International Journal of Engineering Education Vol. 42, No. 2, pp. 293–299, 2026
Printed in Great Britain

0949-149X/91 $3.00+0.00

How To Approach STEM in K–12: A Hands-On Proposal for Teacher Training on Computational Thinking*

JUAN JOSÉ SANTAENGRACIA and LUIS J. RODRÍGUEZ-MUÑIZ
Universidad de Oviedo, C/Jesús Arias de Velasco, s/n, 33005 Oviedo, Spain. E-mail: juanjose@uniovi.es, luisj@uniovi.es

BELÉN PALOP
Universidad Complutense de Madrid, Av. Complutense, s/n, Moncloa-Aravaca, 28040 Madrid, Spain. E-mail: b.palop@ucm.es

This article presents a refined teacher training program in computational thinking as an entry point to STEM education, aimed at fostering STEM vocations from early educational stages. Developed through two iterations of the ADDIE model, the program integrates all components of CT based on the Three Pillar Model – data, problems, and algorithms – and connects them with key ideas in related areas. The training adopts a reflective approach that deconstructs teachers' prior knowledge and beliefs to reconstruct them as professional competence, enabling educators to deliver conceptually grounded CT instruction across all K–12 levels.

Keywords: computational thinking; professional competence; STEM education; teacher training; Three Pillar Model

1. Introduction

Engineering education is facing increasingly complex challenges in a digitally data-driven and interconnected world. The ability to solve complex problems through computational thinking (CT onwards) has become essential not only for computer scientists, but also for all technical professionals [1] and, generally speaking, all citizens [2]. Since Jeannette Wing's seminal work [1] considered CT as a fundamental skill for everyone, all countries have made an effort to include it in K–12 curricula [2–4]. According to her interpretation, CT means formulating problems in a way that a computer can help solve them. It supports key concepts and skills such as abstraction, decomposition, and algorithmic thinking. Research shows that engaging students in CT activities fostering different CT components (e.g., abstraction, algorithmic thinking, modeling) can improve problem-solving capacities, particularly in STEM domains [5, 6]. However, current CT training programs (both for pre-service and in-service teachers) place a strong emphasis on programming and robotics, thereby narrowing the very concept of CT and overlooking many of its connections with the STEM domains [7, 8].

The increasing visibility of STEM careers and vocations highlights a growing social interest in this field. Sustaining this trend necessitates early engagement in STEM education, as such early involvement has a significant impact on students' propensity to pursue STEM-related pathways in the future [9]. Several studies have found that interest in science and engineering often declines during secondary education [9, 10], as shown in the examples that follow: it has been shown that students who do not develop an interest in STEM by middle school are far less likely to pursue scientific or technical careers later, while students who expressed STEM career aspirations as early as seventh or eighth grade were 3.4 times more likely to eventually earn a STEM degree [9]. Longitudinal studies in the U.S. [11] have shown that initial enthusiasm sometimes fades during high school, with many students reporting a decline in interest by graduation, commonly attributing it to feeling incapable in science classes and a general lack of engagement. This disengagement is particularly pronounced among girls, who remain heavily underrepresented among engineering and computing graduates, comprising only about 20% of engineering and less than 15% of computer science graduates in Europe [12–14]. European policymakers have long warned about declining enrollments in engineering and computing, prompting calls for urgent reforms in STEM education to address students' interest [12], especially for young women, who often report feeling discouraged in technical subjects during secondary school, which steers them away from STEM pathways [14, 15]. For these reasons, pre-university teachers play a pivotal role in motivating and retaining students in STEM. Early CT competence nurtures future engineers capable of tackling complex socio-technical problems, thereby feeding the engineering-education pipeline and providing a concrete entry point for the "E" in STEM through the integration of CT with engineering practices. Positive teacher encouragement and engaging teaching methods have been shown to significantly increase students' likelihood of pursuing STEM studies [10, 14]. Thus, well-

* Accepted 24 October 2025.

prepared teachers in primary and secondary education are key to broadening participation – and, particularly, girls' participation – in STEM fields.

