A task for integrating computational thinking in the learning of affine function: An exploratory study with 8th grade students

Uma tarefa de integração do pensamento computacional na aprendizagem da função afim: Um estudo exploratório com alunos do 8.º ano

Andreia Mateus
Instituto de Educação, Universidade de Lisboa
Portugal
amgmateus@edu.ulisboa.pt

Hélia Oliveira
UIDEF, Instituto de Educação, Universidade de Lisboa
Portugal
hmoliveira@ie.ulisboa.pt

João Piedade
UIDEF, Instituto de Educação, Universidade de Lisboa
Portugal
jmpiedade@ie.ulisboa.pt

Abstract. The recent introduction of computational thinking in mathematics curricula requires investment in the design of teaching resources that promote an integrated learning of computational thinking and mathematics in real classroom settings. Developed in the context of a teaching experiment, this exploratory study focuses on discussing how one task that integrates computational thinking into the study of affine function in the 8th grade contributes to the development of students' computational thinking practices and students' functional thinking. A qualitative methodology was adopted, with data collected through participant observation from two classes, supported by audio and video recordings, as well as students' written work. The targeted computational thinking practices were abstraction, decomposition, pattern recognition, analysis and definition of algorithms, and development of habits for debugging and optimizing processes. Functional thinking analysis focused on the function representations, contextual and symbolic generalization, and mathematical modelling. The results show that computational thinking practices and dimensions of functional thinking were integrated during students' task resolution, highlighting the important role of the

Quadrante: Revista de Investigação em Educação Matemática 33(2) 193-222
https://doi.org/10.48489/quadrante.37041
Received: July 2024 / Accepted: December 2024

task's context and the different types of representation, namely the *coded representation* facilitated by Scratch, in supporting students moving from concrete examples to more general situations.
Keywords: computational thinking; functional thinking; affine function; mathematics; middle school.

Resumo. A recente introdução do pensamento computacional nos currículos de matemática exige investimento no design de recursos que promovam uma aprendizagem integrada do pensamento computacional e da matemática na sala de aula. Este estudo exploratório, realizado no contexto de uma experiência de ensino, discute como uma tarefa que integra o pensamento computacional no estudo de funções afins no 8.º ano contribui para o desenvolvimento das práticas de pensamento computacional e do pensamento funcional dos alunos. Adotou-se uma metodologia qualitativa, com dados recolhidos através de observação participante em duas turmas, com áudio e vídeo gravação, e dos trabalhos escritos dos alunos. As práticas de pensamento computacional visadas foram: abstração, decomposição, reconhecimento de padrões, análise e definição de algoritmos e o desenvolvimento de hábitos de depuração e otimização dos processos. A análise sobre o pensamento funcional focou-se nas representações de funções, generalização contextual e simbólica e modelação matemática. Os resultados mostram que as práticas de pensamento computacional e as dimensões do pensamento funcional foram integradas durante a resolução da tarefa pelos alunos, destacando-se o importante papel do contexto da tarefa e dos diferentes tipos de representação, especialmente, a *representação codificada* facilitada pelo Scratch, para apoiar os alunos a progredirem de exemplos concretos para situações mais gerais.
Palavras-chave: pensamento computacional; pensamento funcional; função afim; matemática; 3.º ciclo.

Introduction

The emergence of the term "computational thinking" in education (Wing, 2006) sparked a global discussion regarding its conceptualization and integration in education in general (Caeli & Yadav, 2020; Román-González et al., 2017) and specifically in mathematics teaching (Barcelos et al., 2018; Chan et al., 2023; Ye et al., 2023). Beyond the recognized value of computational thinking as a 21st century competence for young people and essential for prospering in the world (Caeli & Yadav, 2020), its potential to enhance mathematics learning has been receiving growing attention from researchers (Chan et al., 2023). Namely, Weintrop et al. (2016) claim that computational thinking may offer students a more realistic insight into mathematics, and Ye et al. (2023) show that computational thinking based mathematical tasks may support students in developing novel mathematical knowledge, and not only in its application. Recognizing the importance of developing computational thinking in students, various countries have integrated it into their curricula in different ways over the past few years, either through a specific discipline or in a cross-disciplinary manner (Bråting & Kilhamn, 2021). Following this trend, the Portuguese mathematics curriculum for basic education (Canavarro et al., 2021) has introduced computational thinking as one of the six transversal mathematical skills to be promoted. However, significant questions have been raised about how to promote it alongside the mathematical topics in the curriculum (Israel & Lash, 2020). One major concern is how to effectively teach for computational thinking development as well as for mathematics learning, in an articulated way (Ye et al. 2023), as research shows that quite often teaching in that context prioritizes computational thinking outcomes over mathematics learning (Chan et al., 2023). The design features of computational thinking based mathematical tasks, namely if they necessarily must imply the use technology (Caeli & Yadav, 2020), and their nature and level of structure (Cui et al., 2023), are also part of the debate concerning the introduction of computational thinking in mathematics classrooms.

Among the studies that establish a connection between computational thinking and the teaching and learning of mathematical topics, functions are notably underrepresented (Hickmott et al., 2018). Thus, this exploratory study, part of a teaching experiment, aims to contribute to fill a gap in the literature regarding the limited availability of educational resources that promote the integrated development of computational thinking and mathematics learning, specifically concerning the teaching of affine function in the 8th grade, to promote students' functional thinking (Pitta-Pantazi et al., 2020; Smith, 2008). The teaching experiment was conceived to explore how to integrate computational thinking in the regular teaching practice in the classroom, conformed with the actual time restrictions for covering the curriculum and the structure of classes daily timetable, and thus providing clues about the feasibility of promoting computational thinking as a new curricular goal for basic education. Hence, the study inherently reflects the complexity of real-world teaching conditions and assumes the importance to guarantee the research *ecological validity* (Cobb et al., 2003), particularly when curricular innovations are proposed to schools, such as it is the case of the introduction of computational thinking in Portuguese mathematics curriculum.

Based on the above objectives, the following research question is addressed by this study: How does a task that integrates computational thinking into the study of affine function in 8th grade classes contribute to students' practices of computational thinking and students' functional thinking?

Theoretical framework

Computational thinking in mathematics education

Computational thinking was considered by Wing (2006) as a fundamental skill at the same level as reading, writing, and arithmetic, consistently focused on the problem-solving process. Building on the author's ideas, which revisit some of Papert's contributions from the 1980s, the development of computational thinking has come to occupy a prominent place in some studies on mathematics education in which researchers are concerned with the integration of computational thinking in mathematics teaching (Barcelos et al., 2018; Chan et al., 2023; Ye et al., 2023).

In this study, computational thinking is characterized as a way of thinking when faced with problematic situations that involves the enactment of practices such as those outlined in the Portuguese mathematics curriculum for basic education (Canavarro et al., 2021), namely: (i) abstraction, which involves extracting the essential information from a problem; (ii) decomposition, which is based on simplifying a problem by breaking it down into more manageable parts; (iii) pattern recognition, which implies recognizing or identifying patterns and regularities when solving one problem and applying them to similar problems; (iv) analysis and definition of algorithms, which involve solving a problem through a step-by-step procedure, often resorting to technology; and (v) development of debugging and process optimization habits, which encourages testing to identify and correct errors and subsequently refine a given solution.

The characterization of computational thinking in education, especially when incorporated into mathematics, is of utmost importance and it has been the subject of many studies. In research we find reference to computational thinking practices in mathematics, as it is the case of the three studies presented in table 1. These authors agree regarding the notions of abstraction, decomposition and algorithmic thinking, three of the five practices mentioned in the Portuguese basic education mathematics curriculum (Canavarro et al., 2021). Although less frequently mentioned, pattern recognition and debugging are also noted in research, with the latter practice being highlighted in studies based on literature reviews or that are, themselves, literature reviews (Kallia et al., 2021; Ye et al., 2023).

Table 1. Examples from the literature on main practices of computational thinking

Barr and Stephenson (2011)	**Kallia et al. (2021)**	**Ye et al. (2023)**
Data collection, analysis, and representation	Abstraction	Systemic thinking (decomposition, abstraction, and algorithmic thinking)
Abstraction	Decomposition	Data practices (modelling and simulation)
Decomposition	Pattern recognition	Reuse and remix
Algorithms	Algorithmic thinking	Testing and debugging
Testing and validation	Modelling	
Simulation	Logical thinking	
Automation	Automation	
	Generalization	
	Evaluation of strategies and solutions	

Approaches to computational thinking in mathematics teaching

According to Israel and Lash (2020), there are three types of teaching approaches regarding the level of integration of computational thinking and mathematics: (i) no integration, where mathematics or a computational thinking concept are taught isolated; (ii) partial integration, where computational thinking is used to reinforce a mathematics concept or vice versa; and (iii) full integration, where mathematics and computational thinking concepts are taught together. In this study, the authors concluded that fully integrated teaching was the least used by the teachers involved, who mentioned that this approach is more difficult for students due to the increased "cognitive load demands when teaching the CS/CT at the same time as the mathematics curriculum" (p. 374). However, in more recent studies it was found that partial integration is dominant, either with an intensive focus on computational thinking (Chan et al., 2023) or with a predominant focus on mathematical understanding, using computational thinking as a technique (Nordby et al., 2022).

To understand how to outline pedagogical approaches (plugged and unplugged) that best promote the development of computational thinking, several investigations have been carried out. Regarding the unplugged approach, Bell and Vahrenhold (2018) state that it is used "to engage a variety of audiences with great ideas from computer science, without having to learn programming or even use a digital device" (p. 497). Several potentialities have been highlighted for this type of approach, namely, helping students to construct and outline their reasoning (Caeli & Yadav, 2020), allowing access to activities that promote computational thinking when electronic equipment is limited (Evaristo et al., 2022), help to ease fears of programming among teachers and students, and meaningfully engaging the latter with the big ideas of computer science (Bell & Vahrenhold, 2018).

Despite the recognition of their importance, there is a certain consensus that unplugged approaches should not be implemented isolate but in connection with the use of technology (Bell & Vahrenhold, 2018). As such, plugged approaches emerge to complement unplugged ones, contributing to stimulating students' thinking by using technology as a tool (Evaristo et al., 2022).

