Development of Computational Thinking in Brazilian Schools with Social and Economic Vulnerability: How to Teach Computer Science Without Machines

Christian Puhlmann Brackmann, Dante Augusto Couto Barone,
Rafael Marimon Boucinha, Janice Reichert
Universidade Federal da Fronteira Sul – UFFS
Brazil

Abstract

Computational Thinking (CT) has been placing the focus of educational innovation as a set of troubleshooting skills. Unfortunately, there is not a consensus if the teaching methodology and the available materials attend the expectations of the lecturers. To prove the impact that CT training has in primary school, we attempted to evaluate primary school students with a Quasi-Experimental approach and taking Unplugged CT classes in Brazilian Schools with Social and Economic Vulnerabilities. The research happened in two schools to prove if the activities are effective for students who live in areas where there are no electronic devices, Internet or even electrical power can be also benefited. The results show statistically significant improvement. Our study finds shows that we are able to reinforce the claim that CS unplugged is an effective approach and it is an alternative for students who live in unprivileged areas.

Keywords: Computational Thinking Unplugged; Evaluation; Computers in Education; Primary School; South America; Brazil

1. Introduction

In current times, marked by the fluidity of information and the value placed on knowledge, the challenge imposed on users of information and communication technologies has been to create their own systems (for example, programs and games) or modify existing ones according to their personal needs. More than ever, in order to deal with information, to process it and to transform it into competences, the domain of knowledge and skills related to Computational Thinking (CT) has become fundamental (Kologeski, Silva, Barbosa, Mattos, & Miorelli, 2016). Due to this trend, CT has been adopted in several countries in primary schools (C. Brackmann, Barone, Casali, Boucinha, & Munoz-Hernandez, 2016).

Wing (2006) defines CT as a mental activity for the formulation of a problem that can be solved computationally, in other words, it is a thought processes involved in identifying a problem and expressing its solution effectively, so that both machines and people can execute them. Publications and researches led by Code.Org (CODE.ORG, 2015), Liukas (2015) and BBC Learning (2015) merged the elements cited by Grover and Pea (2013) summarizing the so-called "Four Pillars of Computational Thinking" (or dimensions) for a problem-solving approach: Decomposition, Pattern Recognition, Abstraction, and Algorithms. These pillars are very important and are interdependent during the process of formulating computationally feasible solutions.

Therefore, Computational Thinking involves identifying a complex problem and dividing it into smaller, easier-to-manage pieces (DECOMPOSITION). Each of these smaller problems can be analyzed individually in greater depth to identify similar problems that were previously encountered (PATTERN RECOGNITION), focusing on the important details and ignoring irrelevant information (ABSTRACTION). Finally, simple steps or rules can be created to solve each of the sub-problems found (ALGORITHMS). By proposing rules or steps used to create a code, the result becomes understandable for use in computational systems, and consequently, in solving complex problems efficiently.

In this context, different activities were created and adapted for teachers so that they could use and replicate this material in their classes without the need for electronic equipment, internet, or electricity. Such activities enabled children to study computer concepts in schools without appropriate equipment (e.g., ruined, outdated, or lacking) or located in geographically distant areas (e.g. rural or forest areas). It is believed that, by using these activities without the use of machines (hence, unplugged activities), it is possible to teach Computational Thinking in a more accessible manner, using basically paper, scissors, pens, coloring pencils, glue, and other commonly used school materials.

The unplugged approach is the only one possible for a huge number of schools around the world that do not have basic technology infrastructure (Unnikrishnan, Amrita, Muir, & Rao, 2016), such as electricity, Internet, computers, mobile devices, and other electronic devices. According to UNESCO, the use of ICT in education is still at a very early stage in most countries in sub-Saharan Africa, since the percentage of basic infrastructures in primary schools is under 15% in all the region (UNESCO Institute for Statistics, 2015). In other regions, such as Asia, the percentage of schools with basic infrastructure is also far from being close to 100% (UNESCO Institute for Statistics, 2014). But even in most European countries, there are still remote, rural areas with a lack of proper resources.

In the literature, there is little research on the application and evaluation of unplugged students. Seeking to fill this gap, this paper presents research carried out in two primary schools in Brazil, where the objective was to verify the effectiveness of Unplugged Computational Thinking classes in primary education. To accomplish this, pre and post-test questionnaires were applied, before and after Unplugged Computational Thinking classes, in order to verify if the children presented better performance regarding Computational Thinking abilities by doing activities without computers. On that ground, Computational Thinking learning objects were developed, classroom interventions were carried out with unplugged activities and, at last, the evaluation of the Computational Thinking of the students who participated in the intervention and the control group occurred.

Thus, this article is composed of six sections that follow this Introduction. The second section is regarding the history and contextualization of the Unplugged Computational Thinking approach. The third section presents the methods and materials used in the research, followed by the fourth section in which shows the quantitative and qualitative results. Finally, the fifth section presents the conclusion of the paper and recommendations for future work, which is followed by acknowledgements of the institutions supporting the research, as well as the bibliography used.

2. Computational Thinking in Basic Education

In Brazil, the final document of the Base Nacional Comum Curricular (National Curricular Common Base) (BNCC), which is a document created to conduct the teaching of Brazilian schools, from kindergarten to high school, was finally approved by the Minister of Education (MEC) in December 2018. All school institutions in Brazil must, necessarily, implement the BNCC by the end of 2019.
In its paper, BNCC states that throughout basic education, students must develop "Ten General Competences", both cognitive and social-emotional, which include the exercise of intellectual curiosity, the use of digital communication technologies and the appreciation of individuals diversity. One can highlight three competences that follow the line of Computational Thinking, being them (MEC, 2018):

> - To exercise intellectual curiosity and to use the science-based approach, including research, reflection, critical analysis, imagination and creativity, to investigate causes, to elaborate and test hypotheses, to formulate and solve problems and to invent solutions based on knowledge of different areas;
> - Use verbal knowledge (oral and written) or verbal-visual (as in Libras, Brazilian Sign Language), body, multimodal, artistic, mathematical, scientific, technological and digital languages to express and share information, experiences, ideas and feelings in different contexts and, with them, produce meanings that lead to mutual understanding;
> - Use digital communication and information technologies in a critical, meaningful, reflective and ethical way in the various daily practices (including the school ones) by communicating, accessing and disseminating information, producing knowledge and solving problems (MEC, 2018, p. 11).

