Author: Fernando Pinho

2023 International Symposium on Computers in Education (SIIE) | 979-8-3503-2931-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/SIIE59826.2023.10423695

XXV SIIE

Internacional Symposium on Computers in Education

Setúbal, November, 16 – 18, 2023

Editors
Maria do Rosário Rodrigues, Miguel Figueiredo, João Torres

POLITECNICO SETUBAL
POLYTECHNIC UNIVERSITY

IEEE Education Society

IEEE PORTUGAL SECTION

IEEE SECCIÓN ESPAÑA

- Reto de ecuaciones, an app to support mathematics lessons

Célia Mestre
- The emergence of algorithmics in grade 1: a study framed in the new Portuguese Mathematics curriculum

Cristiana Araújo
- Neuroeducation: Guidelines for Computational Thinking training
- OntoCnE, characterizing Learning Resources for training Computational Thinking

Cristina Chuva Costa
- Class Type Preferences in Informatics Engineering: before, during, and after the pandemic

Cristina Martins
- The emergence of algorithmics in grade 1: a study framed in the new Portuguese Mathematics curriculum

Daniela Rocha Bicalho
- The Use of Immersive Virtual Reality in Educational Practices in Higher Education: A Systematic Review

David de Vicente-Peña
- A Tracer for Teacher Development in the Programming Language ScratchJr

David Nadler Prata
- SCHOOL DROPOUT IN THE FEDERAL NETWORK EDUCATION OF BRAZIL: IS IT AN INHERENT INDIVIDUAL ATTRIBUTE OR IT LIES ON SETTING CONDITIONS?

Dimitra Sofianou
- HerStory: combating gender stereotypes by highlighting women's contribution to the history of societies

Dina Rocha
- e-Campus: concept and design of a virtual campus to support Distance Learning in a Polytechnic Institute

Eduarda Ferreira
- Young people's digital competences: does gender matter?

Emília Bigotte de Almeida
- PRO(g)NATURA: technologie and outdoor education as a complement to improve the child learning process

Emily Psara
- HerStory: combating gender stereotypes by highlighting women's contribution to the history of societies

XXV International Symposium on Computers in Education
SIIE 2023
Setúbal, November 16-18

Table of Contents

E-Assessment Systems: An Evaluation Framework from the Perspective of Higher Education Experts - Pedro Isaias, Paula Miranda, Sara Pifano

CONTRADICTIONS IN THE ADOPTION OF DISTANCE EDUCATION IN BRAZILIAN EDUCATION DURING COVID-19 - Cecília C Vieira, Patrícia Ramiro Silva Souza, Fabrício Santos Dias de Abreu

Project-Based Learning in the curricular unit ICT in Professional Contexts – perceptions of Sociocultural Animation students - Sílvia Roda Couvaneiro, João Grácio, Ana Rute Martins

Young people's digital competences: does gender matter? - Eduarda Ferreira, Maria João Silva

Educational robotics and programming in inclusive educational settings: a scoping review
Ana Claudia Loureiro, Manuel Meirinhos

POTENTIALS OF USING GAMIFICATION IN AN OPEN AND MASSIVE ONLINE COURSE
Mariella Berger Andrade, Marize Lyra Silva Passos

The portfolio as a tool for learning and assessment in the Internship in Teaching Informatics
Ana Claudia Loureiro, Ana Isabel Santos

Pedagogical Training of Higher Education Teachers: Communication and Collaboration with Digital Technologies - Ana Luisa Oliveira Pires, Maria do Rosário Rodrigues

Game-Based Learning: A Strategy for Teaching Curriculum Content in the Classroom Context - João M Batista

GeoGebra applet to learn programming and debugging in mathematics lessons - Wahid Yunianto, Guillermo Bautista Jr, Rully Charitas Indra Prahmana, Zsolt Lavicza

Participatory design of a high-fidelity prototype of an essential learning curricular tool - Bianor Valente, Maria João D Silva, Pedro Sarreira

STEM Career Aspirations Among Portuguese Secondary School Students - Teresa Ribeirinha, Marisa Correia, Mónica Baptista

OntoCnE, characterizing Learning Resources for training Computational Thinking - Cristiana Araújo, Pedro Rangel Henriques, João José Cerqueira

2023 International Symposium on Computers in Education (SIIE) | 979-8-3503-2931-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/SIIE59826.2023.10423689

In heaven as on earth: The performance of students is as good as it is the digraph that describes their behavior - Luis Castillo, F. Sánchez-Carrascosa, M. Ruiz-Martínez, J. Fdez-Olivares

Promoting global awareness of the SDGs in the classroom through the SDGCalendar platform Ezequiel Jiménez-García , M. Carmen de Castro-Cabrera, Sara Balderas-Díaz, Gabriel Guerrero-Contreras

