

An analysis of the current situation of teaching programming in Primary Education

Raquel Hijón-Neira, Liliana Santacruz-Valencia, Diana Pérez-Marín, Marta Gómez-Gómez

Departamento de Informática y Estadística

Universidad Rey Juan Carlos, URJC

Móstoles, Madrid, España

{raquel.hijon, liliana.santacruz, diana.perez, marta.gomez}@urjc.es

Abstract—The interest in studying Computer Science has extended worldwide to children in the last years. In Spain, teaching is regulated with RDA (Royal Decree Act) 89/2014. According to this RDA, the subject “Technology and Digital Resources for Learning Improvement” is configured as an Autonomic Free Configuration Subject, and in the case of Autonomous Community of Madrid (ACM), the subject is compulsory. This paper presents the results of a survey sent to 318 educational centers in Madrid. The goal of the survey was to find out the current situation of programming teaching in Primary Education. Forty six schools filled and returned the survey showing that: (a) 100% of the schools consider that teaching programming in Primary Education is useful; and, (b) 39.1% of the schools are unable to teach programming because they find difficult to accommodate the timetable of the subject, and (or) they lack teachers with enough knowledge and training for the subject. In the schools where the subject is taught, most teachers rely on using Scratch or games, but they lack an adequate methodology. When the schools were asked about the possibility of implementing a methodology based on metaphors, 63% of them approved the idea, and 61% were in favor of applying it in their classrooms.

Keywords—Programming teaching; metaphor; Primary Education

I. INTRODUCTION

The interest to teach programming to children has spread worldwide. In Spain, it is legislated by RDA (Royal Decree Act) 89/2014, according to which, Programming should be taught in Primary Education in the subject “Technology and Digital Resources for Learning Improvement”, configured as an Autonomic Free Configuration Subject. It means that every Autonomous Communities (AC) decides whether the subject is compulsory or not.

In the case of the Autonomous Community of Madrid (ACM), the subject is compulsory. However, RDA 89/2014 does not specify how the programming teaching should be distributed along the six courses of Primary Education. Therefore, each center organizes the teaching according to the available material and human resources. In some cases, programming is taught transversally into other subjects such as Natural Sciences or Mathematics; some others, such as Private Schools, teach programming in complementary hours.

This paper presents the results of a survey submitted to 318 Primary Schools in Madrid. The goal of the survey was to explore the current situation of programming teaching and

didactic resources in Primary Education in the ACM. Forty six schools answered the survey, and all them (100%) stated that teaching Programming in Primary Education is useful, but 39.1% of them cannot afford teaching Programming due to either difficult fitting into the current timetables and (or) lacking teachers with enough knowledge and training for the task.

When asked about the teaching methodology of programming, schools answered that they use Scratch or games, but no teaching methodology was proposed, and this seems to be the case worldwide [2-5]. Moreover, during the literature review, we found only a few papers proposing methodologies to teach programming in Primary Education [6]. On the other hand, our literature review reveals the difficulties found when teaching even basic concepts such as program [7], loops [8], control structures and algorithms [9].

In this paper, we propose to overcome these difficulties through the use of metaphors to introduce children into these basic programming concepts, and we analyze the opinions of the surveyed schools about the application of this methodology into their classrooms.

The results gathered show that 63% of the Primary Schools that answered the survey are in favor of using the methodology based on metaphors, and 61% would like to implement it.

It is interesting to present the reasons why some Primary Schools refuse the application of the proposed methodology. Some stated that they consider that students in Primary Education are too young to understand basic programming concepts; others consider that this information is not relevant for students. A different group reasons that they do not have time, or they lack teachers with enough training or, at least, guides to follow such methodology. In our literature review, we did not find estimations of the minimum age considered optimal to begin teaching programming; as there is no age indication in the legal texts either, it seems evident that more research is needed in this direction.

The paper is organized in five sections: Section 2 reviews the state of the art; Section 3 offers some guides to teach programming in Primary Education; Section 4 presents and reviews the survey; and, finally, Section 5 presents the main

conclusions of the paper and future work lines.

II. STATE OF THE ART

A. Worldwide interest in teaching programming

Computer Science is usually compulsory in Primary Education worldwide [10]. Table I resumes the situation in several countries (blank cells indicate that information is not available), and shows the great interest in teaching Computer Science, starting in early ages.

TABLE I. WORLDWIDE INTERST [10]

Countries	Content						Primary Ed.
	Computer Science	Computer Science & IT	Digital technologies	Programming	Programming & IT	Programming & digital competence	
Australia			✓				✓
UK	✓						✓
Estonia				✓			✓
Finland				✓			✓
New Zeland					✓		
Norway				✓			
Sweden						✓	✓
South Korea	✓						✓
United States	✓						✓
Macedonia	✓						✓

B. How to teach programming

Scratch is one of the most used tools worldwide to teach programming to children [11], but there are many other approaches to teach programming and developing computational thinking. Some are plugged, when schools have enough computers available, with activities such as creating programs [3], and using Lego WeDo, or robots Mindstorms EV3 [12]. Schools interested in teaching programming, but lacking enough computers or trained teachers, use unplugged approaches [2]; in this case Computer Science is taught using stories, or free exercises in web sites such as Code.org.

However, we lack enough validation of these approaches, and there are not still statistically significant results to evaluate their impact on the children [13].

C. Use of metaphors

Metaphoric language is used every day in all contexts, and it is considered a key competence in thinking [14]. The use of metaphors to teach computer science concepts at University level has received much interest [15]. Some studies propose particular metaphors to teach abstract concepts such as dynamic memory [16], or matrixes, to handle events in JAVA [17]. However, we did not find in the specialized literature examples of using metaphors to teach basic concepts of

computer science programming in Primary Education.

III. METHODOLOGY BASED ON METAPHORS

This paper summarizes four scripts based on metaphors, as a methodology to teach basic programming concepts to children and to develop their computational thinking. A full explanation is presented in [18], where the issue of timing is also addressed.








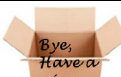
The process is divided in four steps following the normal pace of any ‘Introduction to Programming’ course: (A) The concept of program, sequence, memory and variable; (B) input and output instructions (C), conditionals, and (D) loops.

A. Script to introduce children to their first programming concepts

As an introduction, they are lectured that programs work as a cooking recipe, bringing here, for instance, the widely-known TV contest of ‘Master Chef Junior’. They are made aware that the computer has a *food storage* or *pantry* where it keeps all it needs to work (cook in our case), and that the computer keeps modifying continuously that “pantry” to adapt it to its needs according to the demands of the program. The *pantry* is actually the computer’s memory that allows storing variables, visualized in our course as *little boxes*. Thus, the variables are *boxes* that can be filled up with whatever is needed. For instance (Figure 1): “in the pantry, there is a cup filled with eggs, flour, sugar, etc., i.e. all items necessary to prepare a recipe. The same goes for the computer, which keeps (stores) all the numbers, names, or messages required by the user”.