CT frequently represents the first school explicit encounter with subjects closely linked to engineering and technological disciplines. Over the past decade, many countries have begun incorporating CT – including elements such as programming – into their K–12 curricula [16, 17] to equip students with fundamental skills and provide them with pathways to pursue further studies or careers in computing-related fields [18]. For example, the UK overhauled its national curriculum in 2014 to mandate computer science for all ages; France and Italy have incrementally added programming and algorithmics into math or technology classes in secondary schools; and many other European countries have launched CT initiatives at the basic education level [16, 17]. In [19] three main approaches to integrating CT in European education systems are identified: as a cross-curricular competence; embedded within existing subjects (typically mathematics or technology); or as a standalone subject. While some countries adopt hybrid models, each approach entails different levels of curricular depth, coherence, and teacher preparation requirements. In some systems, these contents are taught through dedicated informatics subjects in secondary and upper-secondary education, while in others they are embedded within mathematics or technology courses. In countries where informatics is merged into the math curriculum, the typical instructor in secondary education is a mathematics teacher who may or may not have specific training in computer science while at the primary level computing lessons are usually delivered by generalist classroom teachers, most of whom have little background in informatics beyond basic digital literacy [17, 19–21]. Despite their commitment and efforts to self-train, many teachers face challenges in confidently addressing STEM and CT-related content, often due to limited previous formal preparation in both content and pedagogical strategies and lacking data-centered training in the pre-service and in-service training [7, 8]. This variability makes teacher preparation a critical issue: ensuring that all students benefit from equitable and meaningful exposure to computing thus requires that teachers – regardless of their background – are well prepared to teach these contents in engaging and inclusive ways.

Considering these challenges, this paper presents a refined proposal for a teacher training program targeting all K–12 educators. Its design followed two consecutive iterations of the ADDIE model (Analysis, Design, Development, Implementation, and Evaluation), a well-established framework in instructional design [21]. The program was initially carried out with a cohort of approximately 100 in-service teachers and subsequently with a second cohort of around 40 participants, encompassing all pre-university educational stages by three facilitators [22] working in parallel, enabling close monitoring of group work and the provision of immediate, personalized feedback. Each iteration informed the redesign of the training based on empirical observations and participant feedback, allowing for progressive refinement. The version presented in this paper incorporates the conceptual and pedagogical improvements resulting from these iterations, thus representing a consolidated and evidence-informed training proposal.

The training program is grounded in the Three Pillar Model (TPM) [23], which conceptualizes CT as a reasoning process built upon three foundational elements: data, problems, and algorithms, defining it as "the way of reasoning that allows people to tackle a problem on some data with the aim of having a computer solve it". This triadic structure underscores the relevance of data not merely as contextual input, but as a core component in computational reasoning – on par with algorithmic design and problem-solving. By aligning with national curriculum frameworks and promoting inclusive practices aimed at increasing female participation in STEM fields, the program equips teachers with both conceptual foundations and didactic strategies to foster meaningful and equitable engagement with STEM learning from early educational stages.

2. Design Methodology

The design of the teacher training program described, grounded in a reflective learning model, aims to deconstruct and transform prior knowledge, experiences, and belief systems into professional competence [24]. The deconstruction process begins with an initial session in the form of a seminal talk, designed to critically surface and question participants' prior beliefs and assumptions about CT. Subsequent knowledge construction unfolds through a series of structured workshops in which participants engage in scaffolded tasks [25] designed to promote conceptual and pedagogical knowledge. These scaffolded tasks take place in small collaborative groups of two to four participants [26]. This group configuration fosters dialogue, co-construction of knowledge (as proposed in [24]), and mutual scaffolding, all of which are central to the development of professional competence.

Following the two initial iterations described in the Introduction, the training program consists of six practical workshops and three seminal talks. The workshops are designed to be adaptable across the entire K–12 spectrum, with suggested modifications outlined in the description of each activity to accommodate different educational stages. Each workshop includes a proposed time allocation, although a flexible margin of up to 30 minutes is allowed to facilitate adaptation to diverse instructional contexts. All workshop activities are systematically aligned with the components of the TPM model [23], ensuring a coherent integration of its three core pillars: data, problems, and algorithms. Furthermore, each activity is explicitly designed to foster both conceptual and pedagogical connections between CT and the foundational ideas of mathematics and science, reinforcing its relevance within broader STEM education frameworks.

