International Electronic Journal of Mathematics Education
2025, 20(2), em0812
e-ISSN: 1306-3030
https://www.iejme.com

MODESTUM

Research Article

OPEN ACCESS

Development and validation of learning objects aimed at mathematical instructions based on computational thinking

Jeanne Dobgenski [1,2]*, Maria Elisabette Brisola Brito Prado [1,3], Angélica da Fontoura Garcia Silva [3]

[1] Centro Universitário Anhanguera de São Paulo, São Paulo, BRAZIL
[2] Universidade do Minho, Braga, PORTUGAL
[3] Universidade Unopar Anhanguera, São Paulo, BRAZIL
***Corresponding Author:** jeanne.jd@gmail.com

Citation: Dobgenski, J., Brito Prado, M. E. B., & Garcia Silva, A. d. F. (2025). Development and validation of learning objects aimed at mathematical instructions based on computational thinking. *International Electronic Journal of Mathematics Education, 20*(2), em0812. https://doi.org/10.29333/iejme/15816

ARTICLE INFO

Received: 19 May 2024

Accepted: 29 Nov. 2024

ABSTRACT

By integrating computational thinking (CT) based mathematical reasoning with early childhood education through programming a dinosaur game in Scratch, this study aims to enhance the pedagogical skills of pre-service teachers by using learning objects (LOs). Also, explores frameworks for LOs construction to align with pedagogical goals, emphasizing the importance of students' mathematical outcomes, understanding of CT concepts, and the process used by teacher to help pupils solve the problem. This research utilizes an assessment grid tool for evaluating digital learning resources, with findings indicating the LOs high-quality as assessed by 57 pre-service teachers. Our study indicate how it is possible to create and use LOs for the pedagogical development of teachers, focusing on teaching methodologies and the application of mathematical knowledge in CT activities.

Keywords: learning object, computational thinking, pre-service teacher training, mathematics instruction, primary education, Scratch

INTRODUCTION

Academic researchers have been studying how it is possible to include computational thinking (CT) skills and concepts in mathematics learning (Benton et al., 2016; Gleasman & Kim, 2020; Silva et al., 2020; Ye et al., 2023). In this sense, it is important to investigate how students learn when facing CT-based mathematics instruction and how pre-service teachers can teach this subject (Dobgenski et al., 2022; Gleasman & Kim, 2020). This understanding is an important matter to Brazilian pre-service teachers of primary education because CT was introduced with mathematics in the nacional common core curriculum (NCCC) in 2018. The NCCC is a federal document that guides school curricula and provides for the development of CT through mathematics teaching (Ministério da Educação [Ministry of Education], 2018). It means that the board of education had to prepare its elementary teachers to teach math and develop CT skills and abilities in their students. However, few attempts have been made to introduce CT to teacher in the education in Brazil and less for pre-service teachers who will act in the first years of education.

Trying to fill this gap, our main doctoral research is focused on analyzing the reflections resulting from pre-service teachers during a CT skills training process. Therefore, to support teaching practices used during the teachers' training process, we decided to investigate how to teach pupils about the elementary mathematics content mentioned in the NCCC and integrate it with CT. We chose to introduce the concept of geometric orientation to children which is presented in a spiral learning pattern from 1st to 5th grade of primary education in Brazil. In this investigation, we involved two students, one from the fourth and another from the fifth grade, in a practical programming situation with Scratch aimed at developing the ideas of geometric orientation.

The literature review shows there is no consensus definition of CT, making it difficult to discuss how CT can be developed, measured, and assessed (Raabe et al., 2015; Rachmatullah et al., 2020; Valente, 2016). Thus, it is hard to find a methodology convergence in CT research that indicates what is important to its development. But the literature common sense observes the programming as a frequent tool used to develop CT in the students and Scratch is pointed as a way of coding (Benton et al., 2016; Gleasman & Kim, 2020; Hsu et al., 2018; Rachmatullah et al., 2020). For this reason, we proposed under the constructionism perspective (Papert, 1986) that these students built a game–which presupposes the creation of algorithms based on the notion of Cartesian plane–developing it in Scratch.

This study is a part of ongoing thesis by the first author.

It is in this investigation scenario that the student's learning process was analyzed in terms of the relevant strategies that were used by the teacher, who, through a constructionism approach, assisted the students to develop CT and to use their mathematical knowledge during a programming activity in Scratch.

Therefore, the investigation described in this article focused on creating learning objects (LOs) derived from the pupils' experience while exploring mathematical ideas when developing their CT skills during the activity of programming a dinosaur game with Scratch–to use these LO in a teachers' training process. In this sense, we sought some frameworks to guide the LO construction aiming to align with our pedagogical expectations about the understanding of how CT-based mathematical activities can be positively integrated into early childhood education. We argue that such LO should be

(1) based on the students' mathematical learning outcomes,

(2) their understanding of CT skills and concepts, and

(3) the process used by the teacher to help pupils solve the problem.

These three conceptual topics was highlighted and evidenced by Ye et al. (2023), Selby and Woollard (2013) combined with Rachmatullah et al. (2020), and by Silva et al. (2020) frameworks, respectively.

This paper provides a foundational theoretical review about LO, shows the development and validation of the LO as a tool to teach CT-based mathematical instructions to pre-service teachers.

THEORETICAL FRAMEWORK

To establish our theoretical framework initially we investigate concepts such as LOs, constructionism, problem-solving and frameworks. This exploration was driven by our aim to establish meaningful connections among these elements.

Learning Objects

LOs refer to discrete, self-contained digital resources that are designed to facilitate learning and instruction on a specific topic. Wiley (2002, p. 4) explains that "learning objects are generally understood to be digital entities deliverable over the Internet" but defines them as "any digital resource that can be reused to support learning" (p. 6) as those that can directly interfere with learning as "digital images or photos, live data feeds (like stock tickers), live or prerecorded video or audio snippets, small bits of text, animations, and smaller Web-delivered applications" (p. 6). These resources are intended to be easily reusable, and adaptable for various educational contexts, making them valuable tools for educators to enhance teaching and learning experiences. LOs typically encapsulate a single learning objective or concept, which can be a small piece of knowledge, a skill, or a complete learning activity.

The concept of LOs in education shares some similarities with the object-oriented paradigm in computer science (Wiley, 2002). Understanding the relationship between LOs and the object-oriented paradigm in computer science provides valuable insights into how the principles of object-oriented programming align with the design and use of LOs. The object-oriented paradigm, commonly used in software development, shares certain conceptual similarities with creating and utilizing LOs. Both concepts revolve around the idea of abstraction, encapsulation, modularity, and reusability, although they are applied in different contexts.

Abstraction

Object-oriented programming relies on abstraction to simplify complex systems by focusing on relevant details while hiding unnecessary complexities. This enhances the understandability and maintainability of the codebase (Deitel & Deitel, 2005; Sebesta, 2002). In education, LOs abstract specific educational content or activities. They provide a focused learning experience on a particular topic while abstracting the underlying complexities of instructional design. This enables educators to present content in a clear and structured manner, similar to how object-oriented programming abstracts complex operations into manageable objects.

Encapsulation

In the object-oriented paradigm of computer science, encapsulation refers to the grouping of data and the methods (functions) that operate on that data into a single unit called an "object". This encapsulation ensures that the internal workings of an object are hidden from the outside world, promoting data integrity and abstraction (Deitel & Deitel, 2005; Sebesta, 2002). Similarly, in the context of LOs, encapsulation involves packaging a specific piece of educational content, such as a learning module, it could be an interactive activity or assessment for example, into a self-contained unit. This encapsulation ensures that the educational content is distinct, well-defined, and can be used independently or combined with other LOs.

Modularity

LOs are designed to be modular, meaning they can stand alone as independent units of learning. Object-oriented programming emphasizes breaking down complex systems into smaller, manageable modules (objects). These modules can be developed and tested independently before being integrated into the larger system. This approach enhances code organization, maintenance, and reusability (Deitel & Deitel, 2005; Sebesta, 2002). In education, LOs exhibit modularity by breaking down complex topics or learning goals into discrete units. Each LO focuses on a single learning objective or concept, making it easier to organize, reuse, and adapt instructional content. As occurs with object-oriented modules that can be reused in different software projects, the same goes with LOs that can also be reused across various educational contexts, because their modularity allows educators to mix and match LOs to create customized learning experiences that align with their teaching goals and the needs of their learners.