The Use of Immersive Virtual Reality in Educational Practices in Higher Education: A Systematic Review - Daniela Rocha Bicalho, João Manuel Nunes Piedade, João Filipe de Lacerda Matos

Artificial Intelligence and Academic Research: An Exploratory Study with Master's Students - Teresa Gouveia, Rui Raposo, Nídia Salomé Morais

Rio Doce Watershed - socio-historical, economic and environmental aspects: hybrid continuing education in the Rio Doce Escolar Project - Manoel Augusto Polastreli Barbosa, Antonio Donizetti Sgarbi

Integration of Large Language Models into Higher Education: A Perspective from Learners - Katerina Zdravkova , Fisnik Dalipi, Fredrik Ahlgren

Assessment of learning Mediated by Digital Technologies in the Context of Environmental Education: Teacher Training through a MOOC Course - George Bassul Areias, Isaura Alcina Martins Nobre, Marize Lyra Silva Passos

Using Microworlds to Teach Basic Programming Concepts to Future Early Childhood Education Teachers - Sergio Cavero, Liliana Patricia Santacruz Valencia

PRO(g)NATURA: technologie and outdoor education as a complement to improve the child learning process - Isabel Duque, Ricardo Almeida, Marlene Migueis, Emília Bigotte de Almeida

SCHOOL DROPOUT IN THE FEDERAL NETWORK EDUCATION OF BRAZIL: IS IT AN INHERENT INDIVIDUAL ATTRIBUTE OR IT LIES ON SETTING CONDITIONS? - Jabson Cavalcante Dias, Tadeu Lucena da Silva, Marco Antônio Juliatto, Adriano Nascimento da Paixão, David Nadler Prata

Digital Educational Games for learning mathematics - Catarina Delgado, Fátima Mendes

Videojuego Educativo "Scratch Estadístico" para la enseñanza de Estadística - Antonio de la Hoz Ruiz, Hijón Neira, Raquel

Reconfigurable Metadata Schemas with Multivalued Elements: Towards Educational Repository Management Through Formal Grammars on the Clavy Platform - Joaquín Gayoso-Cabada, Mercedes Gómez-Albarrán, José Luis Sierra Rodríguez,

Collaborative learning in higher education: the role of Padlet in developing academic writing skills - Mariana O Pinto

Neuroeducation: Guidelines for Computational Thinking training - Cristiana Araújo, Pedro Rangel Henriques, João José Cerqueira

A Tracer for Teacher Development in the Programming Language ScratchJr - J. Ángel Velázquez-Iturbide, David de Vicente-Peña, Eva López-Puente

Teach the importance of logic (programming) in Computer Science and why it is important - Joaquín Arias

HerStory: combating gender stereotypes by highlighting women's contribution to the history of societies - Lucía García-Holgado, Alicia García-Holgado, Sonia Verdugo-Castro, Erika García Silva, Francisco J. García-Peñalvo, Antrea Kosta, Dimitra Sofianou, Emily Psara, Miguel Á. Conde

e-Campus: concept and design of a virtual campus to support Distance Learning in a Polytechnic Institute - Ana Loureiro, Inês Messias, Dina Rocha

The emergence of algorithmics in grade 1: a study framed in the new Portuguese Mathematics curriculum - Célia Mestre, Isabel Guerra, Cristina Martins, Cândida Tourais

Reflections on a technology curricular unit in initial teacher training: the students' perspective - Maria do Rosário Rodrigues, João Vítor Torres, João Carlos Grácio

Reflections on an educational robotics project in kindergarten and in primary school education - João Grácio, João Torres, Maria do Rosário Rodrigues, Ana Chambel, Miguel Figueiredo

Exploring the Potential of Modern Board Games to Support Computational Thinking - Fábio Daniel Machuqueiro; João Piedade

A study on the Motivation of Computer Science Students to Learn Programming - Leonardo Silva , Anabela Gomes, Ana Rosa Borges, Verónica Vasconcelos, António José Mendes

The use of Pixton software as a pedagogical resource in the teaching and learning process - Henrique Gil, Joana Vitorino

Contributions and potential of the WheelDecide digital application in a Basic Education class - Henrique Gil, Rute Mateus,

Class Type Preferences in Informatics Engineering: before, during, and after the pandemic - Cristina Chuva Costa, Anabela Gomes

Teacher training at the Federal University of ABC during the Covid 2019 pandemic: Results, Challenges and Future Perspectives - CAROLINA C CARVALHO, Carla Lopes Rodriguez, Marcella dos Santos Abreu

Application for the academic management of supervised clinical practicals for the Degree in Nursing - Rosa M Carro, Pablo Izaguirre, María Isabel Guzmán Almagro, Ana Isabel Parro-Moreno

