A Framework for teaching programming in High School through Educational Robotics

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, {wilkerson, livia}@computacao.ufcg.edu.br

***Abstract*—This Research-to-Practice Full Paper aims to introduce a methodological framework to foster Computational Thinking (CT) skills by teaching programming with Educational Robotics (ER) in High School. Problem-solving is an essential skill for human development, so it is necessary to introduce actions to improve this skill in High School. It is possible to stimulate problem- solving through computer programming teaching with ER. However, as far as we know, there are no methodologies for teaching programming with ER focusing on CT development in High School that went through scientific validation of the proposal. This study is an Educational Design Research that aims to introduce a methodological framework for teaching programming in High School with ER (named CTProgER) and validate two instances of the framework. It was considered consolidated teaching methodologies with ER, combined with the Anthropological Theory of Didactics and evidence from our ER studies. As a result, we present the CTProgER Framework guideline, two instances of the CTProgER Framework, and these instances' validation through the Delphi method.**

I. Introduction

Problem-solving is an essential skill for human development. Computational Thinking (CT) is a modern approach to improve problem-solving skills [3]. CT can be defined as a problem-solving process exploring common skills in the field of Computer Science [24], [7], [31]. The Computer Science Teachers Association (CSTA) has introduced CT in its Model Curriculum for K-12 Computer Science [38] (Basic Education stages in Brazil), to improve students' problem-solving skills.

Programming education has been explored worldwide to stimulate CT in High School students, becoming no longer an exclusive practice in Computer and Engineering courses [43]. However, teaching programming can be challenging to understand for several reasons: lack of student preparation, lack of adequate methodologies and computational tools that help teachers and students in teaching-learning process [19]. Consequently, computational instruments and methodologies guidelines must be considered to facilitate computer programming teaching and CT development.

Educational Robotics (ER) is one of these computational instruments that is becoming increasingly popular in classrooms to support education at different educational stages that favor developing teamwork, logical reasoning, and creativity. ER has been used to engage students in and improve their CT skills because through a robot it is possible to observe the effects of the programming on the robot's behavior in real-time [5], [8], [25]. ER is used in programming education to allow abstract concepts to be implemented visually and physically to understand Computer Science concepts more quickly and to foster students' CT skills [1], [2], [29].

Educational companies [44], [18], [32] commonly propose a methodology to promote Science, Technology, Engineering, and Mathematics (STEM) teaching through ER [9], [29]. However, these do not support the computing concepts involved in robotics, such as algorithms, conditionals, and loops. These concepts are taught when the robot's programming needs it.

In addition, general methodologies propositions do not show their validation process, demonstrating the need for effectively valid methodologies. However, validating an educational methodology is still a challenge because it involves the solutions to complex problems proposition to academic practice [30], i.e., it is a task that is not limited to statistical procedures. Methodological proposals in education undergo a validation process in typical environments and are based on evidence. Due to the COVID-19 pandemic, face-to-face activities have become limited. In this way, validation methodologies that involve the expert's validation, such as the Delphi Method, become an alternative for the process of validating methodological proposals.

Although the literature recommends ER for teaching programming and fostering CT skills, studies discussing guidelines to implement and validate ER classes with this goal still appear scarce. We conducted previous studies that demonstrated that ER teaching based on CT impacts students and teachers positively [14], [15], [23], [36], [35]. However, our analyses did not consider the teaching methodology validation and did not contemplate programming teaching as a priority. As far as we know, there are no methodologies for teaching programming through ER that focus on CT development in High School that went through a process of scientific validation of the proposal.

This study is an Educational Design Research that aims to introduce a methodological framework for teaching programming in High School with ER, named CTProgER Framework, and validate two CTProgER Framework instances. This study was organized in four steps: 1) CTProgER Framework definition based on methodologies for teaching with ER and on the Anthropological Theory of Didactics [12]; 2) CTProgER Framework presentation to experts following the Delphi method [28]; 3) Instances' validation by experts'

Opinion; and 4) Analysis of data obtained from the experts' opinion.

We conducted two rounds of instances' validation according to the Delphi method. As a result, we observed a positive majority of experts' votes and the absolute agreement percentage (PAA) between the experts' opinions. We performed two rounds of experts' validation to observe the instance's concordance with the CTProgER Framework. In both rounds, we obtained a PPA $> 75\%$, and a positively majority vote of experts.

