Informatics in Education, 2023, Vol. 22, No. 3, 499–523

DOI: 10.15388/infedu.2023.19

Effects of Scratch-Based Activities on 4th-Grade Students' Computational Thinking Skills

João PIEDADE, Nuno DOROTEA
UIDEF – Institute of Education of University of Lisbon, Lisbon, Portugal
e-mail: jmpiedade@ie.ulisboa.pt, nmdorotea@ie.ulisboa.pt

Received: August 2022

Abstract. Computational Thinking (CT) has emerged in recent years as a thematic trend in education in many countries and several initiatives have been developed for its inclusion in school curricula. There are many pedagogical strategies to promote the development of elementary school students' CT skills and knowledge. Unplugged learning tasks, block-based programming projects, and educational robotics are 3 of the most used strategies. This paper aimed to analyze the effect of Scratch-based activities, developed during one scholar year, on the computational thinking skills developed and concepts achieved by 4th-grade students. The study involved 189 students from two school clusters organized into an experimental group and a control group. To assess students' computational knowledge, the Beginners Computational Thinking Test developed by Several Zapata-Cáceres *et al.* (2020) was used. The results indicate statistically significant differences between the groups, in which students in the experimental group (who performed activities with Scratch) scored higher on the test than students in the control group (who did not use Scratch).

Keywords: block-based programming; computational concepts; computational thinking; elementary education; scratch programming.

1. Introduction

Computational Thinking (CT) is currently considered a transversal and essential competence for problem-solving in different areas of knowledge. Wing (2006) considers that CT is essential for everyone, like writing, reading and arithmetic competencies. Presented by Wing (2006, p. 33), CT builds on the fundamental theoretical concepts of computer science involving problem-solving skills, systems design and understanding human behavior.

By engaging in problem formulation and problem-solving, students express and put into practice many other skills needed to design efficient solutions to that problem in a way that allows a computational agent to execute it (Wing, 2017). Skills such as algorithmic, abstraction, generalization, decomposition, pattern recognition, flow con-

Table 1
Computational Thinking Skills

Skill	Definition
Algorithms	The algorithm is a practice of writing a step-by-step sequence of instructions for carrying out a solution or process.
Abstraction	Abstraction is the process of taking away or removing characteristics from something in order to reduce it to a set of essential characteristics.
Generalization	Generalization is transferring a problem-solving process to a wide variety of problems.
Decomposition	Decomposition is about breaking problems down into small parts in order to make them easier to solve.
Pattern recognition	Recognizing a pattern or similar characteristics helps break down the problem, build a construct as a path for the solution, and find a set of patterns or similar characteristics that can be generalized.
Flow control	Process of using different flow control structures.
Parallelism	Process of executing different instruction sets in parallel.
Data representation	Process of selection of the appropriate models for data representation.
Testing and debugging	Process of evaluation and verification of the solution to recognize when instructions correspond to pretended actions. This process allows to find codification errors.

Note. Adapted from Piedade *et al.* (2020)

trol, parallelism, data representation, testing and debugging (Wing, 2006; Brennan and Resnick, 2012). Table 1 presents the main definitions of each CT skill.

However, it is important to note that CT should not be confused with coding and programming activities. As presented by Bocconi *et al.* (2022), CT refers to the "ability to understand the underlying notions and mechanisms of digital technologies to formulate and solve problems", while programming aims to analyze a problem and design its solution, implementing it later in a specific programming language through coding.

Thus, we can consider that "CT is an umbrella term that embodies an intellectual foundation necessary to understand the computational world and employ multidimensional problem-solving skills within and across disciplines" (Fagerlund *et al.*, 2020). Computational Thinking is, in itself, a strong cognitive component, encompassing several concepts and distinct computing-related reasoning, but it is also useful in several other real-life domains and contexts (Machuqueiro and Piedade, 2022, p. 7).

By solving problems through programming and coding, the development of CT skills is stimulated, putting into practice computational concepts such as variables, data types, sequences, iterations, loop structures, conditional structures, parallelism, testing and debugging, among others.

In the Portuguese curriculum, CT, computer science and programming learning are present from primary (K-9) to secondary (K-12) education. However, in primary education there are no specific computer science subjects nor is it compulsory for students to learn about these topics in the subjects that are part of the curriculum. Although, there are guidelines in this respect. Some schools offer in primary school an optional subject where students learn how to create projects and solve problems by programming with Scratch.

The focus of this paper is the development of CT skills of K-9 students through programming in Scratch, the most used block-based language in Portuguese primary schools. Also, it is the block-based language used for learning programming with the highest rank in TIOBE Index[1] (August 2022).

1.1. *Research Questions and Hypotheses*

In this study, we aimed to analyze the impact of Scratch-based activities, developed during one scholar year, on the computational thinking skills developed and concepts achieved by 4th-grade students. According to the main goal defined for the study, the following questions were assumed:

- **RQ1.** What is the level of Computational Thinking Skills of 4th-grade students?
- **RQ2.** What is the effect of Scratch-based activities on 4th-grade students' Computational Thinking Skills?
- **RQ3.** What is the influence of gender and age on 4th-grade students' Computational Thinking Skills?
- **RQ4.** What is the influence of Mathematics, Science and Portuguese Knowledge on 4th-grade students' Computational Thinking Skills?

In addition to the research questions, six conceptual hypotheses were formulated:

- **H1.** Scratch-based activities produce a significant effect on the 4th-grade student's Computational Thinking skills.
- **H2.** There are significant differences in students' Computational Thinking skills by gender.
- **H3.** There are significant differences in students' Computational Thinking skills by age.
- **H4.** There are significant differences in students' Computational Thinking skills considering their knowledge of Mathematics.
- **H5.** There are significant differences in students' Computational Thinking skills considering their knowledge of Science.
- **H6.** There are significant differences in students' Computational Thinking skills considering their knowledge of the Portuguese language.

2. **Background**

2.1. *Scratch-based Programming to Promote Computational Thinking*

The introduction of CT in schools may be conducted through activities related to Computer Science, such as programming problem-solving, which currently includes an endless number of existing learning tools, such as Scratch, Kodu, Blocky, Minecraft, Alice,

---

[1] `https://www.tiobe.com/tiobe-index/`

Tynker, and Ubbu, among others (Machuqueiro and Piedade, 2022). These block-based programming environments are usually designed to be used by children and young people, enabling the development of CT-related processes (Piedade *et al.*, 2019; Ramos and Espadeiro, 2014). Several studies highlight the use of block-based programming tools in primary schools as a way to teach and learn to program, develop CT, or design algorithms to answer specific problems. Piedade *et al.* (2019) analyzed the core characteristics of 26 block-based apps and pointed out that most of the applications are excellent solutions for teaching essential programming content and supporting computational thinking activities.

Scratch is a free and graphical block-based programming environment which allows students to code by drag and drop blocks that correspond to instructions. It is a very popular language for younger learners because of its attractiveness and simplicity of understanding and use.

Scratch is a very accessible language and a good starting point for learning by students with little or no previous knowledge of programming (Sáez-López *et al.*, 2016).

The coding process is facilitated by its representation through blocks (Weintrop and Wilensky, 2015) with different shapes and colors, by creating sequences by joining blocks that represent expressions, statements, control structures and other programming concepts (Montiel and Gomez-Zermeño, 2021).

Being a language based on blocks that are connected to each other, it does not present the difficulties characteristic of text-based programming languages (Durak, 2020), as the learning and memorizing of the vocabulary and of the functions, minimizing traditional syntax errors, instruction parameterization errors and errors in certain instruction sequences (it's not possible to connect two incompatible concepts). The comprehension of the meaning of each block is facilitated by its natural language labels (available in dozens of languages), making it easier to read its function and the whole code.

Thus, Scratch allows students to devote their effort and focus on the solution to problems rather than the programming language. In this way, it facilitates the development of cross-curricular projects, combining programming with subjects such as the arts, sciences, languages, among others, and activities such as simulations, animations, games, music, and stories.

The environment consists of an area where the blocks are available, an area where the blocks are dragged to create the program, and a stage where the execution of the program is visualized. Scratch offers over 100 blocks grouped into nine categories: motion (positioning, movement and interactions of objects on the stage); looks (of the objects and the stage); sound; events (which allows parallelism through the creation of sequences of independent blocks that are executed simultaneously when different events occur); control (flow control structures); sensing (sensor programming); operators, variables, and my blocks (where it is possible to create a new block as functions). Through the addition of extensions, it is possible to add new categories of blocks associated with different activities such as video sensing, text translation, text-to-speech, robotics, and modules with artificial intelligence.