In mathematics teaching, plugged approaches are mostly focused on programming and are recognized as a strong pedagogical approach to developing computational thinking, given the importance of the cognitive aspects behind programming (Kallia et al., 2021). Scratch programming environment is one of the most common digital tools for exploring computational thinking with students and it is also suggested by the Portuguese basic education mathematics curriculum (Canavarro et al., 2021). By using a visual programming language (in blocks), this tool becomes very simple for students. However, other tools are also pointed out as having the potential for promoting computational thinking in conjunction with mathematics, such as spreadsheets, which are considered a "language" that uses text-based code (Chan et al., 2021), or Sketchpad, regarded as a third programming language that employs geometric objects instead of text or block-based code (Ye et al., 2023).

Recognizing the importance of both approaches, Caeli and Yadav (2020) argue that to engage students, unplugged and plugged approaches should be combined. For instance, unplugged tasks may reinforce students' understanding of certain programming concepts, such as in the study by Gomes et al. (2018) where the loop concept was identified as a difficulty in plugged activities. Therefore, it is pertinent to investigate what tasks can be created to combine these two approaches.

Task design for computational thinking in mathematics

Given the central role that tasks play in mathematics teaching (Kieran et al., 2015), they should be designed to engage students and to effectively enhance their mathematical understanding. When designing a task, we should consider the relevance of mathematics in daily life, through its application in other areas of knowledge or in everyday situations, allowing students to recognize the need to use the knowledge acquired in this subject to model real-world scenarios (Brennan & Resnick, 2012). Additionally, one of the main goals in school mathematics is to support students in developing conceptual understanding. Focusing on the design of mathematical tasks with that goal, Swan (2014) proposed four tasks genres, namely: (i) observing, classifying, and defining, involving the manipulation of mathematical objects, and creating and examples and non-examples; (ii) representing and translating mathematical concepts in their various representations; (iii) justifying and/or proving mathematical conjectures and procedures; and (iv) identifying and analysing structure within situations, including studying relationships between variables and comparing mathematical structures. These provide important elements that can be considered in tasks that target mathematical concepts from different topics.

In recent years, the design of mathematical tasks has been extensively discussed by researchers and teachers (Kieran et al., 2015). However, since computational thinking is a relatively recent concept in mathematics curriculum, the pedagogical design of mathematical tasks that address the dimensions of computational thinking has not been widely explored. Kotsopoulos et al. (2017) developed an initial pedagogical framework based on empirical data and rooted in Papert's constructivist learning theory that considers four experiences for developing computational thinking: unplugged, tinkering, making, and remixing. These authors suggest that students should start with less demanding experiences in terms of understanding (*unplugged* and *tinkering*) and progress to more cognitively demanding ones *(making* and *remixing*). The authors also note that the first experience, *unplugged*, should be the initial approach to promote the development of computational thinking, as it introduces concepts without using computers. This approach requires less cognitive load and does not require knowledge of programming languages. In the next experience, *tinkering*, students modify existing objects, analysing the implications of these changes on the results and correcting possible errors, with a focus on applying concepts and simulation. In the *making* experience, students solve problems by planning and selecting tools, as well as establishing connections between concepts. Finally, in the *remixing* experience, students modify an object to adapt it for a different purpose.

Taking into consideration what one knows about ways to promote students' conceptual understanding in mathematics, there is a need of seeking how to combine these four experiences for developing computational thinking in the design of tasks for supporting the development of different concepts in mathematics.

Functional thinking

Functional thinking is a form of generalization, associated with the exploration of sequences from the early years of schooling (Cabral et al., 2022). According to Smith (2008), functional thinking focuses on the relationship between two or more variables and on the evolutionary process from specific relationships to their generalization. Therefore, it is important to identify these relationships, represent them in various ways, and ultimately generalize them.

According to the Portuguese basic education mathematics curriculum, the study of functions is explicitly introduced in the middle school, where it is associated with understanding variation, establishing relationships between quantities/magnitudes, expressing generalizations and representing them in various forms, and modelling (Canavarro et al., 2021). Specifically, in the 8th grade, the focus is on the study of affine function, concerning: their multiple representations (graphical, algebraic, and tabular) and the connections between them; the effect of varying their parameters; the possibility of making predictions; and the interpretation and modelling of real-world situations.

Functional thinking has several dimensions, from which, within the scope of this study, we highlight the following: (i) generalization, (ii) representation of functions and their connections, and (iii) modelling. Generalization refers to extending reasoning beyond specific cases to converge on a general rule, which can be represented through natural language (use of words), syncopate language (a mix of symbols and natural language), or symbolic mathematical language (alphanumeric notation) (Cabral et al., 2022). Depending on how the rule is expressed, there are three types of generalization: factual, using specific cases; contextual, supported by the described context; and symbolic, using symbolic mathematical language (Cabral et al., 2022; Radford, 2006).

In the learning of functions, several authors highlight the importance of promoting different representations (Best & Bikner-Ahsbahs, 2017; Blanton & Kaput, 2011). Blanton and Kaput (2011) emphasize students' understanding of multiple representations to express the relationships between quantities, as well as using these representations to interpret and predict the functions behaviour. Similarly, Best and Bikner-Ahsbahs (2017) consider it essential to relate different representations of functions to ensure that learning is not fragmented. Recognizing the importance of multiple representations of functions, in this paper, we introduce a new category called *coded representation*, which can be presented through pseudocode or programming languages, such as Scratch, and can be associated with purely mathematical contexts or real-life situations.

To promote functional thinking, mathematical modelling is also emphasized, as it involves translating a real-life situation into a mathematical model. This process helps recognizing the relevance of mathematics in various real-world contexts, as outlined in mathematics curriculum (Canavarro et al., 2021). According to Pitta-Pantazi et al. (2020), students must first interpret the context, identify the quantities involved and their relationships, and then represent them using the most appropriate mathematical model. Modelling is considered a cyclical process divided into seven steps: initial understanding of the real situation without context, simplification focusing on what is essential, identification of the mathematical model involved, mathematical manipulation of the model, interpretation of the obtained mathematical results, validation of these results, and translation back to the initial real situation (Blum & Ferri, 2009).

Developing functional thinking is not an easy endeavour, as it is noticeable in many secondary classes (Martins et al., 2023), therefore it is significant to provide students with tasks that challenge them to recognize and articulate structures and relationships.

Methodology

The study context and task selection

The present study was carried out as part of a broader teaching experiment focused on tasks integrating computational thinking with topics related to functions, specifically affine function, literal equations, and systems of two first-degree equations with two variables (graphic solving). The teaching experiment was conducted by the first author, who teaches mathematics at a public secondary school in the Oeiras Municipality (Lisbon metropolitan area). The participants included 49 students aged 13 to 14 from two 8th grade classes.

Following Chan et al. (2021), to avoid adding difficulties in learning mathematics when solving integration tasks, during the current school year, the teacher-researcher had introduced the Scratch programming environment in her classes to reinforce content related to the topics of numbers and algebra. For this study, the first task of the teaching experiment, intitled "Natural Gas vs. Propane Gas" (Appendix) was selected. This task intended to introduce students to the concept of the affine function, $f(x) = ax + b$, with $a, b \in \mathbb{R}$, through the interpretation and modelling of a concrete situation related to different gas consumption plans in Portugal. As students were familiar with direct proportionality function from the previous year, the task takes that knowledge to introduce the students in the exploration of the new concept, alongside with the intended computational thinking practices. By relying on this first task in the teaching experiment, this study allows us to analyse students' functional thinking when they first contact with a new type of function in connection with the task's characteristics.

Data collection and analysis methods

This study followed a qualitative methodology, and as such, the data collected were qualitative in nature and gathered in the natural classroom environment, focusing on the students' reasoning during the completion of the selected task. Different data collection methods were employed, including participant observation, video recordings of the lessons, audio recordings of eight pairs of students, and the collection of students' written solutions to the task. The pairs of students (four pairs from each class) were chosen to maximize the benefits of their interaction in solving the tasks, ensuring more comprehensive data.

To gather empirical evidence on how students show the intended learning outcomes from the proposed task, this study focuses on analysing the students' productions that represent the range of solutions from the target groups. These were complemented, when necessary, with data from the audio recordings of the pairs.

The analysis framework focuses on (i) five practices of computational thinking—abstraction, decomposition, pattern recognition, algorithm analysis and definition, and development of debugging and process optimization habits—and (ii) three dimensions of functional thinking—function representations, mathematical modelling, and contextual and symbolic generalizations—which are central aspects related to the intended learning outcomes of the task that is presented in the next section.

The task for integrating computational thinking and functional thinking

Design of the task

For the design of the task, the specific objectives for learning functions outlined in the mathematics curriculum (Canavarro et al., 2021) were considered, as well as the guidelines on designing tasks related to mathematics learning (Swan, 2014) and the development of computational thinking (Kotsopoulos et al., 2017). Thus, the task was designed to achieve the goal of understanding of affine function and included aspects of the task's genres for conceptual understanding pointed out by Swan (2014) and simultaneously it followed the pedagogical approach suggested by Kotsopoulos et al. (2017) to promote computational thinking. For the latter, this implies to start with unplugged tasks and then move on to modifying existing objects and validating these modifications. Next, students solve problems, requiring them to plan, select tools, and establish connections between concepts. Finally, students adapt what they have learned by applying it to a new problem situation.

Adapting the task design from Kotsopoulos et al. (2017) and following the illustration in Figure 1, the task is divided into four parts according to its level of difficulty.

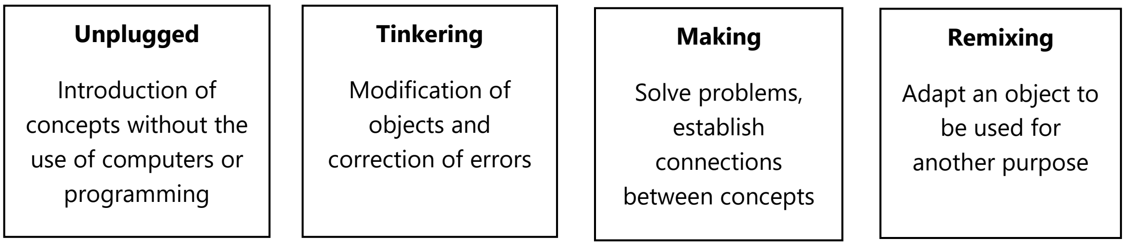


Figure 1. Task design sequence

The construction of the task was also based on the teacher-researcher's experience over 20 years of teaching, which allowed her to develop knowledge regarding the design and implementation of mathematical tasks, both in mathematical and real-world contexts. Additionally, it drew on her previous experience in facilitating lessons for consolidating mathematical content using flowcharts and programming in Scratch and on the literature as previously explained.

