# Understanding CTProgER's Impact on High School Students' Programming Learning

Isabelle M. L. Souza*, Wilkerson L. Andrade*, Lívia M. R. Sampaio*
*Federal University of Campina Grande, Campina Grande, PB, Brazil
isabellemaria@copin.ufcg.edu.br / isamlsouza@gmail.com, {wilkerson, livia}@computacao.ufcg.edu.br

***Abstract*—This research-to-practice full paper describes a study that employs the CTProgER educational process [37] within High School environments to enhance programming skills. Computational Thinking (CT) is recognized as crucial for developing these skills, especially in high school programming education. Teaching programming presents challenges, necessitating improved educational processes and guidelines. Based on the Anthropological Theory of Didactics, CTProgER operates on the principle that human-created entities are interconnected with individuals, forming relationships denoted as R(X,O). This research investigates the efficacy of CTProgER through a rigorous intervention study conducted in a Brazilian Technical and Vocational High School (TV High School). The results reveal a significant performance disparity between the experimental and control groups, with the experimental group showing notable improvements. These findings highlight the potential of CTProgER to guide and enhance programming instruction within high school curricula, offering valuable insights for validating educational processes.**

## I. Introduction

Practical problem-solving skills are vital for personal and academic development, particularly in an increasingly digital world. Computational Thinking (CT) has emerged as an approach to enhance these skills, emphasizing logical thinking and algorithmic problem-solving. While programming education is recognized as a means to foster CT in High School students, the challenge lies in implementing an effective educational process.

The CTProgER educational process, as proposed by Souza *et al.* [37], addresses this challenge by integrating CT principles into programming instruction in High Schools. Grounded in the Anthropological Theory of Didactics (Chevallard, 1989), CTProgER frames programming education within a structured educational process to facilitate meaningful learning experiences for students. This process is guided by the understanding that every human-created entity, known as an Object (O), is associated with at least one individual, forming a connection expressed as R(X,O). The CTProgER comprises six didactic moments: Problematic, Analysis, Discovery, Implementation, Validation, and Assessment.

This research-intervention study aims to evaluate the effectiveness of the CTProgER educational process in improving Technical and Vocational High School (TV High School) students' programming skills, observing the relation between students (X) and programming concepts (O) in TV High School to observe the impact of CTProgER on learning programming. Specifically, it aims to answer the research question: (RQ) What is the effectiveness of the CTProgER educational process in improving TV High School students' programming skills?

The study comprises two main steps: (1) Conducting a research intervention in TV High School to implement CTProgER and (2) Analyzing the effectiveness of CTProgER on students' programming skills. The sample includes 93 students divided into experimental and control groups based on their exposure to robotics. The experimental group receives programming instruction using the CTProgER educational process, while the control group follows conventional programming teaching methods without CT integration. Pre-test and post-test assessments are conducted using pseudocode questions to evaluate programming performance.

Statistical analysis, including the paired Mann-Whitney (U) test and Cohen's effect size (d), is employed to assess the impact of CTProgER on programming learning. Results indicate a significant improvement in post-test scores for the experimental group compared to the control group across all evaluated lessons, highlighting the efficacy of CTProgER in enhancing programming proficiency.

Although statistical significance is not consistently achieved, CTProgER positively impacts programming learning, suggesting its effectiveness in TV High School settings. Future research will delve deeper into the research intervention's effects on individual groups, providing further insights into the efficacy of CTProgER.

This paper is organized as follows: Section II, describes the concepts of CT, Educational Robotics, Anthropological Theory of Didactics, and the CTProER educational process. Section III presents related works. Section IV details the technique used in our research. In Section V, presents the results, discussions, and threats to validity. Finally, Section VI presents the conclusions and future works.

## II. Background

This section presents the fundamental concepts that are the basis of this study.

### A. Computational Thinking

The concept of CT originates from Seymour Papert's 1980 constructionist learning studies [31], [32], which focused on developing thinking skills through computer science principles. Its importance increased in 2006 when Jeannette Wing

highlighted it as a fundamental problem-solving skill in her paper [22]. Wing described CT as a universally applicable mindset and skill for both computer scientists and learners [40].

Today, CT is a key topic in Computer Science education research, though its definition continues to evolve. It is widely recognized for its role in problem-solving processes [22] and is considered vital for future adaptability, supporting its inclusion in early education [49]. Many researchers advocate integrating CT into curricula across all educational levels, from kindergarten to university [27], [43], [44].

In 2011, the International Society for Technology in Education, with the National Science Foundation and the Computer Science Teachers Association, introduced the Model Curriculum for K-12 Computer Science [41]. At the High School level, CT typically involves computing, programming, and technology. Programming education has gained global traction for developing CT skills among High School students, extending beyond computer and engineering courses [48].

### *B. Educational Robotics*

Educational Robotics (ER) refers to a learning environment that integrates various materials or kits containing motors and sensors controllable by computers and software, allowing the construction of models that can be programmed [18]. It can also be defined as creating mechanisms that computers can control for educational purposes [42]. Research in ER explores how robot interactions can facilitate and enhance learning across different age groups, from young children to adults [3], [50]. ER strategies typically fall into two categories: 1) Learning about Robots, which focuses on education in robotics or robotics as a science, and 2) Learning with Robots, which emphasizes using robotics for educational purposes [15]. ER programs can be integrated into the curriculum or offered as extracurricular activities, depending on the educational objectives [28].