In addition, the BNCC predicts the use of concepts of Computational Thinking in Mathematics disciplines to assist the process of solving a problem, according to (MEC, 2018):

> Algebra learning can contribute to the development of students' Computational Thinking, since they need to be able to translate a given situation into other languages, such as transforming problem situations presented in the mother language into formulas, tables and graphs, and vice versa.
>
> Associated with Computational Thinking, it emphasizes the importance of the algorithms and their flowcharts, that can be study objects in the classes of Mathematics. An algorithm is a finite sequence of procedures that solves a given problem. Thus, the algorithm is the decomposition of a complex procedure into its simplest parts, relating and ordering them, and can be graphically represented by a flowchart. Algorithmic language has points in common with algebraic language, especially in relation to the concept of variable. Another ability related to algebra closely related to Computational Thinking is the identification of patterns for establishing generalizations, properties, and algorithms (MEC, 2018, p. 271).

The integration of PC in Basic Education is also analyzed in Valente's research (2016), where he carries out a survey among different authors and defines six categories of approaches in teaching the concepts of Computing in Basic Education, namely: activities without the use of technologies, programming in Scratch, pedagogical robotics, digital narratives productions, game creation and use of simulations.

Each of the approaches has a different characteristic to reach the common goal: the development of Computational Thinking. Notice that all the approaches mentioned in the research require the use of specific equipment and software, except the first. A non-technology approach, also known in the literature as "Unplugged Computational Thinking" or "Offline," has become one of the main focuses of this research, considering its ease of application in different economic and social realities in Brazil.

3. Unplugged Computational Thinking

The literary records about the emergence of Unplugged Computational Thinking are diffuse, since it is known that the need for abstraction to create any software and hardware is an essential part of Computer Science (CS). Instead of participating in an expository class, unplugged activities often occur through kinesthetic learning (e.g., moving, using cards, cutting, pasting, drawing, painting, solving riddles, etc.) and students work together to learn CC concepts.

In relation to elementary education classrooms, the first records refer to Bell *et al.* (1997), with the launch of a book draft in digital format called "Computer Science Unplugged. Off-line activities and games for all ages," intended for teachers interested in differentiated classes for their students, applicable at all academic levels. At the time the idea was well received by the other teachers, as well as by Academia. Due to the quality of the material published, the Association for Computing Machinery (ACM) recommended that the activities contained in the book be part of the curriculum proposed by the Computer Science Teachers Association (CSTA) of the United States of America. Until the publication of this article, the book *CS Unplugged* is in version 3.1 and can be accessed at the project site (Bell, Witten, & Fellows, 2015)

There are several studies that investigate the efficiency of programming languages (visual and coding) with children (Román-González, Pérez, & Carmen Jiménez-Fernández, 2015) (Román-Gonzáles, Pérez-González, & Jiménez-Fernández, 2017) (Shuchi Grover & Basu, 2017) (Franklin et al., 2017)), but they lack unplugged approaches. Other studies have attempted to standardize the evaluation and teaching of CT Unplugged activities, such as (Nishida et al., 2009), in which he presented a proposal for a design pattern, a transversal evaluation of CT at a high school (Feaster, Segars, Wahba, & Hallstrom, 2011), case studies in the process of adopting CT in the classroom (Curzon, 2013), and evaluation of studen points of view regarding CS before and after CT classes (Taub, Ben-Ari, & Armoni, 2009), as well as suggestions on how teachers can assess student progress in performing CT activities (Curzon, McOwan, Plant, & Meagher, 2014). Lambert *et al.*(Lambert & Guiffre, 2009) made a similar attempt, however to identify an increase in interest in the areas of Computation or Mathematics, without checking the increase/decrease in skills related to Computational Thinking.

The solution proposed by (Rodriguez, Stephen, Rader, & Camp, 2017) sought to evaluate students doing unplugged activities at basically three-levels (proficient, partially proficient, and unsatisfactory). In (Campos et al., 2014) there was also the attempt to adopt a test, however without presenting satisfactory results. However, (Scaico, Mychelline, Cunha, & Alencar, 2012) carried out an evaluation of student success, but without the use of a pre and post-test to verify changes in their performance.

The works cited here are part of a large set of studies that try to measure CT skills, but do not use a direct solution that is easy to apply and with a formal validation process to achieve a more precise result as

proposed by (Román-González, 2015; Román-González et al., 2015).
Without proper evaluation, Computational Thinking in the classroom will not be likely to follow the path of success in any curriculum (Grover, 2013), that is, in addition to the need to evaluate the effectiveness of any curricular approach by integrating Thought Computational, it is necessary to define attributes that allow educators to evaluate what students have learned.
It is also important to note that the use of physical examples and school materials are common to simulate the behavior of machines up to the present day in undergraduate courses. Many important Computer Science (CC) topics can be taught without the use of computers. The unplugged CC approach introduces hardware and software concepts that take everyday technologies to non-technical people.

4. Methods

In this section, we describe the sample in our research, and how participants were divided into two different groups-conditions: the experimental group-condition and the control group-condition. Then, we present the instrument used for assessing the CT skills of the participants from both conditions, with a pre-test and a posttest. The pedagogical materials containing the unplugged activities taken by the experimental group along the teaching sessions are then explained. Finally, we report the procedure followed in our quasi-experiment.