The task was intended to introduce the subtopic "affine function", with the following learning objectives: to identify a direct proportionality function with a linear function; to interpret and model real-world situations using affine functions; and to recognize the affine function as a function of the type $f(x) = ax + b$, with $a, b \in \mathbb{R}$, and the linear function as a specific case of the affine function (Appendix).

The first question, framed within the *unplugged* part of the task, aims student to analyse a situation that can be modelled by a linear function. In the first part, they are asked to explore an example of a direct proportionality situation using specific values of gas consumption. The following part requires them to construct a flowchart that identifies the inputs/outputs related to the described situation and outlines the reasoning from the previous part in a more abstract context: the contextual algebraic generalization referred to by Radford (2006). In the final part, students are expected to advance to symbolic algebraic generalization (Radford, 2006), using the variables x and y to write an algebraic expression representing the described situation.

In the first four items of the second question, related to the *tinkering* part of the task, students are expected to interpret situations modelled by a nonlinear affine function, make predictions, and correct errors. In the first two parts, students should interpret a pre-written code, contextualize some of its instructions according to the given situation, and make a prediction about an output (image) resulting from a specific input (object). In item c), students are prompted to compare codes and identify errors, using, for example, the rules of mathematical operation priority to justify their reasoning. Item d) requires students to translate the situation using the variables *x* and *y*, arriving at the algebraic expression of a nonlinear affine function.

The last three items of the second question, which pertain to the *making* part of the task, aim to address more challenging situations involving the nonlinear affine function and to establish connections between concepts. In item e), students are expected to determine an object corresponding to a given image, using, for example, reasoning with inverse operations or solving equations which were studied in previous years. Next, students should generalize their reasoning by completing lines of a code presented in the task and then build the program in Scratch to validate the obtained result.

Finally, in the *remixing* part of the task, students are required to build, in a less guided manner, a program in Scratch to model a real-life situation involving percentages, which is associated with a direct proportionality function. In this part, after interpreting the described situation, students must outline a strategy to follow.

In Table 2 we present the aspects of computational thinking and functional thinking targeted by each question of the task, according to the previously defined analysis framework, which is based on two dimensions: (i) the computational thinking practices mentioned in the Portuguese basic education mathematics curriculum (Canavarro et al., 2021), namely: abstraction (AB), decomposition (DE), pattern recognition (PR), algorithm analysis and definition (AD), development of debugging habits and process optimization (DH); and (ii) dimensions of functional thinking, such as function representations (FR), mathematical modelling (MM), and contextual (CG) and symbolic (SG) generalizations.

Table 2. Aspects of computational and functional thinking targeted by the task

Group of Questions	Question	Computational thinking practices	Dimensions of functional thinking
G1	1. a)	AB	CG, SG, FR
	1. b)	AD	
	1. c)	AD	
G2	2. a)	AB	FR
	2. b)	DH	
	2. c)	DH	
	2. d)	AD	
G3	2.e)	PR	CG, FR
	2. f)	PR, AD	
	2. g)	DH	
G4	3.	AB, DE, AD, DH	MM; FR; CG

The task's enactment in the classroom

Adopting an exploratory teaching practice organized into four phases (Menezes et al., 2015), the implementation of the task began with organizing student pairs and setting time limits for each part of the lesson. This was followed by the presentation and contextualization of the task by the teacher-researcher, who ensured that students understood the objectives (5 minutes). Subsequently, the students worked on questions 1 and 2 of the task (70 minutes). Throughout the completion of the task, students received assistance from the teacher-researcher, who sought to clarify any questions that arose, posed questions to the students, and emphasized the importance of justifying their reasoning. However, the teacher-researcher did not interfere with their strategies or make corrections. Next, the teacher conducted a whole-group discussion from the students' strategies and solutions, encouraging the comparison of different approaches (15 minutes). Finally, based on the students' work and to systematize the learning, a summary table was created where the conclusions regarding the identification of differences between a linear and a non-linear affine function, the algebraic expression of the function, and the determination of an object or an image of the function (10 minutes). In a subsequent class, question 3 of the task was presented (5 minutes), where the potential of programming in creating a gas/electricity bill was discussed. Finally, after the task was completed (35 minutes), the teacher-researcher requested students to share with the class the different strategies they developed (10 minutes).

Results

The following subsections present the analysis of the students' task resolution. These are organized into four parts, according to the group of questions (G1 to G4).

Group 1

In the first question, all the selected student pairs were able to extract the essential information from the problem (AB), correctly interpreting how to relate the 33€ with the 11 m^3, although they used different strategies, as illustrated below. For example, pair TM likely drew connections with other types of situations solved in different contexts to determine the unit price per m^3 (PR), using a simple proportion rule (Figure 2). Pair FJ (Figure 3) started by determining the price per m^3, breaking down the problem into simpler parts (DE) using the rule of three to do so (PR). Pair GT (Figure 4) determined the price per m^3 by dividing 33 by 11 and identified the result of this quotient as a unit price per m^3, demonstrating an understanding of the constant in a direct proportionality situation (PR).

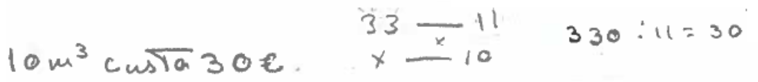


Figure 2. Solution to question 1.a by pair TM

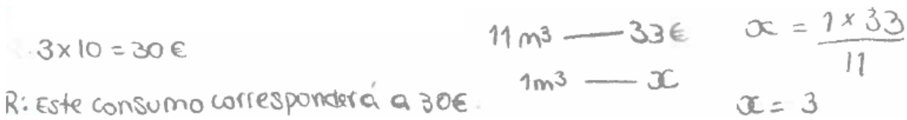


Figure 3. Solution to question 1.a by pair FJ

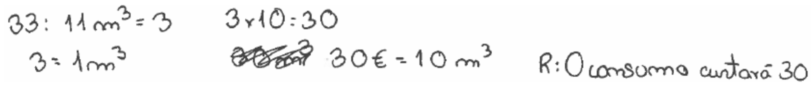


Figure 4. Solution to question 1.a by pair GT

Regarding the next question, most students were successful in the targeted computational thinking practice (AD). However, pair HD did not correctly identify the input data for the flowchart, using the concrete values from the previous question to try to complete it (Figure 5, left). Pair CA successfully outlined the construction of an algorithm to automatically determine the cost of gas (AD), correctly identifying the input data, the procedures to follow, and the output data. They explicitly represented these in the flowchart (Figure 5, centre) using natural language (FR), demonstrating the ability to generalize within the context of the task (CG). In the flowchart created by pair AC (Figure 5, right), although the meaning of the variables is not explicitly defined, the students show a successful sequenced of steps needed to determine the cost of gas (AD), and to generalize this procedure by using variables (SG).

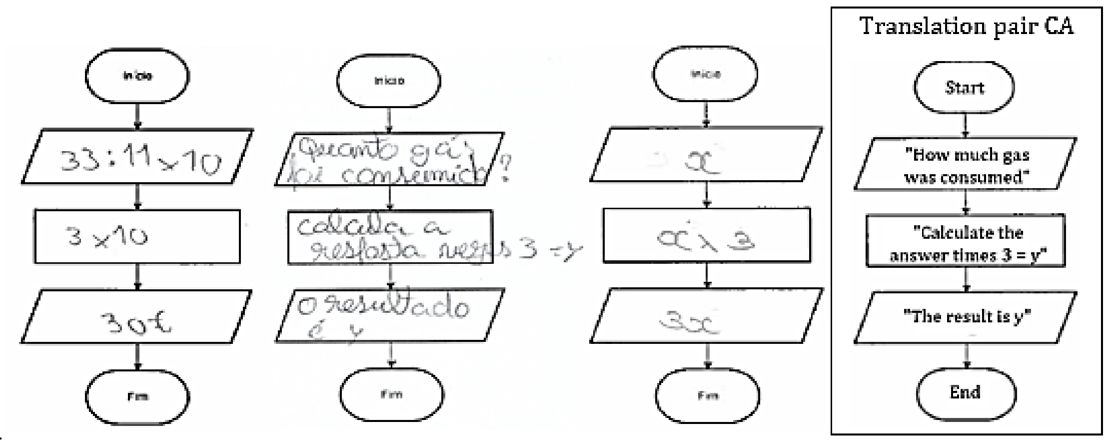


Figure 5. Solution to question 1.b by pairs HD (left), CA (centre), and AC (right)

All the analysed pairs managed to write a correct algebraic expression (AD). The pair AC (Figure 6, left), despite not having identified the value of m^3 in question 1.a, determined it in the current question and used it to generalize their reasoning (SG) and write an algebraic expression (FR). The pair CA (Figure 6, centre), who had identified the fixed and variable values in the rule of three used in the first question, applied that reasoning to arrive at an algebraic expression (SG and FR). In addition to writing the algebraic expression, the pair GT tested it to confirm the results from the first question (DH) (Figure 6, right).

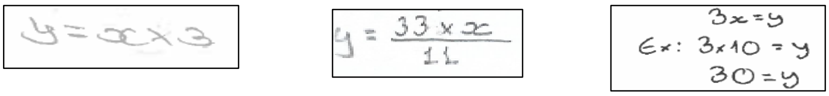
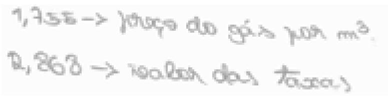


Figure 6. Different algebraic expressions presented by pairs AC (left), CA (centre), and GT (right)

In this first group of questions, the students, starting from the analysis of a concrete example of the described situation, were able to advance their reasoning to reach symbolic generalization by writing an algebraic expression of a linear function.