ER serves various purposes in basic education, including teaching programming, introducing robotics as a scientific field, interdisciplinary science education, computer programming development, participation in competitions, and Olympic events [36]. Whether delivered as part of the curriculum or as extracurricular activities, educational practices with ER often emphasize problem-solving situations, aiming to teach curriculum science and programming logic [36]. This approach typically involves hands-on learning, where students instruct robots to perform specific tasks. Regardless of the particular goals and learning strategies employed, there is a common emphasis on planning, designing, or implementing algorithms to control the behavior of robots, which directly contributes to the development of CT skills [7], [33].

### *C. Anthropological Theory of Didactics*

The Anthropological Theory of Didactic, developed by Yves Chevallard in the field of mathematics education, focuses on studying the processes of didactic transposition [5], [8], [10], [9], [11]. According to this theory, mathematics, in its various dimensions, is fundamentally linked to human activities. Therefore, it can be applied in diverse contexts that facilitate the execution of tasks and the acquisition of necessary knowledge. In the Anthropological Theory of Didactic, the object of study is considered a human activity resulting from constructed knowledge, which can be transmitted, taught, or learned through different technical tasks and technologies [47].

The Anthropological Theory of Didactic comprises structural elements, such as praxeologies, and functional elements, including didactic moments. The realization of didactic moments involves the constitution of praxeologies and their associated elements [T, $\tau$, $\theta$, $\Theta$], ultimately leading to the establishment of a relationship between the Personal (X) (e.g., students and teachers) and the Object (O), denoted as R(X,O). This successful relationship is indicative of learning [14].

### *D. The CTProgER Educational Process*

The CTProgER was initially presented and validated from the perspective of experts in Souza *et al.* [37]. Subsequently, it was validated from the student's perspective by observing the development of CT through an intervention study that applied CTProgER and assessed its impact using the Román-Gonzalez CT test [20] in [38].

The CTProgER, designed for teaching programming through ER with a focus on CT skills, draws upon methodologies established in commercial ER, such as LEGO® Education, and integrates concepts proposed by the Anthropological Theory of the Didactic along with findings from the study [16], [17], [21], [39], [36]. Within this framework, an educational Institution [I] serves as the setting for teaching practices. Here, an Object (O) is provided as input, and the goal is to foster a relation R(X,O) as output (see Figure 1).

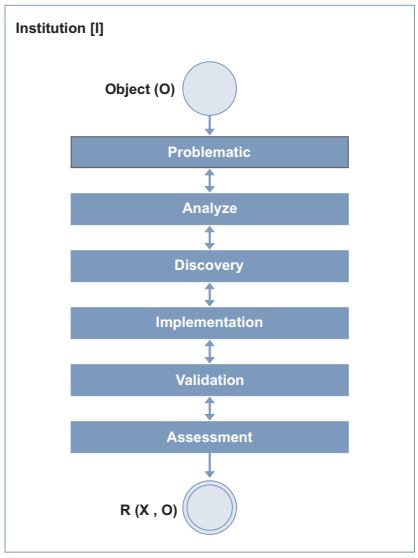


Fig. 1. CTProgER Educational Process by Souza *et al.* [37]

The **input** identifies programming concepts teachers aim to impart, facilitating students' development of specific knowledge. This development encompasses conceptual, procedural, and attitudinal dimensions related to facts, principles, practical skills, and behavioral aspects. The **output** represents the relation between the individual (student) and the object (CT skills cultivated through programming concepts). In the Anthropological Theory of the Didactic, any task arising from human activity is regarded as an object (O), enabling individuals (X) to establish a relationship with it—an action represented by R(X,O). Thus, when R(X,O) $\neq$ 0 indicates a positive relationship, it suggests learning the concepts provided as input.

To support this proposition, Souza *et al.* proposes six didactic moments based on the Anthropological Theory of Didactics and the lessons learned from previous research:

1) **Problematic:** Involves presenting a problem/situation that favors the teaching-learning of concepts given as input. The presentation of the problem must be contextualized and based on natural elements to simplify the adaptation in real environments. At this didactic moment, it is necessary to establish a relation between the problem and the student; for this, the teacher must explore methods and resources that they find appropriate.
2) **Analysis:** Involves a closer connection between the elements presented in the problem/situation and the student's prior knowledge in a practical context. At this didactic moment, the teacher's responsibility is to instigate the student's active participation to promote strategies to solve the proposed problem. The teacher needs to play the mediator, and the student must be the protagonist of their proposed solutions.
3) **Discovery:** Involves the understanding of the concepts studied given as input. At this didactic moment, the teacher must lead the students to connect the new knowledge to the previous ones to clarify the necessary procedures to implement the strategies conceived in the "Analysis" didactic moment. The teacher must be a mediator, facilitator, and articulator of knowledge, provoking students to learn from their proposals.
4) **Implementation:** Involves the conception of the codification strategy. At this didactic moment, the students must implement the concepts defined as input using a programming language to solve the proposed problem. The students must take the lead in their practices, and the teacher should facilitate the application of the new concepts studied.
5) **Validation:** Involves the execution and analysis of the code built in the "Implementation" didactic moment using a robotic assembly provided. At this didactic moment, the student must analyze the code execution, correct possible errors, and validate the proposed strategy. The teacher must mediate on the results achieved by the student to learn the concept given as input.
6) **Assessment:** Involves the application of instruments to observe the student's learning of the concepts presented as input and the CT skills development"

### *E. The CTProgER Educational Lessons*

The CTProgER lessons have been developed by Souza *et al.* [37] to correspond to each 1.5-hour programming class following CTProgER guidelines. Three lessons were available, each lasting 1.5 hours, focusing on teaching programming with ER to enhance CT skills. Students are grouped into teams of five, and a detailed lesson plan is provided in Table I.

