students of the experimental group were selected following a simple random sampling. The statistical tests performed to contrast the initial and final results showed that there is an improvement in the final results, which demonstrates the effectiveness of the interventions carried out.

Keywords: computational thinking, CT, module of solved problems, computer engineering students

Introduction

Information and communication technologies (ICT) are advancing very fast, driven by the scientific and technical development that the world experienced in the second half of the last century, affecting almost all spheres of society. In education, as a fundamental basis for the development of any society, this almost immediate repercussion was evidenced in the gradual evolution of the Teaching Methodologies to be inserted in this process of change (Suárez Granados, Arencibia Rodríguez del Rey & Pérez Fernández, 2016). In this new era of technologies and communications, it is required that skills of different forms of thinking (critical, mathematical and algorithmic, among others), combine and give rise to a new way of reasoning: computational thinking.

The origin of this new way of thinking goes back to the 80s, when Seymour Papert proposes that to build knowledge it is not enough to work exclusively from abstract formulations. The society also must provide students with means and materials with which to experiment and create knowledge (Papert, 1980). In addition, he argues that this experiential learning should focus on problem solving and offers one of the keys, which Wing later assumes, when she states "to solve a problem, look for something similar that you have already understand" (Wing, 2006). That is to say that people go to their previous knowledge when facing a new knowledge or challenge. Wing follows her research and raises the need to train future professionals from multiple areas who, necessarily, will have to have this knowledge to develop in their professional careers, increasingly mediated by technology. So, both Papert and Wing go to the same approach: problem solving using computer skills and computer skills, to be able to model and use complex abstraction skills through intensive practice, using human creativity, so that tomorrow's professionals acquire the ability to solve, effectively and efficiently, new problems by using technology applied to any field.

It was then until 2006, that Jeannette Wing published an article entitled "Computational Thinking" in ACM (Association for Computing Machinery) in which she defended the need to start putting into practice the techniques used by computer scientists in the planning and development of their processes, in scientific areas away from computer science. And she defines it as: "Computational thinking involves solving problems, designing systems, and understanding human behavior, making use of the fundamental concepts of computer science." Thus, the essence of computational thinking would be to think like a computer scientist when one is faced with a problem (Wing, 2006).

In general terms, computational thinking consists of problem solving using basic concepts, procedures and development of programs and algorithms in computer science, and can help to develop skills such as creativity, problem solving, abstract thinking, recursion, iteration, collaborative methods, and patterns, among others. Regarding this, ISTE (2018) describes computational thinking as the ability to develop and utilize strategies to understand and find solutions to problems with the help of computational methods.

When reflecting on the approaches given by different authors on computational thinking, the need to solve problems is constant. Regardless of the scenario where the individual is, there is always something that needs to be solved, improved or invented. It is not only occurs in mathematical problems where the direct solution is an algorithm, but solutions to real-world problems that can be answered with the development of an application (software). So, in computational thinking the logical, systemic and algorithmic thinking are present (Wing, 2011).

Like the definition of computational thinking, talking about its components is difficult due to the lack of consensus of authors who have been talking about the subject such as Jeannette Wing (2006), Barr and Stephenson (2011, p.51), Royal Society (2012, p.29), Olabe et al. (Olabe, Basogain & Basogain, 2015), CSTA & ISTE (2014), Google for Education (2015) and many others. But most of them coincide in describing the components as phases or elements of computational thinking.

In summary, Computational Thinking (CT) is a problem-solving approach that emphasizes the integration of critical thinking, computer concepts and digital technologies. Although each of these authors has raised in their own way the phases, processes, elements or components of the CT, regardless of how they have named it, we can notice that most of them coincide in four aspects: the generalization of patterns and abstraction, the decomposition of the problem, the recognition of patterns, and the algorithmic design.