Group 2

In the first question of this group, the students were able to interpret the values presented in the coded representation (FR), associating their meaning with the context of the described situation (AB), as illustrated by group AC in Figure 7. In contrast, group FJ (Figure 8) did not relate the two values to the real context mentioned, interpreting them in a mathematical context and correctly identifying them as a fixed component and a variable component.



Translation
1,755 → gas price per m^3
2,868 → value of fees"

Figure 7. Solution to Question 2.a by pair AC

Translation
"1,755 is the fixed component and 2,868 is the variable component."

Figure 8. Solution to Question 2.a by pair FJ

In question 2.b, in an initial resolution, pair CP did not consider the fixed tax in their reasoning and only after testing the program using the QR code were they able to identify and correct the error (DH), disregarding the initial result of 1.755, which only accounted for the variable component of the gas price (Figure 9). The remaining pairs correctly anticipated the results of the coded representation, identifying the input and output values and describing their reasoning similarly to what is illustrated by pair FJ (Figure 10).

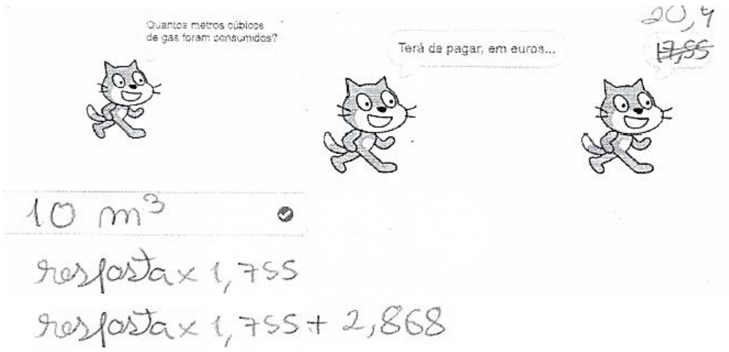


Figure 9. Solution to question 2.b by pair CP

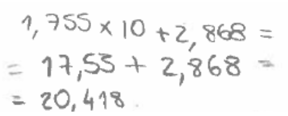
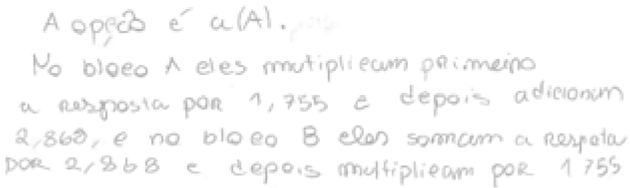


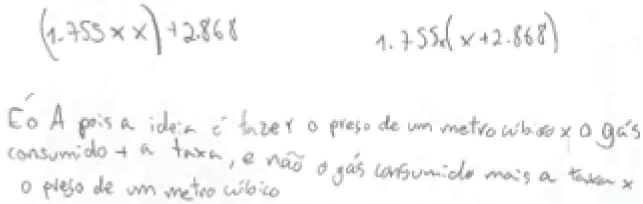
Figure 10. Solution to question 2.b by pair FJ

In question 2.c, all pairs provided a correct answer, accurately identifying and justifying the existing differences between the two blocks of code (DH). Group MM (Figure 11) demonstrates an understanding of the presented codes by explaining what each set of blocks does (DH). Pair HD (Figure 12) first converts the codes into symbolic language and then justifies their response by reinforcing their explanation using natural language in the context of the described situation (AB), showing proficiency in converting between different representations (FR). Going beyond the conversion of the code to symbolic language (SG), group CP (Figure 13) uses the distributive property of multiplication over addition to mathematically validate the difference between the two expressions (DH).



Translation
"The [correct] option is A.
In block A they first multiply the answers by 1,755 then add 2,868, and in block B they add the answer by 2,868 then multiply by 1.755"

Figure 11. Solution to question 2.c by pair MM



Translation
"It's A because the idea is to bring the price of a cubic meter × the consumed gas + the fee, and not the consumed gas plus the fee × the price of a cubic meter"

Figure 12. Solution to question 2.c by pair HD

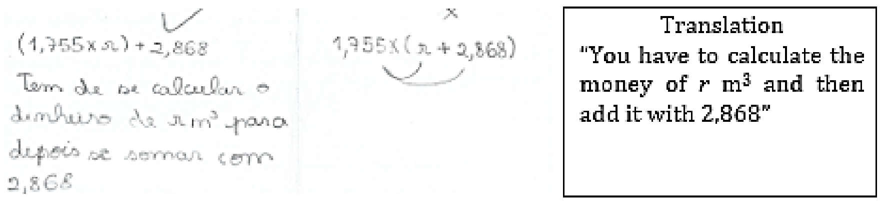


Figure 13. Solution to Question 2.c by pair CP

In question 2.d, all pairs successfully elaborated a flowchart (FR), defining an algorithm where most of them correctly identified the input data, procedures, and output data (AD). This is evident from the geometric shapes used in the flowchart construction (parallelograms for input or output data and rectangles for procedures). Pair HD (Figure 14, left) identifies the input data in the flowchart as a question, using the context of the described situation, and associates it with the variable *x*. For the procedure, it indicates the calculations to be performed in a generalized way, using the variable x (SG), and mentions the output data as a response associated with the result of the previously performed procedure. In the solution by pair CA (Figure 14, centre), which is very similar to the previous pair, there is an important reference to the need to store the value of the variable x, although this is not identified in the flowchart as an input data. Pair TM (Figure 14, right) identifies the values x as an input and y as an output, contextualizing them in parallel with the described situation and mentions the calculation to be performed using the variable x (SG).

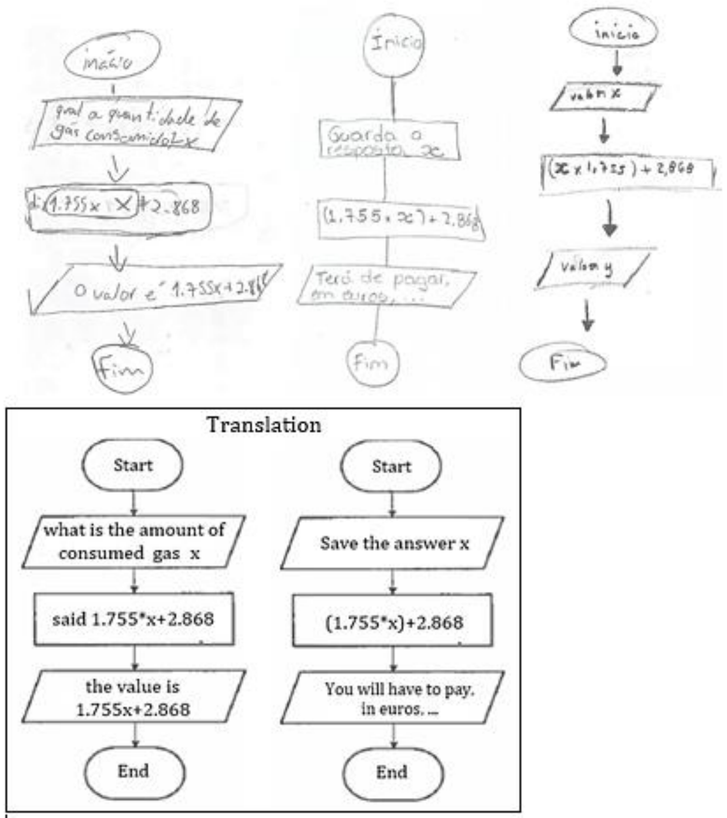


Figure 14. Solution to question 2.d by pairs HD (left), CA (centre), and TM (right)

In this second group of questions, the students managed to analyse a new way of representing a function, different from the usual representations, which we referred to as *coded representation*, contributing to the understanding of the concept of a non-linear affine function.

Group 3

In question 2.e, the pairs that arrived at the correct solution used the inverse operations reasoning, such as pair TM (Figure 15), recognizing similarities with other situations, such as determining term orders in sequences (PR). However, several students solved the question incorrectly, as was the case with pair CA, which attempted to use direct proportionality reasoning without considering the fixed rates (Figure 16, left), or pair JF, which, identifying the need to use inverse operations, did so in the wrong order (Figure 16, right).

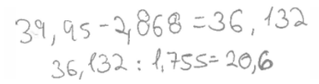


Figure 15. Solution to question 2.d by pair TM

Figure 16. Incorrect solutions to question 2.d by pairs CA (left) and JF (right)

In question 2.f, most of the solutions presented by the students are generalizations of the correct or incorrect reasoning used in the previous question, allowing for the identification of input and output data and the calculations to be performed (AD). Pair GT (Figure 17) correctly and contextually identified the data that needed to be entered into the program, as well as the results to be obtained and how to determine them (CG), taking care to specify the corresponding units. Pair CA (Figure 18), using the incorrect reasoning from the previous question, demonstrated that they were able to transform the simple rule of three into a generalization, despite their answer being wrong. Contrary to what was expected, students in group AC (Figure 19), who had used the simple rule of three without considering the fixed rate in the previous question, were unable to directly translate the rule of three reasoning into code. Instead, these students used the context of the described situation to correctly solve this question, although they did not define the meaning of the variable *x* in their solution, which would prevent the program from working correctly (SG).