TABLE I
LESSONS PLAN

| Lesson | Duration | Contents |
|---|---|---|
| 1 - The Cleaner Robot | 1 Week 1h30min | Concept of algorithms; Problems in the construction and execution of algorithms; Construction of algorithms in natural language. |
| 2 - The Accountant Robot | 1 Week 1h30min | Data input and output; Data types; Variables; Constants; Touch sensor. |
| 3 - The Driver Robot | 1 Week 1h30min | Data input; Condition Control Structure; Ultrasonic sensor. |

The instances were developed for teaching-learning any programming language that could be worked with robotics. Consequently, they do not have code in a specific programming language, only the concepts involved. The teacher needs to perform the programming syntax during the lesson according to the guidelines from the teacher's material. Figure 2 illustrates the format and design of the created instances.


Fig. 2. The Cleaning Robot Lesson Illustration

The CTProgER Educational Lessons are available by Souza *et al.* [37] online[1].

## III. RELATED WORK

Over the past decade, computer programming has been increasingly integrated into Education, either as part of the

[1]https://github.com/isabellelimasouza/CTProgER.git

curriculum or as an extracurricular activity, aiming to foster CT skills in students [23]. CT, which extends beyond coding, forms the foundation for effective programming teaching and learning [19]. Numerous educational proposals have emerged, yet only some have rigorously validated the effectiveness of programming learning outcomes.

Nouri *et al.* [30] conducted interviews with experienced programming teachers to identify the skills students develop through programming. Their findings align with the CT framework by Brennan and Resnick [6], encompassing computational concepts, practices, and perspectives. Similarly, Alegre *et al.* [2] designed an elective course to teach programming and mathematical problem-solving skills to High School students using Codeworld, demonstrating improvements in language proficiency and programming fundamentals.

High School programming education often centers on block-based languages and visual tools [19], [24], [25], [26], [34], [46]. Malizia *et al.* [25] introduced block-oriented programming to develop CT skills, emphasizing the manipulation of tangible objects. Similarly, Noh and Lee [29] found that teaching programming through robotics enhanced CT and creativity among elementary students.

Various methodological proposals aim to integrate CT into Education. Schiavani [35] employed ER in physics education, aligning with LEGO® Education's methodology. This approach offers insights into structuring ER activities for enhanced contextualization. Chevalier *et al.* [7] developed a model to identify CT concepts within ER activities. In contrast, this thesis quantitatively assesses High School students' CT skill development, indirectly applying the Anthropological Theory of Didactics. Azman and Mohamed [4] proposed a framework integrating CT into the innovation process, similar to this thesis's focus on CT development through ER.

This work proposes to observe the CTProgER validates based on teaching programming with ER, rooted in the Anthropological Theory of Didactics [37]. This work explores the relation between students (X) and programming concepts (O) in High School to observe the impact of CTProgER on learning programming and to understand more as an educational process validated can improve students to learn programming.

## IV. Methodology

This study employs a research-intervention method that aims to explore the relation between students (X) and programming concepts (O) in TV High School to observe the impact of CTProgER [37], [38] on learning programming within the Brazilian TV High School context.

### A. Research Design

This research-intervention study aims to explore the relation between students (X) and programming concepts (O) in TV High School to observe the impact of CTProgER on learning programming. The development of this study was divided into two steps: (1) application of the research intervention on TV High School students submitted to CTProgER and (2) analysis of its effectiveness on TV High School students. The following research questions and hypotheses guided the development of this research-intervention study:

**(RQ)** What is the effectiveness of the CTProgER educational process in improving High School student's programming skills?

- *H.0:* There is no significant evidence that the CTProgER educational process improves High School students' programming skills.

- *H.1:* The CTProgER educational process has a significant impact on improving High School students' programming skills.

Table II outlines the research-intervention study design conducted by the primary author. The design includes two student groups, namely the control and experimental groups, each with similar profiles. Both groups engaged with identical programming and robotics content during the research-intervention. However, the experimental group followed the CTProgER educational process guidelines for programming teaching with educational robotics, while the control group underwent traditional programming learning. In the control group, the teacher presented the content expository, followed by students attempting to program the robots using the learned content. The design incorporates several variables. **High School (HS)** is an independent variable representing the actions undertaken during the academic year. **Educational Robotics (ER)** signifies using robotics as a teaching tool. **Programming Teaching (PT)** refers to the approach to programming instruction. **CTProgER** encompasses the guidelines and lessons proposed in [37], [38]. The dependent variable, **Learning Programming Skills (LP)**, measures students' performance in programming language tests, which include pseudocode questions solved by applying the concepts covered in each lesson.

TABLE II
Research-intervention Design

| Groups | Independent Variable | Dependent Variable |
|---|---|---|
| Experimental | HS + ER + PT + CTProgER | LP1 |
| Control | HS + ER + PT | LP2 |

An example of questions in the pseudocode of programming language tests are illustrated in Listing 1. The formulation of pseudocode questions involved collaboration with external researchers and CT experts, ensuring their meticulous construction and review. These questions were intentionally crafted to assess programming logic skills without necessitating proficiency in any specific programming language.

