Introducing robotics and block programming in elementary education

Sáez López, José Manuel; Buceta Otero, Rogelio; De Lara García-Cervigón, Sebastián
Introducing robotics and block programming in elementary education

Available in: https://www.redalyc.org/articulo.oa?id=331464460005

Introducing robotics and block programming in elementary education

La aplicación de la robótica y programación por bloques en la enseñanza elemental

José Manuel Sáez López 1
Universidad Nacional de Educación a Distancia, UNED,
España
jmsaezlopez@edu.uned.es

Rogelio Buceta Otero 2
Universidad Nacional de Educación a Distancia, UNED,
España
bucetaotero@hotmail.com

Sebastián De Lara García-Cervigón 3
Universidad Nacional de Educación a Distancia, UNED,
España
sebasdelara@gmail.com

Abstract:

This study shows the relevance of introducing visual block programming and robotics in primary education. The study describes how robotics are effectively implemented in schools, based on computational concepts and the classroom activities. We describe, apply and present specific resources teachers, who may think of introducing programming and robotics in education must consider. These resources can be adapted to their students' levels and education stages. It is essential to be aware of the resources available and adapt them to students' needs. The analysis involves 107 fifth-grade students in primary education at three schools. The sample of the study was non-probabilistic and intentional. The study is bidimensional. The first dimension is a quasi-experimental design obtaining data from a test. Construct validity was tested by an exploratory factor analysis. The second dimension details the results for four scales previously described: active learning, computational concepts, perceived usefulness and enjoyment. This dimension examines the results of the aforementioned scale, which analyses the pedagogical interactions. Statistically significant improvements were achieved in the understanding of basic computational concepts such as sequences, loops, conditional statements, parallel execution, event handling and use of robotics. Improvements were also noted in didactic interaction, and in greater enjoyment, enthusiasm, efficiency and active participation of students. They also showed stronger motivation, commitment and interest in the process.

Keywords: computational thinking, elementary school, programming languages, teaching skills, technology applications.

Resumen:

Author notes

1   **José Manuel Sáez López.** Profesor contratado doctor en la Universidad Nacional de Educación a Distancia (UNED) España. Su trabajo científico y académico ha sido publicado en 47 revistas revisadas por pares (6 JCR y 11 Scopus). Sus líneas de investigación son la integración de la tecnología educativa, estrategias metodológicas, ludificación y programación en el aula. Está acreditado como "Profesor contratado Doctor" por la Agencia Nacional de Calidad y Acreditación de España. Ha sido reconocido como Microsoft Expert Educator 2014 y Microsoft Innovative Expert 2015. https://orcid.org/0000-0001-5938-1547

2   **Rogelio Buceta Otero.** Doctorando en la Universidad Nacional de Educación a Distancia, UNED (España). Profesor de Enseñanza Secundaria en la Comunidad Autónoma de Andalucía. Trabajando la robótica educativa y el uso del M-bot en las matemáticas, en la etapa de Educación Secundaria, los ejes de coordenadas cartesianas y las aplicaciones educativas de M Block

3   **Sebastián De Lara García-Cervigón.** Doctorando en la Universidad Nacional de Educación a Distancia, UNED (España). Trabajando la fundamentación y análisis de las aplicaciones educativas de la robótica en la enseñanza elemental. Maestro de Educación Primaria en la Comunidad Autónoma de Castilla-La Mancha.

Este estudio demuestra la importancia de un diseño para la educación que incorpore la programación visual por bloques y la robótica en Educación Primaria. Este estudio describe cómo la robótica se implementa de manera efectiva en las escuelas, basándose en conceptos computacionales y las actividades del aula. Describimos, aplicamos y presentamos varios recursos, los docentes que deseen comenzar a trabajar con la programación y la robótica en la educación deben considerar los recursos requeridos y cómo se pueden adaptar a los niveles y etapas de la educación de sus alumnos. Es fundamental conocer los recursos disponibles y adaptarlos a las necesidades de los discentes. El análisis involucra a 107 estudiantes de quinto curso de educación primaria en tres centros escolares, la muestra fue no probabilística e intencional. La primera dimensión es un diseño cuasi-experimental que obtiene datos de una prueba, la validez de constructo se probó mediante análisis factorial exploratorio. La segunda dimensión detalla los resultados para las cuatro escalas descritas anteriormente: aprendizaje activo, conceptos computacionales, utilidad percibida y disfrute, esta dimensión examina los resultados de la escala antes mencionada que analiza las interacciones pedagógicas. Se lograron mejoras estadísticamente significativas en la comprensión de conceptos computacionales básicos: secuencias, bucles, declaraciones condicionales, ejecución paralela, manejo de eventos y uso de robótica. También se observaron mejoras en la interacción didáctica, y en un mayor disfrute, entusiasmo, eficiencia y participación activa, mostrando los estudiantes una mayor motivación, compromiso e interés en el proceso.