Regarding mathematical instruction, different computational thinking environments have been employed to study the eventual impact of programming activities on mathematical learning. Thus, for example, a Logo-like programming environment has been successfully used with kindergarten students to work problem solving strategies and different mathematical skills related to numbers and geometry (Fessakis, Gouli, & Mavroudi, 2013), or an animated programming world, ToonTalk, has proved to be effective to work deep mathematical ideas with young students (Kahn, Sendova, Sacristán, & Noss, 2011).

Mathematics is one of the most important subjects in the education system in computer science, and is studied at all levels of education, from elementary school to university. It is so, because mathematics is a necessary branch of knowledge required for students to support their learning success in the future (Runisah & Dahlan, 2017; Vidal et al., 2019). Mathematics is also one of the tools of scientific thinking that are needed to develop the ability of logical, systematic and critical thinking of students. Moreover, mathematics is required by everyone in daily activities. Therefore, students need to have a good mathematical knowledge to face the future (Prahmana & Kusumah, 2017).

The teaching of mathematics at the University of Cienfuegos remains conventional and tends to be mechanistic. It has not emphasised the development of reasoning, logic and students' thinking processes. Mathematics teaching is dominated by the introduction of verbal formulas and concepts, without sufficient attention being given to the understanding of students (Suastika, 2017). Students just listen, then imitate or copy what the teacher gives without any initiative. Students are not encouraged to develop their thinking and computational skills. Students are not allowed or invited to optimise their potential, to develop their reasoning.

Kathryn, Spaepen, Strickland & Moran (2019), suggest that mathematical and computational thinking revealed synergies and differences that can affect development of integrated mathematics and CT instruction. Discussion of these synergies and differences showed that careful consideration of the relationship between mathematics and CT has significant implications for design of spiral curricula. Effective integration must look beyond surface-level similarities between disciplines to uncover deep connections that enhance learning of both topics. Integration based on surface similarities that develop differently when described in detail could lead to misconceptions and confusion. Integration based on our identified synergies, by contrast, has the potential to lead to instruction that prepares students to engage in activities aligned with Weintrop et al.'s (2016) framework when they reach high school.

By other hand, increasing the quality of university education is an international and regional challenge in order to achieve the training of a more competent professional in today's society (Villalba-Condori, Castro, Guillén, Deco & Bender, 2018). For this reason, it is necessary to develop computational thinking in higher education, particularly to awaken the interest of young people to study computer engineering careers. In the career of Computer Engineering at the University of Cienfuegos, Cuba, it has been found that students have difficulties to organize and analyse data in a logical way in related subjects. For this reason, at the end of each school year there are difficulties in identifying, analysing and implementing possible solutions to several problems, in order to achieve the most effective combination of steps and resources. As a result of this, there are unsatisfactory results that lead to the career having students with difficulties in different subjects and that there are students who leave the engineering career.

Currently the importance of the development of computational thinking and how much it can contribute in the training of the computer engineer is not considered. The components of the CT are worked in isolation in different subjects of the career; nor is there an intentional cohesion that defines a language or a common interdisciplinary approach of the CT as a problem solving methodology. The instruments for measuring computational thinking are not adapted to the context of Cuban higher education and there are also gaps in the didactic and theoretical-methodological foundations to develop the CT through the subjects of the career. Taking into account all of the above, the need for the first-year students of this career to develop computational thinking based on solved problems is identified.

The objective of this research was to develop a "module of solved problems" for the development of computational thinking of first-year computer engineering students. The idea is defended that the application of this module, with the use of the components of computational thinking, will favor learning. The justification for this research is given by the need to have a methodology that guides the development of the solved problem module to develop computational thinking with a representation of these in the Scratch programming language.

Methodology

The research carried out is an experimental design with pre-test and post-test with a control group and an experimental group to which the intervention was applied. The Sign test and the Mann-Whitney test were performed, using the SPSS software for statistical processing.