The instances validation through the Delphi method was part of the validation process of our CTProgER Framework. The instances' validation allowed to align the concordance of the instances with the CTProgER Framework guidelines. We validated the concordance between instantiating and guidelines through experts' evaluation. The first round provided necessary data to improve the concordance of our instances with the CTProgER Framework guidelines. This data was used to adjust the validated instances in the second round. The instances' validation process improves our instances and minimizes future face-to-face studies problems, where we will test our framework validity and effectiveness in our future studies.

This paper was organized as follows: in Section II, we briefly described the fundamental concepts of Anthropological Theory of Didactics and the Delphi method. Section III was dedicated to related works. In Section IV, we detailed the work study process. In Section V we presented the CTProgER Framework, CTProgER Framework Instances, Instance's Validation results, and the threats of validity. Finally, the main conclusions and future works are presented in Section VI.

II. Background

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

A. Anthropological Theory of Didactic

Anthropological Theory of Didactic is based on praxeology and refers to the human behavior study within a given Institution, represented as [I] and where the action takes place, such as school, university, and classroom. According to [11], a praxeology, also known as Praxeological Organization (PO), is represented by [T, τ, θ, Θ], where: T: type of task that can be subdivided into several tasks [τ]; τ: a technique used to perform the task; θ: technology applied to complete the task; Θ: theory involved to carry out the task.

In Anthropological Theory of Didactic, the teaching of an Object (O) (e.g. Physics and Programming) in any Institution [I] needs didactic moments. Didactic moments are class phases. Anthropological Theory of Didactic has six didactic moments: 1) First Meeting; 2) Exploration Moment; 3) Constitution and Development of the technological-theoretical block that aims to base the techniques worked at the Moment of Exploration that responds to the types of tasks explored at the First Meeting; 4) Time of Work of the Technique; 5) Institutionalization Moment; 6) Assessment Time.

These didactic moments do not necessarily have to follow a chronological order, and they can coincide on different occasions, start in one class and end in others, and involves the praxeology constitution and its elements of the set [T, τ, θ, Θ] that, at the end of the process, it is possible to establish a relation between the Personal (X) (e.g., students and teachers) and Object (O) determined as R(X,O). This successful relation is understood as learning [13].

The anthropological Theory of the Didactic is considered because she considers Mathematics a human activity that can be extended to teaching various subjects. Therefore, it is possible to apply it in different activities that favor establishing and executing tasks and the necessary knowledge for their accomplishment. In this theory, the object is a human activity resulting from constructed knowledge. This knowledge can be known, taught, or taught as long as different technical tasks and technologies are applied to work with them [42]. In this context, we believe that programming teaching in High School can be fostered through the Anthropological Theory of the Didactic because the teaching and learning actions are anthropological.

B. Delphi Method

The Delphi method assists judgmental prediction and decision making in various research domains. This method is indicated when a problem can result from cooperative, subjective judgments or decisions and when the expert group does not favor effective communication [6], [16], [33], [39]. The Delphi method allows for a progressive consensus to be reached on a study object. This method occurs through questioning rounds to an experts' group, in which the answers are analyzed to obtain a consensus. The consensus criteria depend on the researcher's study. However, the strictness of the consensus criteria must be respected [40], [26].

The main characteristics of the Delphi method are the specialist's anonymity, the statistical representation of the distribution of results, and feedback from experts' responses for re-evaluation in subsequent rounds [41]. It takes at least two rounds for processes to be characterized as a Delphi method. However, few studies have a rounds number greater than three rounds [20], as there is a tendency to carry out the fourth and later rounds without modifying the experts' opinion.

The reason to use the Delphi method is due to its capacity to establish consensus on best policy [4]. In addition, the Delphi viability is ideal in areas where consensus is missing, process protocols, and other best methods where agreement is essential [21]. It is helpful when there is little knowledge or uncertainty surrounding the area being investigated [22], [27]. For that reason, this method is commonly used as a validation tool based on experts' opinion.

III. Related Work

Some studies propose a kind of model or framework for some teaching [13], [34], [10], [3]. However, very few have discussed how these models or methodological frameworks are validated. As far as we know, no study proposes and validates methodologies for teaching programming through ER that focus on CT development in High School.

Schivani [34] used the Anthropological Theory of the Didactic [34] as a tool to identify essential elements during didactic moments application. For this, the author performed and offered interventions in Physics teaching in High School through the ER. He used student lesson scripts by LEGO® education created for Physics teaching through the ER with LEGO Education methodology in his interventions. With this work, Schivani observes that the ER use, mainly aimed at teaching physics, allows a comprehensive approach, whether related to theory or practice, to the two blocks of Anthropological Theory of the Didactic: practical-technical and technological-theoretical. Schivani's proposal can be seen as a methodological proposal for teaching physics with ER. In addition to proposing a methodological foundation, he applies it and presents the necessary guidelines for its replication.