3. Description of the Teacher Training

3.1 First Lecture: Foundations of CT (1h)

The opening lecture offers a general overview of CT. It introduced the participants to the TPM, detailing each component, and providing illustrative examples of activities corresponding to each. This lecture served to establish a shared conceptual foundation across all participants.

3.2 Workshop 1: Sorting Algorithms through Manipulatives (2h)

This workshop introduces participants to sorting algorithms using physical materials designed to emulate how computers process and order data. The objective is to introduce participants to algorithmic thinking through the design and comparison of sorting strategies, fostering reflection on how computers handle data-ordering tasks working on concepts like comparison operations, memory constraints, algorithm design, and efficiency. This workshop engages teachers in all three pillars of the TPM, addressing most of their associated components. It fosters the development of algorithms and procedures, abstraction, decomposition, and pattern recognition.

The session begins with a whole-group activity using diatonic bells: eight bells are arranged in a random order, and one participant is asked to sort them from lowest to highest pitch. Because the bells can only be compared two at a time and participants do not retain memory of previous comparisons, the activity serves as a concrete analogy of a computational model with limited memory and sequential access – mirroring how computers execute data ordering tasks under constrained resources. Following this demonstration, participants use a manipulative system consisting of small cups with hidden numbers and an eight-cell grid representing computer memory (see Fig. 1). Using this setup, they are tasked with designing their own sorting algorithms, distinct from the one used with the bells. Finally, the participants' algorithms are collected and shared for group discussion, enabling comparative analysis of the various approaches. The facilitator ensures that key classical sorting methods – such as selection, insertion, bubble, and merge – are introduced and discussed, even if they do not emerge spontaneously from the participants' proposals. This concluding phase invites reflection on the intuitiveness, complexity, and efficiency of the different sorting strategies developed.

Fig. 1. Ordering cups. Own elaboration.

3.3 Workshop 2: Stochastic and CT with Dollar Street (2h)

This workshop connects stochastic reasoning and CT by transforming qualitative visual information into structured data. The main objective is to explore how qualitative visual information can be transformed into structured data, connecting stochastic reasoning with CT through image-based inquiry. The CT components associated with this workshop are the data and problem-solving pillars of CT, while also addressing abstraction and pattern recognition.

Participants are introduced to the Gapminder website, specifically the Dollar Street section [27] which presents images of families and everyday items categorized by income and country. Using the category "dishes" (though other categories could be used), the session begins with the question: "How do we convert images into data that a computer can process?" With a selection of dish images on display, participants are guided through questions such as: What are the variables? What values do they take? Who or what are the individuals? How can the information be coded, stored, and structured? Participants progressively refine their data models by creating tables, categories, and, ensuring that chosen variables are applicable to new images. The session is structured through prompts like: Which household has the most dishes? What is the most common shape or color? Can we find dishes in good condition? What questions can be answered with the data? Which cannot? The session concludes with a reflection on data representation in computing, reflecting on the challenges of pattern recognition in images, including a discussion on images as a form of data.

3.4 Workshop 3: Numerical and Algebraic Reasoning through Exploding Dots (3h)

This workshop connects arithmetic, number bases (particularly base 2), and algebraic reasoning with CT through the Exploding Dots number representation [28]. The main objective of this workshop is to explore binary representation and operations through Exploding Dots, reinforcing connections between number bases, arithmetic, and algebraic reasoning while fostering the CT components of data representation, modeling, abstraction, generalization, pattern recognition, and algorithms and procedures.

The session begins by distributing 31 linking cubes and a laminated A3 sheet or whiteboard to each group. Participants are introduced to the main mechanics of Exploding Dots, using a "2-to-1 machine" to work in the binary system. The basic rule is introduced: cubes always enter from the right, and whenever two cube towers are placed in a single cell, they "explode", and one moves to the next cell to the left together. After guided practice to ensure understanding of this process, participants are asked to represent a small number, adding cubes one by one. Later, a larger number is proposed so that entering cubes individually becomes inefficient, prompting participants to adopt a different strategy. This leads to the construction of a representation algorithm and the formalization of the binary number system, identifying cells with 1 (cube present) and 0 (empty). Participants then complete a conversion table of the first 16 decimal numbers and their binary equivalents, identifying the patterns that emerge.