The work was developed at the University of Cienfuegos "Carlos Rafael Rodríguez". The study population is composed of 36 students between 18 and 21 years of both sexes. To select these 36 students, a call was made through oral communication. These 36 students were placed in two groups of 18 students each following a simple random sampling. One of the groups was taken as experimental group (EG) to which the module of solved problems was applied and another of control group (CG) that was not affected by the application of the module of solved problems but continues its learning through traditional problem solving methods. With the students of the experimental group, four interventions of 90 minutes each were developed, and in which several solved problems of the proposed module were explained.

Results

In this research the following statement is then proposed: Computational Thinking is a cognitive process executed by humans for the resolution of diverse problems using computational concepts that involve the following components:

Abstraction: Consists in hiding the inherent complexity of reality to represent only its essential aspects.

Analysis and representation of data: Consists of extracting all the data from the problem, analysing them and representing them appropriately to be used by resolution methods.

Decomposition of the problem: It consists in dividing a task or problem in simpler parts so that they can be solved.

Algorithmic thinking: Consists of defining a task as a set of simple instructions step by step.

Recognition and generalization of patterns: Consists in recognizing specific situations that are repeated and generalize them.

Simulation and Evaluation: It consists of reproducing the functioning of a real world system when it evolves in time or of extending the solution of a problem in all its universe of possible values. In this research, some simulations were performed in Scratch.

Module of Solved Problems (MSP)

The module of solved problems proposed in this research consists of 22 problems that include issues of the subjects: Mathematics I, Mathematics II and Discrete Mathematics. It also presents some search problems without heuristics from problems that appear in simple games.

These problems are grouped into four subgroups in the following order:

- (1) Search problems without heuristics
- Solved Problem #1: Towers of Hanoi
- Solved Problem #2: Missionaries and Cannibals
- Solved Problem #3: The Arriero
- Solved Problem #4: Water jugs (8, 5, 3 liters)
- Solved Problem #5: Water jugs (4 and 3 liters)

Table 1 shows, as an example, the Solved Problem #1: Towers of Hanoi.

Table 1. Example: Solved Problem #1: Towers of Hanoi. (Source: own elaboration)

SOLVED PROBLEM #1: Towers of Hanoi

Problem: There are three towers (A, B, C) and N disks of different sizes. The puzzle starts with the disks in a neat stack in ascending order of size on one tower, the smallest at the top, thus making a conical shape. The objective of the puzzle is to move the entire stack to another tower, obeying some rules: only one disk can be moved at a time, each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty tower, and no larger disk may be placed on top of a smaller disk.

A possible solution will be presented followed by a representation in Scratch.

T						
Abstraction	There are N disks stacked in a tower A (origin).					
	There are two empty towers B (auxiliary) and C (destination)					
	Objective: Bring all the disks from	tower A to tower C, using tower				
	B as intermediate.	_				
Analysis and	Input:	Output:				
representation of data	Tower $A = 3$ disks	Tower $A = empty$				
	Tower $B = empty$	Tower $B = empty$				
	Tower $C = empty$	Tower $C = empty$ Tower $C = 3 disks$				
Decomposition of the	Define restrictions:					
problem	• Number of disks $N = 3$.					
-	• You have to start from the origin tower to the destination					
	tower using the intermediate tower.					
	• Only one disk can be moved at a time.					
	• A disk can never be placed on a disk with a smaller radius.					
Algorithmic thinking	Tower $A = 3$ disks, Tower $B = 0$, Tower $C = 0$					
	Step 1: Is there an empty tower?					
	If yes, make n-1 disk from the origin tower to the tower that is					





- (2) Problems of Mathematics I
- Solved Problem #6: Limit of a real function of a real variable.
- Solved Problem #7: Continuity of a real function of a real variable.
- Solved Problem #8: Derivative from a real function of a real variable.
- Solved Problem #9: Nth derivative of a real function of a real variable.
- Solved Problem #10: Optimization of a real function of a real variable.

Table 2 shows, as an example, the Solved Problem #9: Nth derivative of a real function of a real variable.