Reto de ecuaciones, an app to support mathematics lessons - Pablo Delgado-Rodriguez, Raquel Hijón Neira, Celeste Pizarro Romero

POSITIVE AND NEGATIVE FACTORS AFFECTING THE INITIATIVE FOR THE INCLUSION OF ICT IN THE SECONDARY EDUCATION CURRICULUM IN NAMIBE-ANGOLA - Santana BUNGA

Edited by

Maria do Rosário Rodrigues [1]

Miguel Figueiredo [1]

João Torres [1]

(1) Setúbal Polytechnic University, High School of Education

The emergence of algorithmics in grade 1: a study within the mathematical curriculum

Célia Mestre
Instituto Politécnico de Setúbal
Escola Superior de Educação
Portugal
celia.mestre@ese.ips.pt

Cristina Martins
Centro de Investigação em Educação Básica
Instituto Politécnico de Bragança, Portugal
mcesm@ipb.pt

Cândida Tourais
Agrupamento de Escolas de Azeitão
Portugal
candida.tourais@sapo.pt

Isabel Guerra
Agrupamento de Escolas Miguel Torga, Bragança
Portugal
zabelguerra@gmail.com

2023 International Symposium on Computers in Education (SIIE) | 979-8-3503-2931-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/SIIE59826.2023.10423676

***Abstract*— This article presents a study carried out with two grade 1 classes, in the context of implementing the new Mathematics' curriculum. A sequence of tasks was carried out with the purpose of developing computational thinking, practices articulated with learning about the topic Numbers, and an attempt was made to identify the five practices of computational thinking in the students' activities. We ascertained that the students mobilized all the practices, the emergence of algorithmics coming to the fore and, at the same time, these practices allowed them to delve deeper into their mathematical knowledge in the content explored in the topic Numbers.**

***Keywords*— Computational Thinking, algorithmics, new Mathematics' curriculum, Numbers**

I. Introduction

Computational thinking is presented as a cross-sectional skill in the elementary school mathematics curriculum for the first time in Portugal. The new curriculum, in the format of Essential Mathematical Learnings for Basic Education (Canavarro et al., 2021), henceforth referred to simply as new curriculum, integrates both mathematical knowledge and six transversal mathematical skills, including computational thinking, as learning content.

Considered as a problem-solving process, computational thinking is defined in the new curriculum through a set of five cornerstone practices: abstraction, decomposition, pattern recognition, algorithmics and debugging. In this article we will focus on the deployment of these practices in the development of a sequence of tasks implemented in two 1st grade classes, as part of the operationalization of the new curriculum. Through the activities that the students carried out while exploring the tasks in the sequence, we will try to identify the practices of computational thinking and present evidence of their mobilization, reflecting on how the students developed their computational thinking, in conjunction with the mathematical learning framed in the Numbers topic.

II. Theoretical Framework

According to Wing (2006), computational thinking skills are fundamental and should be developed by any student, alongside with other essential skills such as reading, writing or arithmetic. Computational thinking goes far beyond the ability to program "because it focuses on conceptualization" and requires "thinking at multiple levels of abstraction" (Wing, 2006, p. 35), being a form of thinking that humans use, not computers. It can be regarded as a thought process that involves formulating problems and finding solutions, solutions that can be represented in a way that can be effectively carried out by an information processing agent, such as a computer, but not necessarily (Wing, 2010). Thus, it is assumed that the use of technology or even programming is not an essential condition for the development of computational thinking, although their relevance and purpose are recognized. Similarly, it is also considered that the use of technology does not necessarily imply that the development of computational thinking is being promoted.

The new curriculum presents the development of computational thinking skills as one of the eight general learning objectives, assuming that this "presupposes the development, in an integrated way, of practices such as abstraction, decomposition, pattern recognition, analysis and definition of algorithms, and the development of debugging and process optimization habits" (Canavarro et al., 2021, p. 3), practices already referred to as characteristics of this type of thinking by authors such as Angeli et al. (2016) or Yadav et al. (2019). The document also states that these practices "are essential in mathematical activity and provide students with tools that allow them to solve problems, especially those related to programming" (idem). Also, one of the authors of the new curriculum, Espadeiro (2021) frames and systematizes the understanding bestowed upon each of these practices.

- Abstraction aims at reducing the complexity of a task or problem, or at identifying general principles that can be applied to similar situations or problems;

- Decomposition deals with managing complex tasks or situations by breaking them down into smaller, more manageable parts;
- Pattern recognition involves recognizing regularities and relationships;
- Algorithmics allows one to develop a step-by-step solution to a given problem (resolution steps) or establish rules (conditions) to be followed to solve the problem; and,
- Debugging involves looking for and correcting errors, as well as testing, verifying, refining and optimizing the solution presented (p. 6).