Reusability

Object-oriented programming encourages the creation of reusable code through the development of classes and objects. These reusable components reduce redundancy, save development time, and enhance the consistency and reliability of software systems (Deitel & Deitel, 2005; Sebesta, 2002). Similarly, LOs are designed with reusability in mind. They can be reused across different courses, modules, or educational contexts, reducing the effort required to create new instructional materials. Just as a class in object-oriented programming can be instantiated multiple times, a LO can be reused across various educational scenarios.

Constructionism

Constructionism, as a learning theory, emphasizes the importance of active problem-solving and hands-on learning experiences (Papert, 1986). At its core, constructionism highlights the idea that individuals learn best when they are actively engaged in the process of constructing knowledge rather than passively receiving it. Papert (1986) believed that children should have access to concrete materials and tools that allow them to explore mathematical concepts through direct manipulation. This can include physical objects like blocks, measuring tools, and computers. In this sense, Papert (1986) was against computer-aided-instruction because he did not believe in a computer teaching children, but he intended the children to program the computer, learning how to “talk” with the computer (p. 17-18). Papert (1986) also said that “learning to communicate with a computer can change the way other learning happens [...] and with computers with which children like to communicate when this communication occurs, children learn mathematics as a living language” (Papert, 1986, p. 18), we can say that computers provide a dynamic learning environment in which children can experiment, make mistakes, and iterate. This reflects the idea of “learning by doing” in which children actively build their understanding of mathematics through practical experiences, making learning mathematics more meaningful.

He was a pioneer in the use of computers for educational purposes and developed the Logo programming language, which allowed children to create and manipulate geometric shapes using code. This hands-on experience with coding and geometry helped them understand mathematical concepts in a practical way. But Jonassen (2007, p. 176) pointed two significant limitations of Logo-based microworlds presented very constrained and circumscribed problems that engaged a limited set of skills and although Logo is syntactically simply language, it still requires several months of practice to develop skills sufficient for easily creating microworlds. However, Jonassen (2007) agreed with the idea of the microworld by considering that spaces of exploration and experimentation as problems are indeed a powerful idea (p. 176). Today, similar educational tools and software can be used to integrate technology into mathematics education such as Scratch, which overcomes the limitations of Logo’s microworlds discussed by Jonassen (2007).

Silva et al. (2020, p. 8) state the “constructionism, CT, and problem-solving are closely related” and highlight that “constructionism, in general, is a learning theory involving problem-solving that usually offers ways to explore computer programming in mathematics learning”. Computer programming becomes a helpful tool for exploring mathematical ideas, enabling students to visualize and manipulate mathematical models in ways that traditional teaching methods may not offer.

Problem-Solving

Polya (1945, p. 221) suggested that “solving problems is a fundamental human activity. In fact, the greater part of our conscious thinking is concerned with problems” he added that “to solve a problem is, essentially, to find the connection between the data and the unknown” (Polya, 1945, p. 182) and that “we need a certain amount of previously acquired knowledge” (Polya, 1945, p. 150) to solve a problem. Solving problems is a useful and necessary challenge for learning math well because it’s essential to completely understand its context, constraints, and underlying factors (Säfström et al., 2023). For us, the importance of problem-solving goes beyond finding an adequate or optimal solution to the problem addressed but is focused on the heuristics or process that involves finding the solution.

When we think about math problems, we may see it as a puzzle where the student does not have a ready-made solution. They must figure out the important parts using their own math thinking, that is, they need to use their previous knowledge as said Polya (1945). This decomposition can be understood both for the problem, breaking it down into smaller problems, and for the process of finding the solution to the problem. By dividing this process into phases, it is possible to check which aspects are most confusing or difficult for students when solving a problem. Säfström et al. (2023) developed a diagnostic framework for the difficulties primary and secondary students faced when constructed their own solutions to mathematical problems, focused on diagnosing of students’ specific reasoning difficulties. These authors where based on Schoenfeld’s (1985) six-phase model to solve a problem, which one was inspired by four-phase model of Polya (1945), as shown in **Figure 1**.

Schoenfeld’s (1985) model expands on Polya’s (1945) process by emphasizing analysis and exploration of the problem before planning the solution. It also places importance on metacognition, encouraging students to reflect on their own thinking during the problem-solving process (Schoenfeld, 1985). This involves breaking down the problem into smaller components or sub-problems, but to do this is necessary to understand the problem deeply. The problem-solver can generate multiple solutions and evaluate and compare them to determine which one is the most effective, efficient, and suitable for the given situation. After implementing the solution, it’s essential to assess whether it effectively solves the problem. If not, adjustments and refinements may be necessary. Problem-solving is often an iterative process. If the initial solution does not work or if new challenges arise, the problem-solver may need to revisit earlier steps to find a better solution. This is related to with we discussed about constructionism and reinforced what Silva et al. (2020) said about constructionism, CT, and problem-solving share a close relationship. In the realm of CT and mathematics instruction, it plays a central role in developing critical thinking skills. Valente

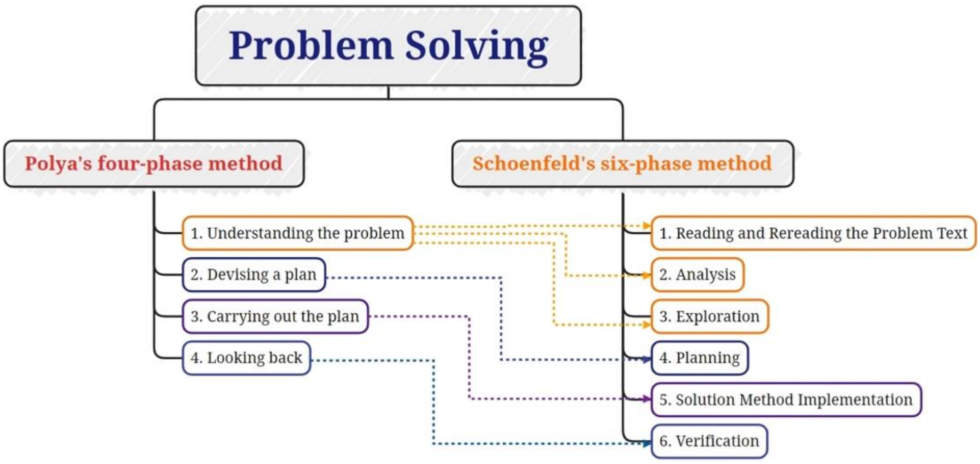

Figure 1. Polya's (a945) and Schoenfeld's (1985) phases models to problem-solving

(2003) highlighted that utilizing the learner's understanding of concepts related to the problem, including those pertaining to the computer, software, and the effective application of strategies, help them to articulate and outline the steps involved in solving the problem, which are considered as a set of rules, and they are called routine tasks in software development. Math problems and routine tasks make students think in different ways, helping them learn at different levels (Säfström et al., 2023). Papert (1986) also reinforces this relationship when suggested that Logo and the turtle geometry provides excellent opportunities to practice the subdivision of the difficulties–when explain why this tool is useful to apply the Polya's (1945) solving problem method because when facing a problem, we should mentally review a set of heuristic questions, asking: can this problem be decomposed into simpler questions? Is there any way to associate it with another problem whose solution is already known to me? (Papert, 1986, p. 88). Furthermore, Papert (1986) states that the Turtle's geometry introduces a new dimension to Polya's (1945) concept: "To solve a problem, look for something similar that you already know and understand." (Papert, 1986, p. 88). Even though the idea is abstract, the Turtle's geometry transforms it into a concrete and methodical principle.