 Table 2. Example: Solved Problem #9: Nth derivative of a real function of a real

variable. (Source: own elaboration)

SOLVED PROBLEM #9 : Nth derivative of a real function of a real variable.				
Problem: Determine the nth	derivative (order n) of the following function: $f(x) = x \cdot e^{-x}$			
Abstraction	To determine the derivative of order <i>n</i> of the function $f(x)$, we must			
	necessarily determine the first derivative of the function $(f'(x))$ the			
	second derivative $(f''(x))$, the third derivative $(f'''(x))$ and other			
	successive derivatives until finding the pattern for the nth			
	derivative.			
Decomposition of the	Determine the following successive derivatives:			
problem	1) Of order 1 (First derivative): $f'(x) = e^{-x} - x \cdot e^{-x}$			
	2) Of order 2 (Second derivative): $f''(x) = -2e^{-x} + x \cdot e^{-x}$			
	3) Of order 3 (Third derivative): $f'''(x) = 3e^{-x} - x \cdot e^{-x}$			
Analysis and	Real function of a real variable: $f(x) = x \cdot e^{-x}$			
representation of data				
Algorithmic thinking	Step 1: $f'(x) = e^{-x} - x \cdot e^{-x}$			

	Step 2: $f''(x) = -2e^{-x} + x \cdot e^{-x}$ Step 3: $f'''(x) = 3e^{-x} - x \cdot e^{-x}$				
	$\begin{bmatrix} \mathbf{S} \mathbf{L} \mathbf{P} \mathbf{S} \mathbf{L} \mathbf{J} \\ \mathbf{X} \mathbf{J} \end{bmatrix} = \begin{bmatrix} \mathbf{Z} \mathbf{L} \\ \mathbf{X} \mathbf{L} \mathbf{L} \end{bmatrix}$				
Recognition and	Pattern # 1: First term				
generalization of patterns	(n=1): e^{-x}				
	(n=2): $-2e^{-x}$				
	(n=3): $3e^{-x}$				
	$\therefore (-1)^{n+1} \cdot n \cdot e^{-x}$				
	Pattern # 2: Second term				
	(n=1): $-x \cdot e^{-x}$				
	$(n=2): x \cdot e^{-x}$				
	(n=3): $-x \cdot e^{-x}$				
	$\therefore (-1)^n \cdot x \cdot e^{-x}$				
	Generalization of patterns:				
	$f^{n}(r) = (-1)^{n+1} \cdot n \cdot e^{-x} + (-1)^{n} \cdot r \cdot e^{-x}$				
Simulation / Evaluation	Derivatives of any higher order are calculated by evaluating in the				
	generalized formula:				
	$f^{8}(x) = (-1)^{9} \cdot 8 \cdot e^{-x} + (-1)^{8} \cdot x \cdot e^{-x}$				
	$n = \delta f^{\circ}(x) = (-1)^{\circ} \cdot \delta \cdot e^{-x} + (-1)^{\circ} \cdot x \cdot e^{-x}$				
	n = 9 $f^{9}(x) = (-1)^{10} \cdot 9 \cdot e^{-x} + (-1)^{9} \cdot x \cdot e^{-x}$				
	$n = 10 f^{10}(x) = (-1)^{11} \cdot 10 \cdot e^{-x} + (-1)^{10} \cdot x \cdot e^{-x}$				

- (3) Problems of Mathematics II
- Solved Problem #11: Geometric interpretation of the partial derivative.
- Solved Problem #12: Physical interpretation of the partial derivative.
- Solved Problem #13: Directional derivative of a real function of several variables.
- Solved Problem #14: Conditioned extremes of a real function of several variables.
- Solved Problem #15: Volume of a solid.
- (4) Problems of Discrete Mathematics
- Solved Problem #16: Representation of deductive structures.
- Solved Problem #17: Translation into natural language.
- Solved Problem #18: Truth Table.
- Solved Problem #19: Factorial problem.
- Solved Problem #20: Fibonacci Succession.
- Solved Problem #21: Recursive addition of the first N natural numbers.
- Solved Problem #22: Problem of the robot.