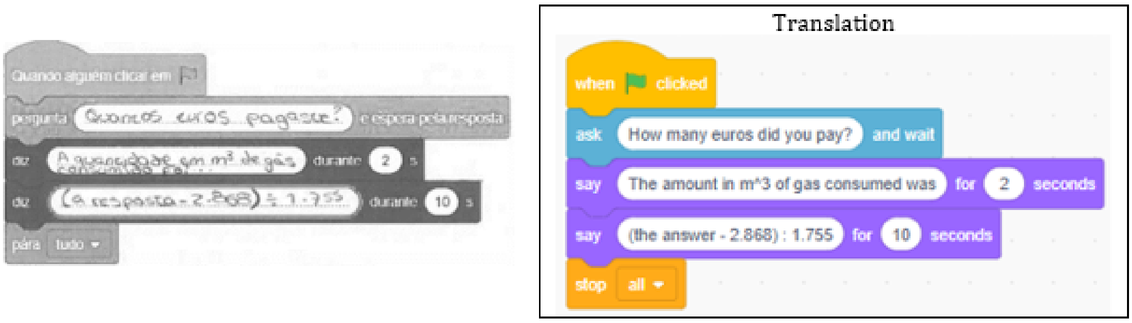


Figure 17. Solution of question 2.e by pair GT

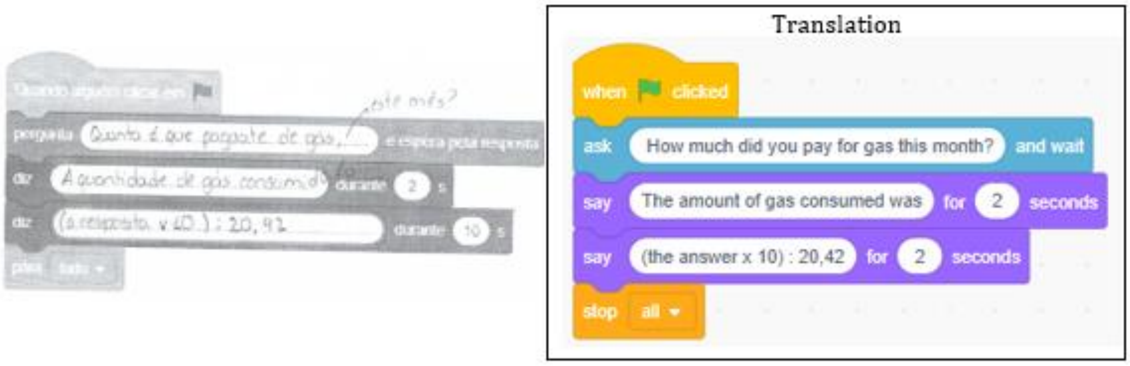


Figure 18. Solution of question 2.e by pair CA

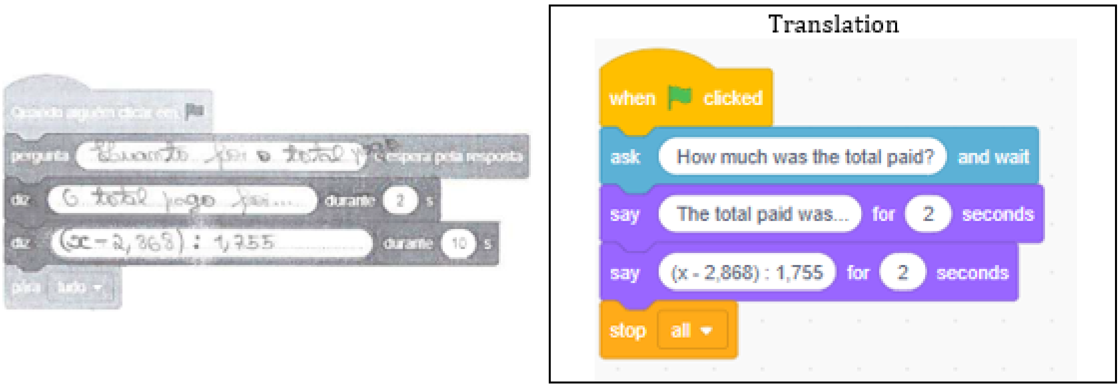


Figure 19. Solution of question 2.e by pair AC

In question 2.g, most students managed to test their reasoning by building the program according to the blocks they completed (DH). However, the students in pair CA, who had generalized the incorrect reasoning from question 2.e, only realized their mistake during the whole group discussion when they noticed they had forgotten to account for the fixed rates. Pair AC, when constructing their program, noticed something was wrong and managed to resolve the issue by creating the variable *x* to store the value of the response.

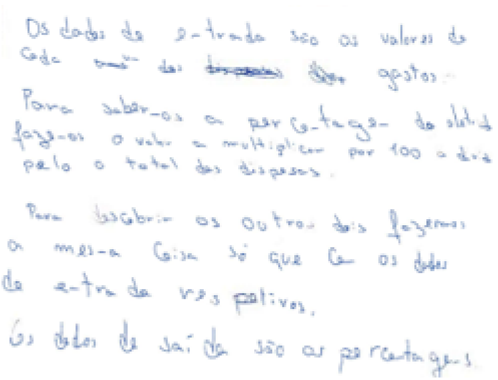
As students involved in this research already had some experience with Scratch and demonstrated code optimization skills (DH) in program construction, some of them mentioned that it was unnecessary to use the "say… for… seconds" block twice, suggesting the use of the "join… …" block instead. They also questioned why a variable had not been defined to store the response. These questions were discussed collectively, and the teacher-researcher explained them that that option was related to the possibility of the task being proposed to students with limited knowledge of Scratch programming.

In this third set of questions, most students, starting from the analysis of a concrete example of the described situation, managed to advance their reasoning to reach a contextualized generalization. They achieved this through the construction of a program to determine an object given an image in a nonlinear affine function and performed testing and validation of their reasoning.

Group 4

Most students exhibited difficulties in understanding the intended objective of this question. They focused on validating the specific situation presented in the task, using the given percentages to confirm the cost associated with each consumption, or using the final value to confirm the given percentages. Thus, initially they were not able to move from the specific values to a generalization. Considering this, the teacher-researcher suggested them to define a strategy using the specific values provided to verify the validity of the percentages presented in the question. They should identify the input and output data and, simultaneously, create a generic schema for the strategy they developed. This schema should allow them to calculate the percentages on any invoice they might come across.

After this, students were able to extend their reasoning beyond a particular case by defining an algorithm to model the described situation and using various types of representation. Pair JF used natural language (FR) to describe the generalization of their reasoning (CG), identifying the input and output data and the steps to follow (AD), as illustrated in Figure 20.



Translation

"The input data are the values of each expense. To know the percentage of electricity, we multiply the value by 100 and divide it by the total expenses. To discover the other two, we do the same thing but with the respective input data. The output data are percentages."

Figure 20. Explanation of the generalization constructed in question 3 by pair JF

Pair AC created a diagram with the concrete resolution (PR) and based on this representation, used a pre-symbolic language, combining natural language with mathematical symbols (FR, CG) to describe the procedures to be followed (AD) to generalize the described situation. They began by identifying the need to first determine (DE) the total expenses to be paid (Figure 21). Demonstrating a strong knowledge of Scratch programming and focusing on the construction of the program, pair HD (Figure 22) presented their explanatory diagram (AD, MM) using the blocks from this programming language. They identified the need for variables to store data related to consumption values, rates, and total expenses, which is the first operation they felt necessary to perform (DC).