PALABRAS CLAVE: pensamiento computacional, escuela primaria, programación y lenguajes de programación, habilidades de enseñanza, aplicaciones tecnológicas.

Concepts related to coding, programming, and Computational Thinking

Papert (1980) described constructionism as an educational method that involved "learning by doing", that took account of the student's prior knowledge and experiences, so children learn better when they work with materials that enable them to design and construct meaningful artefacts (Rogers & Portsmore, 2004). This proposition fits with the idea of working with tools such as robots; and constructionist theory posits that children actively construct their intellect and mega-cognitive skills.

Visual block programming enables experimentation with computational methods that contribute to problem solving, hence active approaches become increasingly important in the educational processes, fostering the development of logical thinking skills. Computational thinking is used in problem solving, designing systems and understanding human conduct by applying the fundamental concepts of computing (Wing, 2006).

Programming is not only a fundamental science skill and an essential tool to support the cognitive tasking involved in computational thinking, but also a demonstration of computer competence. These practices could be extremely beneficial if integrated into pedagogical activities to improve logic, Maths competence, problem solving and critical thinking.

Programming language based on blocks, also known as drag and drop programming, refers to any program language that allows the user to create programmes by handling graphic elements in programming instead of writing coding in text form. Some well-known examples include Blockly and Scratch.

Studies that have focused on computing and computational thinking in elementary education (Bers, González-González & Armas-Torres, 2019; González-González, 2019; Maya, Pearson, Tapia, Wherfel & Reese, 2015; Moreno, Robles, Román & Rodríguez, 2019; Sáez-López, Román-González & Vázquez-Cano, 2016; Sáez-López, 2019; Sengupta, Kinnebrew, Basu, Biswas & Clark, 2013; Wilson & Moffat, 2010), emphasise the positive results of computing (Baytak & Land, 2011, Kwon, Kim, Shim & Lee, 2012; Lambert & Guiffre, 2009; Lin, Yen, Yang & Chen, 2005; Relkin, de Ruiter & Bers, 2020).

Some studies (Sáez-López, Román-González & Vázquez-Cano, 2016; Sáez-López & Sevillano-García, 2017; Sáez-López, 2019; Sáez-López, Sevillano-García & Vázquez-Cano, 2019) underline the advantages of applying visual block programming in primary education, describing cases in Art classes. In these studies, the students created their own content using Scratch, and a relative increase in commitment, motivation and sense of fun was noted.

These studies highlight student improvements in understanding computational concepts by means of active approaches and project work. The implementation of visual block programming language and active methodologies in education instils enthusiasm, commitment and a sense of fun in students throughout the entire process.

Integrating Robotics

Benitti (2012) drew on empirical evidence to suggest that an effective education in robotics could complement learning in general, and that it had considerable potential at elementary level. These computational practices allow students to write programmes and test them in the form of a robot; simple activities involving movements with wheels, humanoid robots that move their arms and legs, and walk. In primary and secondary education, knowledge of computational concepts can go further, with the introduction of algorithms, variables, conditional statements, loops, parallel execution and event handling. In her meta-study that explored the potential of robotics in school, Benetti reviewed 10 articles published in 2006-2009 that suggest that education in robotics generally improves learning, although this is not always the case since other studies point to no improvement in learning via robotics.

Lindh & Holgersson (2007) stated that robotics education helps students to learn, using statistical analysis (ANOVA) to suggest that there were no obvious general effects in students' use of Lego although some significant positive effects were noted in subgroups.

Mitnik, Recabarren, Nussbaum & Soto (2009) recognized that students who did activities with robotics achieved a significant increase in their graphic interpretation skills while Spolaôr & Vavassori-Benitti (2017) presented empirical evidence to suggest the efficiency of robotics as a complementary tool to learning.

For their part Mazzoni & Benvenuti (2015) analysed two experimental conditions (boy/girl-boy/girl and boy/girl-robot) and demonstrated the efficiency of the socio-cognitive conflict to improve children's learning of English. Chen, Shen, Barth-Cohen, Jiang, Huan & Eltoukhy (2017) presented an instrument that was applied at a primary school that had adopted a new study plan to include humanoid robotics in fifth grade. The results showed that the instrument contained good psychometric properties, with the potential to present challenges and highlight improvements in students' learning and in their technological competence.

Teachers who wish to begin working with robotics in education must consider the resources required and how they can be adapted to the levels and stages of their students' education. It is essential to be aware of the resources available and adapt them to their students' needs. A categorization of potential resources appears in "Digitalized human environments" coordinated by Cózar & De Moya (2017), which contains a chapter on educational robots and block programming in proposals for using Bee bot and M bot in infant and primary education (Sáez-López & Sevillano-García, 2017).