Listing 1. Pseudo-code question example

```
What is the result of numAtual?

numAtual = 10
counter = 5
```

```
BEGIN
    FOR counter = 5 to 0 REPEAT
        IF (counter / 2 ) = 0 THEN
            numAtual = numAtual + 20
        ELSE
            numAtual = numAtual + 10
        END
    END
    PRINT ("numAtual = " + numAtual)
END
```

### B. Sample and Data Collection

This study involved a sample of 93 students from the Computing TV High School in Brazil public school. The students were divided into experimental and control groups based on their exposure to the educational process in 2022. The control group comprised 40.86% (38) students, while the experimental group comprised 59.14% (55). Both groups engaged with identical programming and robotics content during the research intervention. However, the experimental group followed the CTProgER educational process guidelines for programming teaching with educational robotics, whereas the control group underwent traditional programming learning. In the control group, the teacher presented the content expository, followed by students attempting to program the educational robots using the learned content.

To determine whether our sample of students was representative, we applied the sample size calculation in Equation 1, where $n$ = calculated sample, $N$ = population size, $Z$ = standardized normal variable associated with the confidence level, $p$ = true probability of the event, and $e$ = sample error.

$$n = \frac{N \cdot Z^2 \cdot p \cdot (1-p)}{Z^2 \cdot p \cdot (1-p) + e^2 \cdot (N-1)} \tag{1}$$

Considering the student population size of 101 students, the sample size calculation results indicate that the sample is represented with an error margin of 2.87% at a 95% confidence level. This means that there is a high probability that the actual sampling error is less than the accepted sampling error.

Data collection was conducted between September, October, and November 2022, encompassing quantitative data on students' cognitive development both in programming language before and after the three programming lessons that composed the research intervention.

The students in the experimental and control groups were selected from various classrooms within the target school using random sampling methods. The sample consisted of 93 students, with 42 (45.16%) males and 51 (54.84%) females.

### C. Data Collection and Analysis

The students' data were collected by administering the test, which comprises pseudocode questions that can be logically solved by applying the concepts covered in each lesson (see Subsection II-E). The control and experimental groups participated in the test and were briefed in advance about its structure, duration, and procedures for resolving it. During the correction process, each question was evaluated, with one point assigned for each correct answer. The results were recorded in a spreadsheet.

Statistical data analysis was performed using the R programming language. This involved graphical analysis procedures and hypothesis testing to assess whether the data parameterization was appropriate and to guide the statistical tests. Given the non-parametric nature of the data, the U Test (both paired and unpaired) was selected to evaluate whether there was a significant difference in the performance of both groups (experimental and control) in programming language based on the test results. Cohen's effect size index was utilized to calculate and analyze the effect of the CTProgER educational process on students' programming language performance [13]. A confidence level of 95%, a statistical significance of $\alpha$ = 0.05, and the U Test were considered for the tests, given the non-parametric nature of the data and adherence to the specific statistical assumptions of the test. The application of these statistical tests aims to understand how the relation between between the Personal (X) and the Object (O), denoted as R(X,O), evolved after each research-intervention class.

### D. Ethics Code

This study adhered to Brazilian National Health Council Resolution 196/96 and was registered on the "Plataforma Brasil" [2] under CAAE 90723918.5.0000.5182. Before beginning, participants received detailed explanations of the procedures and signed consent forms. All data were handled anonymously to ensure confidentiality and privacy.

## V. Results

o address the research inquiry **(RQ)**, we analyzed the mean, median, standard deviation (*SD*), mean difference, and median difference between the experimental and control groups regarding programming performance, as measured by the programming test in each lesson. Three lessons were administered during the research intervention, with tests conducted before and after each lesson.

The results presented in Table III indicate that, in all lessons, the control group demonstrated superior performance in the pre-test of programming compared to the experimental group. This difference is evidenced by the negative mean differences, which represent the superiority of the control group over the experimental one. Specifically, in Lesson 1, the mean difference was 10.64%; in Lesson 2, it was 32.02%; and in Lesson 3, it was 47.52%. It is important to note that the negative values in the table result from the calculation's positioning, which places the experimental group first. Since the control group's performance is higher, the difference is expressed as a negative value, thus demonstrating the control group's superiority over the experimental one.

[2]Plataforma Brasil: http://plataformabrasil.saude.gov.br/login.jsf

The same trend is observed in the median difference only in the pre-test of Lesson 2, where the control group exhibits a median difference of 1% higher than the experimental group. These data demonstrate that the control group is somewhat superior to the experimental group in the pre-tests, i.e., before the research-intervention lessons.

The mean difference in the post-test for all lessons showed that the experimental group outperformed the control group, with a difference of 20.31% in Lesson 1, 6.56% in Lesson 2, and 24.72% in Lesson 3. However, the median difference remained unchanged. Nevertheless, given that the mean difference consistently favored the experimental group, we understand that the research intervention based on CTProgER may positively impact students' performance in programming learning.