This is the strength of coding solutions at the computer using tools such as Scratch, for example. The students may see what is happening, identify the errors and trying new ways to find the solution. But it could not be effective if they just used trial and error, what suggest the importance of being well guided by a teacher to find their own answers to the heuristic questions to solve a problem as Polya (1945) and Schoenfeld (1985) showed.

Pedagogical Validation Frameworks

To support the creation of LOs, we use three distinct frameworks: one to validate students' mathematical knowledge (acquired or used); other to validate the pedagogical method used by the teacher and another, together with the definition of CT, to validate the skills and concepts of CT used by students.

Computational thinking definition, concepts, and skills

In this research we consider CT definition by Selby and Woollard (2013, p. 5) which "is a focused approach to problem-solving, incorporating thought processes that utilize abstraction, decomposition, algorithmic design, evaluation, and generalizations". This approach combines the four CT pillars: abstraction, algorithms, decomposition, and pattern recognition (BBC Learning, n. d.; Raabe et al., 2020, p. 19), which, according to BBC Learning (n. d.), help solve complex problems.

Rachmatullah et al. (2020) developed a concept inventories (CI) using a block-based programming language with 24 multiple-choices items guided by knowledge, skills and abilities focused on programming/CT concepts as variables, loops, conditionals, and algorithms. In our research context we consider these CT concepts used by Rachmatullah et al. (2020), since we are interested in use this CI to assess the CT knowledge of the pre-service teachers before and after the training, in which we use the LO described in this paper.

Mathematical knowledge

To identify how pupils mobilized their elementary mathematical knowledge and how they developed new ones while programing their own game's code, we used the framework by Ye et al. (2023). Ye et al. (2023) conducted a systematic review on the integration of CT in K-12 mathematics education with a focus on CT-based mathematics instruction and students learning under such instruction. They used Web of Science as a database searching studies published from 2006 to 2021. They selected 24 articles to provide illustrations of CT-based mathematics instruction and related student learning, analyzed according to education levels and contexts, programming tools, learning outcomes in CT and mathematics, and the mutual relationship between CT and mathematics learning. They found CT-based mathematics learning entails an interactive and cyclical process of reasoning mathematically and reasoning computationally, which can occur when:

(1) applying mathematics to construct CT artefacts,

(2) applying mathematics to anticipate and interpret CT outputs, and

(3) generating new mathematical knowledge in parallel with the development of CT.

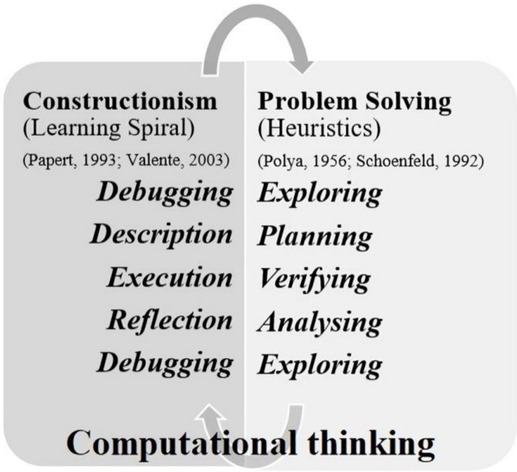

Figure 2. Constructionism and problem-solving components by Silva et al. (2020)

Pedagogical method

Silva et al. (2020) investigated students' CT in mathematics education analyzing teaching experiments and used a framework based on constructionism and problem-solving. For the constructionism approaching they considered user-computer interaction as description-execution-reflection-debugging-description … as a spiral actions. For the problem-solving they suggest heuristics components as exploring, planning, verifying, and analyzing, as shown by **Figure 2**.

They consider two conceptualizations' processes of learning, one regarding actions within a computer environment and another emphasizing heuristic or discovering processes in problem-solving (Silva et al., 2020). They investigated how elementary school and university students approached a mathematical task in a programming environment. It was observed that both groups used constructionism practices and enhanced learning by using computer programming to create and manipulate dynamic models of mathematical relationships through the elaboration and testing of conjectures, experimentation with technology and thinking with media. A significant difference emerged in the final, more complex task, where university students demonstrated more solution attempts and a greater variety of heuristic components compared to 6th year students, evidencing a qualitative difference in the ways of thinking mathematically and acting computationally between the two groups (Silva et al., 2020).

The fact that undergraduate students think in a more sophisticated way in terms of heuristics may be linked to the group's previous experience, that is, their previous knowledge may be broader than that of the other group.

METHOD

When we structured the doctoral research project, we did not expect to use LO in the CT-based mathematical instruction training process for pre-service teachers, but we were interested in acquiring experience and understanding about how early years students could explore mathematical ideas when developing coding activities. After the pupils' intervention and analyzing the data, we visualized how powerful would be to use the pupils' Scratch codes and videos to train pre-service teachers, in terms of our main doctoral research goal. Transforming these codes and videos into LO is a strategic approach that holds great potential for enhancing teachers' teaching proficiency.

This study seeks to address the research question "How can the development of LOs be designed to improve the pedagogical skills of future teachers, enabling them to understand how to teach effectively and facilitate students' learning about CT based on mathematics instructions?".

Participants and Data Collection to Create Learning Objects

We analyzed the activity developed by two primary school students from the São Paulo State in Brazil. The 9-year-old student was in grade-4 and had face-to-face meetings. We referred to him as "F student". The other student was 10 years old and was in grade-5 and had remote meetings. We referred to him as "R student". Both had 1-hour session per week for 2 months in 2021, to carry out programming activities in Scratch. The goal was to produce a game like Google Chrome's dinosaur game.

The dinosaur game[1] consists of making a dinosaur jump some obstacles shown on the screen. These obstacles are always in a horizontal direction and whoever is playing needs to press a key to make it jump–the dinosaur will move in a vertical direction. The game will end if the dinosaur touches the obstacle.

Data were collected from students' protocols, training's video recordings, and Scratch's programming code developed. This research is registered with the Research Ethics Committee (Comitê de Ética em Pesquisa–CEP) under Certificate of Presentation of Ethical Appreciation (Certificado de Apresentação de Apreciação Ética–CAEE) number 40924020.2.0000.5493, opinion number 4.481.578.

[1] To play the game, type chrome://dino/ in the Chrome browser.

Table 1. LO's technical characteristics

LO characteristics	Actor's jumping		Obstacle's movement	
	OA1F (F student)	**OA1R (R student)**	**OA2F (F student)**	**OA2R (R student)**
Format	Video + table	Video	Video + table	Video
Video time	9 m 19 sec	16 m 59 sec	12 m 13 sec	11 m 31 sec
Description	F student demonstrates how to program the actor's jump by presenting the reasoning he used to solve the problem.	R student demonstrates how to program the actor's jump by presenting the reasoning he used to solve the problem.	F student shows how to program the obstacle's movement by presenting the reasoning he used to solve the problem.	R student shows how to program the obstacle's movement by presenting the reasoning he used to solve the problem.
Purpose	It aims to show training teachers how is possible to use: (1) pupil's mathematical knowledge in a programming activity to develop their CT skills, (2) programming activities to generate new math knowledge, and (3) the constructionism and problem-solving approaches to conduct the pupils learning.			
Abstraction (focused on specific activity)	Pupils learning: focus on movement in *y* axis: • Switch costume • Change *y* axis parameter		Pupils learning: focus on movement in *x* axis: • Negative steps • Use of the edge sensor	
Encapsulation (packing a specific piece of educational content)	Pedagogical content knowledge (PCK) (Ball et al., 2008) to teacher: constructionism and problem-solving approaches used to conduct pupils learning.			
Modularity (breaking down complex topics or *learning goals*)	PCK: Problem-solving process described by students–Knowledge of content and students (KCS) from Ball et al. (2008, p. 403); constructionism approach followed by teacher–Knowledge of content and teacher from Ball et al. (2008, p. 403), and Pupils' mathematical knowledge applied to solve the problem–KCS from Ball et al. (2008, p. 403).			
Reusability (reused across different courses, modules, or educational context)	Teacher's training process purposes: (1) to facilitate the understanding about CT skills and concepts, (2) to show math activity developed with coding in Scratch, (3) to show how teacher can lead a problem-solving process with students, and (4) to compare the students learning in a face-to-face or remote class, and others.			