TABLE 1
Programmes, applications and robots. Digitalized human environments (Cózar y De Moya, 2017).

Aplicación		Educación Infantil			Educación Primaria					
		3 años	4 años	5 años	1º	2º	3º	4º	5º	6º
Scratchjr	http://www.scratchjr.org	■	■	■	■	■				
Kodable	https://www.kodable.com/	■	■	■	■	■	■	■	■	■
Lightbot	https://lightbot.com/	■	■	■	■	■	■	■	■	■
Daisy the dinosaur	(//itunes.apple.com/us/app/)	■	■	■	■	■				
The foos	http://thefoos.com/hour-of-code/	■	■	■	■	■				
Code.org	https://code.org/		■	■	■	■	■	■	■	■
Scratch	https://scratch.mit.edu/						■	■	■	■
Blockly/ Blockly games	https://blockly-games.appspot.com/						■	■	■	■
Made with code	https://www.madewithcode.com/						■	■	■	■
Tickle	https://tickleapp.com/						■	■	■	■
Bee bot	https://www.bee-bot.us/	■	■	■	■	■				
Dash and dot	https://www.makewonder.com/dash	■	■	■	■	■	■	■	■	■
ozobot	http://ozobot.com						■	■	■	■
Sphero SPRK	http://www.sphero.com/sphero-sprk						■	■	■	■
M-bot	http://www.makeblock.cc/mbot/						■	■	■	■

Tabla 1: Aplicaciones y robots secuenciados en educación infantil y primaria (Sáez-López, en Cózar, 2017)

On the implementation of computational thinking in various countries, the Computhink Computational Thinking study JRC_UE INTEF (2017) described how some countries such as Austria, Hungry, Israel and Malta had a deeper tradition, greater experience and longer time spent in integrating computational thinking. Other nations want to integrate it as part of their curriculum update or have plans to do so. Germany and Spain, states that are more decentralized than most, have integrated computational thinking across the country to a greater or lesser extent.

Tools to develop programming: M-bot, Dash and Dot, Ozobot, Blockly and Scratch.

The M-bot robot has the great advantage that it can work with visual block programming (http://www.makeblock.cc/mbot/). It can function with makeblock, similar to Scratch 2.0, and is a visual and intuitive language that enables students to experiment with this resource in 3rd and 4th grade in primary education and above. It is an ideal robot for initiation in robotics, programming and electronics based on Arduino.

Motor skills. Following the example and proposals of Sáez-López & Sevillano-García, (2017), they explain the event handling and blocks necessary for giving commands to robot motors. The speed can be 50 (slow), 100 (intermediate) or 255 (fast). In the example, the speed is set at 100. Four events are established, by clicking the arrow on the up, down, left or right keys.

These four events allow the robot's movements to be controlled by the arrows on the keyboard; the robot can be connected by USB, or can function without cables with WiFi or Bluetooth. It is important to consider

the speed value. In the example, the speed is set at 100; if we change the value in all the boxes to 50, the robot would move slowly whereas if we alter the value in all the boxes for the four directions to 255, the robot would move more rapidly in all directions.

The Dash & Dot robots help teach the fundamentals of programming on iPhone, iPad and some devices that carry Android (https://www.makewonder.com/dash). The students programme the robots that use the Blockly language; these robots contain sensors and eternal devices such as microphones, speakers, lights and distance sensors. The use the Blockly visual block language means that the possibilities are considerable.

Dash is designed to attract young thinkers with visual coding and self-guided missions. The robot can respond to voice command and interact with objects, and sing and dance. It has various options and applications that we shall describe in more detail later.

Dot aims to inspire young people with its proposals and manual arts projects, with self-guided coding challenges, disguises and accessories. This robot contains multiple sensors and offers manual arts and creativity project. Students can learn robotics, coding and problem solving with Dot projects, creating and gaming throughout the entire process. Both devices have a series of apps on Android and IOS.

- Go: this app could be used as the first step to introduce users to the full potential of Dash. Dash can be handled and moved with total control of movements and turns. It can experiment with lights, sounds, sensors, movements, even with robot blinking. Go demonstrates the basic controls of Dash and Dot.
- Path (only with Dash): Path starts children off on the fundamentals of robotics and coding before they can read. By drawing a route on their tablet or smart phone, children can send Dash along particular pathways using code. These pathways are situated in cities or on a farm. Special skills practice, sounds and animation can be unveiled as the children progress in their handling of the app. Challenges can be set and obstacle courses created to test the children.
- Wonder: this is a coding language based on images created for children; in this way, they can create detailed behaviours for Dash using smart phones and tablets in order to handle their own gadgets and robots more skilfully in the real world. Wonder is visual and intuitive, and allows students to design behaviours and interactions for the Dash robot by creating sequences and intuitive instructions.
- Xylo: Dash can be programmed to interpret music with the Xylo app and the Xylophone accessory. This enables students to programme Dash to play a favourite song or create their own songs. Xylo introduces the children to programming via music and games, allowing them to compose and interpret; this way they can work with algorithms, command sequences and loops.
- Blockly: this presents advanced coding concepts through fun projects and puzzles. It allows students to work with computational concepts, exploring variables, event handling, conditional statements, loops and sequences. Blockly is a visual block programming tool that enables students to drag and drop blocks in the interface. It works with fundamental programming concepts: algorithm design, sequences, conditional statements and loops; it also works with sensors, event handling and problem solving.

