# Developing a teacher training curriculum including Computational Thinking skills

## Early advances on a study focused on Colombia

Armando Muñoz del Castillo
Universidad de Santander
Pasto, Colombia
armando.munoz@cvudes.edu.co

Mario A. Muñoz
The University of Melbourne
Parkville, VIC 3010, Australia
munoz.m@unimelb.edu.au

Lydia Cruz Acosta Huertas, Edgar Herrera
Escuela Normal Superior de Pasto
Pasto, Colombia
{lydia.acosta,edgar.herrera}@escuelanormaldepasto.edu.co

Javier Jimenez Toledo, Deixy Ximena Ramos
Institución Universitaria CESMAG
Pasto, Colombia
{jajimenez,dxramos}@iucesmag.edu.co

***Abstract*—During training at the Complementary Formation Program (CFP), Colombian primary school teachers do not acquire Computational Thinking (CT) skills, which are considered fundamental for the knowledge economies of the 21st century. In this paper, we describe the early stages of a project aiming to reform the CFP curriculum, such that CT skills becomes an integral part. As a first step, we carried-out four introductory workshops with 64 first- and third-semester CFP students and then surveyed them to gather their perceptions on CT. The results show that the students have a limited understanding of CT and its associated skills, but they recognize their importance. Moreover, they agree that primary school should be the starting point to develop them. The project will continue gathering information from current CFP teachers, current primary school teachers, and university professors who are experts in curriculum development, to develop the curricular proposal.**

***Index Terms*—Computational Thinking, Curriculum design, Teacher training**

## I. Introduction

The term Computational Thinking (CT) was popularized by Jeannette M. Wing in 2006 [1], to describe the set of skills required to formulate and solve problems, through its representation as a logical sequence of basic operations, conditionals, loops, functions, and variables [2]. It involves critical and algorithmic thinking, creativity, pattern recognition, and idea decomposition. Although computer programming is a keystone skill, CT requires thinking at different abstraction levels independently of a device or coding language. As such, CT implies an intelligent and imaginative way to solve problems [3], making it a fundamental set of skills for all citizens in our current digital society [4]. This has led to intensive academic research [5, 6, 7, 8] and public policy aiming to incorporate the development of CT related skills into the official curricula. Key participants in this process should be the teaching staff, who must have received appropriate training to develop the skills themselves.

In Colombia, public policy [9] dictates that teacher training should include the development of skills such as programming and coding. For example, the Escuela Normal Superior de Pasto, a teacher training institution in the southwest of the country, offers a two-year tertiary course called the Complementary Formation Program (CFP). Its aim is to provide foundational training for pre-school and primary school teachers. The CFP curriculum includes two units aiming to develop digital competencies, such as authoring tools, specialized learning software, and learning management systems [10]. However, as recognized by the National Ministry of Education [11], there are no strategies in place to incentivize the development of broader information and communication technology (ICT) skills, or even more advanced CT skills, across the country. This is due to limitations on the training process, poor planning, and inappropriate curriculum structure.

In this paper, we describe the early stages of a project aiming to reform the CFP curriculum to include the development of CT skills. Our long-term aim is to answer questions such as: Which CT skills should be included in the training of primary school teachers? Which ones are the priority? How can these skills be developed? Which knowledge is required to reach competency in these skills? What didactic resources are needed? Which learning and assessment activities are appropriate? What principles, strategies, and criteria will direct the evaluation process? To answer these questions, along with a specialized literature review, we begun working with the participants in the educational process, i.e., the current CFP students and teachers, current primary school teachers, and experts in curriculum development. This paper presents the results from four workshops and a survey carried out on 6 first- and third-semester CFP students. The results show that the students have a limited understanding of what CT constitutes, and which are its associated skills. However, they recognize its importance and acknowledge that primary school should be the starting point to develop CT skills.