TABLE III
GROUPS PERFORMANCE FROM PROGRAMMING LANGUAGE TEST

| Lesson | Test | Experimental | | | Control | | | Mean Difference | Median Difference |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean | Median | SD | Mean | Median | SD | | |
| 1 | Pre-test | 0.89 | 1.00 | 0.87 | 1.00 | 1.00 | 0.88 | -10.64 | 0.00 |
| | Pos-test | 1.72 | 2.00 | 0.54 | 1.43 | 2.00 | 0.65 | 20.31 | 0.00 |
| 2 | Pre-test | 0.42 | 0.00 | 0.58 | 0.61 | 1.00 | 0.56 | -32.02 | -1.00 |
| | Pos-test | 0.69 | 1.00 | 0.75 | 0.65 | 1.00 | 0.55 | 6.56 | 0.00 |
| 3 | Pre-test | 0.24 | 0.00 | 0.49 | 0.47 | 0.00 | 0.63 | -47.52 | 0.00 |
| | Pos-test | 1.12 | 1.00 | 0.67 | 0.90 | 1.00 | 0.71 | 24.72 | 0.00 |

To check the *H.0* null hypothesis (There is no significant evidence that the CTProgER educational process improves High School students' programming skills.), we performed a non-parametric hypothesis test (*Mann-Whitney U*) (paired and unpaired) applied to the two independent samples. Initially, we observed whether there was a significant difference between the experimental and control groups in programming after (post-test) each lesson. We applied the unpaired test (*Mann-Whitney U*) for this. The *p-value* obtained in this test demonstrated a significant difference between both groups in Lesson 1 and 3 (see Table IV).

TABLE IV
HYPOTHESIS UNPAIRED TEST FROM PROGRAMMING LANGUAGE TEST

| Lesson | U Test | *p-value* | Confidence Level | |
|---|---|---|---|---|
| | | | Min | Max |
| 1 | 854 | <0.05 | -0.00006243391 | 0.9999828 |
| 2 | 644 | 0.3769 | -0.00006510172 | 0.00004438377 |
| 3 | 794 | <0.05 | 0.00002163388 | 0.9999935 |

While the results from Lesson 2 did not demonstrate significant differences, the outcomes observed in Lessons 1 and 3 allow for the observation of significant distinctions between the experimental and control groups. This data indicates that the educational process of teaching programming with educational robotics, augmented by the CTProgER, yielded discernible effects on students' programming skills in TV High School, as depicted in Tables IV. Thus, we can reject the null hypothesis with a confidence level of 95% and accept the alternative hypothesis *H.1*: The CTProgER educational process has a significant impact on improving High School students' programming skills.

However, further studies and complementary analyses are warranted to comprehensively understand the factors contributing to the need for significant differences in Lesson 2. These additional investigations can provide insights into the specific dynamics during this instructional session, shedding light on potential areas for improvement or modification in future implementations. Therefore, to verify if there is a significant change between the preliminary and final performance on the programming test of the experimental and control groups, we applied the paired test (*Mann-Whitney U*). This test distinctly identifies the changes in the behavior of the groups, i.e., it verifies if there is a change between the pre-test and the post-test of each group individually. Then, we can see the evolution of each group statistically. The *p-value* obtained in this test demonstrated a significant difference in the experimental and control group's preliminary and final programming performance in Lessons 1 and 2 (see Table V).

TABLE V
HYPOTHESIS PAIRED TEST FROM PROGRAMMING LANGUAGE TEST

| Lesson | Group | U Test | *p-value* | Confidence Level | |
|---|---|---|---|---|---|
| | | | | Min | Max |
| 1 | Experimental | 1676 | <0.05 | 0.0000274809 | 1.0000360000 |
| | Control | 869 | <0.05 | 0.0000193866 | 0.9999997000 |
| 2 | Experimental | 1367 | 0.07632 | -0.0000503064 | 0.0000440930 |
| | Control | 496 | 0.80790 | -0.0000058192 | 0.0000130522 |
| 3 | Experimental | 1992 | <0.05 | 0.9999780000 | 1.0000470000 |
| | Control | 600 | <0.05 | 0.0000161000 | 0.9999836000 |

The significant changes observed in the experimental and control groups' preliminary and final performance can be attributed to several factors. Firstly, it's crucial to note that both groups underwent programming lessons, implying they were exposed to the same content and utilized identical robotics instruments. The sole difference lies in applying the CTProgER educational process to the experimental group. This distinction is pivotal as it allows for the isolation of the impact of CTProgER on the experimental group's performance.

Analyzing such a context solely through statistical measures is challenging, as even minor changes could influence outcomes. However, despite this challenge, the results from both groups align with existing literature [39], [36], [1], [45]. It's widely acknowledged that integrating robotics into programming education and fostering CT can yield favorable outcomes. Therefore, the observed impacts in both groups corroborate with established findings. However, exploring potential reasons behind the lack of significant changes in lesson 2 is noteworthy. This could be attributed to various factors, including the complexity of the lesson content, individual differences in student engagement, or instructional variations. Lesson 2 may have involved concepts or tasks that could have been more conducive to applying CTProgER, leading to less pronounced effects than other lessons. Factors such as the timing of the research intervention or the duration of exposure to CTProgER could also influence the observed outcomes.

To truly discern the impact of CTProgER on programming learning, it's imperative to scrutinize the effects that each

lesson exerts. This requires a more profound analysis beyond mere statistical comparisons. One helpful tool for this purpose is Cohen's effect size, which helps quantify the magnitude of the differences observed between groups. By calculating Cohen's *d* effect size index, we can gain further insights into the practical significance of the observed effects.