It allows programs to be created using all the computational concepts that exist in textual languages, but in an environment that stimulates creativity and facilitates the programming of complex activities existing in other languages.

Delal and Oner (2020) consider that programming may not be a requirement to develop CT skills. However, to Sáez-López *et al.* (2016), the use of Scratch can promote the learning of programming concepts, logic, and computer literacy in young students. More recently, in a study with 217 students from 5th grade, Sigayret *et al.* (2022) conclude that who used Scratch "showed a better mastery of computational concepts and a greater ability to solve algorithmic problems".

According to Fagerlund *et al.* (2022), programming projects with Scratch allows the development of CT skills in at least four dimensions: i) the planning of the solution with the representation of the structure and its functionalities; ii) the codification of the solution, going through all the stages of development, from the understanding and decomposition of the problem to testing and debugging the coded solution; iii) the collaboration between students (pair programming) and the seeking for support and resources; and iv) the evaluation of the designed solution through testing, debugging and corrections of the code.

Considering the CT skills presented in Brennan and Resnick's (2012) framework, beyond computational concepts (presented in Table 1), the Scratch environment allows the development of computational practices (abstracting and modularizing, being incremental and iterative, testing and debugging, and reusing and remixing by access and modifying projects developed by others). In fact, Scratch allows students to learn to think creatively and collaboratively.

Finally, we agree that programming with Scratch is a strategy not only to develop students' CT skills (Rodríguez-Martínez *et al.*, 2019), but also to foster students' understanding between multiple contents from different subject areas beyond computer science (Resnick, 2013).

2.2. *Assessment of Students' Computational Thinking Skills*

Considering the growing interest in the development of CT in recent years, particularly in children's school curricula, several studies have devoted attention to the study of methods and tools for the evaluation and analysis of CT skills (Basso *et al.*, 2018; Brennan and Resnick, 2012; Fagerlund *et al.,* 2020; Kormaz *et al.*, 2017; Moreno-León *et al.*, 2015; Roman-González, 2015; Zapata-Cáceres *et al.,* 2020). In these studies, different assessment approaches are used to analyze students' CT skills, CT concepts, and practices. Some of them define and use frameworks based on the dimensions of CT for the analysis of the Scratch-based projects developed by students. Others report the use of tests or scales to measure the different types of CT concepts. Román-González *et al.,* (2019) highlighted seven CT assessment approaches and tools: (i) diagnostic tools; (ii) summative tools; (iii) formative-interactive tools with automatic feedback; (iv) data-mining tools; (v) skill transfer tools; (vi) perceptions-attitudes scales; and (vii) vocabulary assessment. In the same way, Poulakis and Politis (2021), based on a systematic literature review, defined three categories to classify the CT assessment approaches: (i) Using specific programming environments; (ii) using CT assessment criteria and/ or psychometric tools; and (iii) using multiple forms of assessment.

In the Scratch-based context, Brennan and Resnick (2012) proposed a framework for studying and assessing the development of CT organized in three dimensions: Computational Concepts, Computational Practices and Computational Perspectives. Computational concepts are related to the use of programming concepts (embedded in Scratch programming blocks), that are present in many other languages, such as sequences, loops, events, parallelism, conditionals, operators, and data. Computational practices are related to the practices developed by students when they are engaging with the application of programming concepts and debugging tasks or remixing other's projects. Computational Perspectives are related to the students' perspectives about the computational world around them. This framework is useful to identify CT skills developed in multidisciplinary Scratch-based projects through the articulation between the three perspectives of analysis. Following a similar approach, Fagerlund *et al.* (2020) organized a rubric-based framework, based on the previous literature review, to assess 4th-grade students' CT by the analysis of the 365 Scratch projects. This framework analyzes the programming contents, coding patterns, coding constructs, instance types, core educational principles and CT patterns that can be found in each scratch project.

Moreno-León *et al.* (2015) present Dr. Scratch, a web-based application for the automatic analysis of projects coded in the Scratch environment. The Dr. Scratch application can be used by students and teachers to check if the projects have been properly coded, to identify the mistakes and provide suggestions to improve their Scratch code and to develop their CT skills. This tool classifies CT concepts (abstraction, parallelization, logic, synchronization, flow control, user interactivity, and data representation) on a three-point competence scale: Basic (1 point); Developing (2 points); and Proficiency (3 points). Dr. Scratch is used in both formative and summative assessment, however, according to Hoover *et al.* (2016) it is difficult to properly analyze more complex projects by comparing qualitative and quantitative data.

Several tests and psychometric tools to assess CT concepts, attitudes, and skills were reported in the literature. Ambrosio *et al.* (2014) proposed a Computational Thinking Test that uses multiple-choice questions to analyze programming concepts. Román-González *et al.* (2015; 2017) developed a Computational Thinking Test, based on the results of Ambrosio *et al.* (2014) study, and validated the test with 10–15 years old students.

Other psychometric tools are presented by Mühling *et al.* (2015), Kormaz *et al.* (2017) and Yagci (2019). These tools are referred to in the systematic literature review carried out by Poulakis and Politis (2021).

Finally, Zapata-Cáceres *et al.* (2020, 2021) proposed the Beginners Computational Thinking Test (BCTt). The BCTt includes Brennan's 3D framework basic computational concepts, ordered by increasing difficulty according to the target audience: sequences, simple loops, nested loops, and conditionals (Zapata-Cáceres *et al.*, 2020). It uses multiple-choice items with three options of response for each item, which are represented in two different graphic layouts: canvas or maze type.

In this study, we used the BCTt to assess computational thinking concepts developed by 4th-grade students after undertaking Scratch-based activities during a school year. The reason for choosing it was that it was developed and validated with students from 5 to 12 years old and also because the items are independent of the programming language and environment used. In contrast to the study developed by Fagerlund *et al.* (2020), which focused on the analysis of CT skills present in Scratch programming projects developed by 4th-grade students, in this study our focus was to access, through a CT test, of the students' concrete knowledge after the development of Scratch-based learning activities.

3. Material and Methods

This study was developed according to a quasi-experimental design collecting data through a post-test from experimental and from control groups to analyze the students' Computational Thinking Skills. The choice of a quasi-experimental design was because it was not possible to guarantee randomization in the distribution of the participants between the control and experimental groups and control all the variables in a laboratory (Cohen *et al.,* 2018). The organization of the educational system does not allow for the decomposition of classes and groups in a random way.

The next sections describe the sampling strategy, the characteristics of the participants, the experimental procedures adopted, and the instruments used to collect data from the participants.

3.1. *Sampling and Participants*

In this study, a non-probability sampling technique, based on non-random criteria, was used to select the participants who are most accessible and available to participate.

Two Portuguese School Clusters were selected from the Lisbon district, belonging to different municipalities and with the same economic, demographic, and social characteristics. Both municipalities have a purchasing power above 90% of the national average. The two selected school clusters are integrated into the Portuguese public educational system and have students from preschool to secondary education.

The 186 participants were organized in control (N = 93) and experimental (N = 96) groups, involving the totality of the 4th-grade students at both schools. The control group belongs to school cluster A, consisting of a male (53%) and female (47%) students. The experimental group belongs to school cluster B, consisting of a male (45%) and female (55%) students. The students in both groups were aged between 9 and 10 years old. Thus, it was possible to constitute equivalent groups with similar characteristics.

The students in both groups had no previous experience with scratch programming or other similar computer science activities, accordingly, it was decided not to conduct a pre-test. The validation of the students' lack of previous experience was conducted with the main teachers of each group in a preparatory meeting. We should notice that in primary schools, in Portugal, the students have the same teachers between 1st and 4-grade. Thus, we were able to ensure that in a school context and in an organized and structured way the students had no previous experiences in this field.

3.2. *Experimental Procedure*

This study aimed to analyze the effects of intentional and well-organized Scratch-based activities on 4th-grade students' CT skills compared with another group of students who did not participate in Scratch-based programming activities, or any other activities related to computer science. Thus, we defined a control group consisting of students who did not have contact with introductory programming activities during their schooling. On the opposite, the students in the experimental group attended a weekly hour of programming with the Scratch coding tool throughout the 2020–2021 school year. (Fig. 1)

This weekly lesson is a mandatory complementary program for all 4th-grade students at school B. The learning activities are planned and implemented by a computer science teacher in articulation with the 4th-grade main teacher.