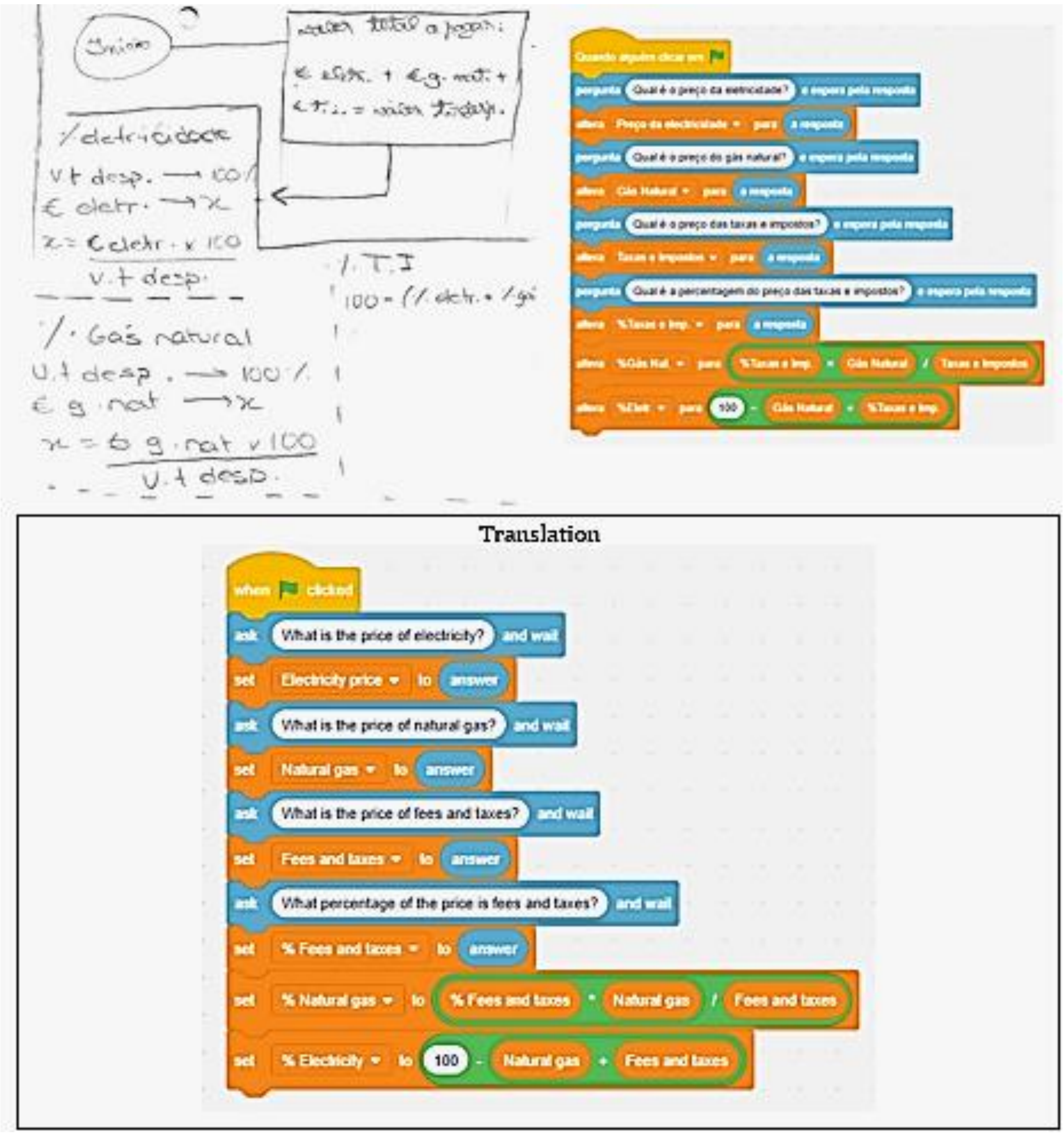


Figure 21. Explanatory diagram and program constructed in question 3 by pair AC

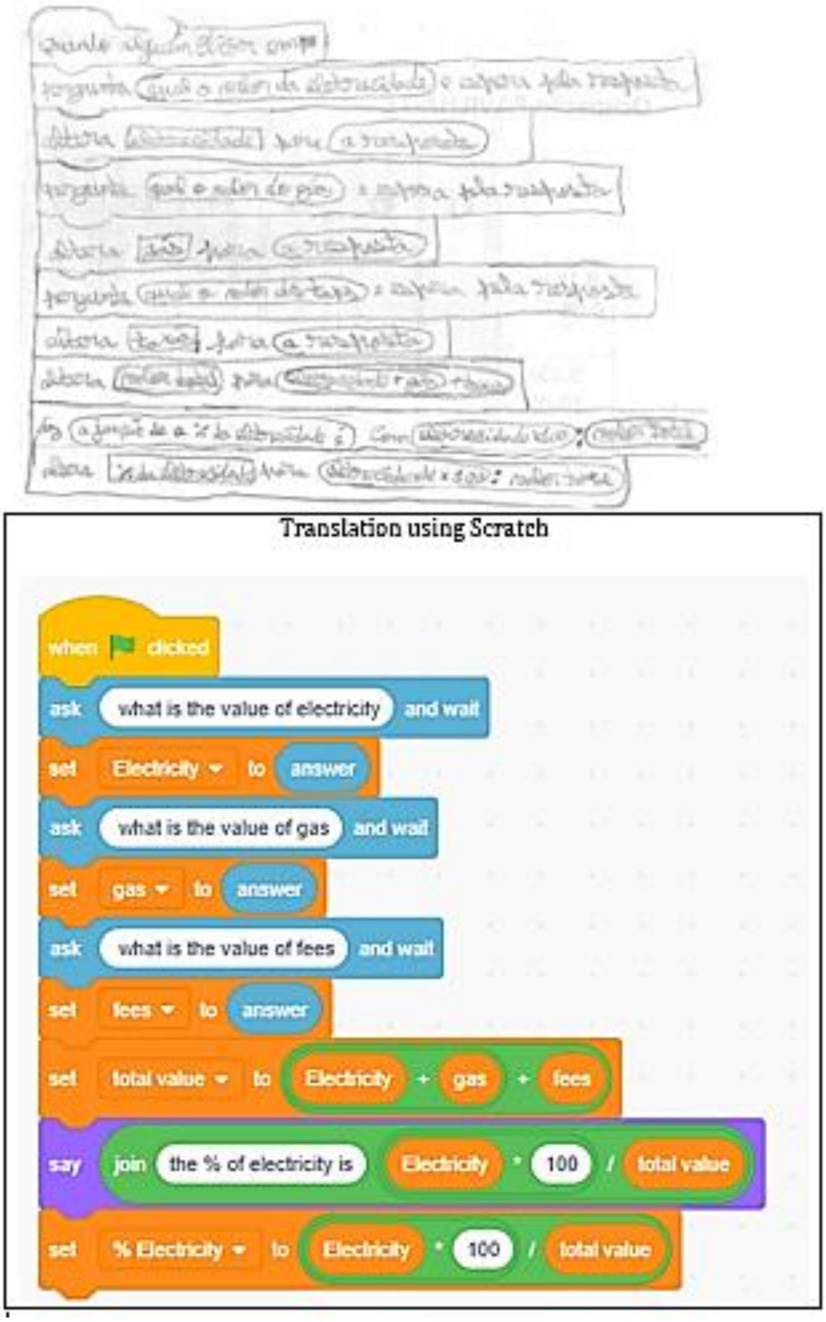
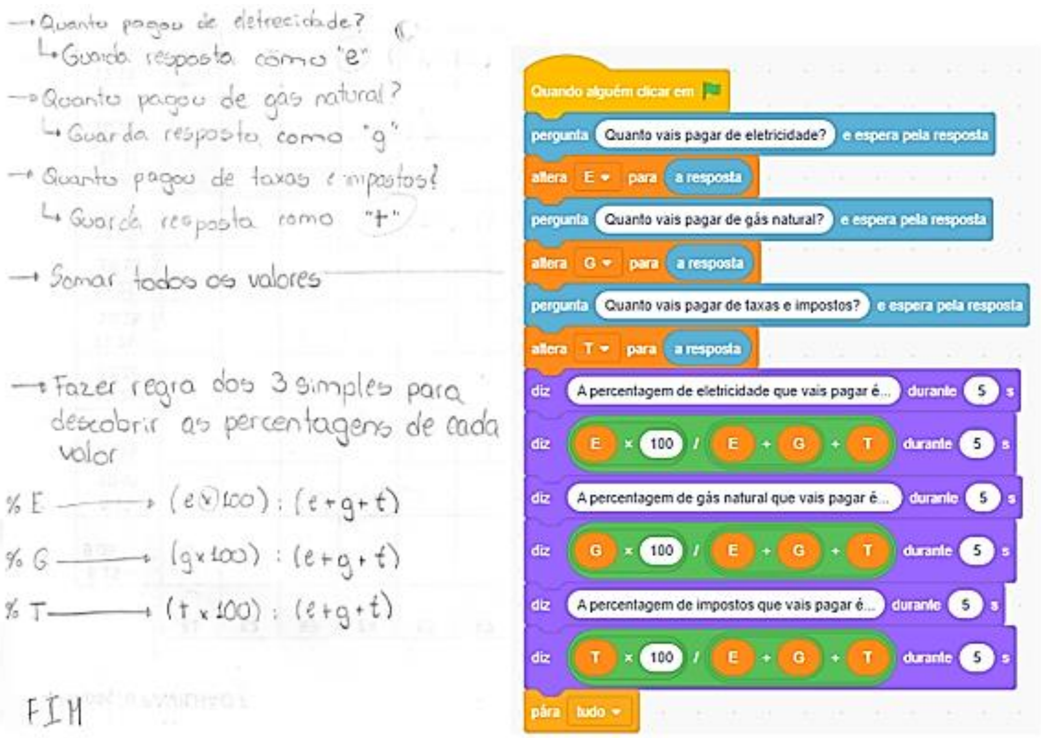


Figure 22. Explanatory diagram constructed in question 3 by pair HD

The pair CA (Figure 23) started by creating an explanatory diagram (AD, MM) using natural language to identify the procedures to be carried out. They also recognized the need to store input data and the result of summing these data into variables (DC). This approach was later used to generalize the direct proportionality reasoning from the rule of three (PR), employing an alphanumeric expression – symbolic mathematical language (FR, SG). After building the program in Scratch, few students tested it, possibly due to a lack of time or because they did not feel the need to test it, as the percentage values were already provided in the given image.



Translation

"→How much did you pay for electricity?
Save answer as "e"

→ How much did you pay for natural gas?
Save the answer as "g"

→ How much did you pay for fees and taxes?
Save the answer as "t"

→Add all values

→Use the rule of three to find out the percentages of each value"

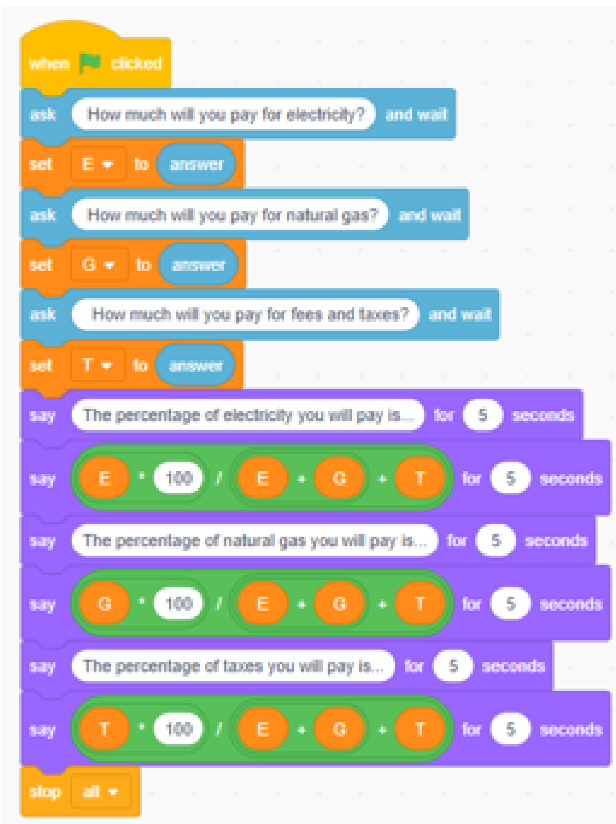


Figure 23. Explanatory diagram and program constructed in question 3 by pair CA

After some initial constraints in this final question of the task that were addressed by the guidance of the teacher-researcher, the students were able to model a real-world situation of direct proportionality (linear function), transforming this model into a program in Scratch. However, most students felt no need to validate the programs they have built based on the concrete example presented.

Discussion of results and conclusions

The task designed within this study addresses the need for diversifying students' learning experiences in mathematics, particularly to promote functional thinking (Pitta-Patanzi et al., 2020) and simultaneously to promote practices of computational thinking. It includes important mathematical processes such as justifying reasoning, expressing ideas, using various representations, and establishing connections between them (Swan, 2014), and it promotes the use of symbolic language, intra and extra-mathematical connections, and modelling real-world situations.

This study provided evidence of the use of various practices of computational thinking and dimensions of functional thinking in solving the integration task, that met the intentionality with which it was designed. However, in the final question of the task, students exhibited difficulties in prioritizing details and extracting the essential information from the presented situation to build a mathematical model. This issue may be related to the higher demands associated with the practice of abstraction in modelling situations, as this corresponds to a higher level of cognitive challenge (Qian & Choi, 2023). It is also noteworthy that students employed some practices of computational thinking not anticipated in questions of group 1 (decomposition, pattern recognition), in questions of group 2 (abstraction), and in question 3 (pattern recognition), as well as dimensions of functional thinking in group 2 (symbolic generalization).