For students to develop computational thinking, it is essential that they mobilize and interrelate these practices (Angeli et al., 2016) when exploring a mathematical task. However, their natural interrelationship and dependence does not make it easy to recognize each one in isolation, nor is it considered necessary from the student's perspective. On the other hand, it will be important for the teacher to recognize each of these practices in the students' work so that they can intentionally promote the development of all of them in an integrated manner. Thus, in this article, we recognize the importance of trying to identify the presence of each of these practices in the activity that students build up when considering a mathematical task, in order to be able to understand the development of this mathematical ability as a whole, in line with the objectives defined in the new curriculum. We also believe that although all these practices may be present in the exploration of a task, this is not an essential condition for promoting the development of computational thinking. For example, one of the practices that might be considered less accessible to students in the first years of school is the practice of algorithmics and, in fact, this is not essential for students to develop computational thinking.

Even so, there are several studies that show the relevance of introducing algorithmic thinking in the early years of schooling (e.g. Mittermeir, 2013; Voronina et al., 2016; Figueiredo et al., 2021). For example, Voronina et al. (2016), in a study with 6- and 7-year-olds, introduced the practice of algorithmics through game-related activities. During said study, they developed a conceptual framework that allows us to understand the development of algorithmic thinking through three stages that represent a progression in the development of algorithmics in pupils of these ages. Thus, while in the first stage students learn to apply given algorithms, in the second they are confronted with a greater diversity of algorithms and already carry out activities to complete these algorithms, rather than just applying them. In the third and final stage, according to the authors, students are already capable of transferring learned algorithms to other similar activities, modifying them to achieve different results, and are even capable of producing new algorithms.

As already mentioned, the use of technology is not a prerequisite for the development of computational thinking. This also applies to the practice of algorithms, which can be worked on using paper and pencil and/or non-digital manipulatives. However, the role of technology is undeniable when used appropriately to promote learning. One example of a technological resource with recognized potential is the visual programming language Scratch. Considering that the development of algorithms requires the use of some kind of language, whether textual or graphical (Mittermeir, 2013), being easy to use and very appealing even to those who have never experienced programming (Resnick et al., 2009), the use of Scratch can enable the development of this practice. Using this resource, students can develop algorithmic thinking in a progression that can include the three stages presented by Voronina et al. (2016). Thus, they can start by experimenting with existing projects from a user-only perspective, then they can reuse them and make small modifications to the programming of these projects until, hopefully, they can create their own projects. Therefore, this technological tool seems to us to have important potential for the development of algorithmic thinking and is even proposed in the teaching guidelines for teachers in the new teaching document.

III. METHODOLOGY

This section briefly describes the methodological options, the participants, the sequence of tasks that was the subject of this study and presents the analytic categories used.

A. *Methodological options*

This study draws on a qualitative methodology, with the main purpose of describing and interpreting the computational thinking practices that emerged from the students' work in the exploration of a sequence of tasks developed in the two 1st grade classes that were operationalizing the new mathematics curriculum. The data was collected during the classroom work through field notes, photographic recording of the students' productions and audio recording of the moments of autonomous work and collective discussion.

B. *Participants*

The participants in this study were the students in the two 1st grade classes that were working on the new curriculum in the 2021/22 school year, before the nation-wide implementation phase. In this context, the four authors of this article formed a collaborative work team where the first two had the task of supporting the class teachers (the other two authors of the article) in the implementation of the new curriculum document. The collaborative work took the form of weekly meetings to plan tasks and analyse their implementation in the classrooms, reflecting on the different aspects of implementing and conducting the lesson, focusing on the task and the work of the teachers and students and the learning and difficulties they both experienced.

The first two authors of this article also monitored the classes on a more regular basis, attending and participating in some lessons. The two 1st grade classes belonged to public schools in different regions of the country, one in the municipality of Bragança and the other in the municipality of Setúbal. The classes numbered 22 and 24 students respectively, all of whom were attending 1st grade for the first time. The data was collected in the 3rd term of the school year and the average age of the students was 7.

C. *The sequencing of mathematical tasks*

The sequence of tasks presented in this article is made up of three interrelated tasks, which we call: 1) How many numbers can the Numi robot write? 2) Complete and correct instructions; 3) Get inside Numi's head. In the first task, a picture of a robot appears in the task statement. This robot was presented to the students as someone who obeyed the orders given to it, which had to be very clear and precise. The following tasks were built on this first one.

D. *Data analysis*

To identify the Computational Thinking practices that emerged from the students' activity while exploring the mathematical tasks, the categories of analysis shown in Table 1 were created.