The learning tasks were planned to challenge the students to develop small Scratch-based projects or games that allow them to learn computational concepts and CT through computational practices (Brennan and Resnick, 2012) in articulation with other disciplines such as Mathematics, Science, and Portuguese Language. The activities in the experimental and in the control group took place in face-to-face classrooms during the covid 19 pandemic. According to that, the students had to work mostly individually on their own laptops however, some challenges were done in pair programming. Several projects were developed related to the study of geometric solids, division and multiplication, solar planets, the water cycle, and other examples. These projects al-

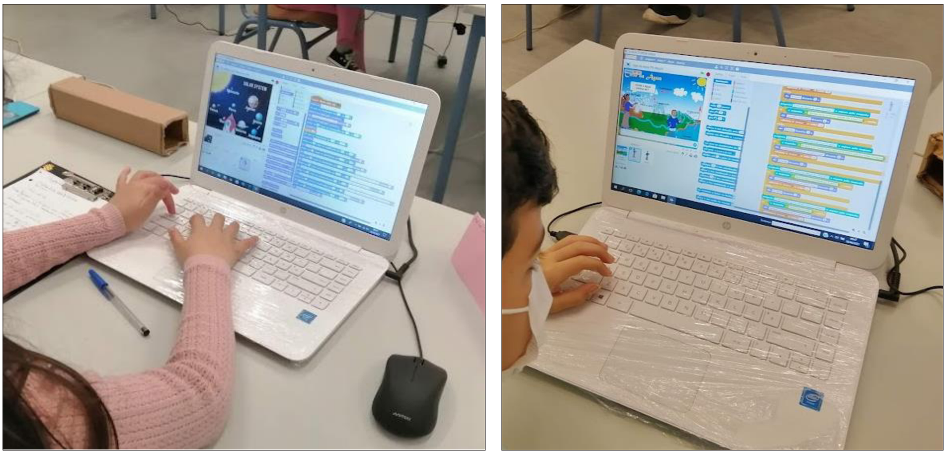

Fig. 1. Students of the experimental group programming in the Scratch Application.

lowed the students to develop basic computational knowledge, such as the notion of algorithms, programs, sequences, arithmetic and logic operators, conditionals, cycles, variables, parallelism and, synchronization in articulation with the curricular contents of mathematics, science, and Portuguese language. Two examples of projects developed by the students were presented in Fig. 2 and Fig. 3. The first one was a project aimed to challenge the students to develop a game/ quiz about the cycle of the water, and the second one challenged the students to create a mathematical game about the multiples of a number.

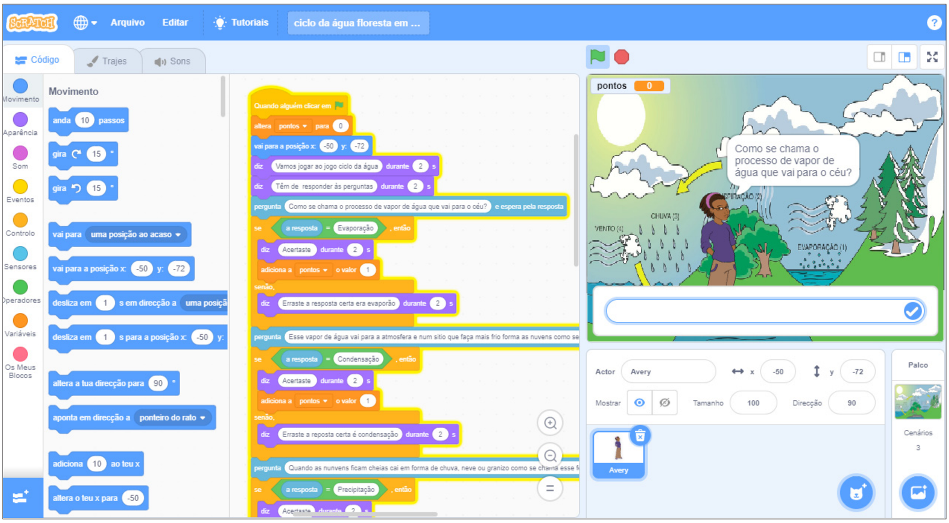

Fig. 2. Examples of Computational Thinking test items for each dimension (computational concepts).

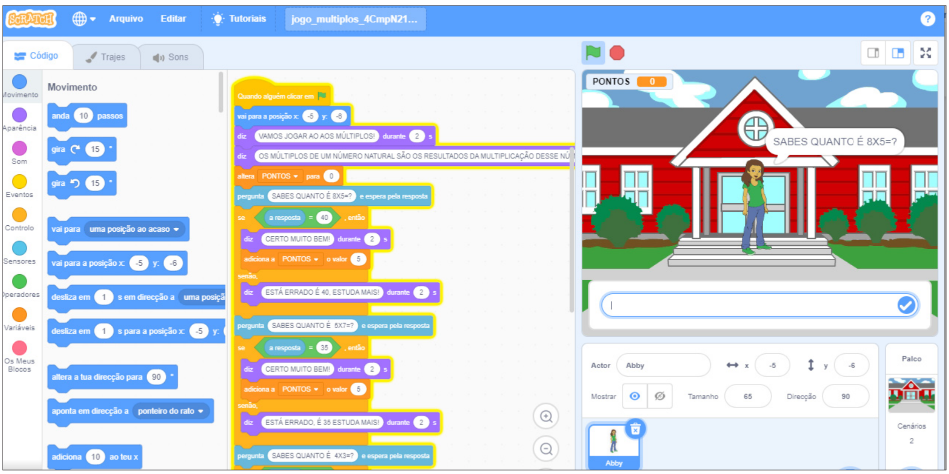

Fig. 3. Examples of Computational Thinking test items for each dimension (computational concepts).

Table 2
List of the variables under analysis

Name	Type	Operational Definition
Results in Computational Thinking Test	Dependent *Discrete*	Dynamic property: total score of the test of each student, applied in both groups after the Scratch activities
Final Results in Mathematics	Independent *Discrete*	Manipulation: final results of the students in Mathematics indicate by the main teacher (1 – Very Unsatisfactory; 2 – Unsatisfactory; 3 – Satisfactory; 4 – Good; 5 – Very Good)
Final Results in Science	Independent *Discrete*	Manipulation: final results of the students in Science indicate by the main teacher (1 – Very Unsatisfactory; 2 – Unsatisfactory; 3 – Satisfactory; 4 – Good; 5 – Very Good)
Final Results in Portuguese language	Independent *Discrete*	Manipulation: final results of the students in Portuguese language indicate by the main teacher (1 – Very Unsatisfactory; 2 – Unsatisfactory; 3 – Satisfactory; 4 – Good; 5 – Very Good)
Age	Control	Static Property: Students with similar age
Gender	Control	Static Property: Students with similar gender
School Cluster	Independent *Discrete*	Manipulation: an experimental group belong to School Cluster A (level 1); a Control group belongs to School Cluster B (level 2)

3.2. *Measures*

In this study, a set of variables were controlled and manipulated during the statistical analysis process. The variables, their type, and their definition are presented in Table 2. The dependent variable is defined around the academic results of the students in the Computational Thinking corresponding to the value obtained from the computational thinking test (BCTt) developed and validated by Zapata-Cáceres *et al.* (2020). The results in Mathematics, Science and Portuguese (independent variables) correspond to the final grade (between 1 and 5) of each student at the end of 4th grade and were indicated by the main teacher of the classes. In the context of this study, it was only possible to access the final results of the students in the experimental group. The School Cluster variable was defined to code the experimental group (School Cluster A) and control group (School Cluster B) and to distinguish between students who participated in the Scratch activities and students who not participated.

3.3. *Instruments – Beginners Computational Thinking Test (BCTt)*

The measure of students' acquisition of CT skills was performed by the application of the Beginners CT Test (BCTt) developed by Zapata-Cáceres *et al.* (2020). The authors developed the BCTt which consists of an autonomous instrument for validating students'

Table 3
Distribution of the items by the computational concept

Itens	Sequencies	Simple Loops	Nested Loops	Contitionals If-Then	Conditionals If-Then-Else	Conditionals While
1 to 6						
7 to 11						
12 to 18						
19 to 20						
21 to 22						
23 to 25						

appropriation of computational concepts, whose application does not depend on any programming environment. It was developed under a psychometric approach and provides evidence of its reliability and content, criterion, and predictive validity (Zapata-Cáceres *et al.,* 2020).

This version of the test was originally targeted at students between 5 and 12 years old. Since the BCTt is intended for early ages, both its content and its form were adapted to the audience in question. Thus, the texts of each item are easy to read and interpret and their visual appearance is intended to be appealing and enlightening for children, to facilitate the understanding of the situation described, taking into account the age of the participants. In our study, this test was used with students between 9 and 10 years old.