Our study is similar to that presented in Schivani [34] in the sense of ER use as a tool for teaching and Anthropological Theory of Didactic application in High School. However, we did not apply the Anthropological Theory of Didactic as a validation tool but rather as a theory indicating how we can organize our framework's didactic moments. We propose our framework's didactic moments based on the Anthropological Theory of Didactic and ER consolidated methodologies to teach sciences like LEGO® Education [44], besides our ER experiences [14], [15], [23], [36], [35]. Our study proposed and validated our instances according to our CTProgER Framework. As our objective is to offer a methodological framework, determining the alignment between the methodology and the application materials can facilitate the next steps of our study. Finally, our study goal favors programming teaching through ER and improves CT development, while Schivani [34] focuses on teaching Physics without associating it with CT.

Azman and Mohamed [3] proposed a framework for the integration of CT in higher education. For this, the author from an analysis of CT research and offers a framework based on four ideation process steps. The Azman and Mohamed framework [3] is organized in four integrated phases of ideation: 1) Crack The Big Issue, 1) Identify Similarities, 3) Compress The Data, 4) Planning Step-By-Step Instruction.

Our study is similar to that presented in Azman and Mohamed [3] in using the ER to promote CT. However, we consider the methodological framework for programming teaching in High School and not the CT skills. We believe that working with specific CT skills remains challenging, so we consider CT a general skill that can be fostered through teaching programming. Besides, we intend to evolve our study to methodological framework validation; we created several educational materials and its validation planning to facilitate a future methodology validation.

IV. Methodology

'This study is an **Educational Design Research** because it aims to develop a "systematic study of designing, developing, and evaluating educational interventions (such as programs, teaching-learning strategies, materials, products, and systems) as solutions for complex problems in educational practice. It also aims at advancing our knowledge about the characteristics of these interventions and the processes of designing and developing them" [30].

We introduced a methodological framework for teaching programming in High School with ER, named CTProgER Framework, and validate two CTProgER Framework instances. For this purpose, the development of this work process was divided into four steps: 1) CTProgER Framework definition; 2) CTProgER Framework presentation to experts follows the Delphi method; 3) Instances' validation by experts' opinion; and 4) Analysis of data obtained from the experts' opinion.

The work process adopted in this study was organized in steps according to the people involved, such as the researcher and experts, represented by horizontal lines (see Figure 1).

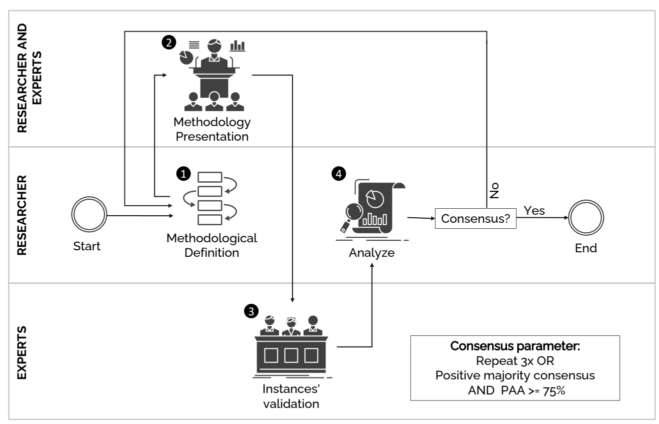

Fig. 1. Work Process

A. *Methodological Definition*

The methodological definition of the CTProgER Framework comprised the following activities: 1) Categorize the methodological steps applied to the context of ER from our previous studies; 2) Raise bibliographies about methodologies for teaching ER as a CT development tool; 3) Propose a methodological framework for teaching programming through ER to High School; 4) Create two methodological framework instances for teaching programming; 5) Investigate and propose instruments to validate the instances created.

In this step, all validation instruments were created as experts' supporting documents, student lesson scripts, lesson plans, and the Barema defined for the instance validation. This process supports the Delphi method, a strategy of instrument validation that uses an expert's systemic opinion to generate a validated product at the end of the process [26]. This step is included in the instance work process because, according to expert consideration, it may be necessary to update the instances created during the definition of the methodology framework described after each round. It is indicated that up to three rounds search for a consensus among the experts. However, it is possible that after the three rounds, an agreement is not reached, which indicates a non validation of the product.