TABLE 1 – ANALYTIC CATEGORIES OF THE COMPUTATIONAL THINKING PRACTICES IN STUDENTS' ACTIVITY

CT Practices	Indicators in the exploration of tasks
Abstraction	- Identifies essential information - Mobilizes essential information
Decomposition	- Divides into smaller parts - Mobilizes this division for intermediate solutions
Pattern recognition	- Recognizes regularities - Mobilizes regularities in generating solutions
Algorithmics	- Recognizes needed steps - Recognizes order of needed steps - Mobilizes the sequence of steps
Depuration	- Identifies mistakes - Corrects mistakes - Optimizes correct solutions

The tasks were developed within the collaborative working group and implemented in both classes. Each task was explored in the classroom using an exploratory model, characterized by the following three phases: 1) presentation of the task; 2) autonomous work by the students in pairs; and 3) collective discussion with the whole class and systematization of learning. This article presents evidence of the students' work on the three tasks, and does not identify any of the two classes, as it is considered that the students' performances were not significantly different.

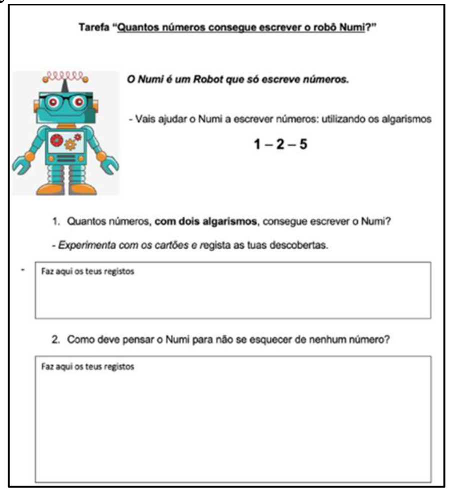
IV. PRESENTATION OF RESULTS

The following is a description and analysis of the students' activity in each of the three tasks of the sequence mentioned above. This analysis provides evidence of the five computational thinking practices.

A. *Task "How many numbers can robot Numi write?"*

In this task (Figure 1) students were given three digits and asked to write down all the possible two-digit numbers, given those three. In addition to the task statement, each pair of students was given a set of cards with the digits written on them, so that they could easily move them around and discover all the possible numbers.

In both classes, the students were able to discover and record the six possible numbers very easily. To achieve this, it was noticeable that the students focused on the given digits and the construction of two-digit numbers, identifying and mobilizing the essential information, thus revealing the use of the practice of abstraction. This practice was therefore easily and naturally mobilized by the students in both classes.


Tarefa "Quantos números consegue escrever o robô Numi?"

O Numi é um Robot que só escreve números.

- Vais ajudar o Numi a escrever números: utilizando os algarismos

1 – 2 – 5

1. Quantos números, **com dois algarismos**, consegue escrever o Numi?

- Experimenta com os cartões e regista as tuas descobertas.

Faz aqui os teus registos

2. Como deve pensar o Numi para não se esquecer de nenhum número?

Faz aqui os teus registos

Fig. 1. Wording of the task "How many numbers can robot Numi write?".

The second question of this task proved to be more demanding and was the one that intentionally promoted the use of some algorithmic thinking, as it asked the students to show "How should the Numi robot think so that it doesn't forget any numbers?". In order to answer this question, the students had to mobilize the procedure they had previously carried out when they discovered all the possible numbers. Moreover, they had to translate this procedure very clearly and precisely so that they could "convey" it to the Numi robot, and it was desirable for them to define a strategy so that no possible number was forgotten. In the first approach to this second question, the students did not mobilize the knowledge and procedures they had easily used in the first question. It was difficult for them to understand the importance of the completeness and precision of the instructions to be given to the Numi robot. Naturally, this question required them to "go back" and think about how they had proceeded, which, of course, is not immediate for students at this level of schooling. At this time, the role of the teachers was very important, leading the students to focus their attention on the need to identify the different steps and the importance of their accuracy. The following quote is an example of a moment in class, during the students' autonomous work phase, where the teacher tries to get the students to understand the need for instructions to be clear and complete.

Teacher: You must give Numi orders. Numi first does like this... with these numbers, first Numi can do what?

Student: He must pick up the numbers.

Teacher: But does he pick them all up?

Teacher: Then what? Tell Numi... Numi first takes the card with the number on it...

Student: One. And then you take the card with the number 2 on it.

Teacher: And what does he do?

Student: And he makes the number 12.

Teacher: But he can take it like she did [put the 2 card under the 1 card]. So, what does he have to do with the 2 card?

Student - Put it next to 1.

Teacher: Which side?

Student - On the right.