When solving the different parts of the task, students used multiple representations (natural language, diagrams, flowcharts, symbols, formulas, and coded representations) to organize and represent their reasoning, an aspect of paramount importance to strengthen students' functional thinking. It is noteworthy that in the final question, which involved constructing a program in Scratch, some groups used the unconventional form of representing generalization introduced in this study (coded representation) to justify their reasoning before building the program, even though that was not required. Thus, these students were able to give meaning to their ideas in increasingly abstract ways, progressing between natural and algebraic language. This new representation may serve as a scaffold to facilitate their transition to symbolic language. Additionally, by introducing the notion of affine function in a real-life context, supported by the use of concrete quantities like fixed and the variable component of the gas consumption situation, the task may contribute to mitigate the difficulties students typically face with functions when introduced in an abstract manner (Warren & Cooper, 2005).

Globally, students demonstrated the ability to make connections between concrete examples and more general situations, which facilitated a deeper understanding of the relationships between variables—something that the research points out as being a difficulty for students (Wilkie & Ayalon, 2018). Finally, the modelling question in the task allowed students to move beyond using the rule of three as a procedure for relating values, enabling them to generalize this procedure and thereby enhance their functional thinking.

From the results of this study, we may argue that the integration task that starts with less demanding and unplugged questions, followed by questions involving the modification of existing objects and their validation, and then cognitively challenging questions that require establishing connections between concepts, provided the necessary support for students' functional thinking in relation to a new type of functions. However, the ultimate purpose of adapting knowledge to new problem situations was not initially achieved as intended, therefore points out to the need of continuous contact of students with this type of tasks.

In line with what Ye et al. (2023) discussed regarding the introduction of computational thinking in mathematics, the task proposed at the beginning of the teaching experiment intends to illustrate a possible way of teaching functions in connection with computational thinking. However, this new teaching approach must be carefully crafted in order not to add to the students' difficulties (Chan et al., 2021) related to programming. Thus, although the proposed task allows students to develop computational thinking practices, it does not require programming knowledge that could detract attention from the task's mathematical focus or to make it too demanding for them.

This study offers insights into the design of tasks that integrate computational thinking and functions, enabling students to move beyond mere algebraic manipulations and fostering the establishment of both intra and extra-mathematical connections. Additionally, similar to other non-conventional representations as those mentioned by Pinto et al. (2022), the coded representation introduced in the task helps enrich students' mathematical learning as it allows them to communicate their reasoning in less abstract way when they are still building their understanding of this type of functions. As a matter of fact, they assumed the use of the response block in Scratch as representing an arbitrary value (in association with a variable) or faced the necessity of defining variables in the program assigning them one meaning that comes from concrete and authentic situation in the task statement, thus developing contextual generalizations (Radford, 2006).

Being an exploratory study, there are some limitations to be considered. First, we recognize that the task's options concerning the use of Scratch were somewhat limited and that future research could go deeper in envisioning new ways of supporting students in develop functional thinking in that context of block-programming. The same can be said concerning promoting computational problem solving as the ones proposed by Cui et al. (2023), since the task in the present study is mainly closed in its structure and some question direct students to specific procedures. Nevertheless, the same authors point out that students often become stuck if tasks demand long procedures and provide no feedback, therefore some balance is needed concerning the tasks' structure.

Centring in only one task, the first of the teaching experiment, the findings of this study are still tentative and limited, but further analysis of students' functional thinking may provide a better understanding of the potentialities of approaching functions in articulation with computational thinking. In this regard, an understanding of how students comprehend variables in programming language and in the context of the mathematical functions is another aspect that was not analysed in this study but that has great importance (Bråting & Kilhamn, 2021).

Corroborating the idea of Hickmott et al. (2018), the introduction of computational thinking in mathematics presents many challenges but also opportunities to enhance students' mathematical understanding. Indeed, integrating computational thinking into mathematics will bring changes in the types of tasks proposed to students (Kallia et al., 2021), as well as in the reasoning processes they use when solving these tasks. This study, conducted in real classroom settings and reflecting the complexity of real-world teaching conditions, may contribute to the development of new educational resources. It may also support further research on functional thinking, which remains underrepresented in studies integrating computational thinking into mathematics learning.

Acknowledgements

This research was supported by the municipality of Oeiras, Portugal, with a scholarship to the first author, and by National Funds through FCT–Portuguese Foundation for Science and Technology, I.P., under the scope of UIDEF–Unidade de Investigação e Desenvolvimento em Educação e Formação, UIDB/04107/2020, https://doi.org/10.54499/UIDB/04107/2020.

References

Barcelos, T. S., Muñoz, R., Villarroel, R., & Silveira, I. F. (2018). A systematic literature review on relationships between computational thinking and mathematics. *Journal on Computational Thinking, 2*(1), 23-35. https://doi.org/10.14210/jcthink.v2.n1.p23

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

Bell, T., & Vahrenhold, J. (2018). CS unplugged—How is it used, and does it work? In *Lecture Notes in Computer Science* (Vol. 11011 LNCS, pp. 497–521). Springer. https://doi.org/10.1007/978-3-319-98355-4_29

Best, M., & Bikner-Ahsbahs, A. (2017). The function concept at the transition to upper secondary school level: tasks for a situation of change. *ZDM - Mathematics Education*, *49*(6), 865–880. https://doi.org/10.1007/s11858-017-0880-6

Blanton, M. L., & Kaput, J. J. (2011). Functional thinking as a route into Algebra in the elementary grades. In J. Cai, & E &. Knuth (eds.), *Early Algebraization*: *A global dialogue from multiple perspectives* (pp. 5–23). Springer. https://doi.org/10.1007/978-3-642-17735-4_2

Blum, W., & Borromeo Ferri, R. (2009). Mathematical modelling: Can it be taught and learnt? *Journal of Mathematical Modelling and Application, 1*(1), 45-58.

Bråting, K., & Kilhamn, C. (2021). Exploring the intersection of algebraic and computational thinking. *Mathematical Thinking and Learning*, *23*(2), 170–185. https://doi.org/10.1080/10986065.2020.1779012

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, 1*. https://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

Cabral, J., Mendes, F., & Oliveira, H. (2022). A capacidade de *noticing* do pensamento algébrico dos alunos: um estudo na formação inicial. *Quadrante*, *31*(1), 28–53. https://doi.org/10.48489/quadrante.27091

Caeli, E. N., & Yadav, A. (2020). Unplugged approaches to computational thinking: A historical perspective. *TechTrends*, *64*(1), 29–36. https://doi.org/10.1007/s11528-019-00410-5

Canavarro, A. P., Mestre, C., Gomes, D., Santos, E., Santos, L., Brunheira, L., Vicente, M., Gouveia, M. J., Correia, P., Marques, P. M., & Espadeiro, R. G. (2021). *Aprendizagens Essenciais de Matemática: 8.º ano*. Direção-Geral da Educação. http://www.dge.mec.pt/sites/default/files/Curriculo/Aprendizagens_Essenciais/3_ciclo/ae_mat_8.o_ano.pdf

Chan, S. W., Looi, C. K., Ho, W. K., Huang, W., Seow, P., & Wu, L. (2021). Learning number patterns through computational thinking activities: A Rasch model analysis. *Heliyon, 7*(9). https://doi.org/10.1016/j.heliyon.2021.e07922

Chan, S. W., Looi, C. K., Ho, W. K., & Kim, M. S. (2023). Tools and approaches for integrating Computational Thinking and Mathematics: A Scoping Review of Current Empirical Studies. *Journal of Educational Computing Research, 60*(8), 2036–2080. https://doi.org/10.1177/07356331221098793

Cobb, P., Confrey, J., diSessa, A., Lehrer, R., & Schauble, L. (2003). Design experiments in education research. *Educationa Researcher, 32*(1), 9–13.

Cui, Z., Ng, O., & Jong, M. S.-Y. (2023). Integration of Computational Thinking with Mathematical Problem-based Learning: Insights on Affordances for Learning. *Educational Technology & Society, 26*(2). 131-146. https://doi.org/10.30191/ETS.202304_26(2).0010

Evaristo, I. S., Terçariol, A. A. L., & Ikeshoji, E. A. B. (2022). Do pensamento computacional desplugado ao plugado no processo de aprendizagem da Matemática. *Revista Latinoamericana de Tecnología Educativa*, *21*(1), 75–96. https://doi.org/10.17398/1695-288X.21.1.75

Gomes, T. C. S., Falcão, T. P., & Tedesco, P. C. D. A. R. (2018). Exploring an approach based on digital games for teaching programming concepts to young children. *International Journal of Child-Computer Interaction, 16*, 77-84. https://doi.org/10.1016/j.ijcci.2017.12.005

Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A Scoping Review of Studies on Computational Thinking in K–12 Mathematics Classrooms. *Digital Experiences in Mathematics Education*, *4*(1), 48–69. https://doi.org/10.1007/s40751-017-0038-8

Israel, M., & Lash, T. (2020). From classroom lessons to exploratory learning progressions: mathematics + computational thinking. *Interactive Learning Environments*, *28*(3), 362–382. https://doi.org/10.1080/10494820.2019.1674879

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

Kieran, C., Doorman, M., & Ohtani, M. (2015). Frameworks and Principles for Task Design. In A. Watson, & M. Ohtani (Eds.), *Task Design in Mathematics Education: an ICMI Study 22* (pp. 19–81). Springer. https://doi.org/10.1007/978-3-319-09629-2_2

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A Pedagogical Framework for Computational Thinking. *Digital Experiences in Mathematics Education*, *3*(2), 154–171. https://doi.org/10.1007/s40751-017-0031-2

Martins, R., Viseu, F., & Rocha, H. (2023). Functional thinking: A study with 10th grade students. *Education Sciences, 13*(4), 335. https://doi.org/10.3390/educsci13040335

Menezes, L., Oliveira, H., & Canavarro, A. P. (2015). Inquiry-based teaching: The case of Célia. In U. Gellert, J. Gimenez Rodrigues, C. Hahn, & S. Kafoussi (Eds.), *Educational paths to Mathematics* (pp. 305-321). Springer International Publishing. https://doi.org/10.1007/978-3-319-15410-7_20