Another interesting resource is Ozobot, a small robot whose main mission is to follow lines. Ozobot enables users to enter colour codes, and is an attractive option for introducing robotics programming in the classroom (http://ozobot.com). This robot provides a unique way to programme by reading colour combinations, which makes it simpler and easier to use. This device can also use Blockly in OzoBlockly (an online editor) and later synchronize its robot by placing the colour scanner on the screen. Its use does not require connection since it functions with Bluetooth or cables. Ozobot can draw letters and pathways with different directions, and it has options to encourage the student to think about the correct code to use.

Blockly and Scratch stand out as visual block programming applications. Blockly is a visual block programming app that allows the user to create programmes by logical block adjustment, which is executed

online and enables the construction of visual programming editors. It can create programmes by logical block adjustment, with each block representing a fragment of code to be executed by the computer.

It is ideal for users who are starting out in programming, as they can translate their ideas into logical statements without worrying about syntax. With Blockly, like Scratch (which we shall examine later) it is totally impossible to make syntax errors, so there are no errors involving commas or full stops in code writing when dragging the blocks; this enables the user to focus solely on the logic. The app is intuitive, and facilitates participation block connection and experimentation. This app is used on various devices and applications (App Inventor, Code.org, Dash and Dot, and Lego Mindstorms).
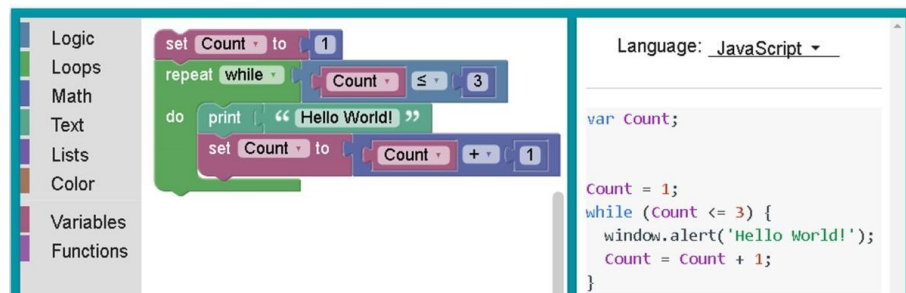

FIGURE 1
The Blockly environment

Blockly Games (https://blockly-games.appspot.com) is a series of seven free activities that allow students to learn computational thinking and acquire basic programming knowledge. The students complete a series of games based on block programming: sequences, conditional statements, loops and equations. It is an excellent introduction for first lessons in coding for 3rd and 4th grade students in primary education, and throughout that stage as the level of complexity progressively increases as they learn.

One fundamental option in block programming languages is Scratch, which is a visual block programming language created by the Lifelong Kindergarten group at the MIT Media Lab. It enables youngsters to control actions and interactions, and create their own interactive stories, games and simulations, which they can then share an online with a community of other young programmers from around the world.

The Scratch visual block programming environment (here we analyse the 2.0 version) offers considerable advantages in terms of language, which aims to provide newcomers with an introduction to computing. It enables users to handle graph blocks to compose simple programmes that allow them to create games and interactive stories.

Scratch programming has more than 100 block programmes grouped in eight categories (movement, appearance, sound, pencil, control, sensors, operators and variables). This programming suite allows young users to create their own interactive stories, games and simulations, which they can then share online with a community of other "programmers" worldwide. The students can programme and share interactive media such as stories, games and animation.

FIGURE 2
The Scratch environment. Example: https://www.youtube.com/watch?v=RVl4TL-o2YA

Scratch encourages children to learn to think creatively and collaboratively. Coding in this interface is easier to handle than traditional language programming since the children play and interact with colour blocks to create command sequences.

This proposal is based on constructivist learning ideas and the "LOGO" project (Papert, 1980). Teachers and students perceive programming to be very complicated due to the highly abstract nature of programming concepts. The creators of Scratch (Resnick et al., 2009) believe that it is possible to incorporate different types of projects within various contexts using a programming language that is fun, meaningful and social. Papert (1980) argued that programming languages should have a "low floor" (easy initiation) and a "high ceiling" (complex projects).