Table 2. LO's approaches

LO approaches	Actor's jumping		Obstacle's movement	
	OA1F (F student)	**OA1R (R student)**	**OA2F (F student)**	**OA2R (R student)**
Constructionism approach (teacher's conduct and questions)	• Set of questions used by the teacher to simulate the actors walking in the same place (switch the costumes) • Set of questions used by the teacher to help students perceive how a vertical movement could be implemented. • After thinking about the questions, students coded their ideas what they understood.		• Set of questions used by the teacher to remind students how the sensor works. • Set of questions used by the teacher to help students perceive how a horizontal movement to the left could be implemented. • After thinking about the questions, students coded their ideas what they understood.	
Problem-solving (Schoenfeld's, 1985 and Polya's, 1945 phases)	Problem-solving phases overlap as students think, strategize, code, and assess solutions during teacher's discussions.			
Math knowledge (Ye et al.'s, 2023 framework)	Pupils' learning in CT-based math instructions: (1) applying mathematical knowledge to construct CT artefacts, (2) applying mathematical knowledge to anticipate and interpret CT artefacts, and (3) generating new mathematical knowledge in parallel with CT development.			
CT concepts (Rachmatullah et al.'s, 2020 framework)	Data, conditional, loop, and algorithm.			
CT skills (CT definition from Selby and Woollard, 2013)	Abstraction, decomposition, algorithm design, pattern recognition, evaluation, and generalization.			

Learning Object Creation

When we are structuring a course, discipline, or specific content area, there is flexibility to employ diverse sizes and arrangements of these LOs. In this sense, there were created four LO intended to be used to instruct pre-service teachers about how the mathematical knowledge of students rises in a programming activity and help them improve their CT skills when conducted by teachers who use constructionism. The LO created from F student activities consists in a video, where the computer screen was recorded diagonally during the intervention, and a table that shows a piece of a talk between teacher and F student to highlight the main math topics they discussed including the Scratch coding. The LO created from R student activities consists only of a video. We did not add a table because the computer screen was completely recorded during the intervention, allowing the viewer to see everything clearly.

To create a LO video type we need first record it, then decompose it into coherent parts that can be cut, and then edit it. The four videos created were edited to students' anonymization and to show only the situations that imply the use or generating math knowledge, considering problem-solving, constructionism, CT skills and concepts. These are the pedagogical content we established as the specific piece of educational content to be encapsulated in a LO. After all, deciding the granularity of a LO is not an easy task, so we support ourselves in Wiley (2002) to determine the piece of educational content when he says that "from an instructional point of view, alternatively, the decision between how much or how little to include in a LO can be seen as a problem scope" (Wiley, 2002, p. 10). Furthermore, we rely on Ball et al. (2008) and their pedagogical theory to classify the pedagogical content, making clear the modularity of LO. **Table 1** summarizes the technical characteristics of LO and **Table 2** shows the approaches used to develop them.

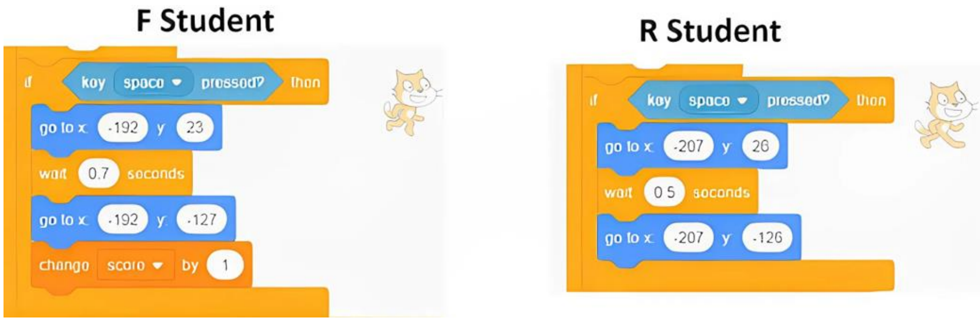

Figure 3. F and R students' coding to move the actor vertically–presented in OA1F and OA1R (Source: Field study)

Data Analysis

We analyzed the students' mathematical learning data outcomes by Ye et al. (2023) framework; their coding was analyzed to highlight the CT skills and concepts used considering Selby and Woollard (2013) CT definition and by Rachmatullah et al. (2020) framework which were combined in this research. Finally, we used Silva et al. (2020) framework to highlight the constructionism process used to solve the problem, which was the pupils' own version of the game.

To assess the quality of the LO[2] created we used a tool developed by El Mhouti et al. (2013) with 20 questions that facilitates two distinct types of evaluations. Firstly, it supports a comprehensive assessment of the quality of digital learning resources in education using a rating scale, providing a broad overview of the assessed products' quality. Secondly, these 20 questions allow for a detailed evaluation of each specific section (academic, pedagogical, didactic, and technical), offering insights into the quality of individual aspects.

The four LO were evaluated by 57 pre-service teachers during the CT-based math training process occurred in October and November of 2023, which took place in a university from Portugal. The pre-service teachers training is also registered with the Research Ethics Committee under Certificate of Presentation of Ethical Appreciation number 54311321.2.0000.5493, opinion number 5.294.198.

RESULTS, ANALYSIS AND DISCUSSION

Besides, to analyze the student's learning process was based on constructionism and problem-solving, we highlighted how mathematical knowledge emerged in this process of coding a game. These characteristics must be inner to the LO created to achieve our expectations when using these LO during the process of training pre-service teachers. Therefore, an illustrative example was given for each aspect of the Ye et al. (2023, p. 17) framework, to demonstrate how students' mathematical reasoning emerges and is facilitated from computational contexts.

We presented in this section the results, the analysis and the discussion of which math aspects were incorporated in the LO, then we highlighted the constructionism and problem-solving approaches, ending with the results from assessment grid of the quality of a digital learning resource (El Mhouti et al., 2013) responded by pre-service teachers about the LO functionality.

Applying Mathematical Knowledge to Anticipate and Interpret CT Artefacts

Ye et al. (2023, p. 17) showed how the students draw upon their mathematics to anticipate and interpret CT outputs and how these processes are tied to their debugging practices. They found other three themes which relate directly to debugging practices, students can

(a) recognize a buggy program by comparing the outputs with the anticipated outcomes,

(b) fix the bugs by reconsidering the mathematical relationships relevant to the situation, or

(c) fix the bugs by modeling the mathematical behavior underlying the program.

To illustrate this situation, we choose to explore the following problem: the students had to program the actor's jump (the actor that stands for the dinosaur in Google's game). This actor's jump was encapsulated at LO called OA1F and OA1R described in **Table 1**.

Figure 3 shows the final code implemented by the pupils to move the actor vertically, or to simulate the actor's jump. It is noticeable how similar the codes from two students are–this was already discussed in another article (Dobgenski et al., 2024).

To implement the actor's jump the pupils had to understand the actor's movement must be in the vertical direction what means they had to realize they needed to change the y axis parameter. To verify if the parameter needed to change to a higher or a smaller number, they changed the actor position in the stage. In the code pupils used the "go to x y" command, but they modified only the y parameter.