B. Script to introduce children to Input/Output Instructions

This script combines the concepts of program, memory and screen and it explains what happens in the computer, representing it on the blackboard, or using the computer screen and memory. In Figure 2, we represent, from left to right: the program instructions, the PC screen, and the memory for each sequential step of the program. For instance, the instruction 1 (I1) produces the creation of a ‘box’ in the memory named *VariableName* empty (Figure 2). The instruction 2 (I2) sends a message to the user asking for an entry, but the state of the box remains the same, as long as the user does not interact.

Pantry		Memory	
Draw a cup		Draw a box (variable)	
Fill it up with eggs		Write ‘Hello’ inside	
Empty it and fill it up with flour		Delete it and write ‘Bye’ in it	
Do not empty it and add chocolate.		Do not delete it and add ‘have a nice day’.	

Now I have a cup with flour and chocolate		Now I have in the box Variable 'bye, have a nice day'	
---	--	---	--

Fig. 1. Examples to show the metaphor of the computer working as a pantry for the program instructions.

The instruction 3 (I3) stores whatever the user inputs (writes using the keyboard), that gets reflected on the screen; on this case the user writes her name "Mary", which is kept in the variable box. Finally, instruction 4 (I4) produces a sentence visualized on the screen as a message partly written by the programmer ("Hello") plus the content of the variable in memory (Mary), which renders the message "Hello Mary" on the screen.









Program	Screen	Memory
I1: create variableName		variableName 
I2: write on the screen ('What's your name?')		variableName 
I3: keep (variableName)		variableName 
I4: write on the screen ('Hello' variableName)		variableName 

Fig. 2. Sequential execution of the instructions (left), output on the screen (center) and state of the box keeping the data (right).

C. Script to introduce children to programming conditional instructions.

To introduce children into how the computer does/executes **conditional instructions**, they are instructed: 'if you are given the grades, if grade is equal or higher than 5, then you have passed, but if it is lower than 5, you have failed'; the computer understands this type of instructions and can make/take decisions too. On the examples in Figure 3, the first one has the box *gradeV* with the value 6, on the execution (in grey) the part of the program that is being executed; note that the PC checks if *gradeV* is greater or equal 5, in which case it executes the branch 'then' and sends the output 'Passed' to the screen (right). The rest of the code corresponds to the 'else' branch and does not execute, because the computer has already taken the decision about which branch will execute. On the last example, *gradeV* has the value 3, the code the PC is going to execute is also in grey color, the PC compares with the value saved in the box in memory *gradeV*, in this case takes the decision that the value is less than 5; therefore it executes the 'else' branch, and sends to the screen the output 'Failed'.





Memory	Program Instructions	PC Output
<i>gradeV</i> 	if <i>gradeV</i> >= 5 then write_on_the_screen ('Passed') else write_on_the_screen ('Failed')	
<i>gradeV</i> 	if <i>gradeV</i> >= 5 then write_on_the_Screen ('Passed') else write_on_the_Screen ('Failed')	

Fig. 3. PC executing conditional instructions. Left: States of the variable kept in memory. Centre (in grey): The program instructions executed by the program. Right: Output on the screen.

D. Script to introduce children to programming loops.

To introduce the **concept of loop**, we suggest the metaphor of laying the table for the number of persons living in the family. Figure 4, shows the drawings that the teacher draws on the blackboard to introduce the loop concept. Left top; the instructions to lay the table for just one person. Left down; the variable containing the number of persons in the family. Right, pictograms of the number of times "n" the instructions are repeated:






Instructions	Number of times that instructions are repeated
Set the plate, fork, knife & spoon 4 times (nPersons)	
Set the plate, fork, knife & spoon 3 times (nPersons)	

Fig. 4. Represents how children repeat several times a set of instructions: to lay the table for the "n" members of their family.

After understanding the concept of loop by means of the previous metaphor, the teacher should present, using the blackboard or on the screen, several examples of how the PC repeats a set of instructions, as demanded by the user. As explained before (Figure 3), the state of the variables in memory (left) includes a new concept: **before execution and after execution**; in figure 4, the central column presents the program instructions which are being executed, and the right side presents the output on the screen.

Memory		Program Instructions	Output on the Screen
Before	After		
execution			
variable 	variable 	While (variable <= 3) do write_on_the_screen ('Hello')	



		Add 1 to variable finMientras	
variable 1	variable 2	While (variable <= 3) do Write_on_the_screen (‘Hello’) Add 1 to variable finMientras	
variable 2	variable 3	While (variable <= 3) do write_on_the_screen (‘Hello’) Add 1 to variable endWhile	

Fig. 5. Metaphors for the loop concept

IV. STUDY OF THE STATE OF TEACHING AND DIDACTICS OF THE SUBJET ‘PROGRAMMING’

An opinion questionnaire about teaching Programming was designed and circulated in the study case, which was filled and returned by 46 teachers from Primary education in the AC of Madrid.

Concerning the professional profile of respondents, 52.2% of them were ITC coordinators, 47.8% of respondents, besides being ITC coordinators, are actually teaching the ‘Technology’ course; 34.8% of teachers acted as directors of centres (Figure 6).

Analyzing the answers to the question: ‘*What do you think about the possibility of teaching Programming to children in Primary Education?*’ it appears that there is consensus on two points: the need of adapting the contents to the age of the students, and the benefits that it provides in several areas: (i) the development of thinking skills, (ii) the organization of ideas, (iii) the ability of abstraction and solving problems, without forgetting its motivational component, and (iv) the opportunities offered by teaching through games (in fact, it is already implemented in some centers).

However, the respondents remark the importance of having properly-trained teachers to teach this subject and also they remark that it should be taught transversally, due to the difficulty of fitting its content in the schedules of other areas (such as some parts of Mathematics or Natural Sciences) through the completion of projects, or extra school activities. However this last option is usually unfeasible, mainly because the economic possibilities of families.

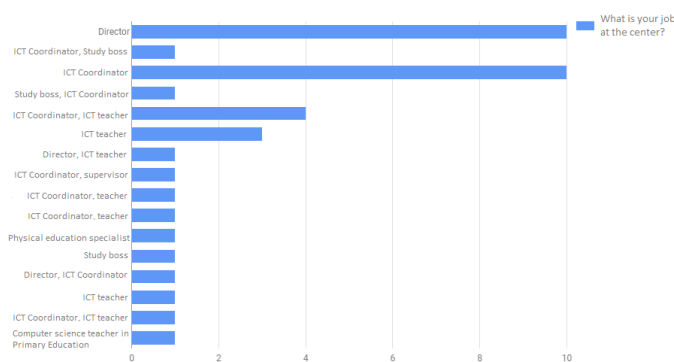


Fig. 6. Profile of respondents.

According to the answers to the question *Do you think that students are interested in learning programming?*, a 38.3% are very interested, while 25.5%, although interested, do not really know what is the use of it. On the other hand, 21.3% does not show much interest in programming, but they are in favor of the use of technology in the classroom, and 10.6% believe that students in primary education are not interested. Only 4.3% admit that they do not know. See Figure 7.