Following this, basic addition and multiplication operations are introduced. Participants are asked to represent a number (usually 7) and multiply it by 2, then repeat the multiplication. They observe results like $111 \rightarrow 1110 \rightarrow 11100$ and reflect on why multiplying by 2 in binary equates to appending a zero. The connection to the decimal system is discussed, emphasizing how 'adding a zero' when multiplying by 10 also corresponds to multiplying by the base. To conclude, using the Thinking Classroom methodology [26], participants are invited to formulate and explain the divisibility rules for 2 and 3 in binary. This activity is designed to help internalize the importance of how numerical data is represented internally by computers, highlighting the role of binary representation, while offering a valuable bridge to the development of number sense in mathematics education.

3.5 Workshop 4: Data Simulation through Spreadsheets (1.5h)

This workshop introduces participants to data-oriented components of CT by engaging them in the simulation of mathematical objects using spreadsheet tools. The objective of this workshop is to develop participants' understanding of data-oriented components of CT through the simulation of probabilistic and geometric phenomena using spreadsheet tools. This workshop foregrounds the data pillar of the TPM. Through these activities, participants explore the intersection of stochastic reasoning and CT by engaging with modeling, simulation, data collection, data representation, and data analysis.

The first task involves the classic probability experiment of rolling two dice and calculating their sum. Participants generate two columns of random integers between 1 and 6 to simulate the outcome of rolling two dice and compute their sums in a third column. They then visualize the distribution of these sums for increasing sample sizes (10, 100, and 1000 trials), allowing for discussions on probabilistic bias, variability, and convergence toward the theoretical distribution.

The second task introduces a Monte Carlo method for estimating the value of π. Participants simulate points within a square and assess whether each point lies within the inscribed circle. By applying the corresponding ratio, they approximate π and reflect on the role of simulation in estimating mathematical constants.

3.6 Second Lecture: The Socio-Affective Dimension of Mathematics and CT (0.5h)

Scheduled between workshops, this session addresses the socio-affective dimension of mathematics [29] and its relevance within STEM education. Beyond technical and conceptual knowledge, the successful implementation of STEM and CT in the classroom also depends on how students relate emotionally to mathematics. Particular attention is given to affective challenges such as math anxiety and gender-related dynamics, which can significantly influence student engagement and participation in CT-related tasks. The session explores how these socio-affective factors are increasingly recognized as integral to understanding students' learning trajectories and to shaping inclusive, supportive educational environments within the broader STEM context.

3.7 Workshop 5: Unplugged CT with Bebras Tasks (1.5h)

This workshop focuses on the resolution and analysis of unplugged tasks from the Bebras initiative [30]. The objective of this workshop is to explore the pedagogical potential of unplugged Bebras tasks for developing CT in K–12 education, and to provide participants with tools for their didactic analysis and classroom implementation. This workshop is designed to address all components of CT, as the selected Bebras tasks are specifically curated to span the full range of CT components.

The workshop begins with a brief introduction to the nature and educational value of Bebras tasks, outlining their format, connections with computer science, CT, and mathematics. Several multilingual online repositories are presented as sources for rich, ready-to-use classroom materials.

Participants then engage with a curated selection of Bebras tasks from the HelloBebras! webpage [31] to become familiar with their structure and the types of reasoning they require. After solving the tasks, facilitators perform a detailed didactic analysis of one of them. This analysis includes a step-by-step solution, a discussion of the underlying informatics concepts, identification of CT components, relevant mathematical processes (in the sense of the NCTM [32]), and potential classroom implementation strategies through guiding and extension questions.

Following this, each in-service teachers group selects one of the previously solved tasks and conducts their own analysis, which is discussed with the facilitators during its development. The session concludes with a presentation of the HelloBebras repository containing 19 fully analyzed tasks across the K–12 spectrum, providing a valuable resource for future classroom application.

3.8 Workshop 6: Programming with Scratch (1.5h)

The objective of this workshop is to introduce participants to foundational concepts of imperative and object-oriented programming through the Scratch environment [33], emphasizing the pedagogical use of scaffolded models to support guided discovery and student autonomy. This workshop primarily supports the development of algorithms part of the TPM with its components (algorithms and procedures, debugging and parallelization), but also relies on decomposition and simulation.

