Qwerty

17/1/2022

Open and Interdisciplinary
Journal of Technology,
Culture and Education

Publisher
Progedit, via De Cesare, 15
70122, Bari (Italy)
tel. 080.5230627
fax 080.5237648
info@progedit.com
www.progedit.com

qwerty.ckbg@gmail.com
www.ckbg.org/qwerty

Registrazione del Tribunale di Bari
n. 29 del 18/7/2005

ISSN 2240-2950

Indice

Developing computational thinking among pre-service teachers

Marta Peracaula-Bosch, Juan González-Martínez**

DOI: 10.30557/QW000049

Abstract

Training pre-service teachers in their own computational thinking is critical to building with them the discourse of CT didactics and its inclusion in the classroom with children in kindergarten and primary education. This research raises possible solutions and offers the results of an intervention carried out with 37 students in the 2nd year of the bachelor's degree in Education. An intensive CT proposal, through a Scratch project, allows all students to reach a sufficient level of CT, regardless of their previous experience and initial level. All students improve their mastery of CT skills: Those who have a lower initial level can develop it from the ground up; those who already have a high level at the beginning improve their efficiency in solving questions where they have to mobilize CT.

Keywords: computational thinking; pre-service teachers; digital skills; education

Introduction

There is no doubt about the astronomical growth that the concept of Computational Thinking (CT) has experienced in the field of ed-

* Universitat de Girona, Spain. Marta Peracaula-Bosch*, orcid: 0000-0003-0871-6583; Juan González-Martínez, orcid: 0000-0002-9175-6369.
Corresponding author: marta.peracaula@udg.edu

ucation in recent years. And, as proof of this, although it may seem superficial, we only need to look at the increase in the number of documents Google Scholar offers for the descriptors "computational thinking" (hereafter, CT) or "educational robotics": Until 1990, 3 and 40; until 2000, 6 and 129; until 2010, 38 and 758; until 2020, 2660 and 4110, respectively. This alone, in a simple way, can confirm that both CT and educational robotics are very widespread realities throughout the world, in all education systems. However, research has not progressed at the same pace, neither in the definition of the concept of CT itself, nor in its educational developments.

In a general way, CT is one of the abilities and skills that can help any citizen cope with the complex situation of the new knowledge society (KS) (Acevedo Borrega, 2016). It is in this situation that the reflection of Wing (2006, 2011) began, emphasizing that CT would become a basic tool for learning in an abstract, algorithmic, and logical way, enabling students to solve complex and open-ended problems.

At a time when CT is spreading in many schools, special training on CT has become a need for teachers; therefore, the most recent focus is on teacher training and teaching models related to CT (Morreale et al., 2012). Many teachers have unknowingly applied these strategies (Barr & Stephenson, 2011; Yadav et al., 2017); however, this cannot be taken for granted. On the one hand, it must be considered that teachers themselves have not been trained in CT (Bustillo Bayón, 2015; Ketelhut et al., 2020); on the other hand, they must have some skills as users of tools especially suitable for CT development (Adell Segura et al., 2017). This can provide the initial content knowledge (i.e., CT concepts, processes, and attitudes), technological knowledge (i.e., tools suitable for CT acquisition), in addition to pedagogical knowledge (i.e., teaching models and didactic approaches) (e.g., Koehler, Mishra, & Cain, 2013). It could also develop their confidence to engage with technological tools for learning and teaching (Sansone et al., 2019).

However, specific training proposals have been quite rare to find within the training modules of faculties of education until recently (Yadav et al., 2017). Moreover, the few documented experiences indicate that very few efforts (and even isolated, concrete actions) yield very interesting results. And it seems plausible to think that before attending to questions of CT didactics stricto sensu, it is important both to address the concept of CT of pre-service teachers (Peracaula-Bosch et al., 2020) and to ensure they reach a level of literacy that allows them to face the design and implementation of learning experiences in this field (Adler & Kim, 2018; Ketelhut et al., 2020). Devoting an intensive first part of a course to the development of CT in teacher training is not easy, so this article is dedicated to analysing the results of this proposal to find out whether pre-service teachers can improve their own CT, in an intensive way, before dedicating the rest of the course to didactic training itself.