978-1-7281-4286-9/19/$31.00 ©2019 IEEE
DOI 10.1109/LACLO49268.2019.00012

## II. Background

Computational Thinking (CT) describes the set of skills that allow sophisticated problem solving, that often emerge from training in diverse disciplines such as mathematics (logic) language (argumentation), arts (creativity) and programming (algorithmic development). However, it is the ability to interconnect these skills that provides a more sophisticated approach that CT refers to. Hence, while programming is foundational, the current focus seems to overly emphasize it, disregarding the connections to other disciplines. Nevertheless, it is useful to observe the experiences in teaching programming, as a basis for the development of the cross-disciplinary skills that CT entitle.

There have been significant initiatives and tools designed to foster programming skills, with LOGO probably being the first one. Created by Seymour Paper at MIT in 1967, it quickly became a popular tool for teaching kids and young adults how to program [12]. Other significant and more modern tools used to teaching programming are: *Micromundos* is a learning environment derived from LOGO, in which projects are constructed by collecting graphs, animated figures, text, sound, and other media, fostering idea exploration and testing [13]. *Alice* is a block-based programming environment designed to teach, besides basic programming skills, logical thinking. It facilitates the creation of animation, interactive stories, and simple 3D games [14]. *Kodu* is a game-developing application that does not require formal programming skills [15]. *Minecraft* is a construction videogame set in an open world. It has an educational version available which can be used in different subject areas [16]. *CodeMonkey* is an environment that helps improving problem solving, logic and creativity skills. In it, the students develop codes using a language called "CoffeeScript", learning concepts such as loops, variables, functions, conditionals, among others [17]. *Scratch* is a block-based programming environment, developed by the Lifelong Kindergarten Group at MIT. Designed for students between eight and 16 years, it helps the development of mental skills through programming [18].

Public policies have also been implemented in several countries to foster CT and, more narrowly, programming skills. For example, in Argentina the *Program.AR* initiative of the Sadosky Foundation aims to introduce the computational sciences at the school level [19]. In the USA, Hadi and Ali Partovi founded the *code.org* Foundation, whose objective is to broaden the access to computational sciences schooling. Their view is that informatics should be taught at the same level as biology, chemistry or mathematics. It proposes a study plan for primary and secondary schools that includes computational sciences. Moreover, code.org organizes a yearly *coding hour* that has managed to involve a large percentage of their users. The foundation is supported by large companies such as Amazon, Facebook, Google, Infosys and Microsoft [20].

In Estonia, the *ProgreTiger* program introduced programming and robotics at pre-school, primary and vocational levels. This program is supported and funded by the Ministry of Education and Research, integrates technology education in the curriculum, offering the teaching material, training opportunities for the teachers, and financial support to the schools, such that they can acquire equipment [21].

In the UK, the *Barefoot Computing Project* was started with the objective of teaching computer sciences in primary school. It has fostered the creation of teacher communities, where ideas and good practices are shared. The Royal Society has recognized that Barefoot is the resource that most teachers choose to develop CT skills in the classroom. By 2018, Barefoot had reached more than 2 million children and more than 70 thousand teachers belong to Barefoot communities, representing more than 60% of the UK primary schools [22]. It is also noteworthy that in the UK, computer science is officially part of the curriculum [23]. This model has been followed by the Autonomous Community of Madrid, Spain, by including a subject called "Technology, Programming and Robotics" for grades first and third of secondary school since the academic year 2015-2016 [24].

Finally, in Colombia, there have been some initiatives to foster CT skills at the school level. The most relevant is the project "Computational Thinking in the Colombian Schools" organized by RENATA and the University of the Basque Country, in collaboration with the Ministry for Information and Communication Technologies and the Ministry for Education, has developed an educative ecosystem based on three basic elements: technology, human talent, and the institutions. The core of this project is a unit that introduces the concepts and basic processes of CT, using *Scratch* as teaching aid [25]. *Coderise* is a foundation aiming to foster CT skills on the students while improving their economic conditions through entrepreneurship [26]. The *Scratch prize*, awarded by the ICESI University and the Gabriel Piedrahita Uribe Foundation from 2012 to 2016, aimed to recognize and disseminate the best projects by teachers of primary and secondary education, that use *Scratch* as programming environment [27]. These are early and limited initiatives that have incorporated in their official curriculum concepts related to CT [3].