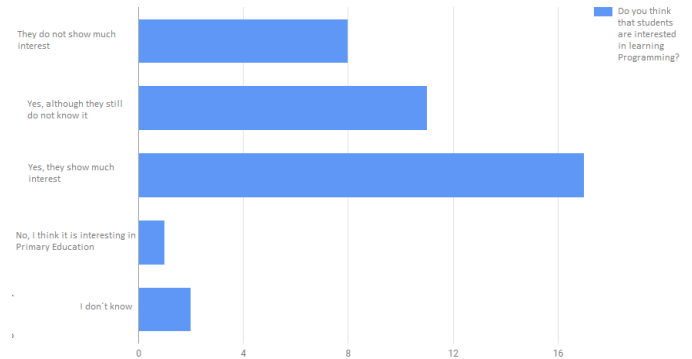


Fig. 7. Do you think that students are interested in learning Programming?

Regarding the question: ‘*In which course Programming is taught in your center?*’, 39.1% of the centers declare that Programming is not taught, while the centers that include it in the curriculum place it in different grades: 15.2% in 4th grade, 10.9% in 5th grade and 13% in 6th grade. On the other hand, 8.7% placed it in 1st, 2.2% in 2nd and 8.7% in 3rd grade of Primary Education.

From the question ‘*If you teach Programming, how do you do it?*’ it results that a 30.43% of respondents do not teach programming, but among those who do a 30.17% use tools such as Scratch [11] and Scratch Jr., and 32.89% use other tools, such as Code.org, Lego Wedo, App Inventor and BlocsCAD. An additional 6.51% of centers teach programming as extracurricular activities and / or through external entities. (Figure 8).

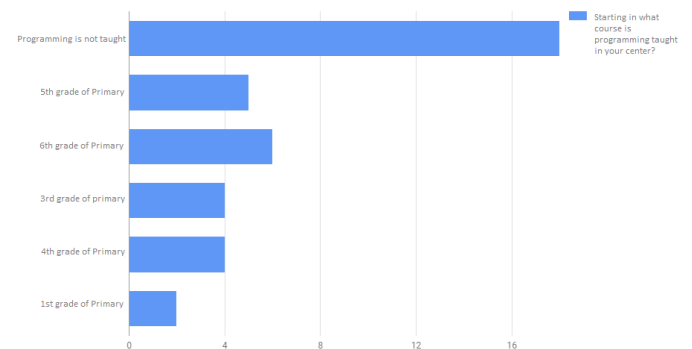


Fig. 8. If you teach Programming, how do you do it?

As for the question, ‘*Would you like to teach Programming through the use of metaphors?*’ for example, ‘Programming is like cooking, i.e., following a recipe step by step’, 63% of respondents considered it a good idea, whereas 19.6% did not like the idea. Still, 8.7% like and favor the idea of using metaphors, and a similar percentage considers that children will not understand metaphors.

With the aim of knowing the acceptance of the use of metaphors to teach some basic concepts related to the Programming, a few more questions were posed. *What do you think comparing the "computer memory" with a pantry?*, that is, in the memory you keep all the data you need in your program and the pantry is the space where you have all the items necessary for cooking, which received more than 60% of acceptance. They were therefore asked the question: *'Would you like us to provide you with scripts with metaphors like the previous ones to teach Programming in your center?'* and 47.8% of respondents answered 'yes', since it would be helpful for their teaching. Additionally, 26.1% believe that, although they do not currently teach it, our assistance will help them in preparing lectures. However, 17.4% indicated that they are not interested in metaphors because they follow their own methods. See Figure 9.

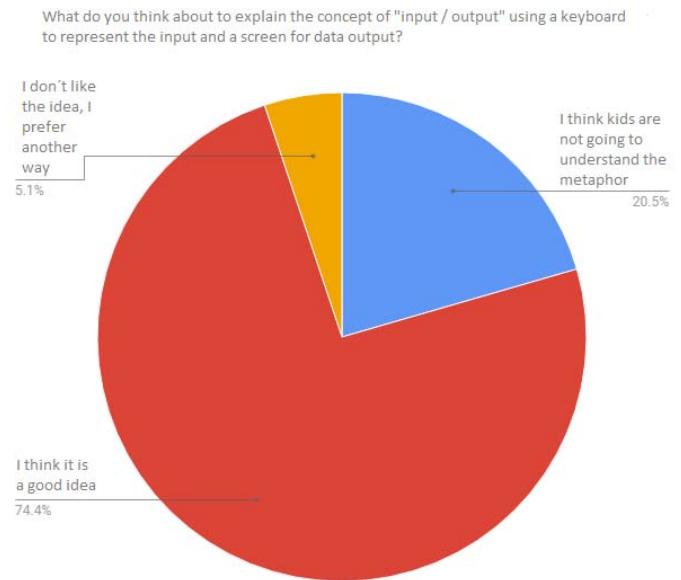
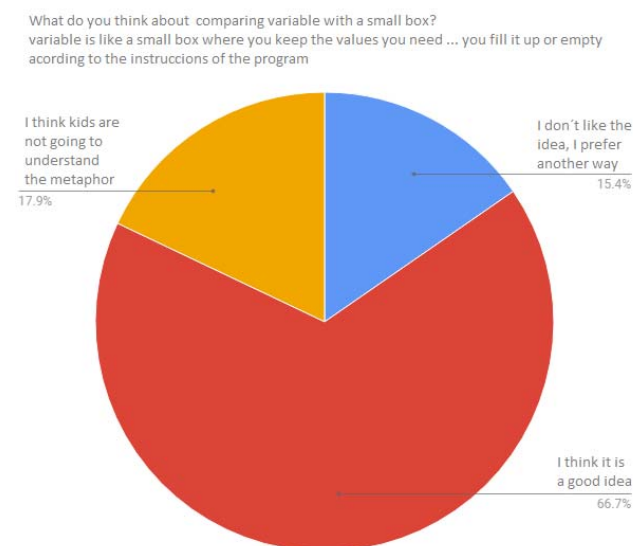
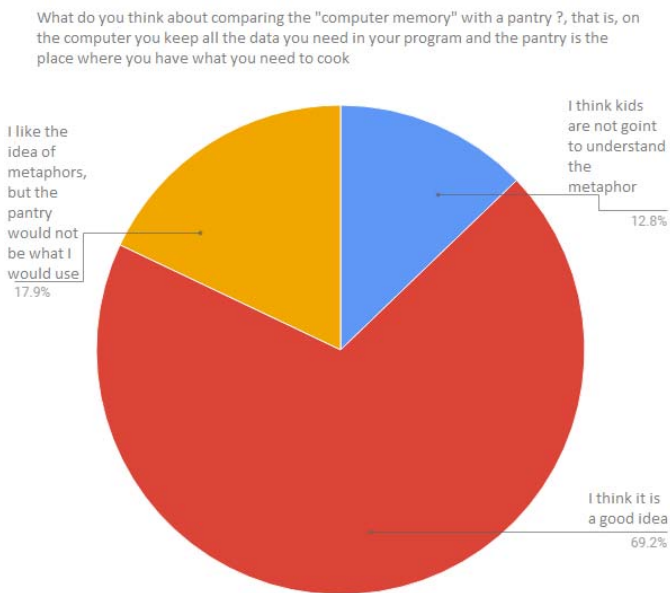


Fig. 9. Results on the acceptance of the use of metaphors

It is gratifying to know that about 61% of the centers would be willing to receive help from us, in implementing the scripts. However, we are aware that some aspects of the proposal need refinement. This is, for instance, the case of using certain metaphors, such as the Three Kings, taking into account the cultural and / or economic diversity of the audience. We also detect a need of rethinking the offer of metaphors, looking for more flexible examples, i.e., more adaptable to a wider specter of interests.