The aim of Scratch is to enable students to use programming concepts by means of a visual intuitive language to situate different coloured blocks and apply commands to make a product. "The Scratch programming system helps users to create intuitions about computer programming because they create projects around issues of interest to the students (Maloney, Resnick, Rusk, Silverman & Eastmong, 2010).

Brennan and Resnick (2012) describe some basic computational concepts:

- Sequences: to create a programme in Scratch, the user needs to think systematically about the order of each step.
- Looping: "Always" and "Repetition" can be used for iteration (repetition of a series of instructions).
- The conditional statement: "If" and "If-If Not" allows the user to check whether there is a condition.
- Threads (parallel execution): The start of two simultaneous actions creates two independent threads that are executed in parallel.
- Event handling: For example, actions that are executed by pressing a key and clicking on an object.
- User interface design: For example, the use of objects to create buttons.
- Keyboard input: some blocks can request the user to write and interact.

Objectives

The main objective of this study is to assess the potential of robotics in primary education with the use of devices that apply visual block programming. The specific objectives are:

- To assess the assimilation of basic computational concepts through visual programming.
- To assess the level of student motivation, commitment and participation.
- To analyse problem solving with the integration of programming and robotics in pedagogical practice.

METHOD

This study used a mix of complementary methods based on an array of data and instruments (table 2). Dimension 1 applied a quasi-experimental design, analysing Student T-test data, and evaluating the results from the test of visual blocks and robotics (TVBR). Dimension 2 details the results obtained from a four-part scale, which will be described in detail.

TABLE 2
Dimensions indicators and instruments

Dimensions	Indicators	Instruments
1.- Computational concepts and practices	Sequence Iteration (Looping)Conditional StatementsThreads (Parallel Execution) Event HandlingRobot programming	Test of visual blocks and robotics (TVBR)Descriptive analysis Student T-test
2.- Pedagogical interactions	Active learningComputational conceptsPerceived usefulnessEnjoyment	ScaleDescriptive analysis Mann-Whitney U-Test

Participants

The study sample was formed of 107 fifth-grade primary school students attending three schools in the Autonomous Community of Castilla-La Mancha in Spain. The total number of fifth-grade primary education students attending state schools of this region is 17,494. The sample was non-probabilistic and intentional. The experimental group was formed of 45.8% girls and 54.2% boys. Normality was assumed based on sample size and the Kolmogorov-Smirnov test scores. The control group consisted of 38 students, from two classes in the same three educational centres.

Instruments and reliability

The intervention, or field work, was developed during academic year 2016-17; the students in the experimental group studied a didactic unit in Sciences using programming and robotics to work through content and activities that required them to handle robots and visual block programming.

The first dimension measured the TVBR test results using a quasi-experimental method. Construct validity was tested by exploratory factor analysis using the eigenvalue > 1 extraction criterion, and the varimax rotation method. In addition, a 7.36 Cronbach value for reliability is deemed acceptable (Hair, Anderson, Tatham & Black, 1998).

The second dimension analysed the values obtained from the four-part scale.

- Scale 1: the active learning scale containing five questions, designed by Hiltz, Coppola, Rotter, and Turoff (2000).
- Scale 2: computational concepts based on a study by (Sáez-López, Román-González & Vázquez-Cano, 2016).
- Scale 3: perceived usefulness consisting of three questions, adapted from Davis, Bagozzi, and Warshaw (2002).

- Scale 4: enjoyment during the learning activities, which consists of five questions adapted from the scale created by Laros and Steenkamp (2005).

The learning processes and the work with computational concepts were analysed during the intervention. This type of research aims to describe the individual experience in specific environments. Non-parametric tests were used to measure the ordinal data. The Mann-Whitney U-test was applied in this dimension. The qualitative validity of the content provided by a panel of nine experts scored an Aiken V value ($V = S / [n (c-1)]$) of more than 0.7 for all items. Therefore, the instrument's relevance and suitability in terms of qualitative validation were acceptable. The construct validity was measured by exploratory factor analysis using the eigenvalue > 1 extraction criterion, and the varimax rotation method. The Cronbach value for reliability was 7.56, which was acceptable.

RESULTS

Dimension 1: Computational concepts and practices

Dimension 1 was based on a quasi-experimental design to compare means by statistical inference using the Student T-test. The pre- and post-test differences were analysed using a paired-samples test. The differences between the control and experimental groups were also analysed.

The Student T-test scores demonstrated that significant improvements had been achieved in the TVBR test results, thus the programme implemented had improved the students' ability to understand computational concepts. The post-test values provided data obtained after the intervention, which underlined the statistically significant differences mentioned (0.000) to a significance level of 99% in the paired samples. In short there are significant improvements regarding computational concepts, as demonstrated in table 3.