The test has 25 questions, organized into 6 sections, where each block evaluates a different computational concept (see Table 3). The concepts present in each section are interrelated, and the more complex concepts imply the mobilization of more elementary concepts. Each BCTt item is presented as a challenge, with four alternative answers for each item; however, only one of them is correct (see Fig. 4). It should be noted that the BCTt does not assess competencies, but rather allows us to measure the students' knowledge and skills regarding concepts associated with Computational Thinking. However, the different challenges that are proposed to students in the BCTt test involve analysis and definition of simple algorithms, problem-solving, decomposition and abstraction.

3.3.1. *Analysis of the Normality*

The normality of the BCTt Score for the entire sample was analyzed by the application of the Shapiro-Wilk test. The results of the test indicated that the BCTt Score variable presents a non-normal distribution ($SW(189) = 0{,}37$; $p < 0{,}001$). Fig. 5 shows the histogram of the variable BCTt score with the respective normality curve, where we can see that the sample follows a non-normal distribution. Despite the violation of normality assumptions, the reduced kurtosis ($ku = -0{,}783$) and skewness ($sk = 0{,}111$) values allow the use of parametric tests in the comparative analysis of the different variables. The ANOVA and t-student parametric tests for comparative analysis of means are sufficiently robust even when the assumptions of normality are not guaranteed, and the samples have a low level of flatness and skewness (Marôco, 2021).

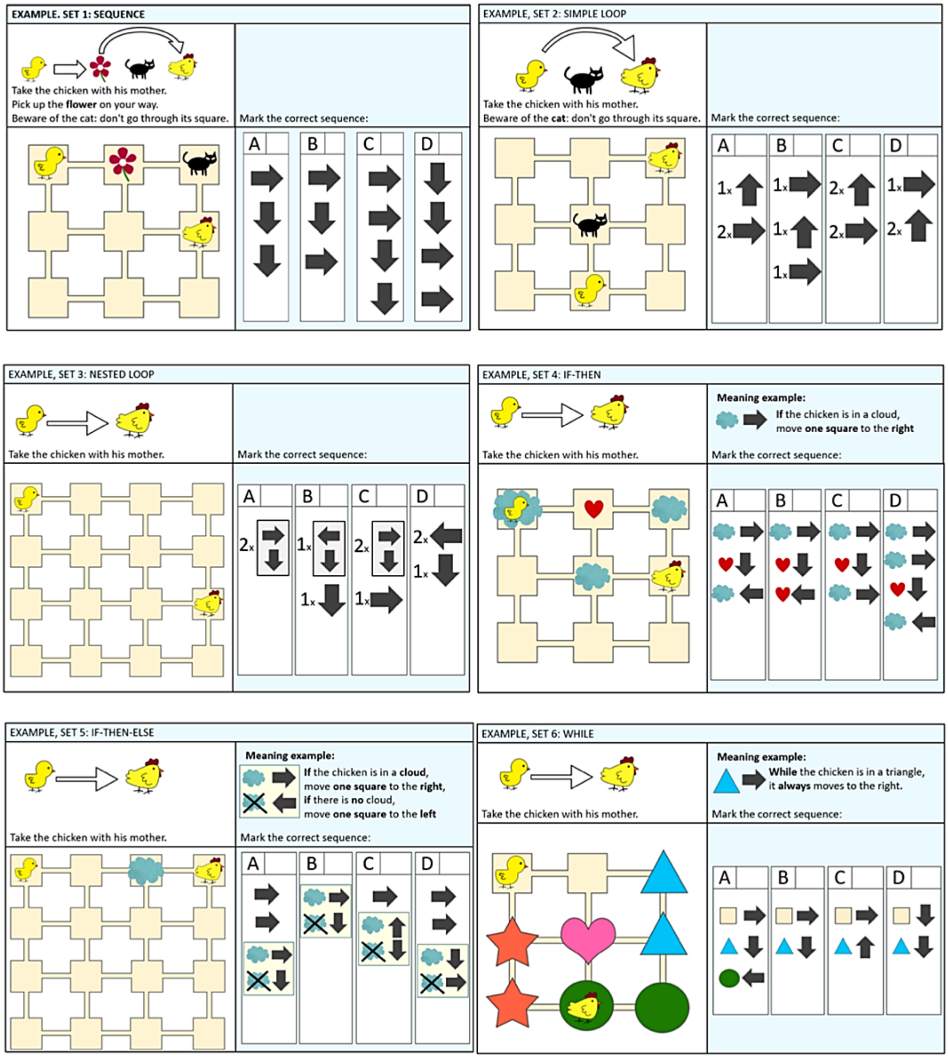

Fig. 4. Examples of Computational Thinking test items for each dimension (computational concepts).

3.3.2. *Reliability Analysis*

Reliability analysis was made to analyze the internal consistency of the BCTt variable scores. The Cronbach's Alpha coefficient indicates that the test has a very good internal consistency ($\alpha = 0{,}85$; $N = 189$). The analysis of the Cronbach's Alpha coefficients of the test from the control group and the experimental group also revealed very good internal consistency (Table 4). These results are consistent with the reliability of the original BCTt proposed by the authors ($\alpha = 0{,}82$; $N = 299$).

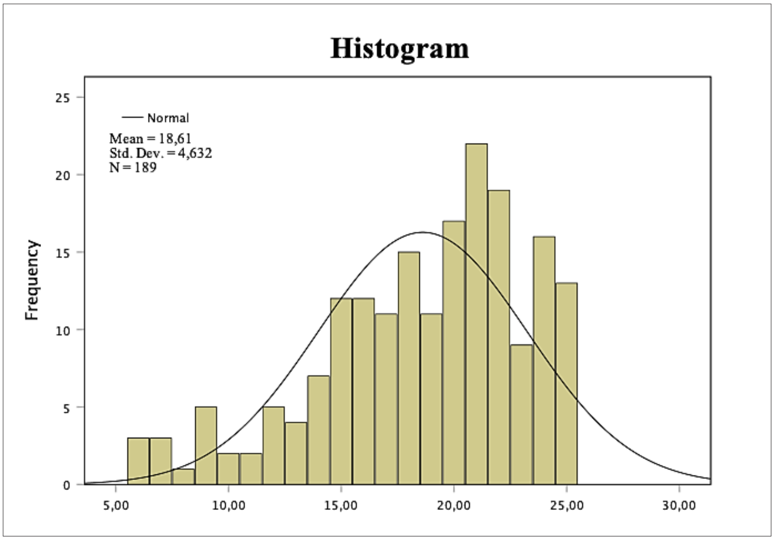

Fig. 5. Histogram of the BCTt Score (N = 189).

Table 4
BCTt Reliability Statistic for the entire sample, for the control group and the experimental group

Sample	Reliability Statistics		Item Statistics	
N	Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	Mean	Variance
189 (Entire sample)	0,85	0,86	0,75	0,04
93 (Control)	0,78	0,79	0,65	0,07
96 (Experimental)	0,84	0,87	0,84	0,02

3.3.3. *Facility Index of the Computational Thinking Test*

The analysis of the results begins with the calculation of the facility index of the test applied to the students considering the entire sample (N = 189), analyzing the statistical significance of the differences between the total correct and incorrect answers. To do this, we used Cochran's Q Test, a non-parametric test for related samples.

The facility index per question described in Fig. 6 confirms the progressive difficulty, particularly in the final questions related to nested loops and conditionals. The average facility index is 0.74. Additionally, the Fig. 7 reports the frequencies of correct and incorrect answers for each question of the test. The results of Cochran's Q test revealed significant differences between the number of correct and incorrect answers, ensuring different degrees of difficulty for each of the test questions (Q (24) = 1013,4; N = 189; p = 0,000).

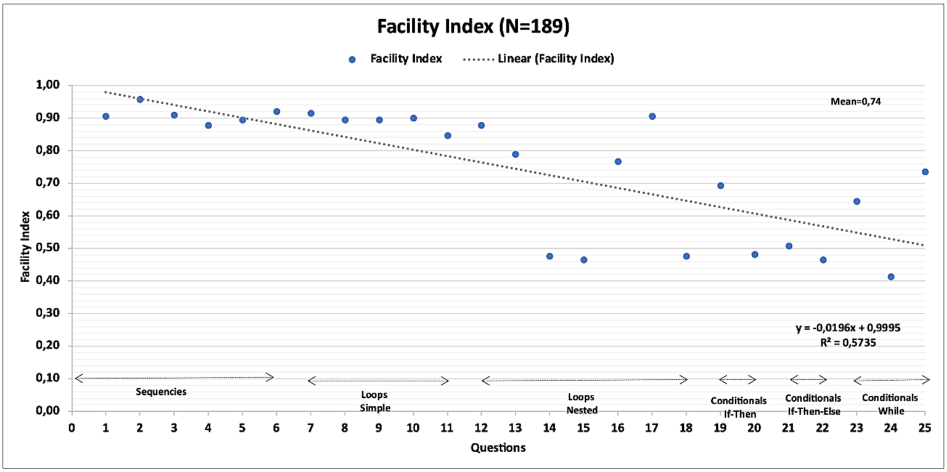

Fig. 6. Index of the facility by question (N = 189).