V. CONCLUSIONS

Our research is based on the premise that the interest for teaching Programming in Primary Education is increasing worldwide. In the case of the Autonomous Community of Madrid, it is mandatory according to RDA 89/2014. However, many schools face serious teaching difficulties derived from the lack of adequate resources or properly-trained teacher. In this article, we present the results of an investigation based on a questionnaire sent to 318 Primary Schools, which received 46 responses.

The staffs of these schools consider useful and appropriate to teach Programming in Primary Education. However, 39.1% of the centers cannot offer the subject of Programming because it is difficult for them to fit it into the center's schedule, or finding properly- trained teachers. Referring to centers with effective Programming offer, most of them use Scratch or games, but without a precise, well worked out, methodology. When they are offered such a methodology by us, 63% of the centers agree and 61% of them would like to put it into practice.

It seems clear that this research topic is very interesting at present, since there are few studies or they do not offer conclusive results either. On the other hand, although the sample of the study is relatively small, as compared with the total number of centers of the Autonomous Community of Madrid, it reveals a growing need for an adequate, well

planned methodology for teaching Programming in Primary Education.

As future work we will continue our study, incorporating more centers, and addressing key questions such as the age at which Programming should be incorporated to the curriculum, how many hours a week are the most suitable, and which should be the impact of the promotion of computational thinking and other skills such as problem solving and creativity.

ACKNOWLEDGEMENTS

This work has been financed with the research projects TIN2015-66731-C2-1-R of the Ministerio de Economía y competitividad and S2013 / ICE-2715 of the Comunidad Autónoma de Madrid.

REFERENCES

- [1] RD 126/2014. Curriculum de Educación Primaria. Available on-line en <https://goo.gl/22goCf> (last visit: 30 marzo, 2017).
- [2] C. Brackmann, D. Barone, A. Casali, R. Boucinha, S. Muñoz-Hernandez. "Computational thinking: Panorama of the Americas", In: International Symposium on Computers in Education (SIIE), 2016, pp. 1-6.
- [3] S. Campe, J. Denner. Programming Games for Learning: A Research Synthesis. Paper presented at the annual meeting of the American Educational Research Association, Chicago, 2015.
- [4] I. Ouahbi, F. Kaddari, H. Darhmaoui, A. Elachqar, S. Lahmine, "Learning basic programming concepts by creating games with scratch programming environment", *Procedia-Social and Behavioral Sciences*, 191, 2015, pp. 1479-1482.
- [5] M. Jovanov, E. Stankov, M. Mihova, S. Ristov, M. Gusev, "Computing as a new compulsory subject in the Macedonian primary schools curriculum", In: Global Engineering Education Conference (EDUCON), 2016, IEEE, pp. 680-685.
- [6] Eliminado por revision anónima
- [7] E. Lahtinen, K. Ala-Mutka, H.M. Järvinen. "A study of the difficulties of novice programmers". In: *ACM SIGCSE Bulletin*, vol. 37, no. 3, 2005, pp. 14-18.
- [8] D. Ginat. "On novice loop boundaries and range conceptions", *Computer Science Education*, 14(3), 2004, pp. 165-181.
- [9] O. Seppälä, L. Malmi, A. Korhonen. "Observations on student misconceptions—A case study of the Build-Heap Algorithm", *Computer Science Education*, 16(3), 2006, pp. 241-255.
- [10] F. Heintz, L. Mannila, T. Färnqvist. "A review of models for introducing computational thinking, computer science and computing in K-12 education", *IEEE Frontiers in Education Conference*, 2016, pp. 1-9.
- [11] M. Resnick, J. Maloney, A. Monroy-Hernandez, N. Rusk, E. Eastmond, K. Brennan, et al. *Scratch: Programming for all*. *Communications of the ACM*, 52(11), 2009, pp. 60-67.
- [12] A. Sović, T. Jaguš, D. Seršić. "How to teach basic university-level programming concepts to first graders?", In: *Integrated STEM Education Conference (ISEC)*, 2014, IEEE, pp. 1-6.
- [13] F. Kalelioğlu, "A new way of teaching programming skills to children: Code. Org", *Computers in Human Behavior*, 52, 2015, pp. 200-210.
- [14] G. Lakoff, M. Johnson. "Metaphors we live by". University of Chicago press, 2008.
- [15] J.P. Sanford, A. Tietz, S. Farooq, S. Guyer, R.B. Shapiro, "Metaphors we teach by", In: *Proceedings of the 45th ACM technical symposium on Computer science education*, 2014, pp. 585-590.
- [16] R. Jiménez-Peris, C. Pareja-Flores, M. Patiño-Martínez, J.A. Velázquez-Iturbide. "The locker metaphor to teach dynamic memory", In: *ACM SIGCSE Bulletin*, Vol. 29, No. 1, 1997, pp. 169-173.
- [17] W.W. Milner. "A broken metaphor in Java", *ACM SIGCSE Bulletin*, 41(4), 2010, pp. 76-77.
- [18] D. Pérez-Marín, R. Hijón-Neira, M. Martín-Lope, "Propuesta de Metodología basada en Metáforas para la Enseñanza de la Programación a Niños". *IEEE RITA* (aceptado para publicación). (in press)

Un análisis de la situación sobre el estado de la enseñanza de la Programación en Primaria y su didáctica

Raquel Hijón-Neira, Liliana Santacruz-Valencia, Diana Pérez-Marín, Marta Gómez-Gómez

Departamento de Informática y Estadística

Universidad Rey Juan Carlos, URJC

Móstoles, Madrid, España

{raquel.hijon, liliana.santacruz, diana.perez, marta.gomez}@urjc.es

Abstract—En los últimos años el interés por el aprendizaje de la programación ha crecido a nivel mundial y en España está legislado en el RD 126/2014. Según este decreto, la asignatura “Tecnología y Recursos Digitales para la mejora del aprendizaje” está configurada como de Libre Configuración Autonómica, y en el caso de la Comunidad de Madrid, se ha fijado como obligatoria. En este artículo se presentan los resultados de un cuestionario realizado a diversos centros educativos, con el fin de indagar sobre el estado de la enseñanza de la Programación en Primaria y su didáctica. Aunque el cuestionario fue enviado a 318 colegios de la Comunidad de Madrid, se recibieron 46 respuestas a través de las cuales se pudo constatar que: (i) a todos los centros les parece útil y adecuado enseñar Programación en dicha etapa y que (ii) un 39,1% de los centros no pueden impartir Programación porque encuentran dificultades en cómo encajarlo en el horario y en encontrar profesores adecuadamente formados. De otra, en aquellos centros en los que se imparte Programación la metodología utilizada suele ser en gran medida soportada en el uso de Scratch o en el uso de juegos, pero con una metodología sin definir. También se les preguntó acerca del uso de otra metodología, como la relacionada con el uso de metáforas, a lo que el 63% de los centros ha respondido que les parece bien y el 61% quisiera ponerla en práctica.