TABLE 3
Paired differences. Related samples. Student T-test

	Mean	Std. Deviation	Std. Error Mean	Upper α = 99%	Lower α = 99%	T	df	Sig
Pre-test - Post-test	-1.112	1.376	.133	-.461	-.763	-8.361	106	.000

In the comparison of means between the control and experimental groups, the Maths test mean seemed to improve considerably (7.45), while the means for the Science test remained at around 6.7, with no apparent differences (table 4).

TABLE 4
Group statistics. Control and experimental group.

		N	Mean	Std. Dev	Std. Error mean
TVBR-Post-test	Experimental group	107	7.30	1.312	.127
	Control group	38	6.63	1.076	.175

Comparing the control and experimental groups, the homogeneity of the variances is confirmed by Levene's test, and homoscedascity is assumed in the Maths test (0.267). Based on these values, the p-value in the TVBR test attained a significance of (0.00) to a level of significance of 99%. Therefore, the research hypothesis is proved and the null hypothesis rejected; there were statistically significant improvements in the application of this test, in the elements measured in this quasi-experimental design.

Dimension 2: Pedagogical interactions.

Dimension 2 details the results for the four scales previously described: active learning, computational concepts, perceived usefulness and enjoyment. Table 5 shows that the material learned and interest in that material scored well in the experimental group, but in the control group, too. Therefore, no statistically significant differences were found in these factors (1.1, 1.2). For active participation (item 1.3), there were statistically significant differences in the experimental group.

In computational concepts, there were statistically significant improvements in the experimental group in all the items (2.1, 2.2, 2.3, 2.4, 2.5, 2.6): sequences, loops, conditional statements, parallel execution, event handling and robotics. Thus, the application is considered to improve the acquisition of computational concepts considerably.

Regarding perceived usefulness, there were statistically significant improvements in the experimental group in items 3.1 and 3.2, therefore, improvements in efficiency and learning can be appreciated, yet there were no significant improvements in usefulness (item 3.3).

Finally, in terms of enthusiasm, motivation, sense of fun and comfort, there were statistically significant improvements in the experimental group in items (4.1, 4.2, 4.3, 4.4).

In all cases, there were evident statistically significant improvements in the integration of computational concepts in the unit applied. The control group showed an improvement in their understanding of sequences, loops, conditional statements, parallel execution, event handling and use of robotics, as demonstrated by the Mann Whitney U-Test, as well as in other elements related to sense of fun, enthusiasm, active participation and efficiency.

PDF generated from XML JATS4R by Redalyc
Project academic non-profit, developed under the open access initiative
104

TABLE 5
Ranges. Mann Whitney U-Test. Statistical comparison.
Group variable: control group – experimental group. p <.01, *

	Exp. Group - Control Group	N	Mean Rank	Mann-Whitney U-Test
1.1.- Materials learned	Experimental G	107	73.35	.859
	Control G	38	72.03	
1.2.- Interest in the subject	Experimental G	107	73.76	.688
	Control G	38	70.86	
1.3.- Active participation	Experimental G	107	78.99	.002*
	Control G	38	56.13	
2.1.- Sequence	Experimental G	107	86.64	.000*
	Control G	38	34.61	
2.2.- Iteration (Looping)	Experimental G	107	85.14	.000*
	Control G	38	38.80	
2.3.- Conditional Statements	Experimental G	107	86.07	.000*
	Control G	38	36.18	
2.4.- Parallel Execution	Experimental G	107	86.96	.000*
	Control G	38	33.68	
2.5.- Event Handling	Experimental G	107	86.04	.000*
	Control G	38	36.29	
2.6.- Robot programming	Experimental G	107	88.41	.000*
	Control G	38	29.62	
3.1.- Efficiency	Experimental G	107	87.01	.000*
	Control G	38	33.54	
3.2.- Learning performance	Experimental G	107	87.00	.000*
	Control G	38	33.59	
3.3.- Useful	Experimental G	107	73.87	.649
	Control G	38	70.55	
4.1- Enthusiasm	Experimental G	107	80.41	.000*
	Control G	38	52.14	
4.2.- Motivation	Experimental G	107	89.03	.000*
	Control G	38	27.87	
4.3.- Sense of Fun	Experimental G	107	89.19	.000*
	Control G	38	27.41	
4.4.- Comfort	Experimental G	107	88.63	.000*
	Control G	38	28.99	

CONCLUSIONS

This study analysed 107 fifth-grade primary school students who worked on content in Science using robotics and a visual block programming language. Using different techniques and analysis instruments, the implementation of programming and robotics was described in detail as achieving progress in

Maths, computational concepts and classroom interaction. Based on the data analysis, this investigation's conclusions are:

1. Statistically significant improvements were achieved in computational concepts in programming and robotics (dimension 1, Student T-test, tables 3 and 4).
2. Positive results observed especially in programming with robots, work with sequences, loops, conditional statements and, in particular, robotics (dimension 2, table 5)
3. Work with computational concepts when programming improved significantly as the unit was implemented. Statistically significant improvements were observed in sequences, loops, conditional statements, parallelism, event handling and robotics (dimension 2, table 5, items 2.1, 2.2, 2.3, 2.4, 2.5, 2.6).
4. In these practices, students showed enthusiasm, motivation, a sense of fun and actively participated (dimension 2, table 5, items 1.3, 4.1, 4.2, 4.3, 4.4).
5. There was a sense of perceived usefulness in the experimental group, with the observation of improvements in efficiency and learning (dimension 3, table 3, items 3.1 and 3.2).