Table 3 shows, as an example, the Solved Problem #22: Robot problem (recursion).

 Table 3. Example: Solved Problem #22: Robot problem (Recursion). (Source: own
 elaboration)

SOLVED PROBLEM #22 Robot problem (Recursion)							
A robot can make steps of 1 or 2 meters. Write a recursive algorithm to calculate the number of							
ways in which the robot c	ways in which the robot can walk <i>n</i> meters.						
Abstraction	Let camina(n	<i>i</i>) be the number of ways in which the rol	oot can walk <i>n</i>				
	meters. We h	nave observed that:					
	camina(1) =	1 $camina(2) = 2$					
	Therefore, th	e algorithm calculates the function define	ed as:				
		$\left(\frac{1}{2}\right)$	n=1				
	camina(n)	= {2	n=2				
		(camina(n-1) + camina(n-2)) $n > 2)$					
Decomposition of the	To find the n	th number of ways in which a robot can	walk, we must				
problem	consider:						
	1) The base	cases that gives us the problem are:					
	camina	c(1) camina(2)					
	2) The gene	ral formula:					
	$camina(n) \rightarrow n > 2$						
Analysis and	Input: <i>n</i> : Number of steps a robot can take.						
representation of data	Output: Nu	umber of ways in which the robot can wa	lk <i>n</i> meters.				
	camina(n)						
Algorithmic thinking	Step 1: Solve	e the base cases					
	camina(1) =	1 $camina(2) = 2$					
	<i>Step 2:</i> To ca	alculate the number of ways in which a ro	bot can walk <i>n</i>				
	meters, takin	g into account that the base cases have al	ready been				
	declared, we will have:						
	$camina(n) = camina(n-1) + camina(n-2) \rightarrow n > 2$						
Recognition and	Pattern #1: $camina(n) = camina(n-1) + camina(n-2)$						
generalization of	This procedure is repeated $\forall n > 2$.						
patterns							
Simulation /							
Evaluation	Distance	Series of steps	Number of				
			ways to do it				
	1	1	1				
	2	1,1 or 2	2				
	3	1,1,1 or 1,2 or 2,1	3				
	4 1,1,1,1 or 1,1,2 or 1,2,1 or 2,1,1 or 2,2 5						

Initial Test

The initial test, consisting of three questions taken from the module of solved problem, was applied to the 36 students. These problems were selected as follows:

- A problem within the group of problems of search without heuristics: "Solved Problem #1: Towers of Hanoi".
- A problem within the group of problems of Mathematics I: "Solved Problem #9: Nth derivative of a real function of a real variable".
- A problem within the group of problems of Discrete Mathematics: "Solved Problem #22: Problem of the robot".

For the evaluation of each test question (initial and final) the following aspects were taken into account: abstraction; analysis and representation of data; decomposition of the problem; algorithmic thinking; recognition and generalization of patterns; simulation/evaluation; and total score.

To verify statistically that the two independent groups (EG and CG) come from the same population and thus demonstrate their randomness, the Mann-Whitney test was applied from the diagnostic data. When applying this nonparametric test for independent groups it turned out that the asymptotic (bilateral) significance is: 0.571 (PI #1_NT: Total score of question 1 in the Initial Test), 0.454 (PI #2_NT: Total score of question 2 in the Initial Test) and 0.819 (PI #3_NT: Total score of question 3 in the Initial Test), as it is shown in Table 4.

	PI#1_NT	PI#2_NT	PI#3_NT
Mann-Whitney U	34.500	35.000	38.000
Wilcoxon W	79.500	80.000	83.000
Z	567	748	229
Sig. A. Bil.	.571	.454	.819
Exact Sig. [2*(1-tailed Sig.)]	.605	.666	.863

Table 4. Contrast statistics (Group variable: EG, CG).

As can be seen, for the total scores of the three questions of the initial test the results were greater than the level of significance α =0.05, therefore, the hypothesis that the two samples (EG and CG) come from the same population, that is they do not have significant differences, is accepted (null hypothesis).