Keywords—enseñanza de la programación; metáfora; Educación Primaria

I. INTRODUCCIÓN

El interés por enseñar Programación se ha extendido a las primeras etapas educativas y está presente a nivel mundial. En España, está legislado en el RD 89/2014 [1], según el cual Programación se enseñaría en Educación Primaria en la asignatura “Tecnología y Recursos Digitales para la mejora del aprendizaje”, determinada como de Libre Configuración Autonómica. Por lo tanto, cada Comunidad Autónoma puede decidir su obligatoriedad.

En el caso de la Comunidad de Madrid, la asignatura se ha fijado como obligatoria. Sin embargo, el RD 89/2014 no especifica cómo distribuir las enseñanzas recogidas en los seis

cursos de Primaria. Por lo tanto, cada centro educativo lo está implantando en función de los recursos materiales y personales que tienen, y en algunos casos de forma transversal en asignaturas como Ciencias Naturales o Matemáticas, o incluso en el caso de colegios concertados o privados en horas complementarias.

En este artículo se presentan los resultados de un cuestionario enviado a 318 colegios de la Comunidad de Madrid con el fin de indagar sobre el estado de la enseñanza de la Programación en Primaria y su didáctica. Se han recibido 46 respuestas de cuyo análisis se desprende que a todos los centros les parece útil y adecuado enseñar Programación en Primaria. Sin embargo, un 39,1% de los centros no pueden impartir programación porque encuentran dificultades para encajarlo en el horario lectivo y en encontrar profesores formados adecuadamente. Entre aquellos que la imparten, al preguntarles sobre la metodología utilizada, la respuesta ha sido Scratch o juegos, pero sin una metodología concreta, lo que es algo representativo de la situación mundial [2-5]. De hecho, durante el proceso de revisión literaria se han encontrado pocos artículos que proporcionen una metodología específica de enseñanza de la Programación en estos niveles [6]. Por el contrario, sí se encuentran dificultades registradas al enseñar conceptos básicos como programa [7], bucles [8], estructuras de control y algoritmos [9].

En este artículo, se propone el uso de metáforas para introducir a los niños en los conceptos básicos de programación y se estudia cuál es la opinión de los colegios encuestados respecto a esta forma de enseñanza y su opinión sobre su aplicación en las aulas. Las respuestas recogidas en el cuestionario indican que al 63% de los centros le parece bien y el 61% quisiera ponerla en práctica. Los motivos por los que algunos centros que contestan que no les parece bien suele ser porque temen que los niños en edades tan tempranas no comprendan estos conceptos o no les resulte necesario, ya que aún no existen investigaciones que indiquen a partir de qué

edad se deberían empezar a enseñar, y el RD no lo contempla; y aún no lo han podido poner en práctica por falta de tiempo, guías o/y profesores.

El artículo se organiza de la siguiente forma: la Sección 2 contiene el estado del arte; la Sección 3 proporciona las guías para la enseñanza en Educación Primaria de programación; la Sección 4 recoge el análisis del cuestionario; y finalmente, la Sección 5 recopila las principales conclusiones y líneas de trabajo futuro.

II. ESTADO DEL ARTE

A. Interés mundial en la enseñanza de la programación

La asignatura de Informática suele ser obligatoria en Educación Primaria a nivel mundial [10]. La Tabla I recoge la situación particular de cada país, con celdas en blanco indicando la falta de información en esos casos. No obstante, la información recogida pone de relieve el interés mundial en enseñar Informática desde edades tempranas.

TABLA I. INTERÉS A NIVEL MUNDIAL [10]

Países	Contenido						Ed. Primaria
	Informática	Informática y TIC	Tecnologías digitales	Programación	Programación y TIC	Programación y comandada	
Australia			✓				✓
Reino Unido	✓						✓
Estonia				✓			✓
Finlandia				✓			✓
Nueva Zelanda					✓		
Noruega				✓			
Suecia						✓	✓
Corea del Sur	✓						✓
Finlandia	✓						✓
Estados Unidos	✓						✓
Macedonia		✓					✓

B. Enfoques para enseñar programación

Una de las herramientas más utilizadas a nivel mundial para la enseñanza de Informática a niños es el entorno de programación visual **Scratch** [11]. Otros enfoques para la enseñanza de Programación y el desarrollo de su pensamiento computacional incluyen crear sus propios programas [3], usar Lego WeDo o robots Mindstorms EV3 [12], y enfoques sin tecnología, para países en los que, aunque consideran que la enseñanza de la Programación es importante, no tienen suficientes ordenadores, o profesores con conocimientos en la materia [2]. En estos enfoques, se enseña Informática mediante el uso de cuentos, o ejercicios gratuitos

proporcionados por sitios web como Code.org. Sin embargo, estos enfoques aún no han sido suficientemente evaluados para poder tener resultados significativos de su impacto en el rendimiento del alumno [13].

C. Uso de metáforas para enseñar programación

El lenguaje metafórico se usa a diario en todo tipo de contextos, y se considera una competencia fundamental de la forma de pensar [14]. El uso de metáforas para enseñar conceptos informáticos a nivel universitario también ha sido objeto de interés [15]. Hay estudios que proponen metáforas concretas para enseñar conceptos abstractos como *memoria dinámica* [16], o matrices para el manejo de eventos en JAVA [17]. Sin embargo, no se encuentra en la literatura ejemplos del uso de metáforas para enseñar conceptos informáticos en la etapa de Educación Primaria.

III. METODOLOGÍA BASADA EN METÁFORAS

En este artículo se resumen los cuatro guiones basados en el uso de metáforas, como una metodología para enseñar conceptos básicos de programación a niños y desarrollar su pensamiento computacional, el desarrollo completo está en [18], donde se detalla también su temporalización.

El proceso se ha dividido en cuatro pasos sucesivos que reflejan el curso normal de introducción a la programación: (A) Concepto de programa, programar, secuencia, memoria y variable; (B) instrucciones de entrada y salida (C), Condicionales, y finalmente (D) bucles.

A. Guión para introducir a los niños a los primeros conceptos de programación

En primer lugar se les dice a los niños que los programas funcionan como las recetas de cocina, recurriendo a programas de televisión actuales y conocidos por los niños como es el caso de Master Chef Junior, por ejemplo. Se les dice que el ordenador tiene un *almacén de comida* o *despensa* donde almacena todo lo que necesita, y lo va modificando para adaptarlo a sus necesidades. Esa *despensa* es la memoria del ordenador que permite almacenar variables como *cajitas*. Las variables son *cajitas* que pueden rellenar con lo que se necesita. Por ejemplo (Figura 1): “en la despensa hay una taza donde se ponen huevos, harina, azúcar, etc., es decir, todo lo que se necesita para seguir una receta; lo mismo sucede en el ordenador donde se guardan números, nombres, o mensajes que se necesita mostrar al usuario”.

B. Guión para introducir a los niños a las instrucciones de Entrada/Salida

En este guion se combinan los conceptos: programa, memoria y pantalla y se explica qué pasa en el ordenador representándolo en la pizarra o la pantalla del ordenador y la memoria (ver Figura 2), de izquierda a derecha: las instrucciones del programa, la pantalla del PC y la memoria en cada paso secuencial del programa. Por ejemplo, como se puede ver en la Figura 2, la instrucción 1 (I1) produce la creación de una “caja” en la memoria llamada

NombreVariable vacía. La instrucción 2 (I2) envía un mensaje al usuario pidiendo una entrada, el estado de la cajita variable permanece igual.