## III. Methodology and Results

We carried out four workshops with 64 first- and third-semester CFP students, to familiarize them with basic CT concepts. The students were enrolled in the units "ICT in the classroom I" and "ICT in the classroom II" during the second semester of 2018. Of these students, 11 were male and 53 were female, 35% had less than 20 years and 32% completed high school at a Normal school. With respect to CT, over 47% declared to have some knowledge about it. After the workshops, the students were asked for their view on four factors that justify the curricular proposal. Each factor had associated a positive and negative statement. A Likert scale was used with the options totally agree (TA), agree (A), indifferent (I), disagree (D) and totally disagree (TD). To facilitate the analysis, the scale was converted to the options very favorable (VF), favorable (F), indifferent (I), unfavorable (U) and very unfavorable (VU). The total favorability was found by averaging the results depending on its meaning. That is, agreeing on positive statements represents a favorable opinion, whereas agreeing on negative statements represents an unfavorable opinion.

*A. Workshop methodology and content*

The four workshops were held over three sessions of two hours across three weeks. The first workshop focused on defining theoretical concepts, such as critical thinking, decomposition, pattern recognition, abstraction, generalization, algorithmic thinking, collaboration, creativity, perseverance and error tolerance. Then, the participants had the opportunity to work in groups with different puzzles, such as the cup tower, whose objective is to construct the tallest and most stable tower with the least number of plastic cups within five minutes. We observed that the students engaged with the puzzles.

The second workshop focused on developing a simple computer program. Initially, we defined theoretical concepts such as algorithms and programming languages. Then, we introduced the *Scratch* environment by presenting the functions that each component carries out. Using an example, we demonstrated how these components can be interconnected to achieve a task. Finally, we provided an exercise in which the students had the liberty to come up with a task and a solution, such that they would explore the programming environment. We observed that the students struggled to comprehend these new theoretical concepts and to work with the programming environment. This resulted in an inability to work out the problem in the allocated time.

The third workshop focused on improving programming skills. This time, we used *code.org* as the environment. Working individually, the students explored the introductory exercises of the platform. Unlike the second workshop, we observed that the students engaged with the environment, and were able to complete the tasks in the allocated time.

Finally, the fourth workshop focused on developing critical thinking through exercises. Initially, we introduced the theoretical concept of critical thinking. Then, the students were shown videos of everyday situations, where conflict arises. The students are asked for their opinion and a solution to the conflict. We observed that the students engaged with the activity. Moreover, we demonstrated how critical thinking as a skill can be applied to a variety of situations.

*B. Survey results*

*1) Understanding CT as a concept:* To the positive statement "CT refers to the problem solving, system design and understanding of human behavior skills that make use of fundamental concepts from informatics" 64.1% responded (TA) and 25,0% (A). To the negative statement "CT is related exclusively to programming and the use of a computer" 26,6% responded (TD) and 18,8% (D). Fig. 1 shows the average results, which have a favorable or better opinion of 67.2% (45.3% VF and 21.9% F), an improvement 20.2% pre-workshops. Moreover, unfavorable or worse opinion was limited to 27.3% (10.9% U and 16.4% VU). The indifferent opinions came at 5.5%. This shows that the students are starting to unlink CT to the use of the computer.

2) *Importance of CT to the students:* To the positive statement "CT is very important to all students as a 21st-century skill, that should be acquired by people, as it gives the ability to solve real-world problems" 62.5% responded (TA) and 28.1% (A). To the negative statement "CT can be categorized as 'complex thinking', hence it should be developed at a late stage, preferably during higher education" 34.4% responded (TD) and 21.9% (D). Fig. 2 shows the average results, which have a favorable or better opinion of 73.4% (48.4% VF and 25.0% F) and an unfavorable or worse opinion of 14.1% (8.6% U and 5.5% VU). This showed recognition from the students that CT skills should be developed in the early education.