Statistically significant improvements are evident in the integration of computational concepts in the unit applied. There is an improvement in the control group's understanding of sequences, loops, conditional statements, parallel execution, event handling and use of robotics, as the Mann Whitney U-Test results show, as well as in other elements such as a sense of fun, enthusiasm, active participation and efficiency.

The conclusions drawn from this study are that there is a clear improvement in the grasp of computational concepts at work in the media and resources used in robotics and visual block programming; this originates in approaches centred on methods to stimulate active participation, with the result that students show greater motivation, commitment, a sense of fun and interest in the process.

These conclusions are a recommendation to the appropriate authorities to implement robotics and programming in primary education contexts. Increased student motivation, a sense of fun, enthusiasm and participation resulting from this pedagogical approach underlines the relevance and suitability of the practices described in this article. The students are totally in favour of this pedagogical design, as results for perceived usefulness and active learning offered by this approach reveal.

The literature review found studies that promoted the inclusion of programming and robotics in the school curriculum, describing their benefits in terms of motivation, commitment and problem solving. On the other hand, some investigations underlined the problems in implementing robotics and programming in schools, as the application of both in Sciences showed in this study. Future research should analyse these obstacles, which could relate to attitudes, teacher training and logistical problems and the school resources available.

Several studies have found positive results in the implementation of computing and information technology in schools (Lambert & Guiffre, 2009; Lin, Yen, Yang & Chen, 2005) and discovered an increase in skills related to computational concepts (Baytak & Land, 2011; Kwon, Kim, Shim & Lee, 2012). And while Maya et al. (2015) described how teachers were initially sceptical about computing in education, in their investigation of teaching practices, in the end they found Scratch to be a valuable teaching option.

Some researchers who have focused on computing in primary schools (Maya, Pearson, Tapia, Wherfel & Reese, 2015) point to growing evidence of support for integrating computing in education in the various school stages (K-12) although some students reject computing as boring, confusing and too hard to handle (Wilson & Moffat, 2010).

Visual programming can be used to solve a range of problems as this can be done in a simple, fun and intuitive setting. Visual block programming has the advantage of being intuitive and easy to handle, and prevents the user from receiving error messages, which is encouraging for students who are new to

programming and helps them avoid the problems and difficulties that can arise when using programming languages in text (Wilson & Moffat, 2010).

The advantages of educational approaches with robotics that fit with the theoretical framework presented here are that they can be operated to function within real life contexts to resolve all types of problems and situations. This amounts to a new alphabetization in the use of technologies, block coding in this case, and the clear development of digital competence in young students.

REFERENCES

Baytak, A., & Land, S. M. (2011). An investigation of the artifacts and process of constructing computer games about environmental science in a fifth-grade classroom. *Educational Technology Research and Development, 59*, 765-782. https://doi.org/10.1007/s11423-010-9184-z

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education, 58*, 978-988. https://doi.org/10.1016/j.compedu.2011.10.006

Bers, M. U., González-González, C., & Armas-Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education, 138*, 130-145. https://doi.org/10.1016/j.compedu.2019.04.013

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*. Vancouver, BC, Canada.

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education, 109*, 162-175. https://doi.org/10.1016/j.compedu.2017.03.001

Computer Science Teachers Association (2003). http://csta.acm.org

Cózar, R., y De Moya, M. del V. (Eds.). (2017). *Entornos humanos digitalizados: experiencias TIC en escenarios educativos*. Madrid: Síntesis.

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (2002). User acceptance of computer technology: a comparison of two theoretical models. *Management Science, 35*(8), 982-1003. https://doi.org/10.1287/mnsc.35.8.982

González-González, C. S. (2019). State of the art in the teaching of computational thinking and programming in childhood education. *Education in the Knowledge Society, 20*, 1-15. https://doi.org/10.14201/eks2019_20_a17

Hair, J. F., Anderson, R. E., Tatham, R. L., & Black, W. C. (1998). *Multivariate data analysis*. (5th ed). Upper Saddle River. Prentice Hall.