In the last stage of this lesson episode, the clear translation of the instructions given by the student through the movement of the cards led him to recognize not only the necessary steps, but also the need to be precise in the instructions he gave. In fact, in this quote, the teacher's action of placing one card under the other led the student to need to indicate exactly where each card should be placed. In a very intuitive way, both teachers felt the need to incorporate Numi, playing the robot and strictly following the orders that the students were giving them, leading them to check whether they were effective. These actions on the part of the teacher covered the two aspects identified as indicators for practicing algorithmics: recognizing the necessary steps and the order of those steps. On the other hand, the fact that they exactly reproduced the instructions given by the students, showing them the result of those instructions, also allowed them to immediately recognize their ineffectiveness and incompleteness. This moment thus exemplifies the mobilization of debugging practice, leading students to identify and correct errors. It should also be noted that the manipulation of the cards facilitated this process, as it made the faithful reproduction of the instructions given by the students' tangible, leading them to identify faults and errors and correct them.

We believe that the difficulties presented by the students are centred around the practice of algorithms, more specifically round the recognition of the necessary steps and their order and are not related to knowledge about laterality or other concepts of spatial orientation. The students were able to recognize or differentiate between left and right, up and down, but they didn't immediately and naturally use these concepts to construct the instructions for writing the numbers. They assumed that these instructions weren't necessary because they translated them into actions very naturally when manipulating the cards. In other words, the difficulty lay in the complete and exact construction of the algorithm.

When they were led to the necessity of using lateralization concepts to construct the instructions, the students easily realized that they only needed to supply the instructions to change the positions of the cards to obtain new numbers with the same digits, as shown in the following excerpt:

Student: Here's 12, now we swap the numbers, 2 goes here and 1 goes here and we make 21.

Teacher: And how many numbers can you build with this strategy?

Student: Two.

Teacher: And then can you build others or not?

Student: Yes, then 25, 52.

This quote shows how the students began to use a strategy to solve the problem, obtaining different numbers by swapping the order of the cards, thus using the regularities of the decimal numbering system, specifically regarding the position of the digit in the number. Therefore, as far as computational thinking practices are concerned, we can note the practice of pattern recognition, with the students having recognized and mobilized the regularities of the decimal numbering system. To show their thinking processes, the students used finger movements, manipulated the cards directly or used arrow diagrams to write the numbers, thus using different representations (Figure 2).

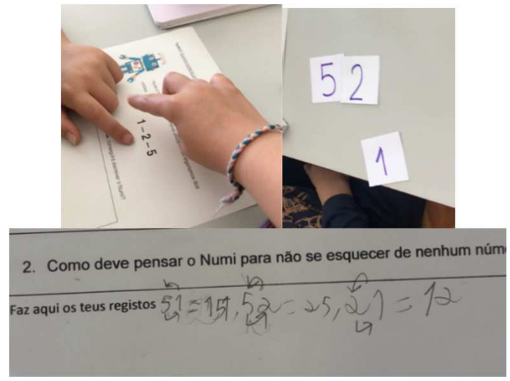


Fig. 2. Strategy of swapping digit position in the numbers.

The students also used other strategies, such as writing down the possible numbers in ascending or descending order or writing down all the numbers that "started" with the same digit, i.e. exhausting all the possible numbers with the digit in a given position (Figure 3). These regularities made it possible to divide the problem into smaller parts, mobilizing this division in smaller partial resolutions, i.e., for example, when starting by writing down all the numbers that had a certain digit in the tens position, the problem was subdivided into minor parts, using the practice of decomposition. In the same way, we proceeded with ordering the numbers.

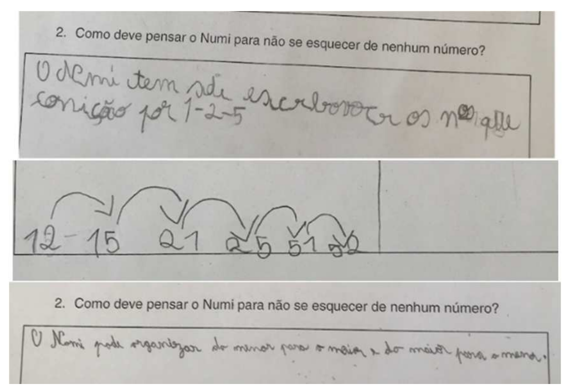


Fig. 3. Students' solutions showing other strategies used.

B. *Task "Complete and correct instructions"*

In this task of the sequence, the aim was for the students to continue to develop their algorithmic thinking in an intentional way. They were asked to complete and correct instructions.

Initially, students were given cards with unfinished instructions to complete. The students began to use the cards, but they very naturally began to create other equally valid instructions, different from those already formulated on the cards given to them. So, apparently, the cards provided the students with a model, but quickly and very autonomously, they were put aside, and the students were able to formulate other instructions, already revealing an understanding of what was intended and showing some creativity in such formulation. Later, in another part of the task, the students were asked to correct wrong or incomplete instructions, not restricting them to a previous formulation and allowing them more freedom in reworking the instructions (Figure 4).