Theoretical frame

When talking about CT, we must undoubtedly pay attention to the work of Seymour Papert, the father of constructionism (a constructivist approach considering that knowledge develops especially during processes of construction of tangible objects). Papert (1980) at the end of his classic *Mindstorms* mentions computational thinking and refers to constructionist learning environments in which the computer is a tool for building and thinking: «an-object-to-think-with» (p. 182). Traditionally, however, the concept is considered to be born with Wing's (2006) seminal definition, a quarter of a century later. Although this is a conceptual approach, it has undoubtedly functioned to a large extent as a definition; indeed, it points to important elements of CT: «Computational thinking involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science» (Wing, 2006, p. 33). However, it is in Wing (2014) where we find a complete definition: «Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out». It is a definition with essential key ideas: A mental process, the formulation of a problem, the expression of its solutions, the need for a computer (human or machine); and this is added to interesting reflections on abstraction or the importance of CT outside the concrete contexts of computer science.

From this point on, many authors believe that CT is a basic competence for the 21st century (Angeli et al., 2016; European Commission/EACEA/Eurydice, 2012; Fluck et al., 2016). It enables us to develop an effective problem-solving and problem-posing procedure (Fluck et al., 2016) thus helping us to understand the world we live in and live in it (Furber, 2012). In that sense, Grover and Pea (2013) emphasize that computation is a human activity; abstraction helps to focus on essentials and neglecting details; and, consequently, CT enables knowledge creation, creativity and innovation, in all senses and at all levels. For this reason, many education systems have decided to include it in the curriculum: Canada, the United Kingdom, Finland, and Australia are some examples of this (Acevedo Borrega, 2016; Bocconi et al., 2016). However, despite Wing's (2014) definition (or perhaps because of the wider dissemination of the 2006 definition), there is no consensus on the priority among the many elements that make up CT: abstraction, generalization, decomposition, algorithmic thinking and debugging (Angeli et al., 2016); automation (Bocconi et al., 2016), information gathering and processing and representation (Barr & Stephenson, 2011), bottom-up and top-down analysis, heuristics, divergent thinking, recursion, iteration, approximation, and metacognition (Zapata-Ros, 2015). And also, attitudinal elements such as confidence, persistence, and collaboration (Bocconi et al., 2016).

When considering the first elements, suggested by Wing (2006, 2011), and developed by Angeli (2016), we can understand why CT has spread in the educational system in the form of robotics or programming experiences (Furber, 2012). However, for economy of space we will not dwell on how CT is landing, in the most diverse forms, within education systems around the world. Instead, what interests us now is to focus not on the children who must develop their CT, but on the teachers who must stand alongside them in this learning process. In this sense, it is generally accepted that teachers need to be trained specifically in CT; and, for more than a decade now, the focus has been on pedagogical models associated with CT (Morreale et al., 2012). As we said in the introduction, one of the current evidences is that teachers themselves have not been trained in the use of tools that help to develop CT and that their experiences in robotics or programming in their initial training have been scarce or non-existent (Bustillo Bayón, 2015). Therefore, the first condition is that the teacher acquires the minimum technical skills with the tools or languages that they want to use. In this sense, we take up a vision proposed by Estebanell et al. (2018), in which they point out that, before facing the question of CT didactics, trainee teachers must consolidate themselves as CT users and as reflective users (i.e., develop their own CT in a reflective way) before going on to develop their dimension as teachers in CT or reflective teachers in CT. There is no doubt that the aim of CT teacher training is to make them competent as designers and implementers of strategies that help to acquire it (Prieto-Rodriguez & Berretta, 2014); but, in order to achieve this, a sufficient initial level of CT is essential.