The session begins with an overview of the software, explaining its design, functionalities, and use. Participants are familiarized with its interface and programming logic before being asked to develop a program based on a pre-established model. This model serves as a scaffold for understanding key programming structures such as sequences, loops, conditionals, events, and object interactions.

Throughout the session, teachers put emphasis on how pre-constructed models can support the teaching of Scratch and foster guided discovery learning. Participants are encouraged to explore and adapt these models, promoting an experiential understanding of programming logic. The workshop concludes with the design and implementation of a free-themed project, aiming to be more inclusive of participants' diverse interests. Projects may include video games, animated stories, artistic creations involving painting or music, or other personally meaningful expressions.

3.9 Final Lecture: Curriculum Integration of CT (1h)

This concluding lecture examines how CT is integrated into the Spanish curriculum and compares it with international curricular models. It analyzes different approaches adopted by various countries, whether as a standalone subject, embedded within mathematics, or treated as a transversal competence. Particular attention is paid to the interpretation of CT within the Spanish curriculum. The session concludes by synthesizing insights from all the workshops, offering a comprehensive reflection on curricular alignment and pedagogical practice.

4. Discussion

The training program has been fully implemented twice through the ADDIE model, while its workshops have also been conducted individually on numerous occasions to refine them. In these implementations, participants reported high levels of satisfaction and perceived improvements in both their understanding of computational thinking and its classroom applicability [34], supporting the robustness and adaptability of the proposal. The training activities span the full K–12 range and are explicitly designed to connect conceptual and pedagogical dimensions of CT, reinforcing its integration with core ideas in related areas like mathematics. The activities are carefully selected to enhance CT components in a balanced way. In Table 1 the degree to which each training activity addresses the different components of CT is visually represented.

As mentioned, teacher training in STEM domains is a critical lever for fostering students' engagement within STEM careers. In this context, CT emerges as a foundational gateway to STEM education. Central to this training is the transformation of teachers' prior conceptions, following a reflective model that first deconstructs existing knowledge and then enables the co-construction of new professional competence [24]. This process supports the development of professional competence, equipping teachers to implement inclusive and engaging CT practices and motivating them to implement these CT practices [34, 35]. By doing so, the training not only enhances teachers' instructional capacity but also positions them as key agents in cultivating future STEM vocations

Table 1. CT component distribution across workshops. Own elaboration.

	Lectures	Order	Stochastic	ED	Spreadsheet	Bebras	Scratch
Data collection	Grey	Grey	Dark	Grey	Dark	Grey	White
Data repr.	Grey	Grey	Dark	Dark	Dark	Dark	White
Data analysis	Grey	Grey	Dark	Grey	Dark	Grey	White
Simulation	Grey	White	White	White	Dark	Grey	Dark
Parallelization	Grey	Grey	White	White	White	Grey	Dark
Algorithms	Grey	Dark	White	Dark	White	Dark	Dark
Debugging	Grey	Grey	White	Grey	White	Grey	Dark
Decomposition	Grey	Dark	Dark	White	White	Grey	Dark
Generalization	Grey	Grey	Dark	Dark	White	Grey	White
Modeling	Grey	White	Dark	Dark	Dark	Grey	White
Abstraction	Grey	Dark	Dark	Dark	White	Dark	White
Patterns	Grey	Dark	Dark	Dark	White	Dark	White

Note: Dark cells indicate full engagement with a given component, grey cells represent partial engagement or likely emergence during the workshop, and white cells denote no direct engagement.

through the early integration of CT in the classroom.

Nevertheless, the study presents limitations concerning the evaluation of CT. The program does not currently include a dedicated workshop on assessment. Given that the training itself follows a competency-based model, and considering that Spanish and other international curriculums are adopting competency-based assessment, the integration of a specific workshop on assessment would constitute a valuable extension. Moreover, due to the absence of validated instruments, no pre–post measures of knowledge were collected, and evaluation relied on self-reported perceptions. Future work should therefore explore the integration of an assessment-focused workshop and the adoption of validated tools to more reliably capture the impact of the training.