Despensa		Memoria	
Dibujo una taza		Dibujo una caja (variable)	
La lleno de huevos		Escribo "Hello" dentro de ella	
La vacío y la relleno de harina		La borro y escribo "Bye" en ella	
No la vacío y le añado chocolate. Ahora tengo una taza con harina y chocolate		No la borro y le añado "have a nice day". Ahora tengo en la cajitaVariable "bye, have a nice day"	

Fig. 1. Ejemplos para ilustrar la metáfora del Ordenador funcionando como una despensa para las instrucciones del programa

La instrucción 3 (I3) almacena lo que el usuario escribe usando el teclado que se refleja en la pantalla, es este caso su nombre "Mary" se almacena en la variable. Finalmente, la instrucción 4 (I4) produce una frase escrita en la pantalla con un mensaje escrito por el programador "Hello" y el contenido de la variable en memoria (Mary), que produce el mensaje "Hello Mary" en la pantalla.







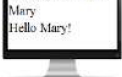

Programa	Pantalla	Memoria
I1: crea nombreVariable		nombreVariable 
I2: escribe_en_pantalla ("¿cuál es tu nombre?")		nombreVariable 
I3: guarda (nombreVariable)		nombreVariable 
I4: escribe_en_pantalla ("Hello" nombreVariable)		VariableName 

Fig. 2. Ejecución secuencial de las instrucciones (izda.), salida en la pantalla (centro) y estado de la cajita en memoria almacenando el dato (dcha.)

C. Guión para introducir a los niños a la programación de instrucciones condicionales.

Para introducir a los niños en cómo el ordenador resuelve condicionales se les dice: "si te dan las notas, si la nota es

mayor que 5, has aprobado y si es menor de 5, has suspendido"; el ordenador entiende ese tipo de instrucciones y puede tomar decisiones también. En los ejemplos que se muestran en la Figura 3, la primera tiene la cajita NotaV con el valor 6, en la ejecución (en gris) la parte del programa que se está ejecutando, se puede ver que el PC comprueba si NotaV es mayor o igual a 5, en ese caso ejecuta la rama "then" y la salida en la pantalla (dcha.) "Passed"; el resto del código correspondiente a la rama "else" no se ejecuta, porque el ordenador ha tomado ya la decisión de qué rama ejecutar. En el siguiente ejemplo, NotaV tiene el valor 3, el código que el PC va a ejecutar está también en gris, el PC hace la comprobación con el valor guardado en cajita de memoria NotaV, en este caso decide que el valor es menor por lo que ejecuta la rama del "else", la salida por pantalla es "Failed".





Memoria	Instrucciones del Programa	Salida PC
NotaV 	if NotaV >= 5 then escribe_en_pantalla ("Passed") else escribe_en_pantalla ("Failed")	
NotaV 	if NotaV >= 5 then escribe_en_pantalla ("Passed") else escribe_en_pantalla ("Failed")	

Fig. 3. PC ejecutando condicionales. El estado de las variables guardado en memoria (izda.). Las instrucciones que se ejecutan del programa (centro en gris). Salida por pantalla (derecha).

D. Guión para introducir a los niños a la programación de Bucles.

Para introducir el concepto de bucle se utiliza la metáfora de poner la mesa para el número de personas de la familia. La Figura 4, muestra los dibujos que el profesor tendrá que hacer en la pizarra para introducir el concepto de bucle, (izquierda arriba) las instrucciones para poner la mesa para una persona (izquierda abajo) variable contenedora del número de personas de la familia; (derecha) pictogramas de las veces que se repiten las instrucciones:



Instrucciones	Veces que repites las instrucciones
pon el plato, tenedor, cuchillo y cuchara 4 veces (nPersonas)	
Pon el plato, tenedor, cuchillo y cuchara 3 veces (nPersonas)	

Fig. 4. Representa cómo el niño repite un conjunto de instrucciones varias veces para poner la mesa para los miembros de su familia

Después de comprender el concepto de bucle con la metáfora anterior, el profesor deberá mostrar ejemplos de

cómo el PC repite un conjunto de instrucciones en la pizarra o la pantalla, como se indicó anteriormente (Figura 3), el estado de las variables en memoria (izquierda) ahora se introduce el concepto de **antes de la ejecución y después de la ejecución**; las instrucciones del programa que se están ejecutando (centro) y la salida en la pantalla (derecha).

Memoria		Instrucciones de Programa	Salida Pantalla
Antes	Después		
ejecución			
variable 0	variable 1	Mientras (variable <= 3) hacer Escribe pantalla ("Hello") Suma 1 a variable finMientras	Hello
variable 1	variable 2	Mientras (variable <= 3) hacer Escribe pantalla ("Hello") suma 1 a variable finMientras	Hello Hello
variable 2	variable 3	While (variable <= 3) do write_on_the_screen ("Hello") Add 1 to variable endWhile	Hello Hello Hello

Fig. 5. Metáforas para el concepto de bucle

IV. ESTUDIO DEL ESTADO DE LA ENSEÑANZA DE LA PROGRAMACIÓN Y SU DIDÁCTICA

Para el estudio de caso se diseñó un cuestionario de opinión sobre la enseñanza de Programación en Educación Primaria¹, al cual respondieron 46 profesores de diferentes centros educativos de la Comunidad de Madrid.

En el perfil de los encuestados un 52,2% corresponde a Coordinadores TIC, seguido de un 47,8% de profesores que imparten la asignatura de Tecnología además de ser Coordinadores TIC y un 34,8% de profesores que se desempeñan como Directores. Ver Figura 6.

De las respuestas a la pregunta *¿Qué opinas sobre la posibilidad de enseñar Programación a niños de Educación Primaria?* Se extrae que hay consenso en la necesidad de adaptar los contenidos a la edad de los alumnos y también en que su enseñanza favorece: (i) el desarrollo de destrezas del pensamiento, (ii) la organización de las ideas, (iii) la capacidad de abstracción y resolución de problemas, sin olvidar su componente motivacional y (iv) las posibilidades de enseñanza a través del juego, en algunos centros ya se imparte.

No obstante, señalan que es importante contar con profesores formados para impartir dicho contenido y que se debería enseñar de forma transversal, debido a la dificultad de encajarla

¹ <https://goo.gl/forms/qLBrhaOqXkgTZ7du1>

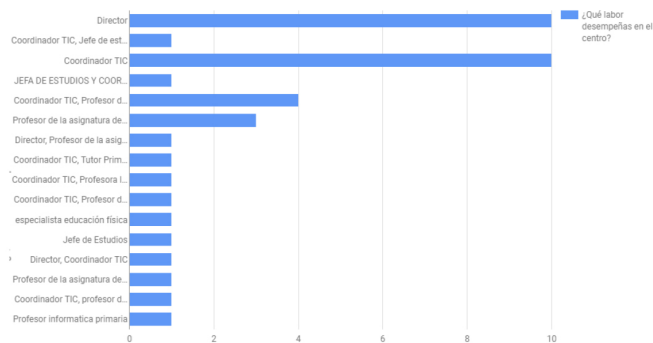


Fig. 6. Perfil de los encuestados

en los horarios de otras áreas, como parte de Matemáticas o Ciencias Naturales, a través de la realización de proyectos, o de forma extra escolar, pero esta última opción suele ser inviable, por limitaciones económicas de las familias.