Final Test

The final test, consisting of three questions taken from the module of solved problems, was applied to the 36 students. These problems were selected as follows:

- A problem within the group of search problems without heuristics: "Solved Problem #5: Water jugs (4 and 3 liters respectively)".
- A problem within the group of problems of Mathematics I: "Solved Problem #8: Derivative of a real function of a real variable".
- A problem within the Discrete Mathematics group of problems: "Solved Problem #20: Factorial Problem".

The final integral results of each group were used to prove statistically that the experimental group (EG) showed better grades than the control group (CG) (see Table 5).

	PF#1_NT	PF#2_NT	PF#3_NT
Mann-Whitney U	7.500	11.500	21.500
Wilcoxon W	52.500	56.500	66.500
Z	-3.042	-2.644	-1.752
Sig. A. Bil.	.002	.008	.080
Exact Sig. [2*(1-tailed Sig.)]	.002	.008	.094

Table 5. Contrast statistics (Group variable: EG, CG)

For this the Mann-Whitney test was applied. This is a non-parametric test used to determine if there are differences between the results achieved by both samples. When applying the test it turned out that the asymptotic (bilateral) significance of the variables of the final grade in questions 1 and 2 is 0.002 and 0.008 respectively, being lower than the level of significance α =0.05. Therefore, the hypothesis (null hypothesis) that there are significant differences between the samples (EG and CG) is accepted. However, the asymptotic (bilateral) significance of the variable of the final grade in question 3 is 0.08, being greater than the level of significance α =0.05. Therefore, we accept the hypothesis (alternative hypothesis) that there are no significant differences between the samples (EG and CG).

Comparison of the results of the Initial Test and the Final Test for the

Experimental Group

The Sign test has been used as a contrast statistic and the binomial distribution for bilateral exact significance.

Result of contrast statistics for question #1

When applying this test to the experimental group (group with which the interventions are performed), it turned out that the asymptotic (bilateral) significance is 0.004, being lower than the level of significance α =0.05. Therefore, we accept the hypothesis that applying the "module of solved problem" is effective (alternative hypothesis) (Table 6).

PF#1_Ab_GEPF#1_AD_GEPF#1_DP_GEPF#1_Al_GEPF#1_P_GEPF#1_S_GEPF#1_NT_GEPI#1_Ab_GEPI#1_AD_GEPI#1_DP_GEPI#1_Al_GEPI#1_P_GEPI#1_S_GEPI#1_NT_GEExact Sig.
(bilateral).0081.000.008.008.004.008.004

Table 6. Contrast statistics for question #1

Result of contrast statistics for question #2

When applying this test to the experimental group, it turned out that the asymptotic (bilateral) significance is 0.008, being lower than the level of significance α =0.05, therefore, the hypothesis that the "module of solved problem" applied is effective is accepted (alternate hypothesis), see Table 7.

	PF#2_Ab_GE	PF#2_AD_GE	PF#2_DP_GE	PF#2_Al_GE	PF#2_P_GE	PF#2_S_GE	PF#2_NT_GE
	- PI#2_Ab_GE	- PI#2_AD_GE	- PI#2_DP_GE	- PI#2_Al_GE	- PI#2_P_GE	- PI#2_S_GE	- PI#2_NT_GE
Exact Sig. (bilateral)	.063	.016	.008	.125	.016	.453	.008

Table 7. Contrast statistics for question #2

Result of contrast statistics for question #3

When applying this test to the experimental group, it turned out that the asymptotic (bilateral) significance is 0.219, being higher than the level of significance α =0.05, therefore, the alternative hypothesis is rejected. This result shows that the initial and final results of the students in question #3 did not have significant differences, the initial results were good and they were maintained (see Table 8).