---

[2] The LO created and research dataset are available in the University of Minho Repositorium at https://doi.org/10.34622/datarepositorium/WIDQQF

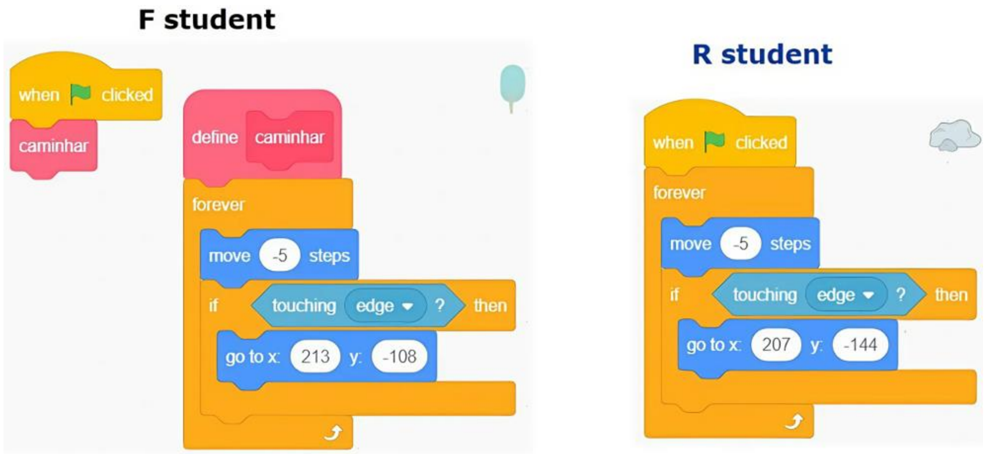

Figure 4. F and R students' coding to move horizontally the obstacle from right to left side of the stage–presented in OA2F and OA2R (Source: Field study)

To implement the actor's jump, they used a sensor that recognizes when a key was pressed. In this case the following steps occurred inside of the conditional command if

(1) the first command changed the y value to a higher number,

(2) the second command made the actor wait a specific time to return to its original position, which was made with

(3) the third command to set up the y value to its original value, what means the actor's position on "the ground".

But it did not work at the first attempt because to make the jump work precisely, they had to verify how high is the obstacle, i.e., its vertical size. With this information the pupils had to calculate the right y parameter to the actor's jump be high enough to not touch the obstacle. In this sense, the pupils had to recognize when the program didn't work by comparing the outputs with the anticipated outcomes and fix the bugs by reconsidering the mathematical relationships relevant to the situation, exactly as described by Ye et al. (2023, p. 21).

This understanding reveals some CT concepts as conditional, data (to set the *y* parameter) and algorithm. The main mathematical knowledge they mobilized in this task were: comparison of numbers, subtraction, addition, percentage, direction, and orientation. It is worth saying these students did not know about Cartesian planes before.

Applying Mathematical Knowledge to Construct CT Artefacts

In this aspects Ye et al. (2023) discussed how students apply their mathematical knowledge to construct CT artefacts, including the challenges they may experience during the application (p. 18).

There are also multiple examples that could be used to represent this situation from the coding process made by the pupils. To further demonstrate it we chose to explore the following problem: the students had to program the obstacle's movement always moving from right to left of scenario side. This obstacle's movement was encapsulated at LO called OA2F and OA2R described in **Table 1**.

Figure 4 shows the final code implemented by the pupils to move the obstacle from right to the left edge of the stage. Again, both students' codes are pretty similar.

To implement the horizontal obstacle's movement from the right to the left side of the scenario, they had to understand the starting position of the obstacle and its movement to the left side to set the parameter, that represents the number of steps, to a negative number. That means that they had to learn in this process how to calculate this parameter by subtracting the length of the steps from *x* values. To understand that they had to visualize when the obstacle was at the right side his position in *x* has a greater value than x value when the obstacle was located at the left side, so if they would subtract a step's length from the initial position the actor will move backwards.

But the obstacle must arrive on the left side, so they had to realize this movement had to be continued until the obstacle reached the left side and not just give one step backwards. Another perception they had was about the size of the steps, if it was a small value than the obstacle moves slowly, if it was a great value the obstacle runs. The parameter was set to "-5" by both pupils after testing the code (**Figure 4**).

Finally, the pupils had to identify if the obstacle touched the left side and if it was true had to set its new position at the beginning of the right side and repeat these operations again, until the game is over. This understanding reveals some CT concepts as loop, conditional, data (to set the x parameter) and algorithm what means to present the right sequence of the commands.

The main mathematical knowledge they mobilized in this task were: comparison of numbers, subtraction, negative numbers, successive subtraction by repetition, unit (step), direction and orientation. It is worth saying these students did not know about negative numbers before.

Generating New Mathematical Knowledge in Parallel With CT Development

Ye et al. (2023) also explain the reciprocal influence in the sense that students' engagement with CT-math integrated tasks are generative to growth in their mathematical knowledge and identified "three avenues through which new mathematical knowledge emerges in CT-based mathematics activities" (Ye et al., 2023, p. 21-22):

1. students can construct mathematical ideas and relationships by reflecting on CT outputs,
2. new mathematical ideas are constructed is reflecting on the programming processes and code features,
3. CT-based mathematics instruction motivates a new form of mathematical knowledge that is represented by computational languages and unique to CT contexts.

We observed some of these conditions in the two problems stated before: obstacle's movement and actor's jump. When pupils were engaged in the activity of programming the horizontal movement of an obstacle in the right-to-left scenario, they need to use a negative number to represent this direction of movement introduced them to a new mathematical concept which they had not learned yet. Reflection on the program's output, that is, the observation that the use of a negative number of results in the desired movement of the actor, leads to the construction of a new mathematical idea: that of negative numbers and their applications.

When the students understood the vertical movement of the actor, to simulate a jump, also constructed new mathematical ideas and relationships about Cartesian plane by reflecting on CT outputs related with the actors' movements in the Scratch stage. They understood the combined value of x and y will indicate the actor location, x for horizontal and y for vertical (Dobgenski & Garcia Silva, 2022). The student was applying mathematical concepts–specifically, the use of coordinates to position and move an object in a two-dimensional space–in a manner that is intrinsic to programming and CT contexts. Even without prior formal knowledge of the Cartesian plane, the students were exploring these mathematical concepts through computational language and logic.

Problem-Solving and Constructionism

Following Silva et al. (2020) framework the pedagogical approach used by teacher was intended to make the pupils think about the problem they were facing at that time. In this learning process the teacher asked how the motion occurred in the original game because they needed to understand it to develop their own version and find a way to program these movements in their algorithms. The teacher's questions were about

(1) if it was the actor (dinosaur) that was running through the scenario or not,

(2) about the actor's physical appearance when he was running and when he touched the obstacle,

(3) about the scenario appearance and so on.

All the questions asked by the teacher were directed to arouses the pupil's attention to the game details. The teacher's targeted questions about the game details are crucial in moving students from a superficial grasp of the game to a deeper strategic comprehension. This process aligns with Polya's (1945) emphasis on understanding the problem but extends it by actively engaging students in the problem's specifics, which is central to Schoenfeld's (1985) methodological approach.

With the constructionism perspective in mind, the teacher's questions and acts were done to guide them to think about their choices when coding the game, and not to give them the answers. The students had to reflect on each part of the game and understand how they could do it, including the mathematical knowledge they had to use as they were not explicitly guided about it. It is worth noting that our aim was for the students to discover their own way to solve the problem.

Figure 5 shows the situation described before, but from the perspective of the frameworks proposed by Silva et al. (2020) and Rachmatullah et al. (2020). At the first column is presented an example from student's game coding which problem was to develop the obstacle's movement in the game–as mentioned before. Both frameworks are presented in the next columns, the 2nd to 4th are about Silva et al. (2020) framework that explores the phases to solve a problem and the last column shows the CT concepts used in Rachmatullah et al. (2020) framework.

At the second column we presented a chat between the teacher and the students, where is possible notice how the students, debugged, explored, reflected, analyzed when they were facing that problem. This structure provides an easy way to understand how the pupils described their misunderstandings and which steps they planned to solve the problem. The execution and verifying phases were highlighted with some prints of the results presented in Scratch from the solutions they implemented. In the last column, there are the CT concepts from Rachmatullah framework. Both students used the same CT concepts.

R student

The R student's conduct described in **Figure 5** was extracted from the LO OA2R. Analyzing the chat between the teacher and the R student, **Figure 5** table, we notice that was inside this situation that the debugging process was initially prompted by the teacher's question about why the movement of the rock didn't work as expected–the R student used a rock as the obstacle in his game version. This question sets the student for exploration, pushing him to examine the code and its outcomes more closely. The student's attempt to reuse the code for the opposite movement without adjusting for directional values indicates an initial lack of understanding of how coordinate systems work in programming. The exploration process is evident when the teacher guides the student to observe and articulate the difference in the x values associated with the rock's movement. This observational step is important in debugging, as it helps in identifying where the logic deviates from the expected.