Hiltz, S. R., Coppola, N., Rotter, N., & Turoff, M. (2000). Measuring the importance of collaborative learning for the effectiveness of ALN: a multi-measure, multi-method approach. *Journal of Asynchronous Learning Networks, 4*(2), 103-125. https://doi.org/10.24059/olj.v4i2.1904

INTEF (2017). *El Pensamiento Computacional en la Enseñanza Obligatoria (Computhink)*. https://intef.es/wp-content/uploads/2017/02/2017_0206_CompuThink_JRC_UE-INTEF.pdf

Kwon, D. Y., Kim, H. S., Shim, J. K., & Lee, W. G. (2012). Algorithmic bricks: a tangible robot programming tool for elementary school students. *Education, IEEE Transactions, 55*(4), 474-479. https://doi.org/10.1109/TE.2012.2190071

Lambert, L., & Guiffre, H. (2009). Computer science outreach in an elementary school. *Journal of Computing Sciences in Colleges, 24*(3), 118-124.

Laros, F. J. M., & Steenkamp, J.-B. E. M. (2005). Emotions in consumer behavior: a hierarchical approach. *Journal of Business Research, 58*(10), 1437-1445. https://doi.org/10.1016/j.jbusres.2003.09.013

Lin, J. M. C., Yen, L. Y., Yang, M. C., & Chen, C. F. (2005). Teaching computer programming in elementary schools: a pilot study. In *National educational computing conference*.

PDF generated from XML JATS4R by Redalyc
Project academic non-profit, developed under the open access initiative
107

Lindh, J., & Holgersson, T. (2007). Does Lego training stimulate pupils' ability to solve logical problems? *Computers & Education, 49*(4), 1097-1111. https://doi.org/10.1016/j.compedu.2005.12.008

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmong, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education, 10*(4), 1-15. https://doi.org/10.1145/1868358.1868363

Maya, I., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: a cross-case qualitative analysis. *Computers & Education, 82*, 263-279. https://doi.org/10.1016/j.compedu.2014.11.022

Mazzoni, E., & Benvenuti, M. (2015). A Robot-Partner for Preschool Children Learning English Using Socio-Cognitive Conflict. *Educational Technology & Society, 18*(4), 474-485.

Mitnik, R., Recabarren, M., Nussbaum, M., & Soto, A. (2009). Collaborative Robotic Instruction: A Graph Teaching Experience. *Computers & Education, 53*(2), 330-342. https://doi.org/10.1016/j.compedu.2009.02.010

Moreno, J., Robles, G., Román, M., & Rodríguez, J. D. (2019). Not the same: a text network analysis on computational thinking definitions to study its relationship with computer programming. *Revista Interuniversitaria de Investigación en Tecnología Educativa, 7*. https://doi.org/10.6018/riite.397151

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic Books.

Relkin, E., de Ruiter., L., & Bers, M. U. (2020). TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education. *Journal of Science Education and Technology, 29*, 482-498. https://doi.org/10.1007/s10956-020-09831-x

Rogers, C., & Portsmore, M. (2004). Bringing Engineering to Elementary School. *Journal of STEM Education, 5*, 17-28.

Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school. A two year case study using scratch in five schools. *Computers & Education, 97*, 129-141. https://doi.org/10.1016/j.compedu.2016.03.003

Sáez-López, J. M., & Sevillano-García, M. L. (2017). Sensors, programming and devices in art education sessions. One case in the context of primary education. *Culture and Education, 29*(2), 350-384. https://doi.org/10.1080/11356405.2017.1305075

Sáez-López, J. M. (2019). *Programación y Robótica en Educación Infantil, Primaria y Secundaria*. Editorial UNED.

Sáez-López, J. M., Sevillano-García, M. L., & Pascual-Sevillano, M. A. (2019). Aplicación del juego ubicuo con realidad aumentada en Educación Primaria. *Comunicar, 61* (XXVII), 71-82. https://doi.org/10.3916/C61-2019-06

Sáez-López, J. M., Sevillano-García, M. L., & Vázquez-Cano, E. (2019). The effect of programming on primary school students' mathematical and scientific understanding: educational use of mBot. *Educational Technology Research and Development, 67*(6), 1405-1425. https://doi.org/10.1007/s11423-019-09648-5

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: a theoretical framework. *Education and Information Technologies, 18*, 351-380. https://doi.org/10.1007/s10639-012-9240-x

Spolaôr, N., & Vavassori-Benitti, F.B. (2017). Robotics applications grounded in learning theories on tertiary education: A systematic review. *Computers & Education, 112*, 97-107. https://doi.org/10.1016/j.compedu.2017.05.001

Wilson, A., & Moffat, D. C. (2010). Evaluating Scratch to introduce younger school children to programming. In *Proceedings of the 22nd Annual Psychology of Programming Interest Group Universidad Carlos III de Madrid*, Leganes, Spain.

Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Additional information