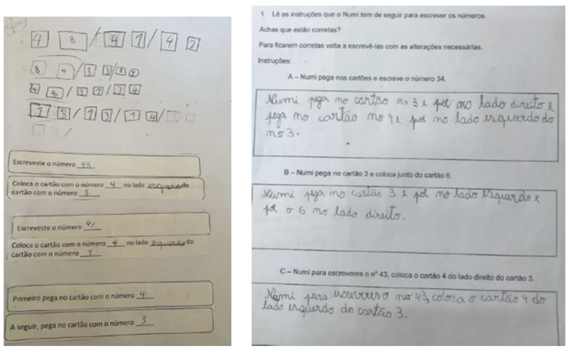


Fig. 4. Students' solutions in the task "Correct and complete instructions".

Regarding computational thinking practices, we present what we consider to be evidence of the mobilization of these practices. When the students needed to identify whether the information in the instructions was sufficient or correct, focusing their attention on the essentials, they mobilized the practice of abstraction. They used decomposition when they differentiated between correct and incorrect instructions and identified the missing elements by analysing the instructions presented to them in slices. They used the recognition of patterns already identified in the previous task and used all of them to complete or correct the instructions given. They clearly worked on algorithmic practice, recognizing which steps were presented, which were missing or incorrect and validating their order. Debugging was used to identify and correct errors in incorrect instructions and to detect their incompleteness.

C. *Task "Get inside Numi's head"*

The third and final task in this sequence was intended as an introduction to the visual programming language Scratch. To this purpose, a program was created around a number game in which the students were asked to indicate a number and next another number would be presented by Numi (in Scratch). The students had to find out what instructions Numi had been given to produce the new number. When the game was played in both classes, the students were able to immediately figure out the instructions given to Numi, which consisted of commands such as adding or subtracting 1 or 10 from the numbers they had indicated (Figure 5).

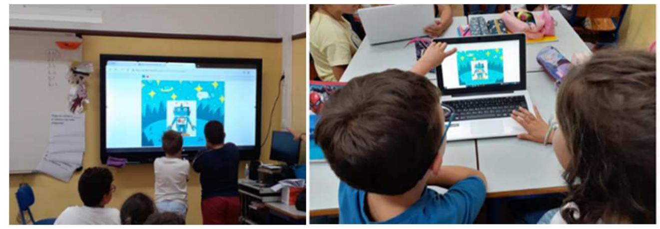


Fig. 5. Student's activity in task "Get inside Numi´s head".

After that, the teachers said that they were going to get inside Numi's head, in other words, to understand how Numi had thought. Naturally, at this stage, the programming done in Scratch was presented and explained in a simple way (Figure 6).

Fig. 6. Scratch programming presented to the students, which they modified.

After that, the students were given access to the Scratch programs on their tablets and were challenged to create new instructions themselves, resulting in new number games that they would play in pairs. All the pairs managed to modify the instructions given, using new commands such as adding or subtracting numbers other than 1 and 10.

In this task, the students mobilized computational thinking practices at different times. For example, by focusing their attention on the number they obtained in Scratch, they mobilized the practice of abstraction. When they identified, in the given algorithm, the parts they had to change to produce new instructions, they did so by decomposing the given algorithm. They used patterns when they recognized the rules presented in Scratch programming and used debugging when they recognized errors in both the rules they identified and those they created. And, of course, they clearly practiced algorithmics when they addressed the given algorithm to discover the rule and when they altered it to produce new rules.

V. Discussion of Results

Considering the indicators relating to computational thinking practices presented and the evidence reflected in the students' activity when exploring the tasks, we present Table 2, which summarizes this evidence over the three tasks analysed. The five practices were present in all the tasks in the sequence described and mobilized aspects related to understanding the subject of Numbers. For example, pattern recognition rendered the regularities of the decimal numbering system evident, both by recognizing the position of a digit in the number and its value, and by recognizing the order of numbers. Abstraction also allowed students to deal with specific digits and numbers or to focus their attention on the instructions to be completed or corrected, as did the numbers requested and obtained from the programming in Scratch. Decomposition allowed for different approaches to the same problem, also enhancing other aspects of the mathematical content, such as the ordering of numbers. The practice of debugging, which is extremely important in mathematical activity, was evident in the three tasks described, when the students corrected or improved presented instructions.