Table 8. Contrast statistics for question #3

	PF#3_Ab_GE	PF#3_AD_GE	PF#3_DP_GE	PF#3_Al_GE	PF#3_P_GE	PF#3_S_GE	PF#3_NT_GE
	-	-	-	-	-	-	-
	PI#3_Ab_GE	PI#3_AD_GE	PI#3_DP_GE	PI#3_Al_GE	PI#3_P_GE	PI#3_S_GE	PI#3_NT_GE
Exact Sig. (bilateral)	.016	.453	.219	.125	.016	.375	.219

The Sign statistical test made to contrast the initial and final results of the experimental group showed that there is an improvement in the final results, which shows the effectiveness of the interventions carried out with the "Module of solved problems".

The Mann-Whitney statistical test performed to contrast the final results of the experimental group with those of the control group showed the improvement of the results of the experimental group.

Discussion

Computational thinking has gained great attention in the field of education in recent years, especially after the launch of Code Hour in December 2013 in the EE.UU. England implemented its computer education in 2014 (García-Valcárcel and Caballero-González, 2019). In the present work as in previous research cited above, Computational Thinking is defended as a problem solving technique in which 4 phases appear: generalization of patterns and abstraction, decomposition of the problem, pattern recognition and algorithmic thinking. Although in this research the concept is defended from the perspective of the interaction between 6 components since in addition to the previous 4 are added: analysis and representation of data, and simulation/evaluation.

Mathematics instruction in computer science at University of Cienfuegos, by educational experts is known as teacher-centred learning, and there are some teachercentred learning weaknesses. This knowledge tends to make students inactive and not creative using computational thinking. The mathematics learning system must be changed following the development of mathematics learning in the world (Runisah & Dahlan, 2017), especially changing the paradigm of mathematics education creative using computational thinking.

The development of 22 problems solved from the perspective of computational thinking in several subjects such as mathematics, discrete mathematics and search problems constitutes a great strength and a practical contribution of this research because never before problems of this type has been presented under the 6 components of computational thinking. According with Schott, D. (2014) reasonable curriculum should not only supply problem-solving competencies in mathematics and engineering, but also including computational thinking.

Results obtained provide opportunities for both teachers and students. In the case of teachers for providing them with a working tool in complex subjects such as the contents proposed from the point of view of teaching and learning. In the case of students to provide them, in addition, all the advantages offered by the development of computational thinking in their comprehensive professional training.

The results suggest, as occur in Weintrop et al (2016), three main benefits for the approach of embedding computational thinking in these contexts: (1) it builds on the reciprocal relationship for learning between computational thinking and mathematics, (2) it addresses practical concerns of reaching all students, and having proficient teachers, and (3) it brings science and mathematics education more in line with current professional practices in these fields.

This paper is a step in the process of bringing computational thinking into mathematics education. Achieving this goal requires the support of a diverse set of stakeholders to be successful. This includes teachers becoming comfortable teaching the material and receiving professional development in computation-based lessons and technology and policy makers prioritizing computational thinking as a part of mathematics education; curriculum and assessment developers producing computational thinking materials targeted for mathematics classrooms; and the broader community supports the effort to bring computational thinking into these educational spaces.

The inclusion of computational thinking, as suggest our results, is a core scientific practice where the next generation science standards and similar language in mathematics' standards are important milestones. But there is still much work to do (Runisah & Dahlan, 2017) toward addressing the challenge of educating a technologically and scientifically savvy population and preparing the next generation of world scientists.

The limitations of the research are present in the simulation/evaluation component which could not be completed in many of the problems solved and therefore it was pending in this first study to obtain a computer tool that contains all the simulations. Researchers are already working on this line and soon results of simulations developed in Python language will be published.

Conclusions

A contextualized definition of computational thinking is proposed: Computational thinking is a cognitive process performed by humans to solve the problems of using computational concepts that involve, in a related way, the components of the following activities: abstraction, analysis and representation of data, decomposition of the problem, thinking algorithm, pattern recognition and generalization, simulation / evaluation. A module of solved problems was elaborated as follows: 5 problems of search without heuristics, 5 problems of Mathematics I, 5 problems of Mathematics II and 7 problems of Discrete Mathematics. The statistical tests performed (Signs and

Mann-Whitney) to contrast the initial and final results of the experimental group and the control group, showed that there is an improvement in the final results, which shows the effectiveness of the interventions carried out with the "Module of solved problems".