5. Conclusions

The training program demonstrated effectiveness in enhancing teachers' understanding and application of CT in the classroom. Participants expressed high satisfaction and perceived meaningful improvements in both conceptual and pedagogical dimensions of CT. The program's design successfully integrated CT across the K–12 range maintaining balance across CT components. The iterative implementation of the workshops contributed to refining the program structure and ensuring adaptability to different educational contexts. The findings indicated that the program supports the development of teachers' professional competence and promotes the inclusion of CT as a core element of STEM education. These outcomes suggest that the proposed training model can serve as a scalable framework for fostering CT within teacher education.

Funding – This research was funded by: Grant FPU21/05874–Ministerio de Ciencia, Innovación y Universidades (J.J.S.). PID2024-155358NB-I00–Agencia Estatal de Investigación (J.J.S., L.J.R.-M., B.P.). GRUPIN-IDE/2024/000713–Agencia de Ciencia, Competitividad e Innovación Empresarial Asturiana SEKUENS (J.J.S., L.J.R.-M.).

References

1. J. M. Wing, Computational thinking, *Communications of the ACM*, **49**(3), pp. 33–35, 2006.
2. 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), pp. 48–54, 2011.
3. K. Brennan and M. Resnick, New frameworks for studying and assessing the development of computational thinking, *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada, 2012.
4. S. Grover and R. Pea, Computational thinking in K–12: A review of the state of the field, *Educational Researcher*, **42**(1), pp. 38–43, 2013.
5. V. J. Shute, C. Sun and J. Asbell-Clarke, Demystifying computational thinking, *Educational Research Review*, **22**, pp. 142–158, 2017.
6. P. Sengupta, J. S. Kinnebrew, S. Basu, G. Biswas and D. Clark, Integrating computational thinking with K–12 science education using agent-based computation: A theoretical framework, *Education and Information Technologies*, **18**(2), pp. 351–380, 2013.
7. L. Knie, B. Standl and S. Schwarzer, First experiences of integrating computational thinking into a blended learning in-service training program for STEM teachers, *Computers Applications in Engineering Education*, **30**(6), pp. 1423–1439, 2022.
8. S. L. Mason and P. J. Rich, Preparing elementary school teachers to teach computing, coding, and computational thinking, *Contemporary Issues in Technology and Teacher Education*, **19**(4), pp. 790–824, 2019.
9. R. H. Tai, C. Q. Liu, A. V. Maltese and X. T. Fan, Planning early for careers in science, *Science*, **312**(5777), pp. 1143–1144, 2006.
10. A. V. Maltese and R. H. Tai, Pipeline persistence: Examining the association of educational experiences with earning a STEM degree, *Journal of Science Education and Technology*, **20**(5), pp. 702–716, 2011.