Reflection is motivated by the teacher's questions, which encourage the student to think critically about the relationship between the direction of movement and the numerical values that represent the positions on the x-axis. By reflecting on the

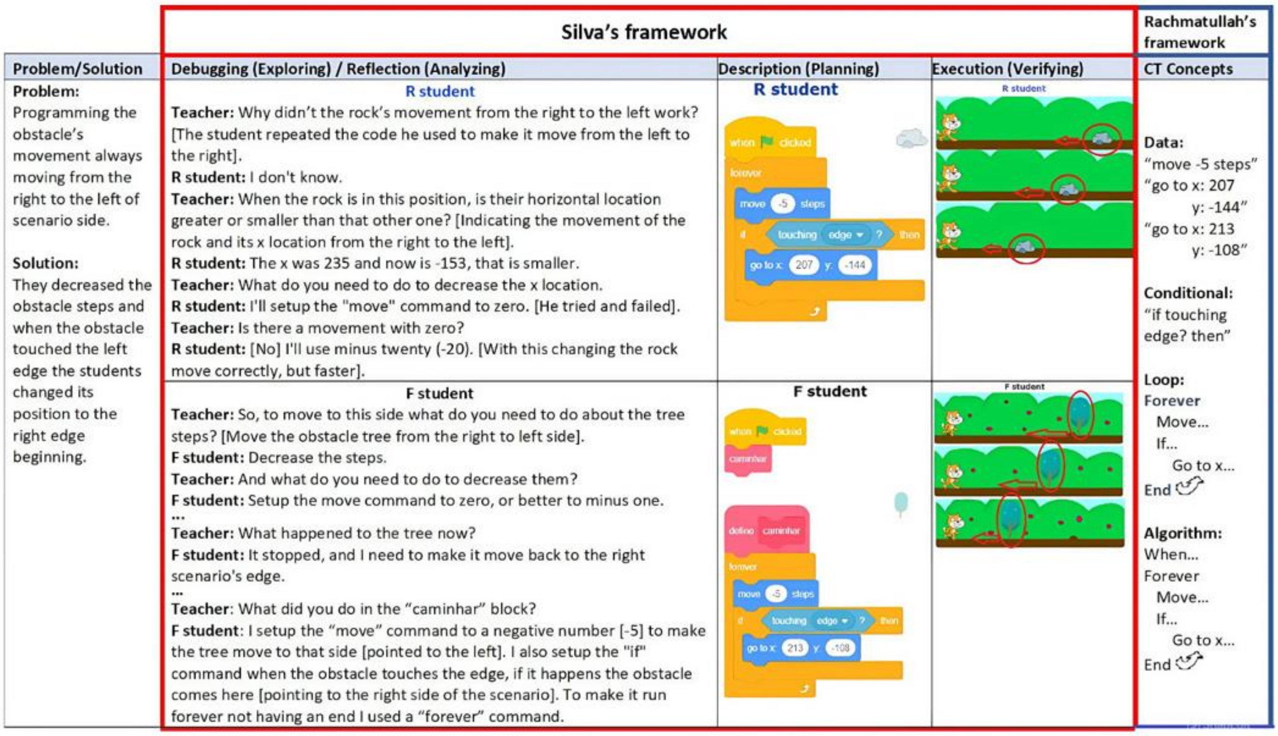

Figure 5. R and F student's codes interpreted by Silva et al. (2020) and Rachmatullah et al. (2020) frameworks

teacher's question regarding the comparison of *x* values, the student can recognize how negative numbers are used to represent direction within a coordinate system. The analyzing process becomes more apparent as the student attempts to rectify the issue by setting the "move" command to zero, which leads to a failed attempt. This failure prompts further analysis and a deeper understanding of the problem at hand. The teacher's follow-up question about the effect of a "move" command with zero value serves as a critical moment for the student to analyze why their attempted solution was ineffective. When the student suggests using "-20" to achieve the desired movement, it shows an application of their reflective and analytical process, adjusting their approach based on the understanding that negative numbers can represent movement in the opposite direction.

This dialogue highlights several important aspects of learning through a constructionism strategy as described in the sequence.

Active learning: The student is actively engaged in solving problems through experimentation and observation, which are the keys to effective learning in CT and programming (Wing, 2006).

Math conceptual understanding: Through debugging and teacher-guided reflection, the student moves beyond rote memorization to a better understanding of how numbers can represent direction and movement in programming stablishing relationships with Cartesian plane. This situation is like that described by Ye et al. (2023) about a study where students "used the command *repeat* to model the relationship between total distance and step size […] constructing a multiplicative relationship between the quantities involved" (Ye et al., 2023, p. 18).

Problem-solving skills: The process of identifying the problem, hypothesizing solutions, testing, and revising approaches enhances the student's problem-solving skills (Polya, 1945; Schoenfeld, 1985).

Teacher's role: The teacher plays a crucial role in guiding the student's thought process without directly providing the answer, facilitating a learning experience that encourages independence and resilience in facing challenges. Polya (1945) had explained the teacher role:

> One of the most important tasks of the teacher is to help his students. This task is not quite easy; it demands time, practice, devotion, and sound principles. The student should acquire as much experience of independent work as possible. But if he is left alone with his problem without any help or with insufficient help, he may make no progress at all. If the teacher helps too much, nothing is left to the student. The teacher should help, but not too much and not too little, so that the student shall have a reasonable share of the work (Polya, 1945, p. 1).

What Polya (1945) said about the role of the teacher is one of the deepest and most important questions about being a teacher. Devotion, practice, and solid principles are fundamental for these professionals to continually train and seek better performance in classes. There is no end to learn in this profession. Tardif (2002) emphasizes the importance of teachers' knowledge and skills, not just in academic terms, but also in the personal and social development of students. Therefore, teachers who teach future teachers could be committed to discussing these aspects of the profession with students to get them to reflect on their future profession. After all, the impact of a teacher on a student's life is unimaginable. The LO described in this paper were created with this purpose: to help future teachers or teachers that are students to improve themselves by showing the knowhow about helping pupils to think about their academic challenges.

Table 3. LO's assessment by El Mhouti et al. (2013) tool

Section of the assessment grid	Learning objects			
	OA1F	OA1R	OA2F	OA2R
Academic quality	22.8	22.6	22.5	22.5
Pedagogical quality	22.1	22.0	22.0	22.0
Didatic quality	19.8	20.6	19.8	20.0
Technical quality	21.6	22.0	21.6	21.5
Total	86.3	87.2	85.9	86.0

F student

The F student's conduct described in **Figure 5** is the LO OA2F table. We can see the chat between the teacher and the F student starts similarly to the occurred between the teacher and the R student in OA2R. By questioning the F student about how to decrease the steps, the teacher encourages exploration and experimentation. When the student suggests setting the move command to zero and then revises it to minus one, it indicates an engagement with trial and error–one part of learning programming and CT.

After the student implements the change, the teacher asks about the outcome. "What happened to the tree now?" – the F student used a tree as the obstacle in his game version. This prompts the student to reflect on the effect of their code modifications, recognizing that while the tree stopped, it did not achieve the intended continuous movement across the scenario. Such reflection is the key to understanding the cause-and-effect relationships in programming.

In the final part of the conversation, the student's explanation reveals his logical reasoning–utilizing negative values for directional control, conditional logic for boundary detection and response, and looping to sustain the action indefinitely. This narrative confirms the student's comprehension of the task and their ability to devise a sequence of operations that were aligned with their strategy to solve the problem. The understanding demonstrated by the student indicates a thoughtful integration of concepts to achieve a desired interactive behavior within the program. This segment of the dialogue underscores the student's capability to synthesize various programming constructs, reflecting in a significant cognitive and computational development.