4.1 Participants and Test Groups

The research was developed in years 2016 through 2017. The valid sample of our quasi-experiment, that is, the set of individuals who were assessed both in the pre-test and post-test time, is composed by 63 students enrolled in the 5th and 6th grade (10-12 years old). The CT Tests and the classes were applied in a public-school system in the city of Santa Maria, an inland city of the Rio Grande do Sul state. The children who participated in the research were chosen randomly by the school management and they participated in the research activities on a voluntary basis. None of the participants had formal programming experience. The distribution of the participants by gender, level, age, and class (group) are shown on **Table 1.**

Table 1. Distribution of research subjects

	Grade	Age	Cond	Gender		Total
				Boys	Girls	
School A	5th	10-11	C	7	3	10
			E	7	8	15
School B	6th	11-12	C	13	6	19
			E	8	11	19
Total				35	28	63

Author

4.2 Evaluation Instrument: The Computational Thinking Test

The Computational Thinking Test (CT Test) (Román-González, 2015, 2016; Román-Gonzáles, Pérez-González, et al., 2017) was the instrument used to assess the level and development of the participants' CT skills. The CT Test was selected for our research because of its precise (although necessarily reductionist) operational definition of CT, which may shed some light on the controversy surrounding this often blurry construct (S. Grover & Pea, 2013; Kalelioglu, Gülbahar, & Kukul, 2016). The CT Test was also elected due its quantitative and aptitudinal approach, and because it has already undergone a rigorous validation process, which has stated its content validity (Román-González, 2015), criterion validity (Román-Gonzáles, Pérez-González, et al., 2017), and convergent validity (Román-Gonzáles, Moreno-León, & Robles, 2017). This test attempts to identify the skills to form and solve problems, based on the fundamental concepts of computation, in addition to using the logical syntax used in programming languages. All the items that assemble the test involve, to a greater or lesser extent, the four-pillar cognitive processes of CT: decomposition, pattern recognition, abstraction and algorithmic design. Thus, when a student tries to solve an item (e.g., item #8, see **Figure 1**), the student must: break down the steps that the Pac-Man should follow; recognize the visual patterns on the marked path (e.g, in the item #8 there is a repeated pattern that consists of advancing four squares and then turning to the right); abstract the core elements of the problem and ignore the irrelevant details (e.g., such as the color of the path or the features of the characters); and design an algorithm to solve the problem, which involves some computational concepts (e.g., in item #8, nested loops must be used along the algorithmic design).

The psychometric studies of the CT Test support that this test is reliable ($\alpha \approx .80$) and valid for assessing the level of CT in students from 10 to 16 years old. The instrument is composed of 28 multiple choice questions, each of which has four alternative answers of which only one is correct. It is divided basically into three parts, the first uses arrows to move the character, the second makes a move related to the position and direction of the character using blocks and the last one uses a pencil to make drawings also using blocks. The test can be applied using any browser (e.g., Chrome, Firefox, Edge) and may be accessed from any device. The study only used the school's computer lab equipment. Three examples of the CT Test are show in Figure 1, Figure 2 and Figure 3 (Anonymous, 2007).

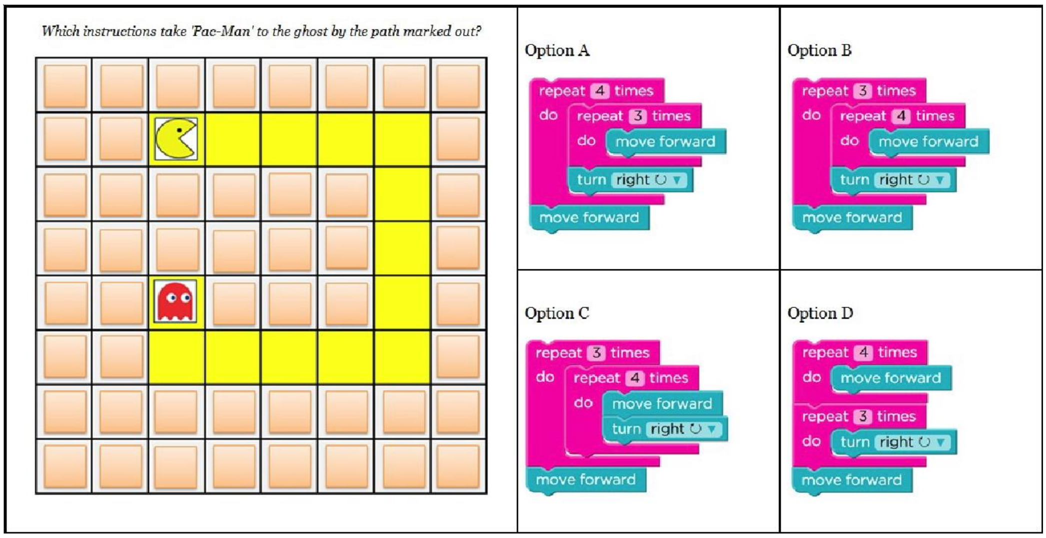


Figure 1. CT Test, question #8 ('maze'): loops 'repeat times' (nested); 'visual blocks'; 'sequencing'.