We also analyzed the effect of the educational process on student performance in the experimental and control groups. Calculating Cohen's *d* effect size index, we obtained $d = 0.49$ in Lesson 1, 0.06 in Lesson 2, and 0.33 in Lesson 3 (see Table VI). Cohen's *d* allows us to interpret the proportion of students who exhibit a higher mean performance in the experimental group compared to the control group.

TABLE VI
EFFECT SIZE WHEN EXPERIMENTAL POST-TEST X CONTROL POST-TEST IS CONSIDERED

| Lesson | Effect (d) | Effect Size |
|---|---|---|
| 1 | 0.49 | Small |
| 2 | 0.06 | Insignificant |
| 3 | 0.33 | Small |

As *d* is equivalent to the tabulated *Z-score* of a standard normal distribution [12], we can deduce that approximately 69% of students in Lesson 1, 52% in Lesson 2, and 63% in Lesson 3 of the experimental group displayed a higher mean performance than their counterparts in the control group. Following Cohen's definition [13], Lesson 1 and 3 effect sizes are considered small since $d \geq 0.35$, and Lesson 2 is considered insignificant since $d \geq 0.20$.

The application of Cohen's effect size provides a nuanced understanding of the impact of CTProgER across different lessons. While Lesson 2 yielded a smaller effect size than Lessons 1 and 3, it's essential to interpret this in conjunction with other contextual factors. For that, we further investigated the effects of the research intervention on the experimental and control groups separately. This allowed us to gain deeper insights into the specific impacts of the CTProgER educational process on each group by lessons.

To better understand the observed small effect sizes when considering the experimental and control groups together, we analyzed the effects of the research intervention on each group individually. The results revealed notable differences in the magnitude of the impact between the experimental and control groups across different lessons (see Table VII). In the experimental group, the effects of the research intervention were substantial, with a large effect size observed in Lesson 1 ($d = 1.15$), a small effect size in Lesson 2 ($d = 0.41$), and another large effect size in Lesson 3 ($d = 1.51$). Conversely, in the control group, the effects were more modest, with a medium effect size observed in Lesson 1 ($d = 0.56$), an insignificant effect in Lesson 2 ($d = 0.06$), and another medium effect size in Lesson 3 ($d = 0.65$).

These findings suggest that the research intervention based on the CTProgER educational process significantly impacted the experimental group more than the control group across all lessons. In contrast, the intervention that did not adhere to the CTProgER had a minor effect on the control group, highlighting the effectiveness of the CTProgER approach in enhancing programming outcomes.

TABLE VII
EFFECT SIZE BY GROUPS

| Lesson | Comparison Situation | Effect (d) | Effect Size |
|---|---|---|---|
| 1 | Experimental Pre-Test x Experimental Post-Test | 1.150 | Large |
| | Control Pre-Test x Control Post-Test | 0.559 | Medium |
| 2 | Experimental Pre-Test x Experimental Post-Test | 0.405 | Small |
| | Control Pre-Test x Control Post-Test | 0.058 | Insignificant |
| 3 | Experimental Pre-Test x Experimental Post-Test | 1.512 | Large |
| | Control Pre-Test x Control Post-Test | 0.645 | Medium |

These data further reinforce the acceptance of the alternative hypothesis and shed light on how CTProgER impacts the teaching and learning process of programming in the context of TV High School. Understanding the impacts of an educational process is a complex endeavor, as numerous variables come into play. However, it's well-documented that certain factors, such as the integration of robotics, can significantly enhance the learning process. In this context, the observed impacts of CTProgER underscore its effectiveness in improving programming skills among TV High School students.

### *A. Implementation Challenges and Impacts of the CTProgER Process*

Implementing the CTProgER educational process in a public High School faced several challenges. Adapting the process to the resource-limited and dynamic environment of public schools was difficult. Limited infrastructure and resources constrained the consistent application of CTProgER. High student turnover and sporadic attendance, known as selective attrition, further complicated the intervention and affected data reliability. Additionally, varying levels of student engagement posed a challenge. Although students were part of a computing-focused TV High School, their interest in programming and robotics varied widely. Efforts were needed to engage disinterested students while maintaining the motivation of enthusiastic ones.

Conducting the intervention in a real educational setting involved complex factors, including the need for significant investments in materials and infrastructure, logistical issues, and managing diverse student needs. Despite these challenges, the CTProgER intervention achieved positive outcomes. Notably, it influenced students' final-year projects, with several choosing robotics as their theme. This indicates that the CTProgER process not only improved programming skills but also fostered a broader engagement with robotics and CT, demonstrating the value of such methodologies in real-world educational settings.

### *B. Threats of Validity*

The study's findings face several validity concerns due to its focus on a specific Brazilian TV High School, which limits generalizability to other educational contexts or student

demographics. Data collection involved manually correcting programming test results and introducing potential human errors. To address this, steps were taken to minimize errors, such as double-checking data and considering using automated tools in future studies. The programming test questions might not have been as straightforward as needed, potentially affecting student performance and statistical outcomes. This issue was partly mitigated by validating the content and conducting pilot tests to refine questions. However, uncontrollable factors such as variations in students' prior experiences, the quality of elementary education, and extracurricular activities were not adjusted for, which could impact the results. Future research should aim to control these variables more rigorously and apply findings to a broader range of student populations to enhance the validity and generalizability of the results.