According to Butler & Leahy (2021), the lack of research on the training processes of pre-service teachers justifies that we should focus on it in order to improve the training of trainers and, consequently, the training of children. In this training, of course, a first part dedicated to the development of CT is essential before addressing didactic issues; and, based on what we already know, few efforts in this direction yield positive results: Previous research confirmed both improvement of CT and attitudes towards it (González-Martínez et al., 2018; Peracaula-Bosch et al., 2020).

Although the importance of developing pre-service teachers' own CT is commonly acknowledged, research has focused more on the conceptual part than on the skills themselves (Yadav et al., 2017; Muñoz del Castillo et al., 2020). The question is certainly not unimportant, as knowing what computational thinking is allows us to jump from the user to the reflective user with our pre-service teachers (what I know about my own CT, what elements I identify when I apply it, what processes are involved, etc.). It helps also to dissociate relationships that are not implicit, such as those of the CT with robotics or programming, in particular; or with technology, in general (Sadik et al., 2017).

Nevertheless, there are already some promising results when analysing the impact of technical skills development. Thus, for example, the implementation of robotics activities with LEGO WeDo can lead to a direct improvement in CT skills when applied in programming contexts, with a particular improvement in abstraction (Jaipal-Jamani & Angeli, 2017); and the design of activities with Arduino for pre-service teachers also improves the CT level of students (which does not happen, on the contrary, if C++-based strategies are applied) (Pala & Türker, 2021. However, the solution is not straightforward, and there are many issues that still need to be clarified. On the one hand, as in any subject of disciplinary didactics, it is important not to stop at the development of disciplinary knowledge, but to jump into didactic issues; therefore, the time devoted to disciplinary knowledge should be reasonably limited and applied. On the other hand, precisely for this reason, it seems sensible to think that, in terms of efficiency, the effort we devote to the development of CT should allow us to learn elements that we can then also use in the didactic approach. Therefore, the development of CT through conceptualizing reflective processes and Scratch projects can be a very useful way of approaching this challenge, so that, from the beginning, we are projecting not only the development of CT, but also its possible didactic use as future teachers. At the end of this reflection, we come to our research question: Is it possible to develop pre-service teachers' CT, at the beginning of a subject, before moving on to the didactic aspects?

Methods

For this research, we decided to use an existing instrument, the Computational Thinking test (CTt), created and validated by Román-González (2016) and Román-González et al. (2018) to a whole group of students of the optional subject of Computational Thinking and

Programming in the 2nd year of the bachelor's degree in Teacher Training at the Universitat de Girona. The course was running during one term in 12 face-to-face sessions, 1.5 hours long each. Students were also supposed to carry out about 36 hours of autonomous work with support of the educators (resources, tutorials, tutoring and on-line contact when requested). The test was applied at the beginning of the course and at the end of the first half dedicated to different activities to develop students' CT, before tackling the second part, dedicated to the development of didactic skills to teach CT to children in early and primary education. Summarizing, the activities between the two tests were the following:

- A first spontaneous unplugged activity to create algorithms for a "human robot" to build cup towers (adapted and extended from curriculum.code.org/csf-18/coursee/1/). Before the creation of the algorithm, the different student teams had to create their own programming language (using symbols or words).
- An introductory session devoted to CT concept and approaches and practical examples of key elements of CT in our day-to-day life
- A first contact with block-based languages, as a coding approach for education purposes.
- A guided workshop on Turtlestich coding language of how to draw different geometrical patterns on screen.
- Scratch introductory activities related to different CT concepts and development of small individual projects.
- Five weeks of autonomous work with guidance through a specific class forum and the elaboration and distribution of tutorials by the trainers. During these weeks, every student was developing an open complex Scratch project related to an educational science simulation of their choice (with some requirements related to core programming concepts like: The use of loops, conditionals, sensors, variables, operators, simultaneous coding of different elements, taking care of initial conditions, etc.; and others of a metacognitive nature).