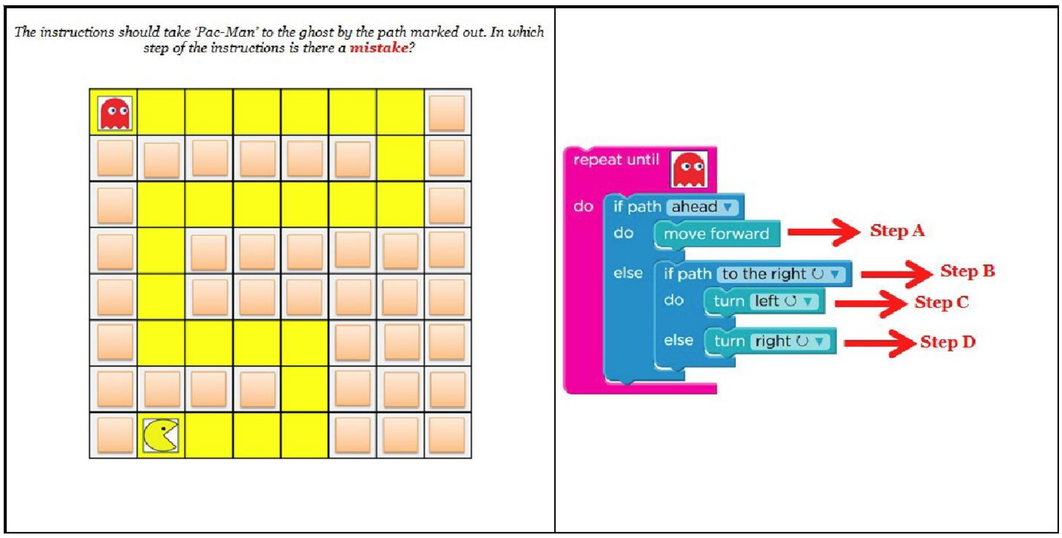


Figure 2. CT Test, question #16 ('maze'): loops 'repeat until' + if/else conditionals (nested); 'visual blocks'; 'debugging'.

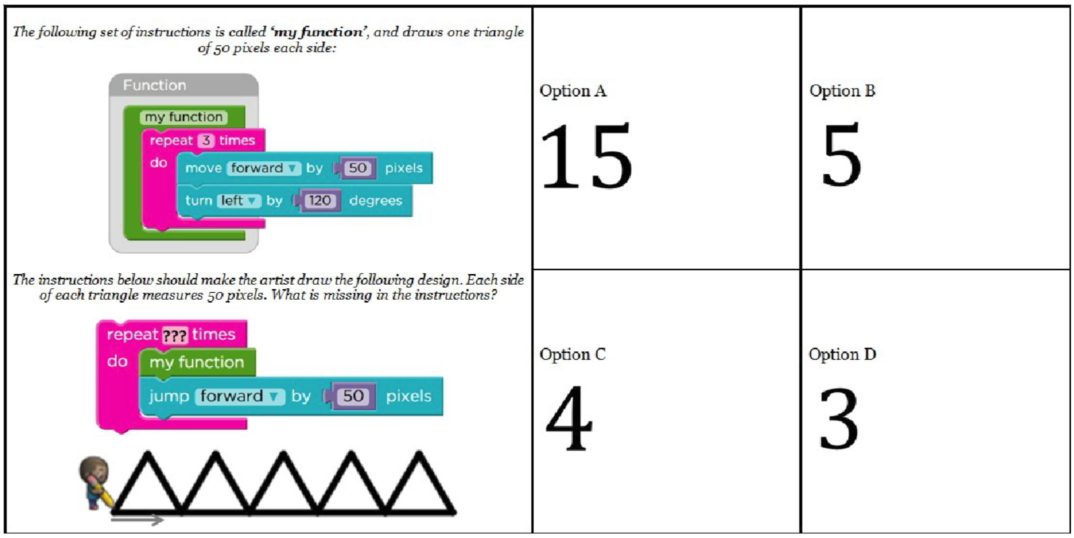
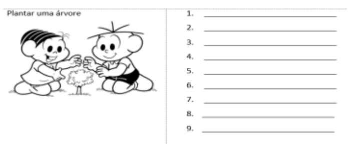


Figure 3. CT Test, question #26 ('canvas'): loops 'repeat times' + simple functions; 'visual blocks'; 'completing'.

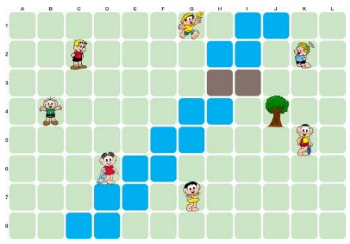
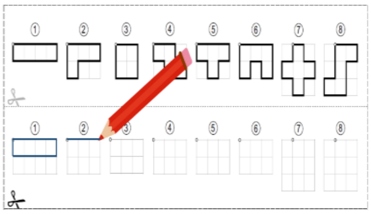
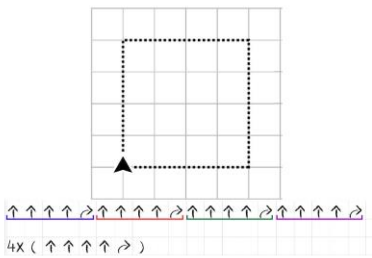
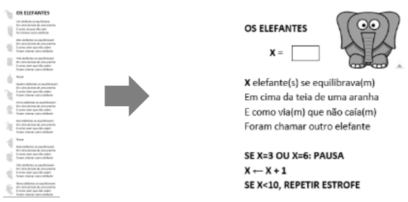
4.3 Development of the Activities

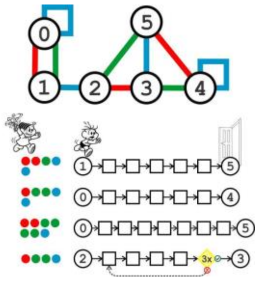
Most of the pedagogical unplugged materials used with the experimental group were developed by the authors for this study, while the rest were translated into Portuguese and adapted from pre-existing activities such as the book *Hello Rubby* (Liukas, 2015) and the board game "Code Master", created by (Engelberg & Thinkfun, 2015). In an attempt to attract the children's attention, popular characters were also used in the activities. For the reader to become familiar with the activities, some are presented in **Table 2** and others activities are available in the "*Pensamento Computacional*" website[1]. It is important to mention that the main objective of the activities is to assist in the teaching/learning process of Computing concepts for elementary school children. Its application procedure is described in greater detail in the following section.