## VI. Concluding Remarks and Future Works

To examine the relation between students (X) and programming concepts (O), R(X,O), in high school to assess the influence of CTProgER [37] on programming learning, we conducted a research-intervention study comprising two main phases: (1) Implementing a research-intervention in a TV High School setting to implement CTProgER and (2) Evaluating its efficacy on high school students.

The findings underscore the effectiveness of the CTProgER educational process, designed to teach programming through educational robotics, in enhancing students' programming skills [37]. Significant disparities between participants who engaged with the CTProgER highlight its potential benefits. However, it's important to note that the results cannot be extrapolated to other student demographics, as the study solely focused on data from students enrolled in a Brazilian public TV High School.

The statistical analyses presented in this study offer quantitative guidelines for assessing the effectiveness of the CTProgER educational process [37], particularly in the context of programming learning and the relation between students (X) and programming concepts (O), R(X,O). These analytical procedures, in conjunction with the outcomes of this investigation, contribute valuable insights to the scientific community, guiding and validating educational processes.

Moving forward, a qualitative study will be conducted using observational data from the research intervention to provide a deeper understanding of the impact and efficacy of the CTProgER educational approach. This follow-up study, as suggested by Souza *et al.* [37], aims to offer further insights into the effectiveness of the educational process and to guide future educational practices. To achieve this, the study will incorporate artificial intelligence (AI) techniques to analyze recordings of the intervention sessions. The focus will be on employing syntactic analysis to examine the dialogues and interactions occurring during the lessons. The syntactic analysis involves breaking down the sentence structures in the recorded dialogues to understand how students and teachers interact and respond to the CTProgER process. By analyzing these interactions, it will be possible to identify patterns in communication, student engagement, and instructional effectiveness. This approach will help to uncover qualitative aspects that are not easily captured through quantitative measures alone, thus providing a more comprehensive evaluation of the CTProgER process.

## Acknowledgment

The authors also would like to thank all college professors and researchers who helped develop this study. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES).

## References

[1] A. Alam. Educational robotics and computer programming in early childhood education: A conceptual framework for assessing elementary school students' computational thinking for designing powerful educational scenarios. In *2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, pages 1–7, 2022.

[2] F. Alegre, J. Underwoood, J. Moreno, and M. Alegre. *Introduction to Computational Thinking: A New High School Curriculum Using CodeWorld*, page 992–998. Association for Computing Machinery, New York, NY, USA, 2020.

[3] D. Alimisis. Educational robotics: Open questions and new challenges. *Themes in Science and Technology Education*, 6(1):63–71, 2013.

[4] S. M. S. Azman, M. Arsat, and H. Mohamed. The framework for the integration of computational thinking in ideation process. In *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. IEEE, 2017.

[5] M. Bosch and J. Gascón. Twenty-five years of the didactic transposition. *ICMI bulletin*, 58(58):51–65, 2006.

[6] K. Brennan and M. Resnick. 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*, volume 1, page 25, 2012.

[7] M. Chevalier, C. Giang, A. Piatti, and F. Mondada. Fostering computational thinking through educational robotics: A model for creative computational problem solving. *International Journal of STEM Education*, 2020.

[8] Y. Chevallard. On didactic transposition theory: Some introductory notes. In *Proceedings of the international symposium on selected domains of research and development in mathematics education*, pages 51–62. Comenius University Bratislava, Czechoslovakia, 1989.

[9] Y. Chevallard. Fundamental concepts in didactics: Perspectives provided by an anthropological approach. *Research in didactique of mathematics: Selected papers*, pages 131–168, 1992.

[10] Y. Chevallard. A theoretical approach to curricula. *Journal fuer Mathematik-didaktik*, 13(2):215–230, 1992.

[11] Y. Chevallard. L'analyse des pratiques enseignantes en théorie anthropologique du didactique. *Recherches en didactique des mathématiques*, 19(2):221–266, 1999.

[12] R. Coe. It's the effect size, stupid: What effect size is and why it is important. In *Annual Conference of the British Educational Research Association*. Education-line, 2002.

[13] L. Cohen, K. Morrison, and L. Manion. Research methods in education. *IEducation, Research methods. Routledge.*, 2011.

[14] J. Colomb. Chevallard (yves).—la transposition didactique: du savoir savant au savoir enseigné. *Revue française de pédagogie*, 1986.

[15] L. O. da Silva Petini. Conhecimentos matemáticos mobilizados por alunos no desenvolvimento de projetos de robótica. *Semana da Matemática do Instituto de Matemática*, 2018.

[16] I. M. L. de Souza, R. da Silva Rodrigues, and W. Andrade. Explorando robótica com pensamento computacional no ensino médio: Um estudo sobre seus efeitos na educação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, 2016.

[17] I. M. L. de Souza, R. da Silva Rodrigues, and W. Andrade. Introdução do pensamento computacional na formação docente para ensino de robótica educacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 2016.

[18] M. Ebenezer Takuno de and S. Thais Helena dos. Dicionário interativo da educação brasileira-educabrasil. *São Paulo: Midiamix Editora*, 2015.

[19] F. J. García-Peñalvo. Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, 80:407–411, 2018.

[20] M. R. González. Computational thinking test: Design guidelines and content validation. In *Proceedings of EDULEARN15 conference*, pages 2436–2444, 2015.

[21] S. Isabelle M. L., W. L. Andrade, and S. Lívia M. R. Analyzing the effect of computational thinking on mathematics through educational robotics. In *2019 IEEE Frontiers in Education Conference (FIE)*, 2019.