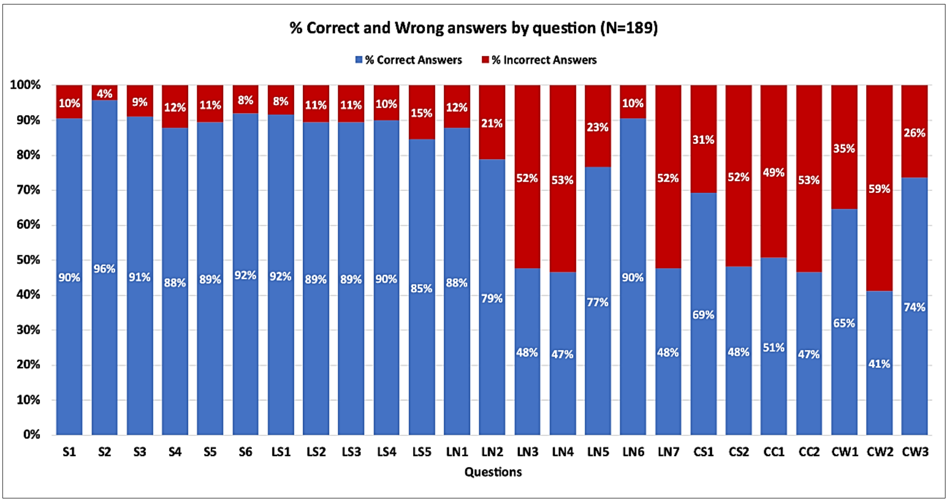

Fig. 7. Percentage of correct and incorrect answers by question (N = 189).

Likewise, analyzing the results of the control group (N = 93) and the experimental group (N = 96) separately, it was possible to confirm the increasing index of difficulty of the questions, with an average facility index of 0,65 and 0,84 for each group, respectively.

3.4. *Application Protocol*

The test was administered to students at the end of the school year (May 2021) following the protocol proposed by the original authors (Zapata-Cáceres *et al., 2020*). Each student received a printed colored copy of the test and an answer sheet where they should indicate the answer option for each of the questions. As suggested by the authors, before the application of the test the teacher explains the test and presents an example of a standard question and answer type to be indicated by the students.

Students should indicate on the answer sheet, in addition to their answers to the questions, their name, gender and age, which are essential variables for comparative analysis of the results.

All students started the test at the same time and finished after 50 minutes, within the originally defined interval between 45 and 60 minutes. Subsequently, after all the students finished, the teacher collected all the answer sheets for assessment and analysis.

3.5. *Data Analysis*

The statistical analysis of the quantitative data collected from the participants was developed with IBM SPSS Statistics, Version 27 (Armonk, New York, NY, USA) and Microsoft Excel. In the first phase, descriptive statistics of the variables in the studies were analyzed considering the different independent samples (experimental group and control group) and, in the second phase, inferential statistics were developed to identify the effects of different independent variables on the students' performance in the BCTt. The parametric tests t-Student and ANOVA were performed to analyze the existence of significant differences between different groups of samples, Levene's test to assess the equality of the variances ($p > 0,05$) and Shapiro-Wilk test to analyze the normality of the samples. For the acceptance of the formulated hypotheses, a p-value < 0.05 was considered as an indicator of statistical significance in the differences found.

3.6. *Ethical Issues*

Regarding the ethics of the research procedures, the schools and students' parents and legal tutors gave informed consent to participate in the research. The schools were previously contacted to request consent and authorization for their participation. The study was previously approved by the ethical commission of the Institute of Education of the University of Lisbon, and by the Office for The Monitoring of Surveys in Schools of the Ministry of Education. The informed consent document referred to the research goals, and the procedures used for data collection and analysis and guaranteed the anonymity of all participants and schools (Tuckman, 2012). Additionally, participants and their legal tutors were assured that the data collected would only be used for research purposes, that they could withdraw at any time, and that they would not be punished for the results obtained.

The use of the BCTt test, in this study, was formally authorized by the authors (Zapata-Cáceres *et al.,* 2020) through a previous contact by email, who shared with the research team the version translated into Portuguese.

4. Results

4.1. *Overall Computational Thinking Skills*

The first research question aimed to analyze the global results of the students in the computational thinking test (BCTt). A statistical analysis of the results obtained by all students in BCTt, considering the total score of each student in the test as the sum of the correct answers, is organized in Table 5. The overall average is moderate (*M* = 16,61; *SD* = 4.63; *Me* = 20). A graphical presentation of the overall results in the BCTt is illustrated in the boxplot of Fig. 8. This entire sample group includes students with previous experiences in computational thinking pedagogical activities (experimental) and students with no previous and formal contact with this type of activity (control). In the following subsection, the differences between the experimental and control group are detailed.

Table 5
Descriptive Statistics for the overall results of the students in BCTt (N = 189)

Descriptive Statistics		Statistic	Std. Error
Computational Thinking Skills Score	Mean	16,61	0,34
	Median	20,00	
	Variance	21,45	
	Std Deviation	4,63	
	Minimum	6,00	
	Maximum	25,00	
	Skewness	-0,78	0,17
	Kurtosis	0,11	0,35

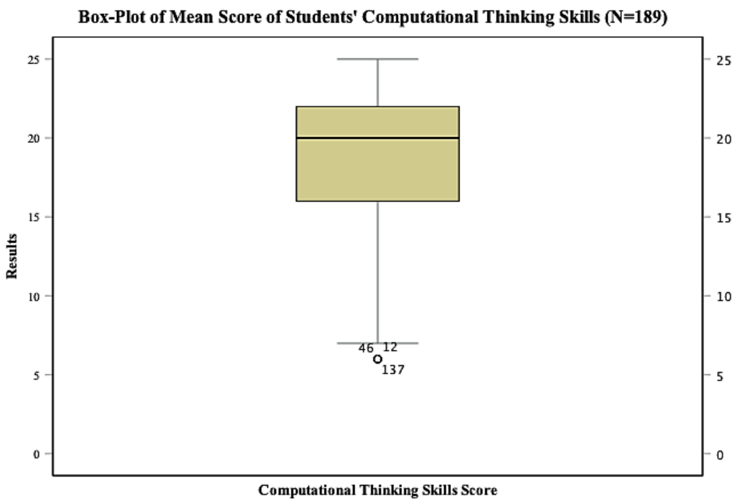

Fig. 8. Boxplot of overall results of students in the BCTt (N = 189).

A deeper analysis of the overall results, splitting BCTt items in different computational concepts that are present in the test, are represented in the Table 6 and the mean scores for each concept, normalized from 0 to 5, are reported in Fig. 9. The results showed that students were most successful on items measuring concepts related to sequences ($M = 5{,}47$; $SD = 1{,}07$; $Max = 6{,}0$) and Simple Loops ($M = 4{,}45$; $SD = 1{,}87$; $Max = 5{,}0$). In the opposite, it was on the items concerning the concepts of composite conditionals (if-the-else) ($M = 0{,}97$; $SD = 0{,}83$; $Max = 2{,}0$) and While ($M = 1{,}65$; $SD = 0{,}99$; $Max = 3{,}0$) that students showed the least success. These data confirm the progressive degree of difficulty of the test and of understanding the concepts involved in each item, with the final items of the test appealing to higher levels of analysis and thinking (CT skills).

To analyze the relationship between the total score and the scores obtained by the students in each computational concept, Spearman's correlation coefficient was calculated. This non-parametric test should be used when the assumption of normality of the distribution is not guaranteed.