In the students' work, the three stages identified by Voronina et al. (2016) regarding the practice of algorithmics were evident, showing progressive situations of completing and changing algorithms. Regarding the use of the Scratch visual programming environment, it should be noted that, in addition to the factors relating to the motivation and interest of the students, it was possible to work with the mathematical operators of the environment itself to discover or change the rules given, which explored arithmetic regularities. It should however also be noted that the practice of algorithms was not only evident in the task where Scratch was used, but also in the previous ones which did not use any technological tool. Although we don't consider it essential to work on the practice of algorithmics in all tasks that promote computational thinking, as Wing (2010) points out, this practice seems to be the aggregating pillar of all the others and, in this study, we show how it summarizes and makes evident the mobilization of all other practices. Thus, it seems possible to develop tasks in which the other practices are mobilized, without intentionally leading to the practice of algorithmics, but the opposite is not so obvious, i.e. the development of algorithmics seems to require the presence of all the others.

TABLE 2 – EVIDENCE OF PRACTICES OF COMPUTATIONAL THINKING EMERGING FROM THE WORK OF STUDENTS IN THE THREE TASKS OF THE SEQUENCE

Tasks / Practices of CT	Evidences of students' work		
	"How many numbers can robot Numi write?".	**"Complete and correct instructions"**	**"Get inside Numi's head"**
Abstraction	- Focus attention on the three digits proposed and in constructing 2 digit numbers.	- Focus attention just on the instruction to complete or correct.	- Focus attention on the number submitted and on the number given by Numi both when the algorithm is given and when the possibility to change the algorithm is given.
Decomposition	Three different approaches arise: - Start by forming numbers using just of the 3 digits and, next, swap the position of these digits to create new numbers; - Find out all the possible numbers for each of the digits dealing with each one individually; - Find out the numbers ordering them increasingly or decreasingly.	- Differentiate the correct from the incorrect elements of the instructions; - Identify the missing elements.	- Decompose the given algorithm identifying which part or parts to change.
Pattern recognition	- The position swap of the digits in the number allows to discover two different numbers, and this always occurs (concept of positional value); - To form numbers with two digits given three digits, with each digit we can write four different numbers and, excluding repetitions, we have 6 different numbers.	- To correct or complete the instructions it is always necessary to use the positional values of digits.	- Discover that the same rule is given for 4 different numbers; - Discover regularities in the given algorithms and that they must change (where they change and how and what results from this change).
Algorithmics	- Define in sequential and complete steps the process of creating the 6 possible numbers.	- Correct or complete the sequence of steps presented in the instructions, obeying to the correct order and including all needed elements.	Understand the algorithm presented ("what is inside Numi's head"); - Change the algorithm given in such a way as to present different rules.
Depuration	- Correct attempts not complying with the given instructions (forming two-digit numbers being given three digits) - Exclude repeated numbers; - Correct wrong instructions; - Make instructions clearer and more specific.	- Detect the error and correct it: - Detect the incompleteness and complete it.	Identify eventual mistakes when the given algorithm is changed and correct them.

It should also be noted that, in the case of this study, working with numbers seems to be conducive to the emergence of algorithms, as it allows the regularities of the decimal numbering system to be thus translated. Other mathematical topics or subtopics may not be so easy, especially in the early years of schooling.

We conclude by mentioning the relevance of exploring computational thinking in an integrated manner with other mathematical topics and reinforce the importance of intentionally working on each of its practices and looking for evidence in the students' work to understand how they mobilize and develop such practices.

REFERENCES

[1] Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. *Educational Technology & Society*, 19(3), 47–57.

[2] 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, 13-17. http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

[3] Canavarro, A.P., Mestre, C., Gomes, D., Santos, E., Santos, L., Brunheira, L., Vicente, M., Gouveia, M. J., Correia, P., Marques, P., & Espadeiro, G. (2021). Aprendizagens Essenciais de Matemática no Ensino Básico. ME-DGE. https://www.dge.mec.pt/sites/default/files/Curriculo/Aprendizagens_Essenciais/1_ciclo/ae_mat_1.o_ano.pdf

[4] Espadeiro, R. G. (2021). O pensamento computacional no currículo de Matemática. *Educação e Matemática*, 162, 5-10.

[5] Figueiredo, M., Amante, S., Gomes, H., Gomes, C.A., Rego, B., Alves, V. & Duarte, R. P. (2021). Algorithmic thinking in early child-hood

education: opportunities and supports in the portuguese context. In *Proceedings of EDULEARN21 Conference*.

[6] Mittermeir, R. T. (2013). Algorithmics for preschoolers—A contradiction? *Creative Education*, 4(09), 557.

[7] Voronina, L.V., Sergeeva, N. N., Utymova, E. A. (2016). Development of algorithm skills in preschool children. *Procedia – Social and Behavioral Sciences*, 233, 155-159.

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

[9] Wing, J. (2010). Computational Thinking: What and Why? https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

[10] Yadav, A., Larimore, R., Rich, K., Schwarz, C. (2019). Integrating computational thinking in elementary classrooms: Introducing a toolkit to support teachers. *Society for Information Technology & Teacher Education International Conference*. https://par.nsf.gov/biblio/10095783.