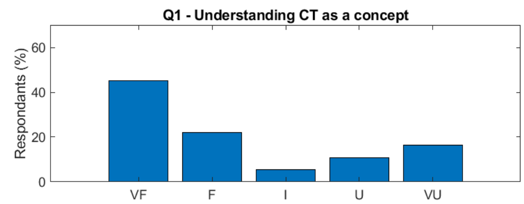


Fig. 1. Students' understanding of CT as a concept

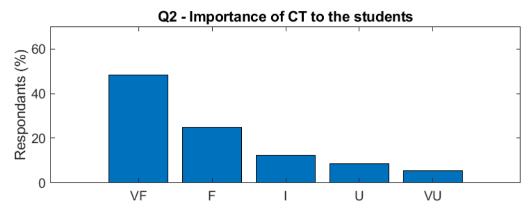


Fig. 2. Importance of CT for the students

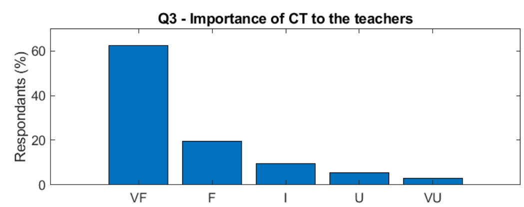


Fig. 3. Importance of CT for the teachers

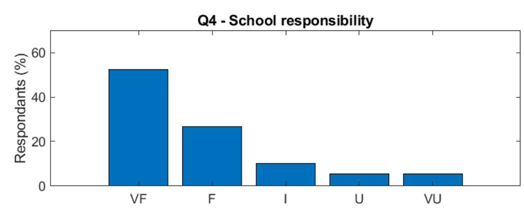


Fig. 4. Role of the schools on facilitating CT skills

3) *Importance of CT to the teachers:* To the positive statement "The teachers should have a broad understanding of CT, independently of their area of specialization or schooling level in which they work, so they can motivate and direct their students" 82.8% responded (TA) while 14.1% (A). To the negative statement "Knowledge on CT is only required for teachers that work in the technology, informatics or related areas" 42.2% responded (TD) and 25% (D). Fig. 3 shows the average results, which have a favorable or better opinion of 82.0% (62.5% VF and 19.5 F) and an unfavorable or worse opinion of 8.6% (5.5% U and 3.1 VU). This shows that the

students recognize that CT skills are broad-based skills that should be fostered for their own personal development.

4) *School responsibility:* To the positive statement "The foundational programs for teacher training offered by the normal schools or the faculties of education should contemplate the study of CT in their curriculum" 65.6% answered (TA) and 26.6% (A). To the negative statement "Most of the teachers do not understand CT, hence, it is not appropriate to implement these topics at the school level. It is better for them to develop these skills while at work" 39.1% responded (TD) and 26.6% (D). Fig. 4 shows the average results, which have a favorable or better opinion of 78.9% (52.3% VF and 26.6% F) and an unfavorable or worse opinion of 11.0% (5.5% U and 5.5% VU). These results indicate that the students think the schools have a large role to play in developing CT skills in their students.

## IV. Conclusions and Further Work

Computational Thinking (CT) describes a set of skills that are becoming fundamental for the future labor force. These skills, such as algorithmic thinking, allow the individual to manage the complexities of an ever more sophisticated work environment. Training individuals with these capacities require an equally capable teaching force. Unfortunately, in the Colombian case, while there are public policies in place to foster skills such as programming in teachers, there is a long way ahead, if the country wants to have in place a training program that fosters CT skills. Our first step towards improving teacher training is to develop a curricular proposal in which CT takes center stage. This paper presented the results of working with the students of the Complementary Formation Program (CFP). We observed that the students agree that CT skills are fundamental, and they need to incorporate them in their training. Our work in the development of this curricular proposal will continue by collecting the opinion of the other actors in this process with the following activities: (a) workshops and a perception survey of the current teachers of primary school; (b) workshops and a perception survey of the current teachers of the PCF; (c) conversation with a group of professors from the Information Education degree and Systems Engineering; (d) literature review about other proposals developed in other institutions at the national and international levels; and (e) analysis of the information and development of this proposal.