También se obtiene información interesante de las respuestas a la pregunta *¿Crees que los alumnos están interesados en aprender programación?* Ya que un 38,3% está muy interesado, mientras que a un 25,5% le interesa, aunque no sabe realmente cuál es su utilidad. De otra parte, un 21,3% no muestra mucho interés en Programación, pero está a favor del uso de la tecnología en el aula y un 10,6% cree que en educación primaria aún no les interesa. Solo un 4,3% no lo sabe. Ver Figura 7.

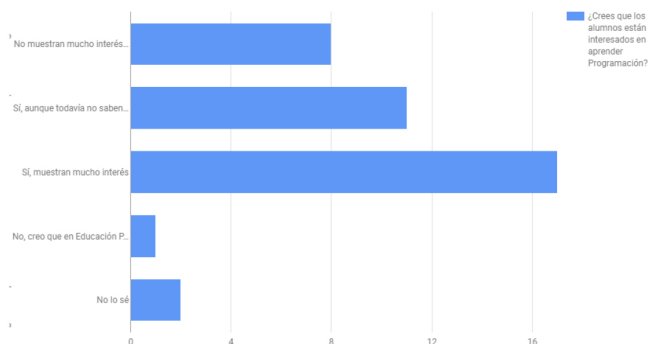


Fig. 7. ¿Crees que los alumnos están interesados en aprender Programación?

En cuanto a la pregunta *¿A partir de qué curso de enseña Programación en tu centro?* Los resultados indican que en un 39,1% de los centros no se enseña Programación, mientras que los centros que si la imparten lo hacen un 15,2% en 4º, un 10,9% en 5º y un 13% en 6º de Primaria. Se encontró además que un 8,7% lo imparte en 1º, un 2,2 % en 2º y 8,7% en 3º de Primaria.

De la pregunta *Si se enseña Programación ¿Cómo lo hacéis?* Se desprende que un 30,43% de los encuestados no enseña Programación, pero los que si lo hacen utilizan herramientas como Scratch [11] y Scratch Jr un 30,17% y otras herramientas como Code.org, Lego Wedo, App Inventor y BlocsCAD un 32,89%. Se observa también que un 6,51% opta por la enseñanza a través de actividades extra escolares y/o entidades externas. Ver Figura 8.

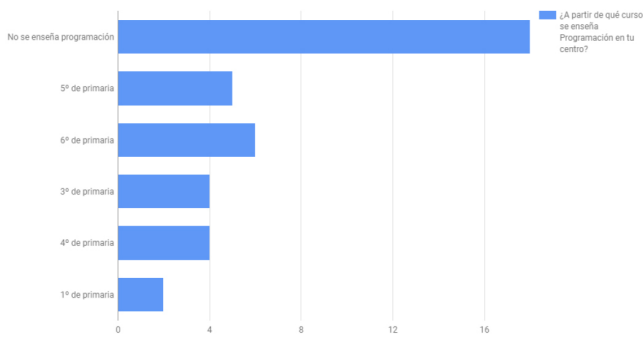
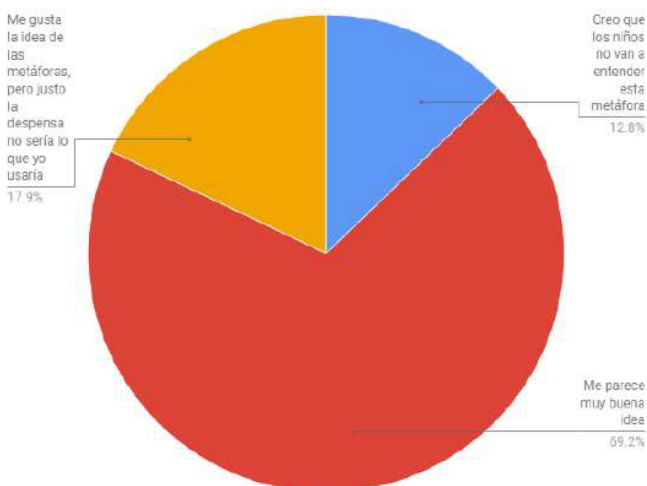


Fig. 8. Si se enseña Programación ¿Cómo lo hacéis?

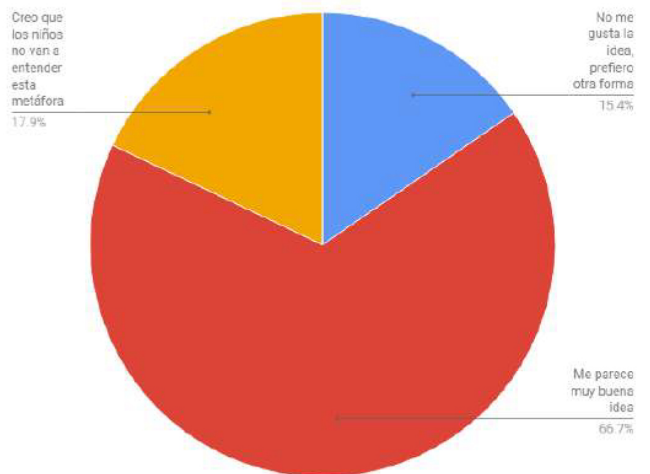
A la pregunta *¿Qué te parece enseñar Programación mediante el uso de metáforas? Por ejemplo, Programar es como cocinar, es decir, seguir paso a paso una receta*, el 63% considera que es una buena idea, al 19,6% no le gusta la idea, por su parte, un 8,7% apoya la idea de utilizar metáforas siempre que no se utilice la de cocinar y el mismo porcentaje considera que los niños no van a entender las metáforas.

Con el ánimo de conocer la aceptación del uso de las metáforas para enseñar algunos conceptos básicos relacionados con la Programación, se plantearon preguntas del estilo *¿Qué te parece comparar la "memoria del ordenador" con una despensa? es decir, en la memoria guardas todos los datos que necesitas en tu programa y la despensa es el espacio donde tienes lo que necesitas para cocinar*, las cuales recibieron un alto porcentaje de aceptación, más del 60%. Se les planteó por tanto la pregunta *¿Te gustaría que te facilitásemos unos guiones con metáforas como las anteriores para enseñar Programación en tu centro?* De la que el 47,8% cree que sí, puesto que les vendría muy bien para cuando tengan que impartirla y un 26,1% cree que, aunque actualmente no la impartan les servirá para irse preparando. Un 17,4% sin embargo indica que utiliza un método propio, con lo cual las metáforas no son de su interés. Ver Figura 9.

¿Qué te parece comparar la "memoria del ordenador" con una despensa? es decir, en la memoria guardas todos los datos que necesitas en tu programa y la despensa es el espacio donde tienes lo que necesitas para cocinar



¿Qué te parece comparar el concepto de "variable" con "cajita"? La variable es como una caja donde vas guardando los valores que necesitas...la vacías y la llenas según va marcando el programa



¿Qué te parece explicar el concepto de "entrada/salida" del programa con un teclado para representar la entrada y una pantalla para representar la salida de datos?