Table 2. Activity Examples

Image	**Description of Activity**	**Main pillars involved**
	"Decomposition" activity: Students had to break down many problems (e.g. Plant a tree) identifying all the steps necessary to solve it. Other examples were: Wash Hands, Prepare breakfast, Take an elevator, Tie a shoe, etc.	Decomposition Algorithms

[1] http://www.computacional.com.br/

	"Monica's Map" activity: A map with many characters is shown to the students and they have to find the shortest route between them using only up, down, left and right arrows (→, ←, ↑, and ↓). On a second moment, they should use multipliers (i.e. → → → → → = 5x→) to write down the solutions.	Pattern Rec. Algorithms.
	"Tetris" activity: some drawings of Tetris pieces are presented to one of the students who gives instructions to its partner. The student who got the upper part of the paper had to hide the images from the partner so it would be possible only to hear the instructions without looking to the answers. The instructions are limited to "start", "up", "down", "left", "right", and "stop". No other words can be used to describe how the figure is drawn.	Abstractions Algorithms
	"Repetition Drawing" activity: allows the students to understand the use of repetitions on Tetris-like figures. In this case, the students need to use instructions based on the perspective of the direction of the arrow and try to use the most amount of multipliers in their command. Differently from the "Tetris" activity, the students do it individually and only the use of turn left, turn right and forward are available (↑, Å, and ¼). The pillars of abstraction, pattern recognition and algorithm are mainly developed.	Decomposition Abstractions Pattern Rec. Algorithms
	The Elephants: the activity uses a popular children's song to exemplify how a song can become an algorithm. This particular song works on the concepts of repetition, variables, and conditional statements.	Pattern Rec. Algorithms

	"Monica's Automata": The last activity is a simpler remake of the Code Master board game developed by the ThinkFun company. In this activity the student is supposed to find a route between two nodes using the allowed colors for each path. All the colors had to be used, leaving no blank spaces. The number located on the left side is the start point and on the right side the finish point.	Decomposition Abstractions Algorithms

Author

4.4 Procedure

To apply this research, contact with the schools and the project presentation were made up to one month in advance of the beginning of the tests and the classes. The researchers were very well received by both institutions, which offered all the necessary support. Each of the schools had at least two classes of a specific grade (School A: two fifth grade groups and School B: two sixth grade groups). Among the available classes, the Experimental Class and the Control Class were randomly selected, respecting the existing grouping of the subjects in their natural classrooms (i.e. the individuals were not randomly assigned to the conditions).

The classes with Unplugged CT activities were applied to the Experimental Group after the pre-test and before the post-test, totaling 10 class hours. However, in order for the Control Group to have the same opportunity to benefit from the proposed activities, CT classes were given after the post-test, without generating the quantitative and qualitative data. Thus, both groups did the same activities in a different order, as shown in Figure 4.

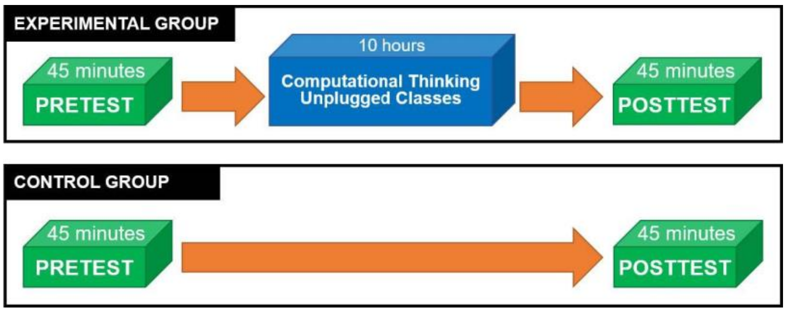


Figure 4. Research Stages

During the first week of the research, students from all four classes were invited to participate in the experiment as part of their regular classes during the first semester of 2017. For the application of the pre-tests, the students were accompanied by their teachers to the school's computer lab, where they remained for up to 60 minutes to carry out an individual test developed described in section 4.2.

During the next five weeks, once a week (approximately two hours per week) CT classes were given to the experimental group using the materials presented in section 3.3. In each weekly session, it was possible to work on an average of two activities.
In the seventh week, students from both groups (Experimental and Control) were sent back to the computer lab so they could take the post-test in the same way as described before. After six weeks elapsed between the pre-test and the post-test, it is a sufficient time to avoid the undesirable ’memory-effect’ of using an identical set of items at both administrations.
All student responses were recorded in the Google Cloud so they could be viewed, retrieved, and converted, and then tabulated and analyzed statistically by XLStat 2018.3 and Past 3.20 (Hammer, Harper, & Ryan, 2001). Results and discussions on the collected data are available in the next section.

5. Results and Discussions

This section briefly described the findings of our quasi-experimental research. At the first moment, we report the qualitative results which include informal observations of the researchers during the application of unplugged activities and the CT Test. In the next section, we present the quantitative results which intends to answer the research question “Did que unplugged activities improve the Computational Thinking skills of the students?”. Remembering that none of the students had any previous contact with formal programming classes.

5.1 Qualitative performance of student activities

During the application of the tests and activities, the researchers made several notes related to minor adjustments or corrections of the activities to make them easier to understand. Due to space limitations, only the most relevant records are listed below (please use **Table 2** as reference):

- All activities were well accepted by the students, depending on the class level (e.g., "The Elephants" activity uses children's music and did not appeal to older students). The most motivating activity was the "Monica’s Automata" because it involved several steps (cutting, pasting, and strategy creation). Since the activity also had more than one correct answer, the students enjoyed comparing and discussing with their colleagues the various possibilities.
- During the "Repetition Drawing" activity, more than half of the students had difficulty understanding the position and direction (perspective) of the arrow. The activity had to be explained several times and in different ways until the students could understand the behavior of the arrow and the commands needed to control it as expected. The most appropriate solution was to stand and walk/rotate according to the commands the students wrote on paper.
- Very large deficiencies in basic mathematical concepts and even literacy were identified. There were cases of students who were unable to read or understand an activity, thus not being able to complete the tasks in full. There were also several cases of serious Portuguese errors and sentences lacking concordance at both grade levels.