Table 6
Descriptive Statistics for the overall results of the students in BCTt, organized by Computational Thinking Concepts (N = 189)

	Sequences	Simple Loop	Nested Loop	If-Then	If-then-else	While
N of Items	6	5	7	2	2	3
Mean	5,47	4,45	4,76	1,17	0,97	1,65
Median	6,00	5,00	5,00	1,00	1,00	2,00
Variance	1,15	1,28	3,50	0,63	0,70	0,99
Std Deviation	1,07	1,13	1,87	0,80	0,83	0,99
Minimum	0,00	0,00	0,00	0,00	0,00	0,00
Maximum	6,00	5,00	7,00	2,00	2,00	3,00
Skewness	-2,758	-2,40	-0,46	-0,34	0,50	-0,356
Kurtosis	8,76	5,37	-0,63	-1,35	-1,56	-0,93

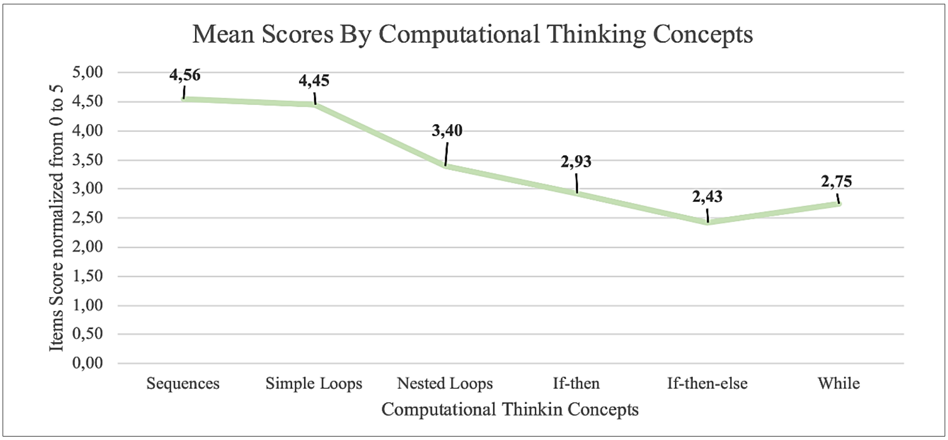

Fig. 9. Overall results of students in the BCTt organized by Computational Concepts (N = 189).

Spearman's correlation revealed that there is a positive correlation between the overall score and the scores obtained in each concept with moderate to high intensity ($0.55 < \text{rho} < 0.85$; $p < 0.001$).

4.2. *Effects of Scratch-Based Activities on Computational Thinking Skills*

The second research question intended to analyze the effect of pedagogical Scratch-based activities on Scores of Computational Thinking Skills, splitting the entire sample into two subsamples consisting of experimental and control groups (Fig. 10). The students in the experimental group scored higher (M = 20,91; SD = 3,96) than the students in the control group (M = 16,25; SD = 4,06), with a mean difference of 4,66.

The differences between the experimental and control group are further confirmed by the percentage of correct answers on the items related to the computational thinking concepts assessed. The Fig. 11 shows that in both groups the students were more successful in the concepts of sequences and simple loops and less successful in the concepts of conditionals. It also shows that in the experimental group the percentage of correct answers was higher in all concepts. For example, in the conditional If-then and If-then-else, the difference between groups is 31% and 35% respectively.

To analyze the statistical significance of the differences between the mean test scores obtained in the two groups, we applied the t-Student test for independent samples. The application assumptions of this statistical test were analyzed, in particular the normality of distribution, for both groups, through the Shapiro-Wilk test ($SW(96) = 0{,}82$; $p = 0{,}001$; $SW(93) = 0{,}97$; $p = 0{,}13$), and the homogeneity of variances through Levene's test based on the median ($F(1{,}187) = 1{,}34$; $p = 0{,}25$). Although the dependent variable did not as-

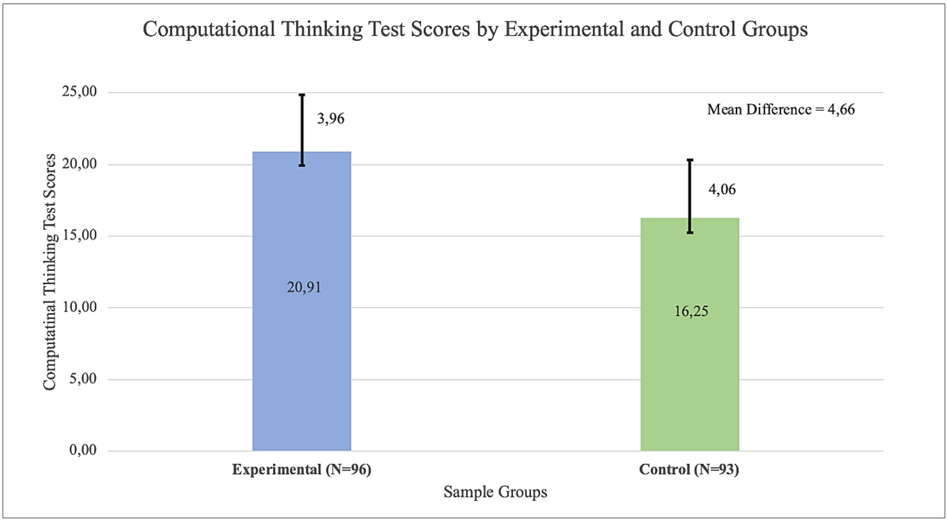

Fig. 10. Mean results on the Computational Thinking Score calculated for the experimental group and the control group. Values are presented as Mean ± Standard Deviation.

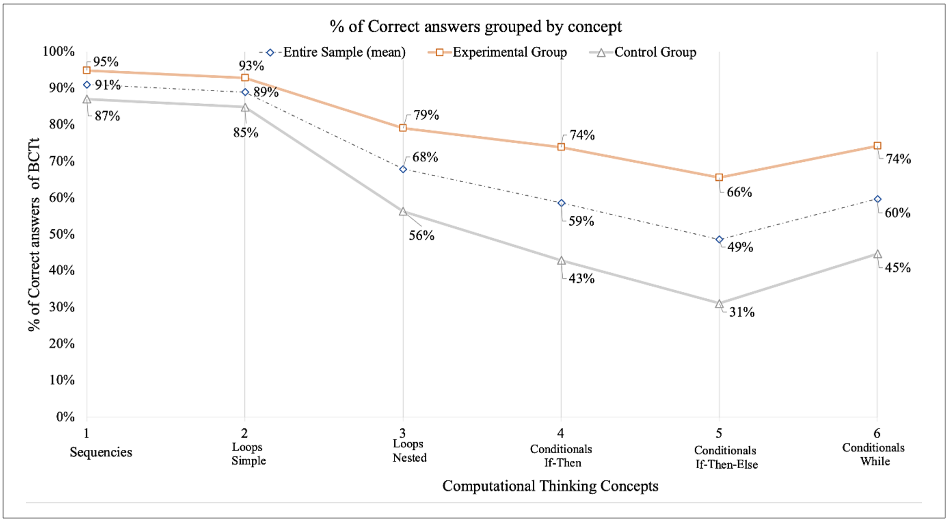

Fig. 11. Comparative analysis of the percentage of correct answers in each computational concept for the experimental, control and entire sample group.

sume a normal distribution in the experimental group, the t-Student test was sufficiently robust to the violation of the normality assumption when the asymmetry values are small (Mâroco, 2021).

The application of the t-test showed that the differences are statistically significant ($t(187) = 7{,}99$; $p < 0{,}001$). Thus, the hypothesis defined that Scratch-based pedagogical activities have a significant effect on students' knowledge and computational thinking skills (H1) was accepted. The magnitude of the effect calculated using Cohen's d is high ($d = 1.16$) and, according to the 95% confidence interval]0.85; 1.47[, students in the group that had contact with computational thinking activities through programming in Scratch present higher scores, on average, between 0,85 and 1,47 than students in the control group.

4.3. *Effects of Gender and Age on Computational Thinking Skills*

The third research question aimed to analyze the effects of gender and age on the computational thinking scores of the students in the experimental and control group. According to that two hypotheses are defined and tested (H2 e H3). Fig. 12 shows the mean scores obtained by the students considering gender and age. The results indicate that male students have higher mean scores than female students, and the differences are greater in the experimental group. We found similar results for the groups organized by age, and in this case, the differences between mean scores are very small.