LO Functionality Assessment

The El Mhouti et al. (2013) proposed an assessment grid that emerges as a tool designed to guide the creation and evaluation of digital learning resources in educational settings. Its primary purpose is to serve as a benchmark for educators and pedagogical agents in developing activities that incorporate digital resources effectively into their teaching methodologies. This tool may help them to reflect on their use of digital resources, ensuring these tools not only supplement traditional teaching methods but also enhance the learning experience for students.

Table 3 shows the results of the four LO we created which were assessed by 57 pre-service teachers who participated in a CT-based mathematics training program that took place in October and November of 2023.

All four LOs (OA1F, OA1R, OA2F, OA2R) are categorized within the 81 to 100 range, indicating they are excellent educational resources. This means they offer diverse functionalities and satisfy the established quality criteria effectively, i.e., "Excellent educational resource with various functionalities that meet the required quality criteria" (El Mhouti et al., 2013, p. 30).

Academic quality

Scores range from 22.5 to 22.8.

This category shows very consistent high performance across all LO, indicating that from an academic standpoint, the content is of high quality, relevant, and likely aligns well with educational goals.

Pedagogical quality

Scores are consistently around 22.

Like academic quality, the pedagogical quality scores are high and consistent, suggesting that the LO are designed with effective teaching strategies, supporting learning processes efficiently.

Didactic quality

Scores vary more significantly in this category, ranging from 19.8 to 20.6.

This category shows slightly lower scores compared to academic and pedagogical quality, indicating space for improvement in how the content is presented didactically.

Technical quality

Scores range from 21.5 to 22.0.

The technical quality also scores highly, though there's a slight variation that suggests some LO may have better technical implementation than others. Factors such as user interface design, accessibility, and reliability could influence these scores.

Despite the variations in scores across different categories (academic quality, pedagogical quality, didactic quality, and technical quality), the overall performance of each LO suggests a high level of quality following El Mhouti et al. (2013, p. 30) rating method. These objects have successfully integrated key educational aspects and technical features, suggesting them as good tools for educational purposes.

This observation gets stronger by the pre-service teachers' perception about their point of view about what they learned from LO used in their training.

Comprehension about how to develop CT using math instructions

> I think that a good way to encourage students to develop aspects of the CT while learning mathematics would be to present them with a complex problem so that they can break it down into smaller parts and solve it in parts through logical reasoning (student 15–11/06/2023).

Student 15 recognizes decomposition as a method that enhances students' problem-solving and logical reasoning skills and focuses on analytical methods for tackling mathematical problems. It is important to highlight that decomposition is one of the four CT pillars which are considered fundamental for the solving problem process in CT (BBC Learning, n. d.). Also, this thought is aligned with the results of Kallia et al. (2021) found in their research with 25 mathematics and computer science experts regarding the opportunities for addressing CT in mathematics education. The participants understand that integrating CT into math education isn't just about using technology but involves cultivating a deep understanding of problem-solving processes that are enhanced by computational methods.

Comprehension about the solving problem process

> I admit that I do not feel prepared to stimulate the development of CT skills while allowing the student to learn mathematical content, even so, I recognize that perhaps stimulating the resolution of problems in parts (decomposition), or problems that require applying the same reasoning to different variants (algorithms and patterns), allowed students to indirectly develop these CT skills (student 11–11/07/2023).

Student 11 is unsure about integrating CT into math education but sees potential in teaching problem-solving through methods like decomposition and applying consistent reasoning to different problems as ways to indirectly develop these skills in students. This student understanding is according to Kallia et al. (2021) outcomes, since their research participants perceive problem-solving in the context of CT as a structured process that often involves breaking down complex problems into simpler, more manageable parts, i.e., decomposition. This understanding also includes recognizing patterns, abstracting problems to their essential elements, and employing algorithmic thinking to devise solutions.

Comprehension about the teaching method

> The teacher played a more passive role, that is, she allowed the student to discover and try. However, she was always there to help, guide and question if something was not going as expected. The teacher helped and guided the student, but always without intervening too much, that is, she allowed him to think, considered his answers and try and then question himself in case something went wrong (student 5–11/07/2023).

> The teacher helped the student to overcome his difficulties, but also let him try so that, if he made a mistake, he would learn from his mistakes (student 10–11/07/2023).

> The class, in my opinion, was quite dynamic and allowed the student to acquire knowledge independently. […] This teaching practice made mathematical and CT concepts easier for the student as it allowed the student to learn by doing (student 8–11/07/2023).

The students noted that the teacher's approach of allowing students to explore and try things on their own, while being available for guidance and to provoke thought when errors occur, effectively aids in understanding and applying mathematical and CT concepts. This constructionism method encourages learning by doing, enabling students to learn from their mistakes and develop their problem-solving skills autonomously (Papert, 1986). Also, the participants from research of Kallia et al. (2021) recognized an active learning environment where students can engage with both theoretical and practical aspects of mathematics as effective for integrating CT into math instruction. This involves using computational tools and languages to model real-world problems mathematically and then exploring these models to understand and solve the problems. Educators are seen as facilitators who guide the exploration and application of computational methods within mathematical contexts.

How LO was useful for their understanding

> My learning experience with CT and the Scratch platform was extremely enriching. During class, we had the opportunity to watch videos in which students explored the platform with the teacher's guidance. It was inspiring to see how students actively engaged in solving challenges and creating projects using Scratch. Throughout the class, the importance of the teacher's guidance in promoting CT was demonstrated and also how the platform can be a powerful tool for developing programming and logical reasoning skills. The subsequent questionnaires helped us reflect on the strategies and approaches used by the students and the teacher, encouraging us to apply different methods in our future (student 5–11/07/2023).

> Regarding the use of LO, I understood that these are digital educational resources that serve to support the teaching and learning process, whose main objective is to provide an effective means to facilitate understanding and acquisition of knowledge. My experience with this resource to understand CT, mathematics teaching and teaching practice was quite pleasant as I did not experience any difficulty in identifying the mathematical themes covered in the activities (student 15–11/06/2023).

> I had a good impression of the teacher who was always available to clarify our doubts and greatly facilitated the process of learning the very concept of CT, which, despite being a complex concept, was much better understood by we students (student 7 11/07/2023).

The usefulness of digital learning resources, according to El Mhouti et al. (2013), depends significantly on its design and its alignment with pedagogical objectives, focusing on its educational quality. According to students' thoughts, they appreciated the LO for its ability to facilitate the understanding of CT, mathematics, and teaching practices through interactive and guided learning, which makes them useful, as El Mhouti et al. (2013) explained.

These reports show that the LO used in the pre-service teachers training facilitated the understanding of the aspects that were the objective of that class:

(1) how to develop a CT based math activity in Scratch could be useful for the students' mathematical learning,

(2) the pupils' understanding of CT skills and concepts, and

(3) the process used by teacher to help pupils solve the problem.

Ye et al. (2023) highlighted a significant gap in research regarding "CT-based mathematics instructions taking place in formal and interdisciplinary education settings" (Ye et al., 2023, p. 24). They emphasized the critical necessity for continuous professional development in emerging educational competencies such as CT-based mathematics reasoning. This study makes a significant contribution by addressing this gap in literature, offering valuable insights and resources for professional development programs aimed to promote teachers' competencies in areas like CT. Furthermore, it contributes to start the pre-service teacher's development of their PCK (Ball et al., 2008) with focus on knowledge of content and students and knowledge of content and teacher in PCK theory, which one will be studied in a future work.

CONCLUSIONS

We aimed to develop LO as tools to help enhance the pedagogical skills of pre-service teachers, with a focus on teaching methods and content domain. This goal prompted us to consider the technical specifications and pedagogical concepts essential for crafting effective LO. Specifically, we sought to instruct pre-service teachers on how students can apply or expand their mathematical knowledge through CT-based activities. Our approach emphasized constructionism and problem-solving methodologies, which seems to be a good strategy to develop mathematical concepts within programming tasks to foster students' CT skills, since the LO were assessed by pre-service teachers with a high-quality score for our educational purpose.