- During the post-test application, the researchers noticed that intervention groups took longer, on average, to complete the test. According to the students, they took longer because, as they had worked on the concepts previously in class, they paid more attention to the test questions.

5.2 Quantitative Performance of Computational Thinking Tests

The score of the CT tests was calculated according to the amount of questions answered correctly, remembering that the test is composed of 28 questions. As explained in section 4.4, the test was applied in both the Experimental and Control Groups.

The results obtained with this instrument were submitted to statistical procedures in order to test the null hypothesis, in other words, to evaluate if there was a difference between the results of the pre-test and post-test and if this difference was significant. The **Table 3** shows the number of participants (N), results of the means, standard deviations and Median of the two classes and the pre- and post-test scores performance alteration.

To verify if the mean improvement was statistically significant or occurred at random, the T-Test for paired samples was used, considering a 95% confidence interval. In the Experimental Group, the result found were P (T <= t) two-tailed = 0.013 and 0.020. Considering that these values are less than 0.05, there is a significant difference between the averages from a statistical point of view. The same data treatment was used in the Control Group and P (T <= t) two-tailed = 0.484 and 0.916, in other words, in these groups there were no improvements in performance. Taking into account the p-value (0.484) obtained in the 5th year of the School A, higher than the nominal value in the control group, it was not evidenced the hypothesis rejection of equality of the group for the Test, with that, without any performance improvement. The opposite occurred in the experimental group, where a p-value lower than the nominal value (0.013) was observed, indicating a significant difference between the pre- and post-tests, evidencing the rejection of the null hypothesis. For the 6th year of School B, a p-value (0.916) was higher than the nominal level in the control group, and the rejection of the hypothesis of equality of the group for the test was not evidenced, proving that there was no significant performance improvement. The opposite occurred in the experimental group, where a p-value lower than the nominal value (0.020) was found, indicating a significant difference between the pre- and post-tests, as well as evidence of rejection of the null hypothesis. This fact reinforces the theory that the improvement in Computational Thinking in the Experimental Group was in fact motivated by the unplugged activities. Thus, the data presented shows that there was an improvement in the student performance in the Experimental Group and stagnation in the Control group. The results are also in **Figure 5** where the data was plotted using error bars with 95% confidence intervals for the means of the CT Test.

Table 3. Statistical Results

		N	Test	Mean	SD	Median (IQ)	p-value	Performance Alteration
School A (5th grade)	Control	10	Pre-test	9.30	3.59	8 (6,75-11,25)	0.484	+1.00 (+10.75%)
			Post-test	10.30	3.09	11 (7,5-12,25)		

	Experimenta l	15	Pre-test	9.93	3.26	9 (8-11)	0.013	+1.80 (+18.12%)
			Post-test	11.73	4.03	10 (9-14)		
School B (6th grade)	Control	19	Pre-test	9.68	4.07	10 (7-11)	0.916	-0.11 (-1.09%)
			Post-test	9.58	3.45	10 (7-3)		
	Experimenta l	19	Pre-test	9.16	3.10	9 (7-11)	0.020	+1.89 (+20.69%)
			Post-test	11.05	4.54	10 (7-15)		

Author

We consider that these finding have two additional implications: 1) demonstrates that the CT Test is a valid and sensitive tool to measure improvements on CT skills of the participants not only after practicing with “plugged” activities, but also in “unplugged” activities; 2) The results give evidence that unplugged activities can be used as part of the school regular classes, enabling the guidance for future curriculum decisions of teachers and policy makers. Overall, the results allow us to answer the research question through our quasi-experimental approach.

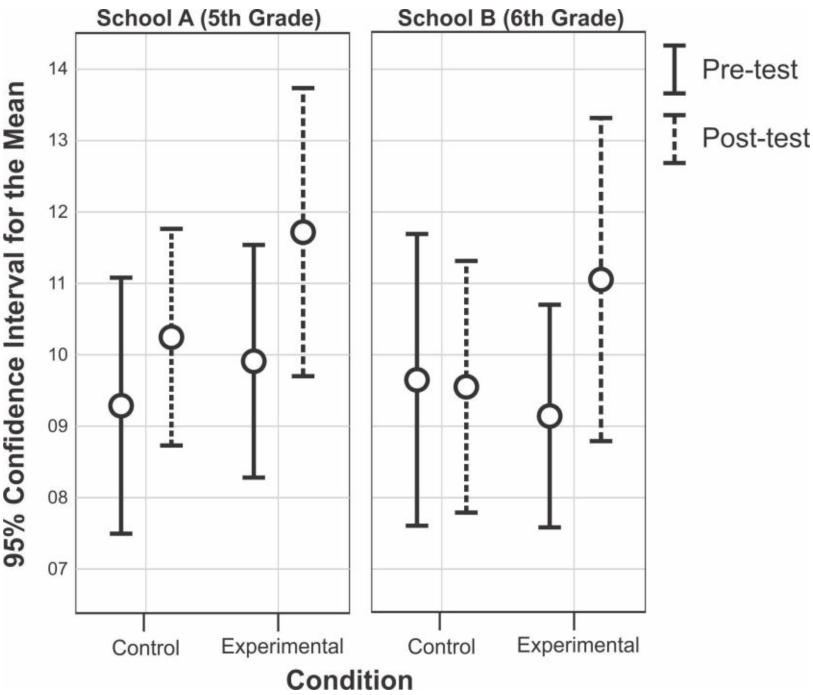


Figure 5. Error bars with the 95% confidence intervals for the means of the test score, split by school and grade, for both groups-conditions, and in pre-test and post-test times.