To analyze the statistical significance of the differences found, we used the t-Student test, considering the samples distributed by gender and age. The test results revealed that the differences are not statistically significant considering the variables gender ($t(94) = 1{,}92$; $p = 0{,}57$; $t(91) = 0{,}87$; $p = 0{,}38$) and age ($t(94) = 1{,}89$; $p = 0{,}06$;

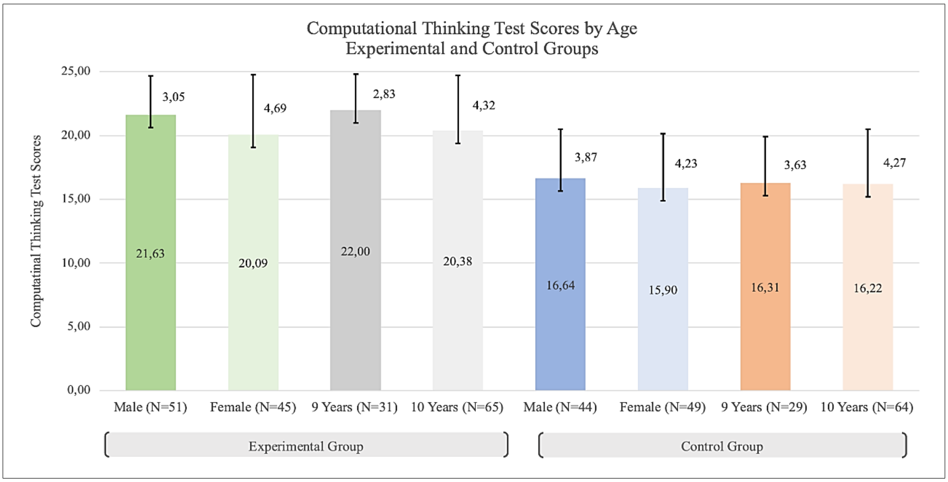

Fig. 12. Mean results on the Computational Thinking Score by age calculated for the experimental group and the control group. Values are presented as Mean ± Standard Deviation.

$t(91) = 0{,}10$; $p = 0{,}92$). Therefore, hypotheses H2 and H3 are rejected, and the variables gender and age did not have a significant effect on the students' score in the computational thinking test.

4.4. *Effects of Mathematics, Science and Portuguese Knowledge on the Computational Thinking Skills*

The fourth research question intended to explore the effects between the students' achievements in the 3 main curricular subjects of the 4th grade, in case, Mathematics, Sciences and Portuguese Language (mother tongue) and the students' computational thinking skills. This analysis involved the students of the experimental group because the final academic results only were available for this group.

For the comparative analysis, the students' academic results in the three subjects were organized qualitatively, between levels 1 (Very Unsatisfactory) and 5 (Very Good). The Table 7 shows the distribution of the sample by level for the 3 curricular subjects.

Table 7

Descriptive Statistics for the overall results of the students in BCTt (N = 189)

Level	Mathematics		Portuguese		Sciences	
	N	%	N	%	N	%
Very Unsatisfactory	0	0	0	0	0	0
Unsatisfactory	0	0	0	0	0	0
Satisfatory	24	25,00%	23	24,00%	13	13,50%
Good	36	37,50%	35	36,50%	36	37,50%
Very Good	36	37,50%	38	39,60%	47	49,00%

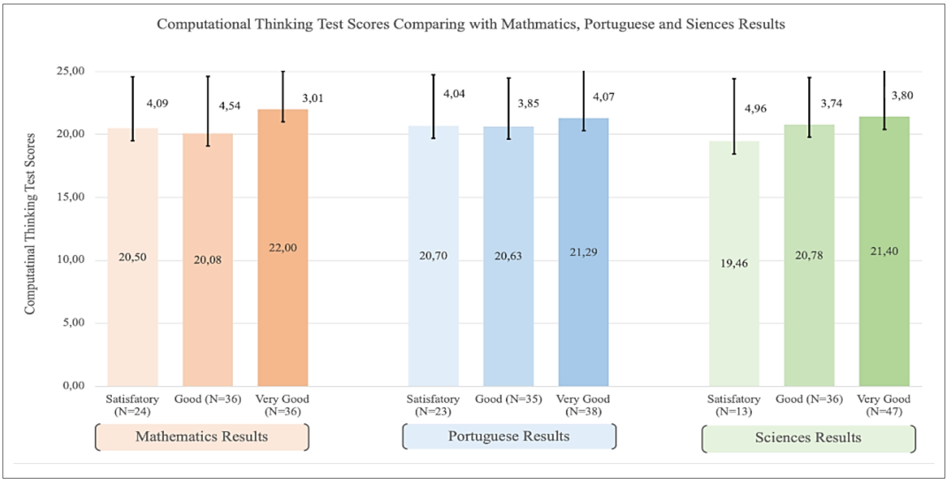

Fig. 13. Mean results of the Computational Thinking Scores considering Portuguese, Mathematics and Science Results (N = 96). Values are presented as Mean ± Standard Deviation.

It was noted that all students have positive results, and most of the students have good levels of knowledge in Math, Portuguese, and Sciences (levels 4 and 5).

The comparative analysis (Fig. 13) has shown that on average the students obtained equivalent total scores in computational thinking in the different groups. However, the students with better results in the different subjects (level 5) performed better on the computational thinking test ($M_{(Math)}$ = 22,00; $M_{(Port)}$ = 21,29; $M_{(Sci)}$ = 21.40). The comparative analysis of mean was implemented by ANOVA One-Way Test and no significant differences between groups were founded ($F(2)_{Math}$ = 2,34; p = 0,10; $F(2)_{Port}$ = 0,29; p = 0,75; $F(2)_{Sci}$ = 1,26; p = 0,29). Considering the results of ANOVA, hypotheses H4, H5 and H6 were rejected, because it is not possible to prove that students' scores in the 3 curricular subjects have a significant effect on their performance on the computational thinking test.

5. Conclusions

This study was directed to analyze the effect of scratch-based programming activities on the 4th-grade students' computational thinking skills and knowledge. In addition, the effects of age, gender, and the results of the students' achievements in Mathematics, Science, and Portuguese Language on the computational thinking test scores were analyzed.

The analysis of the results of the entire sample of students on the computational thinking test was positive and students were found to have acquired basic-level computational concepts. Confirming the first research hypothesis, it was found that students who had programming experiences with scratch showed higher results in BCTt than students who did not develop these activities. The differences in the results are statistically significant, which reveals that the activities developed in the experimental group have a positive effect on the students' results. Thus, the importance of students' engagement in regular activities involving the development of projects, problem-solving, and game creation using block-based programming languages is evident. These activities not only promote the development of computational thinking skills but also contribute to the application of knowledge from other curricular areas. Programming activities in primary schools could promote projects that gamify or simulate other curricular topics (Fagerlund *et al.,* 2020). Scratch has proven to be an excellent tool for developing projects that promote computational concepts (Brennan and Resnick, 2012), particularly in younger students.

No significant differences were found in the students' performance in the computational thinking test considering gender, age, and the students' results in the three curricular areas of the 4th-grade. However, it was found that male and 9-year-old students have higher performance.

Students in both groups showed better performance on questions involving sequences, simple and nested loops and lower performance on questions requiring a higher cognitive level related to conditional decisions that imply decision-making based on certain conditions that students must predict and analyze. In these questions, the differences in the students' performance between the control group and experimental groups were higher. Although the focus of the CT test (BCTt) has been the assessment of basic computational concepts, the problems presented in each question have implicit computational thinking concepts such as decomposition, abstraction, pattern recognition, and analysis of algorithms in symbolic language.

The computational thinking test selected for these studies (Zapata-Cáceres *et al.*, 2020) proved to be adequate for this group of students, confirming itself as a valid instrument to analyze computational knowledge. The results regarding the reliability and facility index of the test proved to be similar to the results of the original authors' validation of the test. These results make available a valid instrument that can be used in other studies in Portuguese language contexts, which is also a relevant contribution.

Despite possible limitations in the constitution of the groups and the lack of pre-testing, this study produced relevant results about the impact of scratch programming projects on the development of knowledge and computational thinking skills of primary school students.

It is important that in future studies, in addition to analyzing basic knowledge of computational concepts, students' practices and perceptions of computational thinking be evaluated. The assessment of these three dimensions proposed by Brennen and Resnick (2012) in their framework will allow for an improved understanding of the skills and knowledge developed by students. To this end, it will be important to mobilize problems and challenges that allow students to explore computational practices such as testing and debugging, reusing and remixing, abstracting, and expressing their computational perspectives on the impact of computing in the real world and in everyday activities. Projects or problems that allow working the more computational and algorithmic aspects (Denning and Tedre, 2019) in articulation with more creative and self-expression aspects (Brennan and Resnick, 2012). Corroborating Fagerlung *et al.* (2022, p. 17), Teachers and Educators need evidence-based pedagogical knowledge increasingly to support students' CT learning through programming in primary school classrooms.

Acknowledgments

This research received national funding from the FCT – Fundação para a Ciência e a Tecnologia, IP, within the scope of the UIDEF – Unidade de Investigação e Desenvolvimento em Educação e Formação – UIDB/04107/2020".

References

Ambrosio, A.P., Almeida, L. da S., Franco, A., Macedo, J. (2014). Exploring Core Cognitive Skills of Computational Thinking. In: *PPIG 2014 – 25th Annual Workshop*. Sussex: University of Sussex.