The applied version of the test was TPC-RA-B (code.intef.es/wp-content/uploads/2019/12/Impacto_EscueladePensamientoComputacional_Curso2018-2019.pdf). This version combines the more discriminative items of the original CTt as a diagnostic tool for algorithmic thinking (some of them adapted to older students), with 6 *Bebras Tasks* (Dagiene & Futschek, 2008). The CTt items correlate with programming achievement in block-based environments, and the Bebras tasks focus on the measure of CT transfer to real-life problems (Román-González et al., 2019). Both algorithmic thinking and CT transfer development were expected from the course.

Thirty-seven students enrolled in the subject voluntarily answered both the test before and after the training. Since it was delivered as a Moodle test (also in pre- and post-training versions), we could also register the time students used to complete it.

Analysing the reliability coefficients, they are considered acceptable for the ranges commonly accepted in the educational field. They are detailed in Table 1.

Table 1. Reliability levels

Scale	Crombach's Alpha
Pre-training test	0.860
Post-training test	0.755

Results

In this section, we offer a synthesis of the most relevant data found in the analysis. Regarding the CTt results and the time students needed to solve the test, we can show the Figure 1, where the increasing level of CT (from 19.03 to 22.41 in a scale of 30 points) and decreasing time (from an average of 45.46 minutes to 41.32):

If we analyse the difference between the pre-training and the post-training CTt, we see that these differences are significant, which indicates that relevant learning has taken place in relation to the computational thinking of the participating subjects (sig. < 0.001). The difference between the initial and final CTt scores is therefore not due to chance, but rather that the learning experience has produced a significant change in the subjects' CT level.

Having previous experience in programming or robotics activities makes a significant difference in pre-training CTt results, but not in post-training results. As it can be seen in the Figure 2, previous experience can make a difference at the beginning of the learning process (sig = 0.024), but those differences, fewer, are not relevant at the end (sig = 0.221).

On the one hand, this means that the difference between those groups at the beginning is more important than at the end; and, on the other hand, that learning experience neutralizes the influence of previous experience on the level of CT. And this is especially important because, in every group (with or without prior experience), the results of the post-training CTt are always higher than those of the pre-training. At this point, it is important to note that prior experience is only shown to have incidence at the CTt pre-training scores, but not in the time informants use to answer the test. Furthermore, in the post-training CTt the differences (in CT level or time) are not significant. In the pre-training test, subjects with prior experience spend an average of 44.44 minutes and those without prior experience 45.42 minutes (p=0.707); in the post-training test, informants with prior experience spend an average of 39.22 minutes and those without prior experience 43.32 minutes on average (p=0.111). Beyond the interpretation linked to the learning experience, it seems reasonable to think that previous experience only conditions the initial level of CT but nothing more (the difference is not sustained over time); therefore, as we said, the learning experience neutralises its effect.

From here we can analyse the post-training CTt results considering three sample groups created from the pre-training CTt levels, to see what their CT improvement is (difference between the post-training CTt and the pre-training CTt). And it is very interesting to see how the progression is evident in all the groups (that is why we said before that learning occurs from a general perspective), but it is especially intense in those students who had started the semester with a lower level of computational thinking: In them, learning is much more intense (and the difference between those who in the pre-training CTt have lower and higher values regarding learning is statistically significant, with sig. = 0.10). Data are shown in Figure 3:

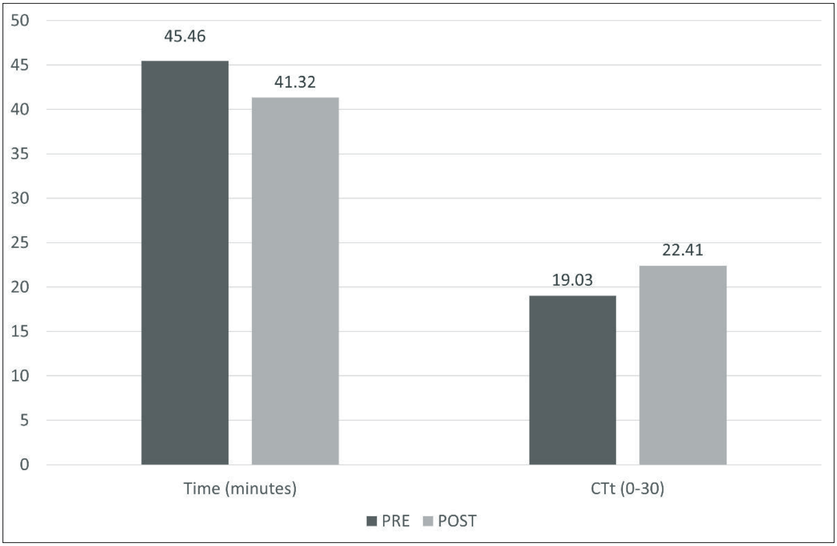

Figure 1. Pre-training and post-training CTt results and needed time in minutes

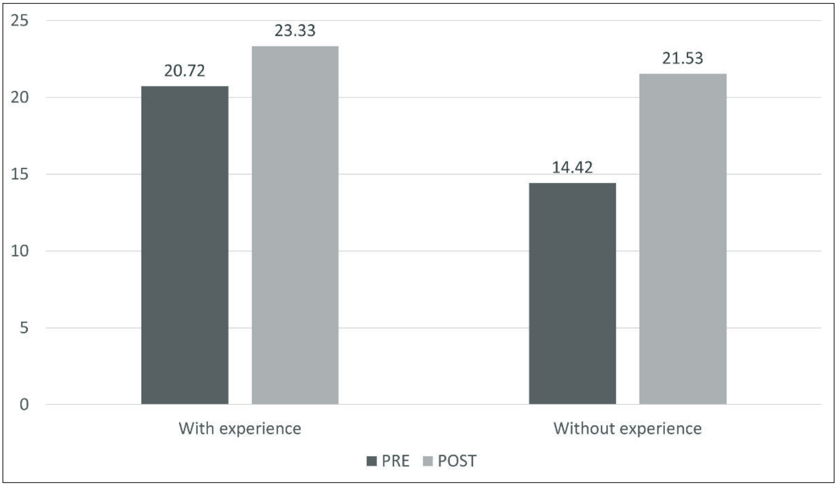

Figure 2. Pre-training and post-training CTt results according to previous experience

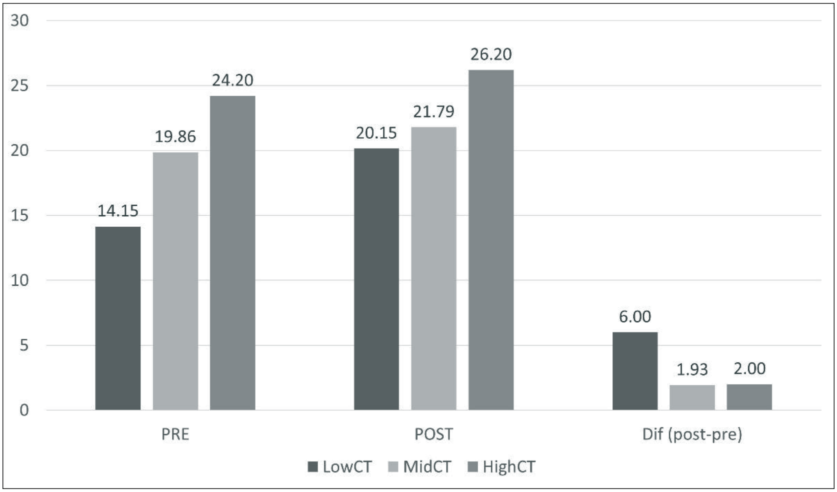

Figure 3. Pre-training, post-training and differences in CTt results according to initial results