## References


[1] C. Selby and J. Woollard, "Computational thinking: The developing definition." *SIGCSE 2010*, pp. 6 – 9, 2010.

[2] M. González Román, "Codigoalfabetización y pensamiento computacional en educación primaria y secundaria: Validación de un instrumento y evaluación de programas," Ph.D. dissertation, UNED, 2016.

[3] J. Valverde-Berrocoso, M. R. Fernández-Sánchez, and M. C. Garrido-Arroyo, "El pensamiento computacional y las nuevas ecologías del aprendizaje," *Revista de Educación a Distancia*, 46, 2015.

[4] L. Giraldo Gómez, "Competencias mínimas en pensamiento computacional que debe tener un estudiante aspirante a la media técnica para mejorar su desempeño en la media técnica de las instituciones educativas de la alianza futuro digital Medellín," Master's thesis, Universidad EAFIT, 2014.

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

[6] G. Fletcher and J. Lu, "Human computing skills: Rethinking the K-12 experience," *Communications of the ACM*, 52(2)23–25, 2009.

[7] S. Grover and R. Pea, "Computational thinking in K-12," *Educational Researcher*, 42(1)38–43, 2013.

[8] M. Guzdial, "Paving the way for computational thinking," *Communications of the ACM*, 51(8)25–27, 2008.

[9] Ministerio de Educación Nacional, "Apropiación de TIC en el desarrollo profesional docente – Programa nacional de uso de medios y nuevas tecnologías." 2008.

[10] Escuela Normal Superior de Pasto, "Proyecto pedagógico institucional," 2008.

[11] Ministerio de Educación Nacional, "Competencias TIC para el desarrollo profesional docente. oficina de innovación educativa con uso de nuevas tecnologías." 2013.

[12] "Lenguaje de programación LOGO." [Online]. Available: https://www.ecured.cu/Lenguaje_de_programaci%C3%B3n_Logo

[13] "Logo Computer Systems Inc." [Online]. Available: http://www.micromundos.com/

[14] "Alice," 2003. [Online]. Available: http://alice.org/index.php

[15] M. Corporation, "Kodu Game Lab," 2011. [Online]. Available: https://www.kodugamelab.com/

[16] "Minecraft: Education Edition," 2019. [Online]. Available: https://education.minecraft.net/

[17] "Codemonkey," 2014. [Online]. Available: https://www.playcodemonkey.com

[18] "Scratch," 2008. [Online]. Available: https://scratch.mit.edu/

[19] "Fundación Sadosky," 2013. [Online]. Available: http://www.fundacionsadosky.org.ar/programas/programar/

[20] "code.org," 2019. [Online]. Available: http://www.code.org/

[21] Information Technology Foundation for Education, "Progetiger programme," 2012. [Online]. Available: https://www.hitsa.ee/it-education/educational-programmes/progetiger

[22] "Barefoot computing project," 2014. [Online]. Available: https://www.barefootcomputing.org/

[23] Department for Education UK, "Statutory guidance – national curriculum in England: computing programmes of study," 2013. [Online]. Available: https://www.gov.uk/government/-publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study

[24] J. S. López and R. C. Gutiérrez, "Pensamiento computacional y programación visual por bloques en el aula de primaria." *EDUCAR*, 53(1)129–146, 2017.

[25] X. Basogain, M. Olabe, J. Olabe, M. Rico, L. Rodríguez, and M. Amortegui, "Pensamiento computacional en las escuelas de Colombia: colaboración internacional de innovación en la educación," in *VirtualEduca*, 06 2017.

[26] "Fundación Coderise Colombia," 2012. [Online]. Available: HTTP://www.coderise.org/

[27] "Premio Scratch Colombia," 2016. [Online]. Available: http://edukatic.icesi.edu.co/2016/premio2016.php