[22] W. Jeannette M. Computational thinking. *Commun. ACM*, 49(3):33–35, Mar. 2006.

[23] R. P. Lai. Beyond programming: A computer-based assessment of computational thinking competency. *ACM Transactions on Computing Education (TOCE)*, 22(2):1–27, 2021.

[24] S. Y. Lye and J. H. L. Koh. Review on teaching and learning of computational thinking through programming: What is next for k-12? *Computers in Human Behavior*, 41:51–61, 2014.

[25] A. Malizia, T. Turchi, and K. A. Olsen. Block-oriented programming with tangibles: An engaging way to learn computational thinking skills. In *2017 IEEE Blocks and Beyond Workshop (B&B)*, pages 61–64. IEEE, 2017.

[26] M. J. Marcelino, T. Pessoa, C. Vieira, T. Salvador, and A. J. Mendes. Learning computational thinking and scratch at distance. *Computers in Human Behavior*, 80:470–477, 2018.

[27] P. Mongeon and A. Paul-Hus. The journal coverage of web of science and scopus: a comparative analysis. *Scientometrics*, 106(1):213–228, 2016.

[28] O. Mubin, C. J. Stevens, S. Shahid, A. Al Mahmud, and J.-J. Dong. A review of the applicability of robots in education. *Journal of Technology in Education and Learning*, 1(209-0015):13, 2013.

[29] J. Noh and J. Lee. Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development*, 2020.

[30] J. Nouri, L. Zhang, L. Mannila, and E. Norén. Development of computational thinking, digital competence and 21st century skills when learning programming in k-9. *Education Inquiry*, 11(1):1–17, 2020.

[31] S. Papert and I. Harel. Situating constructionism. *Constructionism*, 36(2):1–11, 1991.

[32] S. A. Papert. *Mindstorms: Children, computers, and powerful ideas*. Basic books, 1980.

[33] M. Pivetti, S. Di Battista, F. Agatolio, B. Simaku, M. Moro, and E. Menegatti. Educational robotics for children with neurodevelopmental disorders: A systematic review. *Heliyon*, 6(10):e05160, 2020.

[34] M. Sarıtepeci and H. Durak. Analyzing the effect of block and robotic coding activities on computational thinking in programming education. *Educational research and practice*, pages 490–501, 2017.

[35] M. Schivani. *Contextualização no ensino de física à luz da teoria antropológica do didático: o caso da robótica educacional. 2014. 220f.* PhD thesis, Tese (Doutorado em Ensino de Ciências e Matemática)–Faculdade de Educação ..., 2014.

[36] I. Souza, W. Andrade, and L. Sampaio. Aplicações da robótica educacional para o desenvolvimento do pensamento computacional no contexto do ensino médio integral. In *Anais do Simpósio Brasileiro de Educaçã em Computação*, Porto Alegre, RS, Brasil, 2021. SBC.

[37] I. M. Souza, W. L. Andrade, and L. M. Sampaio. A framework for teaching programming in high school through educational robotics. In *2022 IEEE Frontiers in Education Conference (FIE)*, pages 1–9. IEEE, 2022.

[38] I. M. Souza, L. M. Sampaio, and W. L. Andrade. Analyzing the effectiveness of an educational process for teaching programming through educational robotics in a brazilian technical and vocational high school. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, pages 389–401. SBC, 2023.

[39] I. M. L. SOUZA, W. L. ANDRADE, and L. M. R. SAMPAIO. Educational robotics applications for the development of computational thinking in a brazilian technical and vocational high school. *Informatics in Education*, 2022.

[40] M. Tekdal. Trends and development in research on computational thinking. *Education and Information Technologies*, 26(5):6499–6529, 2021.

[41] A. Tucker. *A model curriculum for k–12 computer science: Final report of the acm k–12 task force curriculum committee*. ACM, 2003.

[42] C. Viviane Gurgel de. Roboeduc: especificação de um software educacional para ensino da robótica às crianças como uma ferramenta de inclusão digital. Master's thesis, Universidade Federal do Rio Grande do Norte, 2008.

[43] J. Voogt, P. Fisser, J. Good, P. Mishra, and A. Yadav. Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4):715–728, 2015.

[44] A. Yadav, N. Zhou, C. Mayfield, S. Hambrusch, and J. T. Korb. Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 465–470, 2011.

[45] W. Yang, D. T. K. Ng, and H. Gao. Robot programming versus block play in early childhood education: Effects on computational thinking, sequencing ability, and self-regulation. *British Journal of Educational Technology*, 2022.

[46] H. Yildiz Durak. Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences. *Journal of Computer Assisted Learning*, 34(6):740–752, 2018.

[47] D. C. Zanardi. *A análise praxeológica de atividades experimentais subsidiando a elaboração de situações-problema no ensino de física*. PhD thesis, Universidade de São Paulo, 2013.

[48] H. Zanetti and C. Oliveira. Práticas de ensino de programação de computadores com robótica pedagógica e aplicação de pensamento computacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 2015.

[49] M. Zapata-Cáceres, E. Martín-Barroso, and M. Román-González. Computational thinking test for beginners: Design and content validation. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 1905–1914, 2020.

[50] K. Zawieska and B. R. Duffy. The social construction of creativity in educational robotics. In *Progress in Automation, Robotics and Measuring Techniques*, pages 329–338. Springer, 2015.