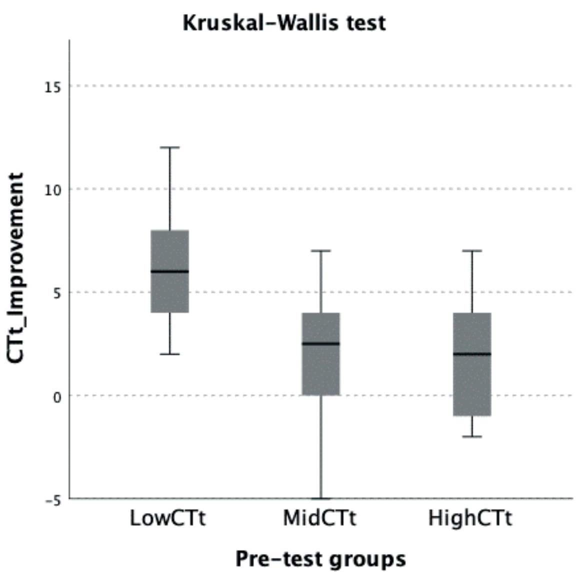

Figure 4. Differences in CT level improvement for pre-training groups

Beyond what can be seen in the graph above, the Kruskal-Wallis test for independent samples (Figure 4) shows significant differences in CT level improvement between the LowCT and HighCT (sig.=0.012) and LowCT and MidCT (sig.=0.007) groups, but not between the two higher groups (sig.=0.941).

And, although we do not represent it due to space limitations, something similar could be said about the time spent by the different groups to solve the post-training CTt, which is inversely proportional to the result obtained (44.08 minutes on average for the LowCT, 40.64 minutes for the MidCT, 38.70 minutes for the HighCT). However, in this case only the difference in time between the LowCT and HighCT groups is significant (sig.=0.008).

At this point, we would like to analyse this progression of each of the groups in more detail, in order to better understand what is happening in each of them. To do this, we analyse the two values that can give us the most information: The results of the CTt and the time they use to answer it (both pre-training and post-training). As it can be seen in Table 2, the improvement in the CTt is significant in the LowCT and MidCT groups (in this group, the SD is higher than in the HighCT); on the other hand, the decrease in time is significant in the MidCT groups (reducing the duration of the test by an average of 6.50 minutes) and, above all, in the HighCT group (reducing the duration of the test by an average of 9.40 minutes).

Table 2. Significance levels of the differences between pre-training and post-training time and results according to pre-training groups

Group	CTt improvement (sig.)	Time difference (sig.)
LowCT	0.022	0.594
MidCT	0.049	0.007
HighCT	0.081	0.008

It is plausible to think that in a group, the HighCT, in which the values are so high (initial mean of 24.20/30, SD=2.846) the possibilities of improvement for some subjects are difficult to achieve (some of them reached values of 27 and 28 points in the first test); therefore, learning in them would not be in terms of effectiveness (scoring high in the test, which they had already done) but in terms of efficiency (passing it with greater comfort, less effort, more agility, etc.; that is, passing it in less time).

Discussion

As we have seen, defining a first part of a teacher training course for the development of CT before going through the didactic approach can seem a good decision also in practical terms. In addition to a necessary base of content knowledge (Koehler et al., 2013), teachers experience and reflect upon learning activities that can inspire them to design according to the curricular needs and context of their teaching practice (Adler & Kim, 2018). Literature suggests the need of that approach (Butler & Leahy, 2021, Peracaula-Bosch et al., 2020); and it is particularly interesting for assuring pre-service teachers reach the user and reflective user needed levels (Estebanell et al., 2018). Our results confirm that this approach is not only interesting from a theoretical perspective, but also affordable: All students improve their CT level, regardless of their CT initial level or their previous experience in robotics or programming activities (students learn CT). Since acquiring skills in programming is one of the tools to develop and learn about CT when accompanied by reflection on how to apply these skills beyond their original context (Voogt et al., 2015), this learning strategy is helpful before moving them on to didactic aspects.

Moreover, our experience is especially interesting because those students with lower levels of CT at the beginning improve them more (and those with higher levels of CT from the very beginning, with little real chance of major improvements, improve significantly in the efficiency with which they apply TC, which is also a form of learning): If all improve, but those who need it most improve the most, a first part of the course dedicated to CT itself allows facing the second part of the course (dedicated to CT didactics) with a larger degree of evenness in the understanding of the concepts and skills for which the pedagogical approaches are going to be designed as a whole group.