11. P. R. Aschbacher, E. Li and E. J. Roth, Is science me? High school students' identities and aspirations in science, engineering, and medicine, *Journal of Research in Science Teaching*, **47**(5), pp. 564–582, 2010.
12. European Commission, *Science Education Now: A Renewed Pedagogy for the Future of Europe (Rocard Report)*, European Commission, Brussels, 2007.
13. European Commission, *She Figures 2021: Gender in Research and Innovation*, European Commission, Brussels, 2021.
14. S. Cheryan, S. A. Ziegler, A. K. Montoya and L. Jiang, Why are some STEM fields more gender balanced than others?, *Psychological Bulletin*, **143**(1), pp. 1–35, 2017.
15. M. Papastergiou, Are computer science and information technology still masculine fields? High school students' perceptions and career choices, *Computers & Education*, **51**(2), pp. 594–608, 2008.
16. European Schoolnet, *Computing Our Future: Computer Programming and Coding in Schools in Europe*, European Schoolnet, Brussels, 2015.
17. Informatics Europe and ACM Europe, *Informatics Education: Europe Cannot Afford to Miss It*, Report of the Joint Informatics Europe & ACM Europe Working Group on Informatics Education, 2013.
18. J. Voogt, P. Fisser, J. Good, P. Mishra and A. Yadav, Computational thinking in compulsory education: Towards an agenda, *Education and Information Technologies*, **20**(4), pp. 715–728, 2015.
19. S. Bocconi, A. Chioccariello, G. Dettori, A. Ferrari and K. Engelhardt, Review on pedagogical practices for computational thinking in teacher education: Characterizing an emerging field, *Informatics in Education*, **21**(2), pp. 235–251, 2022.
20. P. Hubwieser, M. Giannakos, M. Berges, T. Brinda, I. Diethelm, J. Magenheim, E. Jasute and colleagues, A global snapshot of computer science education in K–12 schools, *Proceedings of ITiCSE 2015 Working Group Reports*, Vilnius, Lithuania, pp. 65–83, 2015.
21. R. M. Branch, *Instructional Design: The ADDIE Approach*, Springer, New York, 2009.
22. P. Giadas, M. I. Pascual-Martín, L. Muñiz-Rodríguez, L. J. Rodríguez-Muñiz and L. C. Contreras, Perfiles de formadores de profesores de matemáticas en el contexto universitario español, *PNA*, **18**(5), pp. 439–466, 2024.
23. B. Palop, I. Díaz, L. J. Rodríguez-Muñiz and J. J. Santaengracia, Redefining computational thinking: A holistic framework and its implications for K–12 education, *Education and Information Technologies*, **30**, pp. 13385–13410, 2025.
24. Á. Alsina and I. Mulà, Advancing towards a transformational professional competence model through reflective learning and sustainability: The case of mathematics teacher education, *Sustainability*, **11**(15), p. 4039, 2019.
25. C. E. Hmelo-Silver, R. G. Duncan and C. A. Chinn, Scaffolding and achievement in problem-based and inquiry learning: A response to Kirschner, Sweller, and Clark (2006), *Educational Psychologist*, **42**(2), pp. 99–107, 2007.
26. P. Liljedahl, *Building Thinking Classrooms in Mathematics Grades K–12: 14 Teaching Practices for Enhancing Learning*, Corwin Press, Thousand Oaks, 2020.
27. Dollar Street, https://www.gapminder.org/dollar-street, Accessed 15 May 2025.
28. Exploding Dots, https://globalmathproject.org/exploding-dots/, Accessed 23 January 2025.
29. D. B. McLeod, Research on affect in mathematics education: A reconceptualization, in D. A. Grouws (ed), *Handbook of Research on Mathematics Teaching and Learning*, Macmillan, New York, pp. 575–596, 1992.
30. V. Dagiene and V. Dolgopolovas, Short tasks for scaffolding computational thinking by the global Bebras challenge, *Mathematics*, **10**(17), p. 3194, 2022.
31. HelloBebras!, https://educaixa.org/es/-/hellobebras, Accessed 9 February 2025.
32. National Council of Teachers of Mathematics, *Principles and Standards for School Mathematics*, NCTM, Reston, 2000.
33. Scratch, https://scratch.mit.edu/, Accessed 7 March 2025.
34. J. J. Santaengracia, B. Palop and L. J. Rodríguez-Muñiz, Percepciones del profesorado sobre pensamiento computacional. Estudio de una formación, in C. Jiménez-Gestal, Á. A. Magreñán, E. Badillo and P. Ivars (eds), *Investigación en Educación Matemática XXVI*, SEIEM, Badajoz, pp. 491–498, 2023.
35. J. J. Santaengracia, A. Aguilar-González, B. Palop and L. J. Rodríguez-Muñiz, Conocimiento especializado de estudiantes para maestro/a en una tarea sobre pensamiento computacional, in N. Adamuz-Povedano, E. Fernández-Ahumada, N. Climent and C. Jiménez-Gestal (eds), *Investigación en Educación Matemática XXVII*, SEIEM, Córdoba, pp. 489–496, 2024.

Juan José Santaengracia is a PhD candidate in Mathematics Education at the University of Oviedo, with a focus on mathematics education and computational thinking in primary education. His research explores the design of learning trajectories that integrate computational thinking and mathematics.

Luis J. Rodríguez-Muñiz is a full professor in Mathematics Education and coordinator of the Mathematics Education Research Group at the University of Oviedo. He is the first Vice President of the Royal Spanish Mathematical Society. His research focuses on Statistics education and mathematics teachers training.

Belén Palop is an associate professor at the Complutense University of Madrid (Spain), with a hybrid academic background in computer science and mathematics. Her research spans from computational geometry to mathematics education, with an intense focus on the integration of computational thinking into school curricula.