Future work

It is proposed to go on researching in the development of computer thinking of computer engineering students from subjects of other disciplines of the career and from other academic years (second to fifth year). Also, to complete the simulation/evaluation component in some problems of the proposed module by performing simulations in a high-level language such as Python is proposed. Based on our experience, Python is an excellent first language. It is used by many scientific disciplines, it allows us to teach modern concepts of programming, and it can be used interactively, giving students immediate feedback and giving them a convenient way to experiment with different constructs.

References

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

CSTA (Computer Science Teachers Association) / ISTE (International Society of Technology in Education) (2014). *Operational Definition of Computational Thinking for K–12 Education*. Retrieved from: http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87–97. doi:10.1016/J.COMPEDU.2012.11.016

García-Valcárcel, A., y Caballero-González, Y.A. (2019). Robotics to develop computational thinking in early Childhood Education. *Comunicar*, n. 59, v. XXVII, 63-72. DOI: https://doi.org/10.3916/C59-2019-06

Google for Education, (2015), *Exploring Computational Thinking?*. [Online course]. Retrieved from: https://edu.google.com/resources/programs/exploring-computational-thinking/.

Olabe, X. B., Basogain, J. C. O., & Basogain, M. Á. O. (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje [Computational Thinking through Programming: Learning Paradigm], *RED-Revista de Educación a Distancia*, 46(6). Papert, S. (1980). *Mindstorms. children, computers, and powerful ideas*. USA: Basic Books, Inc.

Prahmana, R. C. I., & Kusumah, Y. S. (2016). The hypothetical learning trajectory on research in mathematics education using research-based learning. *Pedagogika*, 123(3).

Rich, K., Spaepen, E., Strickland, C. & Moran, C. (2019): Synergies and differences in mathematical and computational thinking: implications for integrated instruction, *Interactive Learning Environments*, DOI: 10.1080/10494820.2019.1612445

Royal Society, (2012). *Shut down or restart: The way forward for computing in UK schools*. Retrieved from http://royalsociety.org/education/policy/computing-in-schools/report/

Runisah, H. T., & Dahlan, JA (2017). Using the 5E learning cycle with metacognitive technique to enhance students' mathematical critical thinking skills. *International Journal on Emerging Mathematics Education*, 1(1), 87-98.

Schott, D. (2014). Context-dependent statements and consequences for the mathematical education of engineering students. *World Trans. on Engng. and Technol. Educ*, 12(3), 437-442.

Suárez Granados, J. J., Arencibia Rodríguez del Rey, Y., & Pérez Fernández, A. (2016). Metodología para desarrollar un sistema tutor inteligente basado en la web, para estudiantes de ingeniería [Methodology to develop a smart web-based tutor system for engineering students]. *Revista Universidad y Sociedad*, 8(4), 108-115. Retrieved from http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-

36202016000400014&lng=es&tlng=es.

Suastika, K., Mathematics learning model of open problem solving to develop students' creativity. IEJME-Math.Educ., 12, 6, 569-577 (2017).

Vidal, M.J., Avello, R., Rodríguez, M.A., Menéndez, J.A. (2019). *Revista Cubana de Educacion Medica Superior*. 33(4):e2085.

Villalba-Condori, K., Castro, S., Guillen, E., Deco, C. & Bender, C. (2018). Approaches of Learning and Computational Thinking in Students that gets into the Computer Sciences Career, *In Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, Salamanca, Spain, October 24-26, 36-40.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147.

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

Wing, J. M. (2011). Cuaderno de investigación: el pensamiento computacional-¿Qué y por qué? [Research notebook: computational thinking-What and why?]. *El Boletín Enlace*, *6*, 1-32.