Therefore, if we agree with Adell Segura et al. (2017) and Peracaula-Bosch et al. (2020) that pre-service teachers should be trained in their own CT, our data support that doing so can give us good results in a short time.

References

Acevedo Borrega, J. (2016). *El pensamiento computacional en la educación obligatoria. Una revisión sistemática de la literatura*. Universidad de Extremadura.

Adell Segura, J., Esteve-Mon, F. M., Llopis Nebot, M. Á., & Valdeolivas Novella, M. G. (2017). El pensamiento computacional en la formación inicial del profesorado de Infantil y Primaria. In V. Abella García, V. Ausín Villaverde, & V. Delgado Benito (Eds.), *Actas de las XXV Jornadas Universitarias de Tecnología Educativa* (pp. 151-158). Red Universitaria de Tecnología Educativa.

Adler, R. F., & Kim, H. (2018). Enhancing future K-8 teachers' computational thinking skills through modeling and simulations. *Education and Information Technologies*, *23*(4), 1501-1514. doi.org/10.1007/s10639-017-9675-1.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society*, *19*(3), 47-57. www.jstor.org/stable/jeductechsoci.19.3.47.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?. *ACM Inroads*, *2*(1), 48-54. doi.org/10.1145/1929887.1929905.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education. Implications for policy and practice*. doi.org/10.2791/792158.

Bustillo Bayón, J. (2015). Formación del profesorado con scratch: análisis de la escasa incidencia en el aula. *Opción*, *31*(1), 164-182. www.redalyc.org/pdf/310/31043005010.pdf.

Butler, D., & Leahy, M. (2021). Developing pre-service teachers' understanding of computational thinking: A constructionist approach. *British Journal of Educational Technology*, *52*(3), 1-18. doi.org/10.1111/bjet.13090.

Dagiene, V., & Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. In R. T. Mittermeir & M. M. Sysło (Eds.), *International Conference on Informatics in Secondary Schools-Evolution and Perspectives* (pp. 19-30). Springer. doi.org/10.1007/978-3-540-69924-8_2.

Estebanell, M., López, V., Peracaula-Bosch, M., Simarro, C., Cornellà, P., Couso, D., González-Martínez, J., Alsina, À., Badillo, E., & Heras, R. (2018). *Teacher training in computational thinking. Teaching guide*. pecofim.wixsite.com/pecofim/guia-didactica.

European Education and Culture Executive Agency, Eurydice (2015). *Developing key competences at school in Europe: challenges and opportunities for policy*, Publications Office. https://data.europa.eu/doi/10.2797/93204

Fluck, A., Webb, M., Cox, M., Angeli, C., Malyn-Smith, J., Voogt, J., & Zagami, J. (2016). Arguing for computer science in the school curriculum. *Educational Technology & Society*, *19*(3), 38-46. www.jstor.org/stable/jeductechsoci.19.3.38.

Furber, S. (2012). *Shut down or restart? The way forward for computing in UK schools*. The Royal Society.

González-Martínez, J., Estebanell, M., & Peracaula-Bosch, M. (2018). *¿Robots o programación? El concepto de Pensamiento Computacional y los futuros maestros. EKS. Education in the Knowledge Society*, *19*, 29–45. https://doi.org/http://dx.doi.org/10.14201/eks20181922945

Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Jaipal-Jamani, K., & Angeli, C. (2017). Effect of robotics on elementary pre-service teachers' self-efficacy, science learning, and computational thinking. *Journal of Science Education and Technology*, *26*(2), 175-192. doi.org/10.1007/s10956-016-9663-z.

Ketelhut, D.J., Mills, K., Hestness, E. *et al.* Teacher Change Following a Professional Development Experience in Integrating Computational Thinking into Elementary Science. *J Sci Educ Technol* **29,** 174–188 (2020). https://doi.org/10.1007/s10956-019-09798-4

Koehler, M. J., Mishra, P., & Cain, W. (2013). What is technological pedagogical content (TPACK)?. *Journal of Education*, *193*(3), 13-19. doi.org/10.1177/002205741319300303.