Nordby, S.K., Bjerke, A.H., & Mifsud, L. (2022). Computational thinking in the primary mathematics classroom: A systematic review. *Digital Experiences in Mathematics Education, 8*, 27–49. https://doi.org/10.1007/s40751-022-00102-5

Pinto, E., Cañadas, M. C., & Moreno, A. (2022). Functional relationships evidenced and representations used by third graders within a functional approach to Early Algebra. *International Journal of Science and Mathematics Education, 20*, 1183–1202. https://doi.org/10.1007/s10763-021-10183-0

Pitta-Pantazi, D., Chimoni, M., & Christou, C. (2020). Different types of algebraic thinking: An empirical study focusing on middle school students. *International Journal of Science and Mathematics Education*, *18*, 965-984. https://doi.org/10.1007/s10763-019-10003-6

Qian, Y., & Choi, I. (2023) Tracing the essence: ways to develop abstraction in computational thinking. *Education Technology Research and Development, 71*, 1055–1078. https://doi.org/10.1007/s11423-022-10182-0

Radford, L. (2006). Algebraic thinking and the generalization of patterns: A semiotic perspective. In S. Alatorre, J. L. Cortina, M. Sáiz, & A. Méndez (Eds.), *Proceedings of the Twenty Eighth Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education* (Vol. 1, pp. 2–21). PME. http://www.luisradford.ca/pub/60_pmena06.pdf

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, *72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Smith, E. (2008). Representational thinking as a framework for introducing functions in the elementary curriculum. In J. Kaput, D. W. Carraher, & M. L. Blanton (Eds.), *Algebra in the early grades* (pp. 133-160). Lawrence Erlbaum Associates.

Swan, M. (2014). Designing tasks and lessons that develop conceptual understanding, strategic competence and critical awareness. In A. Boavida, C. Delgado, E. Santos, F. Mendes, J. Brocardo, J. Duarte, L, Santos, M. Baía, & M. Figueiredo (Eds.), *Encontro de Investigação em Educação Matemática* (pp. 15-31). Sociedade Portuguesa de Investigação em Educação Matemática.

Warren, E., & Cooper, T. (2005). Young children's ability to use the balance strategy to solve for unknowns. *Mathematics Education Research Journal, 17*(1), 58-72. http://dx.doi.org/10.1007/BF03217409

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Wilkie, K.J., Ayalon, M. (2018). Investigating Years 7 to 12 students' knowledge of linear relationships through different contexts and representations. *Mathematic Education Research Journal, 30*, 499–523. https://doi.org/10.1007/s13394-018-0236-8

Wing, J. M. (2006). Computational thinking. In *Communications of the ACM* (Vol. 49, Issue 3, pp. 33–35). Association for Computing Machinery. 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*(3). https://doi.org/10.1186/s40594-023-00396-w

Appendix

Task 1: "Natural gas versus propane gas"

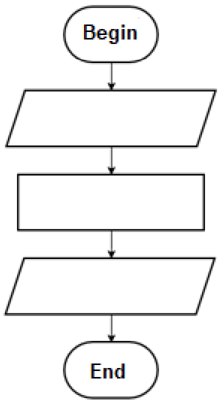
Gas is a very common fuel in our homes, however, the type of gas available for consumption varies depending on the region of the country. In Portugal, the use of two types of gas prevails: bottled/piped propane gas, obtained from petroleum refinement processes, which are quite pollutant, and piped natural gas, which despite being a fossil fuel, emits less CO_2 in its combustion.

https://www.edp.pt/particulares/content-hub/gas-natural-ou-gas-propano/

1. In the interior of the country, there is no piped gas available, so propane gas bottles are used. Each bottle costs approximately €33 and contains the equivalent of 11 m^3 of natural gas.

a) In this context, if we consume the equivalent of 10 m^3 of natural gas in a month, what cost will this consumption correspond to? Justify your answer.

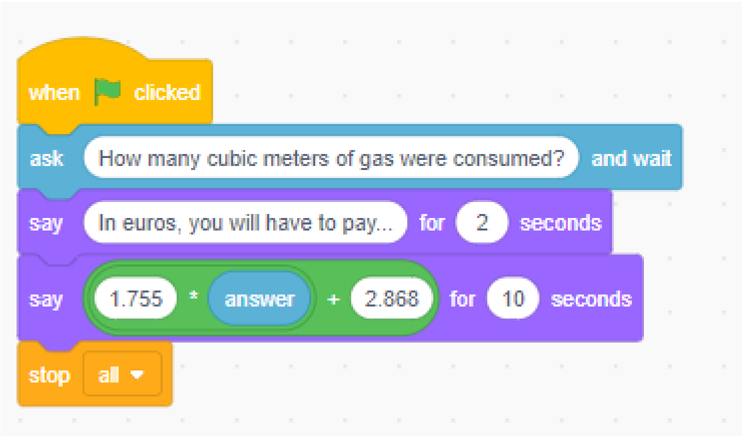
b) Complete the following flowchart that allows you to determine the cost of gas, in euros, depending on the amount of gas consumed, in cubic meters.



c) Write an algebraic expression that allows you to determine the cost of gas, y, in euros, depending on the amount of gas consumed, x, in cubic meters.

2. Due to convenience, safety and price, in recent years piped natural gas has reached more and more homes. In the natural gas bill, there is a variable component, associated with gas consumption in cubic meters, and a fixed component, relating to several fees, namely the fixed term, access to the network and occupation of the underground.

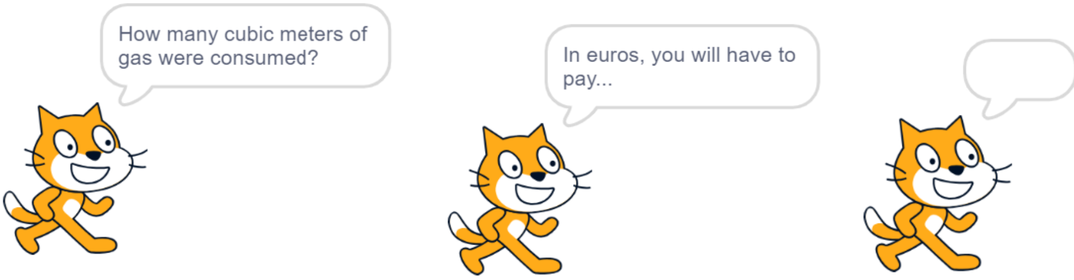
For the company that supplies the gas to automatically issue gas invoices, it is necessary to enter the number of cubic meters that were consumed in a month, real or estimated. Regarding this situation, the following program was built in Scratch.



a) In this context of gas services, what do the values 1,755 and 2,868 mean?

Note: In Scratch the dot means a comma.

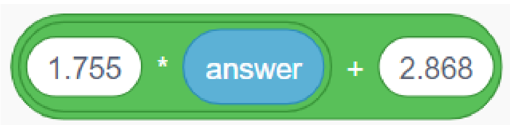
b) When the program is executed, a dialog box for data entry appears at the beginning and, at the end, a data output dialog box, as illustrated in the images below. Fill in the blanks, knowing that the images correspond to the situation of a consumption of 10 m^3 of natural gas this month. Justify your reasoning.



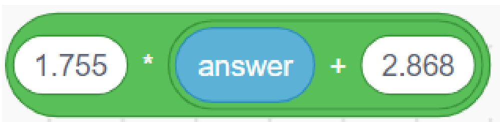
Access the QR code beside to confirm your answer.

c) When building the program presented above in Scratch, two colleagues found that their programs were not reaching the same conclusion, however, for both of them everything seemed to be correct. Consider the two sets of blocks in the penultimate line of code of the program built by colleagues. Choose the correct option and explain the difference between them.

Student A blocks



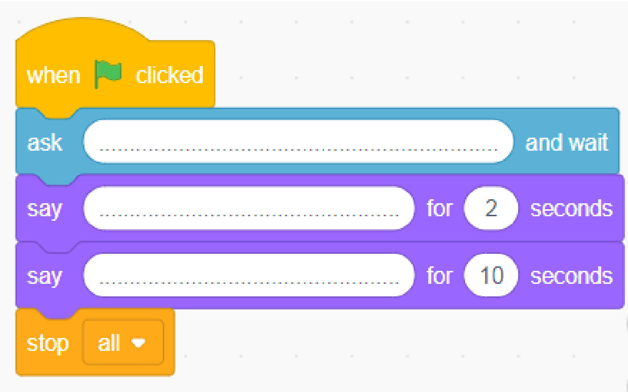
Student B blocks



d) Build a flowchart that reflects the presented situation in this program, using the variable *x* to represent the amount of natural gas consumed this month, in cubic meters.

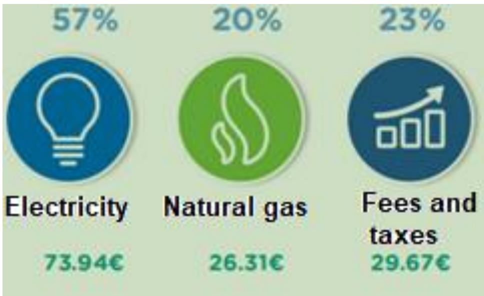
e) If the monthly fee is **€39,95**, how many cubic meters of natural gas have been consumed on that month? Explain your thoughts.

f) Complete the blanks below in order to build a program that automatically determines the amount of cubic meters of gas consumed during a month, knowing the total amount paid in euros.



g) Access the Scratch website (https://scratch.mit.edu/) to create this program. To do this, click "Create" and build the lines of code. Confirm the result obtained in paragraph e).

3. Currently, it is very common for consumers to opt for a single company that simultaneously provides electricity and natural gas services. The invoice below contains details of electricity and natural gas consumption and the fees and taxes associated with both. Furthermore, at the beginning of the invoice, the total values of electricity and natural gas consumption and the respective fees and taxes are indicated, as well as the percentages associated with these costs, as illustrated in the figure on the side.



Build a program in Scratch that allows you to verify the percentages shown in this figure. *Suggestion*: Start by drawing a flowchart or an explanatory diagram for the described situation.