6. Limitations and Threats to Validity

Some limitations and threats to validity of our research can be pointed out:

- The CT Test has some limitations, since it is heavily focused on computational concepts, only partially covers computational practices, and ignores computational perspectives (Brennan & Resnick, 2012). Moreover, the CT Test has a (deliberately) reductionist conception of CT, which puts over-emphasis on path-finding algorithms;

- Most of the unplugged activities carried out along the research might be considered as excessively and artificially aligned with the items of the CT Test. Therefore, if a different set of unplugged activities had been used, we would probably have obtained different results;
- The small size of the sample should be noted (N < 120), in order to consider the limited generalization power of our results.

7. Conclusions and Future Efforts

This paper presented a brief introduction to Computational Thinking, its history and the quasi-experimental research carried out in two Brazilian schools to attempt to test the efficiency of unplugged activities to allow countries with economic and social vulnerabilities to teach Computer Science during their formal education without the need of a machine or another equipment.

The study was carried out in two public schools that are located in unprivileged areas. All students who participated in the research were divided into two groups (control and experimental). The experimental group, after taking the pre-test, participated in classes about Computer Science without the use of machines (unplugged), while the control group didn't have the lessons. After the post-test, it was possible to compare the results and identify that there was an improvement in the student performance in the Experimental Group and no alteration in the Control group' score. Consequently, these findings provide empirical evidence about the effectiveness of the unplugged approach to develop CT skills and contribute to reaffirm that Computational Thinking as a cognitive variable which mainly consists in problem-solving ability or process whose development doesn't necessarily is connected only to computer programming (Wing, 2006).

The review of studies that provide evidence on the utility of unplugged computing to develop CT skills has identified the importance of deepening empirical research, especially when it comes to its use in primary schools. Consequently, with the research reported in this article, we sought to add relevance to the list of evidence.

Based on the experience gained during the process, it was possible to conclude that the children were very enthusiastic and motivated during the CT classes. Teachers also expressed great satisfaction with the opportunity given to their students. In general, schools were remarkably receptive to CT classes, and did not create any barriers to the project.

As can be seen in **Table 3**, there was a considerable improvement in students' scores with highly significant statistical results in the experimental group after 10 hours of Unplugged Computational Thinking, unlike the Control group which maintained a post-test score very close to their pre-test score. These results proved the efficacy of the unplugged approach and met the main goal of this study. The positive data could also be understood as a very small variation in performance improvement, but it is essential to consider that there were only 10 hours of classroom activity.

The unplugged approach has its limitations, and therefore, its use in the introduction of Computational Thinking is recommended. The unplugged approach could be an alternative for countries with social and economic difficulties, taking into consideration that a just a few basic office materials were needed (paper, pencil, eraser, scissors, etc.) and very little cost to print or copy the materials. This will allow the students to access the Computer Science fundamentals, independently of its future professional career, to have greater opportunities and development.

For future studies, more detailed research is necessary to identify the point of convergence for the plugged and unplugged approach or when the unplugged approach loses its effectiveness and it becomes necessary to migrate to the machines. In addition, some more questions can be inserted in the research, being:

- Will the unplugged material have the same effect on students in private schools?
- From what age or school year should one introduce Computational Thinking?
- How can Computational Thinking be developed and measured in early school years?
- At what point does the unplugged approach begin to lose its intended effect and the use of machines is recommended?

8. References

[1] BBC Learning, B. "What is computational thinking?", 2015. Retrieved from http://www.bbc.co.uk/education/guides/zp92mp3/revision. Access: 01/04/2019.

[2] Bell, T., Witten, I. H., & Fellows, M. Computer Science Unplugged... - Off-line activities and games for all ages (draft) (1st ed.), 1997, pp. 1- 240.

[3] Bell, T., Witten, I. H., & Fellows, M. "Computer Science Unplugged", 2015, pp. 1-243. Retrieved from http://csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf. Access: 01/04/2019.

[4] Mec. Base Nacional Comum Curricular (BNCC), 2018. Retrieved from http://basenacionalcomum.mec.gov.br/wp-content/uploads/2018/12/BNCC_19dez2018_site.pdf. Access: 5/2/2019.

[5] Brackmann, C., Barone, D., Casali, A., Boucinha, R., & Munoz-Hernandez, S. "Computational thinking: Panorama of the Americas". IEEE, 2016, pp.1-6. https://doi.org/10.1109/SIIE.2016.7751839.

[6] Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. "Development of Computational Thinking Skills through Unplugged Activities in Primary School", ACM Press, 2017, pp. 65-72. https://doi.org/10.1145/3137065.3137069.

[7] Brennan, K., & Resnick, M. "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, 2012, pp. 1–25. Retrieved from http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf. Access: 5/2/2019.

[8] Campos, G., Cavalheiro, S., Foss, L., Pernas, A., Piana, C., Aguiar, M., … Reiser, R. "Organização de Informações via Pensamento Computacional: Relato de Atividade Aplicada no Ensino Fundamental". 20º.

Workshop de Informática Na Escola (WIE 2014), 2014, pp. 390-399. http://dx.doi.org/10.5753/cbie.wie.2014.390.

[9] CODE.ORG. Instructor Handbook - Code Studio Lesson Plans for Courses One, Two, and Three. CODE.ORG, 2015. Retrieved from https://code.org/curriculum/docs/k-5/complete.pdf. Access: 5/2/2019.

[10] Curzon, P. "cs4fn and computational thinking unplugged", ACM Press, 2013, pp. 47–50. https://doi.org/10.1145/2532748.2611263.