Morreale, P., Jimenez, L., Goski, C., & Stewart-Gardiner, C. (2012). Measuring the impact of computational thinking workshops on high school teachers. *Journal of Computing Sciences in Colleges*, *27*(6), 151-157.

Muñoz del Castillo, A. S., Torres, G. P., & Salazar Losada, J. C. (2020). Desarrollo del pensamiento computacional en la formación inicial de maestros de básica primaria. *Ciencia, Tecnología y Competencias*, 68-70. memoriascimted.com/wp-content/uploads/2020/12/Memorias-CIITA-2020-28-de-Dic.pdf.

Pala, F. K., & Türker, P. M. (2021). The effects of different programming trainings on the computational thinking skills. *Interactive Learning Environments*, *29*(7), 1090-1100. doi.org/10.1080/10494820.2019.1635495.

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

Peracaula-Bosch, M., Estebanell-Minguell, M., Couso, D., & González-Martínez, J. (2020). What do pre-service teachers know about computational thinking?. *Aloma. Revista de Psicologia, Ciències de l'Educació i de l'Esport*, *38*(1), 75-86. www.revistaaloma.net/index.php/aloma/article/view/392.

Prieto-Rodriguez, E., & Berretta, R. (2014). Digital technology teachers' perceptions of computer science: It is not all about programming. In *2014 I.E. Frontiers in Education Conference (FIE) Proceedings* (pp. 1–5). Madrid (Spain). IEEE Computer Society Digital Library. https://doi.ieeecomputersociety.org/10.1109/FIE.2014.7044134

Román-González, M. (2016). *Codigoalfabetización y pensamiento Computacional en educación Primaria y Secundaria: validación de un instrumento y evaluación de programas* (Doctoral Dissertation, Universidad Nacional de Educación a Distancia, Madrid, Spain). dialnet.unirioja.es/servlet/tesis?codigo=65247.

Román-González, M., Moreno-León, J., & Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In S. C. Kong & H. Abelson (Eds.), *Computational thinking education* (pp. 79-98). Springer. doi.org/10.1007/978-981-13-6528-7_6.

Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction*, *18*, 47-58. doi.org/10.1016/j.ijcci.2018.06.004.

Sadik, O., Leftwich, A.-O., & Nadiruzzaman, H. (2017). Computational thinking conceptions and misconceptions: Progression of pre-service teacher thinking during computer science lesson planning. In P. J. Rich & C. B. Hodges (Eds.), *Educational communications and technology-issues and innovations* (pp. 221-238). Springer International Publishing. doi.org/10.1007/978-3-319-52691-1_14.

Sansone, N., Cesareni, D., Bortolotti, I., & Buglass, S. L. (2019). Teaching technology-mediated collaborative learning for trainee teachers. *Technology, Pedagogy and Education*, *28*(3), 381-394, doi.org/10.1080/1475939X.2019.1623070.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715-728. doi.org/10.1007/s10639-015-9412-6.

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

Wing, J. M. (2011). Computational Thinking: What and why?. *The Link Magazine*. www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why.

Wing, J. M. (2014, January 10). Computational Thinking benefits society. *Social Issues in Computing*. Academic Press. socialissues.cs.toronto.edu/index.html%3Fp=279.html.

Yadav, A., Gretter, S., Good, J., & Mclean, T. (2017). Computational thinking in teacher education. In M. Spector, M. J. Bishop, & D. Ifenthaler (Eds.), *Emerging research, practice, and policy on computational thinking* (pp. 205-220). Springer. doi.org/10.1007/978-3-319-52691-1.

Zapata-Ros, M. (2015). Pensamiento computacional: una nueva alfabetización digital. *Revista de Educación a Distancia (RED)*, *46*, 1-47. doi.org/10.6018/red/46/4.