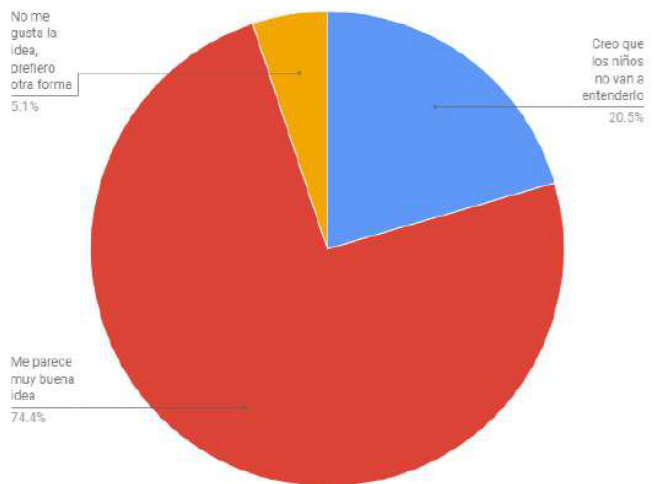


Fig. 9. Resultados sobre la aceptación del uso de las metáforas

Es gratificante saber que cerca de un 61% de los centros estarían dispuestos a recibir ayuda de nuestra parte, para poner en práctica los guiones. No obstante, hay aspectos que refinar en la propuesta, como el uso de ciertas metáforas, como la de los Reyes Magos, puesto que hay que tener en cuenta la diversidad cultural y/o económica y por tanto pensar en metáforas mucho más flexibles para poder aplicarlas con éxito.

V. CONCLUSIONES

Partiendo de la premisa de que la enseñanza de la Programación tiene un interés creciente a nivel mundial en Educación Primaria. En el caso de la Comunidad de Madrid es obligatorio según el RD 89/2014. Sin embargo, muchos colegios encuentran problemas a la hora de impartirla por falta

de recursos o de formación del profesorado. En este artículo, se ha realizado un cuestionario a 318 colegios, recibándose 46 respuestas.

A todos los centros les ha parecido útil y adecuado enseñar Programación en Primaria. Sin embargo, un 39,1% de los centros no pueden impartir programación porque encuentran dificultades en cómo encajarlo en el horario y en encontrar profesores formados adecuadamente. En el caso de que lo impartan, cuando se pregunta sobre la metodología usada, las respuestas suelen ser usar Scratch o juegos, pero sin metodología, y cuando se les propone una metodología, al 63% de los centros le parece bien y el 61% quisiera ponerla en práctica.

Se ha puesto de manifiesto que este tema de investigación es muy interesante actualmente ya que existen poco estudios o sin resultados concluyentes. Por otro lado, aunque la muestra es aún pequeña en relación a todos los centros de la Comunidad de Madrid, es representativa de la necesidad de metodologías para la didáctica de la Programación en Educación Primaria.

Como trabajo futuro se quiere seguir completando el estudio con más centros, y poder responder a preguntas que han realizado los centros cómo la edad a partir de la que se debería empezar a enseñar programación en Primaria, cuántas horas a la semana, y su impacto en el fomento del pensamiento computacional y otras competencias clave como resolución de problemas o creatividad.

AGRADECIMIENTOS

Este trabajo se ha financiado con los proyectos de investigación TIN2015-66731-C2-1-R del Ministerio de Economía y Competitividad y S2013/ICE-2715 de la Comunidad Autónoma de Madrid.

REFERENCIAS

- [1] RD 126/2014. Currículum de Educación Primaria. Disponible on-line en <https://goo.gl/22goCf> (última visita: 30 marzo, 2017).
- [2] C. Brackmann, D. Barone, A. Casali, R. Boucinha, S. Muñoz-Hernandez, "Computational thinking: Panorama of the Americas",

- In International Symposium on Computers in Education (SIIE), 2016, pp. 1-6.
- [3] S. Campe, J. Denner, Programming Games for Learning: A Research Synthesis. Paper presented at the annual meeting of the American Educational Research Association, Chicago, 2015.
- [4] I. Ouahbi, F. Kaddari, H. Darhmaoui, A. Elachqar, S. Lahmine, "Learning basic programming concepts by creating games with scratch programming environment", *Procedia-Social and Behavioral Sciences*, 191, 2015, pp. 1479-1482.
- [5] M. Jovanov, E. Stankov, M. Mihova, S. Ristov, M. Gusev, "Computing as a new compulsory subject in the Macedonian primary schools curriculum", In Global Engineering Education Conference (EDUCON), 2016, IEEE, pp. 680-685.
- [6] Eliminado por revision anónima
- [7] E. Lahtinen, K. Ala-Mutka, H.M. Järvinen, "A study of the difficulties of novice programmers", in *ACM SIGCSE Bulletin*, vol. 37, no. 3, 2005, pp. 14-18.
- [8] D. Ginat, "On novice loop boundaries and range conceptions", *Computer Science Education*, 14(3), 2004, pp. 165-181.
- [9] O. Seppälä, L. Malmi, A. Korhonen, "Observations on student misconceptions—A case study of the Build-Heap Algorithm", *Computer Science Education*, 16(3), 2006, pp. 241-255.
- [10] F. Heintz, L. Mannila, T. Färnqvist, "A review of models for introducing computational thinking, computer science and computing in K-12 education", *IEEE Frontiers in Education Conference*, 2016, pp. 1-9.
- [11] M. Resnick, J. Maloney, A. Monroy-Hernandez, N. Rusk, E. Eastmond, K. Brennan, et al., *Scratch: Programming for all*. *Communications of the ACM*, 52(11), 2009, pp. 60-67.
- [12] A. Sović, T. Jagušt, D. Seršić, "How to teach basic university-level programming concepts to first graders?", In *Integrated STEM Education Conference (ISEC)*, 2014, IEEE, pp. 1-6.
- [13] F. Kalelioğlu, "A new way of teaching programming skills to children: Code. Org", *Computers in Human Behavior*, 52, 2015, pp. 200-210.
- [14] G. Lakoff, M. Johnson, "Metaphors we live by". University of Chicago press, 2008.
- [15] J.P. Sanford, A. Tietz, S. Farooq, S. Guyer, R.B. Shapiro, "Metaphors we teach by", In *Proceedings of the 45th ACM technical symposium on Computer science education*, 2014, pp. 585-590.
- [16] R. Jiménez-Peris, C. Pareja-Flores, M. Patiño-Martínez, J.A. Velázquez-Iturbide, "The locker metaphor to teach dynamic memory", In *ACM SIGCSE Bulletin*, Vol. 29, No. 1, 1997, pp. 169-173.
- [17] W.W. Milner, "A broken metaphor in Java", *ACM SIGCSE Bulletin*, 41(4), 2010, pp. 76-77.
- [18] D. Pérez-Marín, R. Hijón-Neira, M. Martín-Lope, "Propuesta de Metodología basada en Metáforas para la Enseñanza de la Programación a Niños". *IEEE RITA* (aceptado para publicación).