Regarding our research question we understood that the fact of designed the LO with a focus on fostering essential pedagogical skills such as teaching methods, student engagement, and the integration of CT into mathematics, effectively prepare the pre-service teachers to handle alternative classroom settings with the use of the Scratch. Besides that, through the structured scenarios and problem-solving tasks embedded within the LO, the pre-service teachers could visualize and reflecting about teaching strategies that fostered a better comprehension of CT concepts among students. This hands-on approach showed teachers' instructional techniques what could improve their confidence in deploying innovative teaching methods that make mathematical concepts more accessible and engaging for students.

Finally, the LO provided insights to the pre-service teachers into how the pupils assimilated and applied CT-based mathematical instructions. The LO allowed teachers to observe students' learning processes in real-time, including how they approached problem-solving, how they overcome challenges, and how they applied CT to solve math problems and to learn mathematics.

Author contributions: JD: conceptualization, data curation, investigation, methodology, writing – original draft; **MEBBP & AdFGS:** funding acquisition, resources, supervision. All authors agreed with the results and conclusions.

Funding: This study was supported by the Coordination for the Improvement of Higher Education Personnel–Brazil–Financing Code 001.

Ethical statement: The authors stated that the research made with the pupils was approved by the 347–Centro Universitário Anhanguera de São Paulo on 22 December 2020 with approval number 4.481.578 and the research made with the pre-service teachers was approved by the 347–Centro Universitário Anhanguera de São Paulo on 16 March 2022 with approval number 5.294.198. Written informed consents were obtained from the participants.

Declaration of interest: No conflict of interest is declared by the authors.

Data sharing statement: Data supporting the findings and conclusions are available in Dobgenski (2024).

REFERENCES

Ball, D. L., Thames, M. H., & Phelps, G. (2008). Content knowledge for teaching: What makes it special? *Journal of Teacher Education, 59*(5), 389-407. https://doi.org/10.1177/0022487108324554

BBC Learning. (n. d.). Introduction to computational thinking. *BBC*. http://www.bbc.co.uk/education/guides/zp92mp3/revision

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2016). Building mathematical knowledge with programming: Insights from the ScratchMaths project. In A. Sipitakiat, & N. Tutiyaphunegprasert (Eds.), *Proceedings of constructionism* (pp. 25-32).

Deitel, P. J., & Deitel, H. M. (2005). *C++ how to program* (5th ed.). Pearson.

Dobgenski, J. (2024). *Development and validation of learning objects aimed at mathematical instructions based on computational thinking*. Repositório de Dados da Universidade do Minho, V1, UNF:6:t28BN2qIQqbcZQDX+cGXjA== [fileUNF]. https://doi.org/10.34622/datarepositorium/WIDQQF

Dobgenski, J., & Garcia Silva, A. (2022). A matemática e elementos do pensamento computacional contidos em atividades programadas em Scratch [Mathematics and computational thinking elements contained in activities programmed in Scratch]. *Educação e Matemática, 164*(2), 33-35.

Dobgenski, J., Brito Prado, M. E. B., & Garcia Silva, A. D. S. (2024). *Desenvolvimento do pensamento computacional com programação Scratch: Um estudo comparativo entre aprendizado presencial e remoto* [Development of computational thinking with Scratch programming: A comparative study between face-to-face and remote learning]. In *Proceedings 6th Meeting on Games and Mobile Learning*.

Dobgenski, J., Garcia Silva, A. F. & Mazzarino, F. D. (2022). Conhecimentos sobre o pensamento computacional: Um olhar a partir da prática reflexiva docente [Knowledge about computational thinking: A look from the reflective teaching practice]. *Jornal Internacional de Estudos em Educação Matemática, 15*(1), 27-35. https://doi.org/10.17921/2176-5634.2022v15n1p27-35

El Mhouti, A., Nasseh, A., & Erradi, M. (2013). Development of a tool for quality assessment of digital learning resources. *International Journal of Computer Applications, 64*(14), 27-31. https://doi.org/10.5120/10703-5613

Gleasman, C., Kim, C. (2020). Pre-service teacher's use of block-based programming and computational thinking to teach elementary mathematics. *Digital Experiences in Mathematics Education, 6*, 52-90 https://doi.org/10.1007/s40751-019-00056-1

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296-310. https://doi.org/10.1016/j.compedu.2018.07.004

Jonassen, D. H. (2007). *Computadores, ferramentas cognitivas: Desenvolver o pensamento crítico nas escolas* [Computers, cognitive tools: Developing critical thinking in schools]. Porto Editora.

Kallia, M., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Characterising computational thinking in mathematics education: A literature-informed Delphi study. *Research in Mathematics Education, 23*(2), 159-187. https://doi.org/10.1080/14794802.2020.1852104

Ministério da Educação [Ministry of Education]. (2018). Base nacional comum curricular [Common national curriculum base]. *MEC*. http://basenacionalcomum.mec.gov.br/

Papert, S. (1986). *Constructionism: A new opportunity for elementary science education*. National Science Foundation.

Polya, G. (1945). *How to solve it*. Princeton University Press. https://doi.org/10.1515/9781400828678

Raabe A., Brackman C., & Campos, F. (2020). Currículo de referência em tecnologia e computação: Da educação infantil ao ensino fundamental [Reference curriculum in technology and computing: From early childhood education to primary education]. *Centro de Inovação para a Educação Brasileira*. https://curriculo.cieb.net.br/javascript(0)

Raabe, A., Vieira, M., Santana, A., Gonçalves, F., & Bathke, J. (2015). Recomendações para introdução do pensamento computacional na educação básica [Recommendations for introducing computational thinking into basic education]. In *Proceedings of the IV Workshop on Challenges of Computing Applied to Education* (pp. 141-150). SBC. https://doi.org/10.5753/desafie.2015.10049

Rachmatullah, A., Akram, B., Boulden, D., Mott, B., Boyer, K., Lester, J., & Wiebe, E. (2020). Development and validation of the middle grades computer science concept inventory (MG-CSCI) assessment. *Eurasia Journal of Mathematics, Science and Technology Education, 16*(5), Article em1841. https://doi.org/10.29333/ejmste/116600

Säfström, A. I., Lithner, J., Palm, T., Palmberg, B., Sidenvall, J., Andersson, C., Boström, E., & Granberg, C. (2023). Developing a diagnostic framework for primary and secondary students' reasoning difficulties during mathematical problem solving. *Educational Studies in Mathematics, 115*, 125-149. https://doi.org/10.1007/s10649-023-10278-1

Schoenfeld, A. (1985). *Mathematical problem solving*. Academic Press.

Sebesta, R. W. (2002). *Concepts of programming languages*. Addison Wesley.

Selby C., & Woollard, J. (2013). Computational thinking: The developing definition. *University of Southampton*. http://eprints.soton.ac.uk/356481/

Silva, R. S. R., Gadanidis, G., Hughes, J., & Namukasa, I.K. (2020). Computational thinking as a heuristic endeavour: Students' solutions of coding problems. *Pro-Posições, 31*. https://doi.org/10.1590/1980-6248-2018-0034

Tardif, M. (2002). *Saberes docentes e formação profissional* [Teaching knowledge and professional training] (17th ed.). Editora Vozes.

Valente, J. A. (2003). In service teacher development using ICT: First step in lifelong learning. In C. Dowling, & K. W. Lai (Eds.), *Information and communication technology and the teacher of the future. IFIP–The International Federation for Information Processing, vol 132* (pp. 97-108). Springer. https://doi.org/10.1007/978-0-387-35701-0_10

Valente, J. A. (2016). Integração do pensamento computacional no currículo da educação básica: Diferentes estratégias usadas e questões de formação de professores e avaliação do aluno [Integrating computational thinking into the basic education curriculum: Different strategies used and issues of teacher training and student assessment]. *Revista e-Curriculum, 14*(3), 864-897.

Wiley, D. A. (Ed.). (2002). *The instructional use of learning objects*. Agency for Instructional Technology and Association for Educational Communications & Technology.

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

Ye, H., Liang, B., Ng, O.-L., & Chai, C. S. (2023). Integration of computational thinking in K-12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education, 10*, Article 3. https://doi.org/10.1186/s40594-023-00396-w