Basso, D., Fronza, I., Colombi, A., Pahl, C. (2018). Improving assessment of computational thinking through a comprehensive framework. In: *ACM Proceedings of the 18th Koli Calling International Conference on Computing Education Research*, a15.

Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M.A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V., Stupurienė, G. (2022). *Reviewing Computational Thinking in Compulsory Education*, Inamorato Dos Santos, A., Cachia, R., Giannoutsou, N. and Punie, Y. editor(s), Publications Office of the European Union, Luxembourg, 2022, ISBN 978-92-76-47208-7. https://doi.org/10.2760/126955, JRC128347

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*. Vancouver, Canada, pp. 1–25.

Cohen, L., Manion, L., Morrison, K. (2018). *Research Methods in Education* (8th edition). Routledge, New York, NY.

Delal, H., Oner, D. (2020). Developing middle school students' Computational thinking skills using unplugged computing activities. *Informatics in Education*, 19(1), 1–13.

Denning, P., Tedre, M. (2019). *Computational Thinking*. Mit Press Ltd, Cambridge, MA.

Durak, H. (2020). The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. *Tech Know Learn*, 25, 179–195. https://doi.org/10.1007/s10758-018-9391-y

Fagerlund, J., Häkkinen, P., Vesisenaho, M., Viiri, J. (2020). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29, 12–28. https://doi.org/10.1002/cae.22255

Fagerlund, J., Häkkinen, P., Vesisenaho, M., Viiri, J. (2020). Assessing 4th grade students' computational thinking through Scratch programming projects. *Informatics in Education*, 19(4), 611–640. http://dx.doi.org/10.15388/infedu.2020.27

Fagerlund, J., Vesisenaho, M., Häkkinen, P. (2022). Fourth grade students' computational thinking in pair programming with Scratch: A holistic case analysis. *International Journal of Child-Computer Interaction*, 33, 100511. https://doi.org/10.1016/j.ijcci.2022.100511

Korkmaz, Ö., Çakir, R., Özden, M.Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558–569. https://doi.org/10.1016/j.chb.2017.01.005

Machuqueiro, F., Piedade, J. (2022). Development of computational thinking using board games: A systematic literature review based on empirical studies. *Prisma Social*, 38, 5–36. https://revistaprismasocial.es/article/view/4766

Marôco, J. (2021). *Análise Estatística com o SPSS Statistics (8th Edition)*. ReportNumber, Pero Pinheiro.

Moreno-León, J., Robles, G. (2015). Dr. Scratch: A web tool to automatically evaluate scratch projects. *ACM International Conference Proceeding Series*, 09-11-Nove(October), 132–133. https://doi.org/10.1145/2818314.2818338

Montiel, H., Gomez-Zermeño, M. (2021). Educational challenges for computational thinking in K–12 education: A systematic literature review of "Scratch" as an innovative programming tool. *Computers*, 10, 69. https://doi.org/10.3390/computers10060069

Mühling, A., Ruf, A., Hubwieser, P. (2015). Design and first results of a psychometric test for measuring basic programming abilities. In: J. Gal-Ezer, S. Sentance, J. Vahrenhold (Eds.), *WiPSCE'15: Proceedings of the Workshop in Primary and Secondary Computing Education*. New York: ACM, pp. 2–10.

Piedade, J., Dorotea, N., Pedro, A., Matos, J.F. (2020). On teaching programming fundamentals and computational thinking with educational robotics: A didactic experience with pre-service teachers. *Education Sciences*, 10(9), 214. https://doi.org/10.3390/educsci10090214

Piedade, J., Dorotea, N., Ferrentini, F.S., Pedro, A. (2019). A cross-analysis of block-based and visual programming apps with computer science student-teachers. *Education Sciences*, 9(3). https://doi.org/10.3390/educsci903018.1

Poulakis, E., Politis, P. (2019). Computational thinking assessment: Literature review. In: Tsiatsos, T, Demitriadis, S, Mikropoulos, A, Dagdilelis, V. (eds.), *Research on E-Learning and ICT in Education*. Springer, Singapura, 111–128.

Ramos, J.L., Espadeiro, R.G. (2014). Os futuros professores e os professores do futuro. Os desafios da introdução ao pensamento computacional na escola, no currículo e na aprendizagem. *Educação, Formação e Tecnologias*, 7(2), 4–25. http://hdl.handle.net/10174/14227

Resnick, M. (2013). *Learn to Code, Code to Learn*. EdSurge, May.

Rodríguez-Martínez, J., González-Calero, J., Sáez-López J. (2020). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. *Interactive Learning Environments*, 28(3), 316–327, https://doi.org/10.1080/10494820.2019.1612448

Román-González, M. (2015). Computational thinking test: Design guidelines and content validation. EDULEARN15 at: Barcelona.

Román-González, M., Moreno-Léon, J., Robles, G. (2019). Combining assessement tools for a comprehensive evaluation of computational thinking interventions. In: Kong, S.C., Adelson, H. (eds.), *Computational Thinking Education*. Springer, Singapura, 79–98.

Román-González, M., Pérez-González, J., Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*, 72, 678–691.

Sáez-López, J.-M., Román-González, M., Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97, 129–141. https://doi.org/10.1016/j.compedu.2016.03.003

Sigayret, K., Tricot, A., Blanc, N. (2022). Unplugged or plugged-in programming learning: A comparative experimental study. *Computers & Education*, 184, 104505. https://doi.org/10.1016/j.compedu.2022.104505

Tuckman, B.W. (2012). *Manual de Investigação em Educação (4th Edition)*. Fundação Calouste Gulbenkian, Lisboa.

Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.

Wing, J. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7–14. https://doi.org/10.17471/2499-4324/922

Yagci, M. (2019). A valid and reliable tool for examining computational thinking skills. *Education and Information Technologies,* 24, 929–951. https://doi.org/10.1007/s10639-018-9801-8

Zapata-Cáceres, M., Martín-Barroso, E., Román- González, M. (2020). Computational thinking test for beginners: Design and content validation. In: *IEEE Global Engineering Education Conference (EDUCON'20)*, pp. 1905–1914. https://doi.org/10.1109/EDUCON45650.2020.9125368

Zapata Cáceres, M., Martín-Barroso, E., Román-González, M. (2021). BCTt: Beginners computational thinking test. In: *Understanding Computing Education* (Vol 1). Proceedings of the Raspberry Pi Foundation Research Seminar series.

J. Piedade (Institute of Education of University of Lisbon (IE-ULisbon), Lisbon, Portugal) received his PhD degree in ICT in Education from the University of Lisbon in 2017, the Master in Technologies and Methodologies in E-learning in 2010 and the Bachelor degree in Informatics Engineering (2006). Actually, he assistant professor, member of the R&D Unit Education and Training Research and Development Unit (UIDEF), member of the School Council of Institute of Education and coordinator of Scientific Committee of Master in Teaching Informatics and coordinator of postgraduate course in Technologies and Robotics in Basic Education. Their main research interests are concern with Computational Thinking in Education, Didactics of Informatics, Computer Science Teacher Education, Educational Robotics, Technology Enhanced Learning, Initial Teacher Training, Innovative Learning Spaces and Virtual and Immersive Learning Environments. He is member of the National Association of Informatics Teachers and consultant at several Portuguese institutions in the field of digital technologies and education. Author and co-author of several scientific papers published on national and international journals in the field of digital technologies in education and didactic of informatics. More information at `https://jmpiedade.pt`.

N. Dorotea (Institute of Education of University of Lisbon, Lisbon, Portugal) holds a Ph.D. in Education in the specialization area of Information and Communication Technologies in Education, Education, from the Institute of Education of the University of Lisbon, Portugal (IE-ULisbon), a Master degree in E-learning Technologies and Methodologies and a postgraduate degree in Educational Administration from the same institution. He is an invited assistant professor and researcher at the Institute of Education of Lisbon University (IEUL), where integrates the R&D Unit Education and Training Research and Development Unit (UIDEF). Member of the Network of Portuguese Centers of ICT Competences of the Ministry of Education. Design and conceives training actions in the scope of professional teacher development in the field of digital technologies in education and in the formative assessment of learning with technologies. His research interests include technology-enhanced learning, online learning, computer science initial teacher training, computer science education, teacher training in digital technologies, educational programming and robotics, computational thinking, internet safety and digital citizenship, assessment of learning with digital technologies, adaptive platforms in learning and assessment, artificial intelligence in education, assessment of the impact of the Apps for Good project in the national context, and digital technologies in the management processes of educational organizations. More information at `http://www.ie.ulisboa.pt/docente/nuno-dorotea`.