[11] Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. "Introducing teachers to computational thinking using unplugged storytelling", Presented at the WiPSCE, ACM Press, 2014, pp. 89–92. https://doi.org/10.1145/2670757.2670767.

[12] Engelberg, M., & Thinkfun. Code Master Programming Logic Game. Thinkfun. 2015. Retrieved from http://www.thinkfun.com/products/code-master/. Access: 5/2/2019.

[13] Feaster, Y., Segars, L., Wahba, S. K., & Hallstrom, J. O. "Teaching CS unplugged in the high school (with limited success)" , Proceedings of the 16th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE, Darmstadt, Germany, June 27-29, 2011, pp. 248-253. https://doi.org/10.1145/1999747.1999817.

[14] Franklin, D., Skifstad, G., Rolock, R., Mehrotra, I., Ding, V., Hansen, A., … Harlow, D. "Using Upper-Elementary Student Performance to Understand Conceptual Sequencing in a Blocks-based Curriculum", ACM Press, pp. 2017, pp. 231–236. https://doi.org/10.1145/3017680.3017760.

[15] Grover, S., & Pea, R. "Computational Thinking in K-12: A Review of the State of the Field", Educational Researcher, *42,* (1), 2013, pp. 38–43. https://doi.org/10.3102/0013189X12463051.

[16] Grover, Shuchi, & Basu, S. "Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic", ACM Press, 2017, pp. 267–272. https://doi.org/10.1145/3017680.3017723.

[17] Hammer, O., Harper, D. A. T., & Ryan, P. D. "Past: Paleontological Statistics Software Package for Education and Data Analysis", Palaeontologia Electronica, 4(1), 2001, pp. 1-9.

[18] Kalelioglu, F., Gülbahar, Y., & Kukul, V. "A Framework for Computational Thinking Based on a Systematic Research Review", Baltic Journal of Modern Computing, 4(3), 2016, pp. 583–596.

[19] Kologeski, A., Silva, C., Barbosa, D., Mattos, R., & Miorelli, S. "Desenvolvendo o Raciocínio Lógico e o Pensamento Computacional: Experiências no Contexto do Projeto Logicando", Revista Novas Tecnologias Na Educação (RENOTE), 14(2), 2016, pp. 1-10. https://doi.org/10.22456/1679-1916.70686.

[20] Lambert, L., & Guiffre, H. Computer science outreach in an elementary school. Journal of Computing Sciences in Colleges, 24(3), 2009, pp. 118–124.

[21] Liukas, L. Hello Ruby: adventures in coding. Feiwel & Friends, 2015, pp. 112.

[22] Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., & Kuno, Y. "A CS Unplugged Design Pattern", New York, NY: ACM, 2009, pp. 231–235. Retrieved from http://dx.doi.org/10.1145/1508865.

[23] Rodriguez, B., Stephen, K., Rader, C., & Camp, T. "Assessing Computational Thinking in CS Unplugged Activities", In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, Washington, USA: ACM, 2017, pp. 501–506. https://doi.org/10.1145/3017680.3017779.

[24] Román-Gonzáles, M., Moreno-León, J., & Robles, G. "Complementary Tools for Computational Thinking Assessment", Presented at the International Conference on Computational Thinking Education 2017 (CTE 2017), Hong Kong: In press, 2017.

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

[26] Román-González, M. "Computational Thinking Test: Design Guidelines and Content Validation", In Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015), Barcelona, Spain: IATED, 2015, pp. 2436–2444 https://doi.org/10.13140/RG.2.1.4203.4329.

[27] Román-González, M. "Codigoalfabetización y Pensamiento Computacional en Educación Primaria y Secundaria: Validación de un Instrumento y Evaluación de Programas". Tesis Doctoral. Universidad Nacional de Educación a Distancia, Madrid, Spain, 2016, pp. 1-720.

[28] Román-González, M., Pérez, J. C., & Carmen Jiménez-Fernández. Test de Pensamiento Computacional: diseño y psicometría general, Octubre 14-16, 2015, Madrid, ESPAÑA III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015) , 2015. https://doi.org/10.13140/RG.2.1.3056.5521.

[29] Scaico, P., Mychelline, H., Cunha, F., & Alencar, Y. "Um Relato de Experiências de Estagiários da

Licenciatura em Computação com o Ensino de Computação para Crianças", Revista Novas Tecnologias Na Educação, (RENOTE), 10(3), 2012, pp. 1-10. https://doi.org/10.22456/1679-1916.36377.

[30] Taub, R., Ben-Ari, M., & Armoni, M. "The effect of CS unplugged on middle-school students' views of CS", ACM Press, 2009, pp. 99. https://doi.org/10.1145/1562877.1562912.

[31] UNESCO. Institute for Statistics. Information and Communication Technology (ICT) in Education in Asia: A comparative analysis of ICT integration and e-readiness in schools across Asia. Information paper; 22. UNESCO Institute for Statistics, 2014. https://doi.org/10.15220/978-92-9189-148-1-en.

[32] UNESCO. Institute for Statistics. Information and Communication Technology (ICT) in Education in sub-Saharan Africa: A comparative analysis of basic e-readiness in schools. Information Paper No. *25*. UNESCO Institute for Statistics, 2015. https://doi.org/10.15220/978-92-9189-178-8-en.

[33] Unnikrishnan, R., Amrita, N., Muir, A., & Rao, B. "Of Elephants and Nested Loops: How to Introduce Computing to Youth in Rural India", ACM Press, 2016, pp. 137–146. https://doi.org/10.1145/2930674.2930678.

[34] Valente, J. A. "Integração do Pensamento Computacional no Currículo da Educação Básica: Diferentes Estratégias Usadas e Questões de Formação de Professores e Avaliação do Aluno", Revista e-Curriculum, v. 14, n. 3, 2016, pp. 864-897.

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