B. Methodology Presentation

The methodology presentation presents the information necessary to conduct the validation process by the experts. An online website[1] containing a research presentation video and the main concepts that support the validation process execution should be available. In addition, the website will also make available the documents related to the instance's validation.

C. Instances' Validation

This validation aimed to determine the agreement of the instances with the CTProgER Framework guidelines. We used the Delphi method [28] to validate objects through human judgment. For this, validation rounds were conducted with the experts, which registered their impressions in the Barema, which was prepared considering the CTProgER Framework guidelines.

Due to the COVID-19 pandemic, the instances validation was made remotely to avoid physical contact between experts and researchers. From June to December 2021, six experts carried instances' validation through the Delphi method to reach a progressive consensus on a study object.

Google Form was used to obtain the data anonymously. The Google Sheets tabulated the experts' responses and identified the majority voting. The percentage of the absolute agreement of each instance was calculated separately. The R programming was used with the IRR (Internal Rate of Return) package and the AGREE function.

For this, two validation rounds of the instances were conducted with the experts, which registered their impressions in the Barema (see Table I), considering nine criteria around CTProgER Framework guidelines and the analysis parameters presented in the Subsection IV-D.

The validation process used several documents to support the experts' activities. The documents to support the experts' activities are:

- **Supporting Material for Experts (video):** Presentation of the main concepts underlying the study.
- **Supporting Material for Experts (text):** detailed proposal document with theoretical concepts and procedures to be followed.
- **Student Material:** two documents that will be made available to the student during programming classes having robotics as a teaching tool. The instances should be two programming lessons to 1st year of High School.
- **Teacher Material:** two documents with teaching guidelines for applying programming lessons with robotics as a teaching tool. The instances should be two programming lessons to 1st year of High School.
- **Barema:** anonymous form for recording the experts' answers to nine criteria for evaluating the methodology instances (see Tables I).

[1]The website is available online: https://bit.ly/3mIT10j

D. Instances' Validation Data Analysis

The analysis aims to tabulate the results produced by the experts consisting of responses into positive and negative answers and observe a possible consensus among the experts' opinions about the methodology framework guideline. The consensus was based on majority voting such that 50% of positive answers plus one in Barema. In addition to the majority voting, the consensus also needs a PAA $> 75\%$ [37].

The PAA is obtained through Equation 1, where NC is the number of times experts agree on a rating, and TA is the total number of ratings.

$$PAA = \frac{NC}{TA} * 100 \quad (1)$$

Each instance should be considered valid if consensus was reached in Barema. However, if no consensus was reached, it is possible to have up to three new validation rounds according to the Delphi method [26]. If a new round was necessary, the researcher conducting the process should observe the experts' feedback and make the required modifications to carry out a new round. The Delphi method says that if consensus is not reached within four rounds, there is a high probability that it will not be achieved in later rounds [17].

The instances' Validation considers two types of data: quantitative and qualitative. Quantitative data are obtained by tabulating the experts' responses into positive (2 – Partially and 3 - Yes) and negative (0 - No and 1 - Very little) answers and observing a possible consensus among the experts' opinions about the nine criteria on the Barema (see Table I). This data is considered for the validation or not of the CTProgER Framework instances and to determine when to stop the validation process by experts. On the other hand, qualitative data were considered to align the instances with the framework rules better. These data were obtained through a Barema text field that asked the experts to evaluate their assessment of each criterion. These considerations were used between the evaluation rounds to improve the instances.

V. Results

This section presents the CTProgER Framework, CTProgER Framework instances, instance's validation results, and the threats to validity.

TABLE I
BAREMA CRITERIA

Criteria	Judgment
Q01: Is it possible to identify the content this lesson wants to teach?	0 - No; 1 - Very little; 2 - Partially; 3 - Yes
Could you please briefly explain your answer regarding criterion Q01?	
Q02: Is it possible to identify the lesson objective (what does the lesson intend to promote in the student)?	0 - No; 1 - Very little; 2 - Partially; 3 - Yes
Could you please briefly explain your answer regarding criterion Q02?	
Q03: Is it possible to identify the development lesson objective (What does the teacher want the student to develop in)?	0 - No; 1 - Very little; 2 - Partially; 3 - Yes
Could you please briefly explain your answer regarding criterion Q03?	
Q04: Is it possible to identify the lesson problem?	0 - No; 1 - Very little; 2 - Partially; 3 - Yes
Could you please briefly explain your answer regarding criterion Q04?	
Q05: Is it possible to identify prior knowledge needed in this lesson (student-centered)?	0 - No; 1 - Very little; 2 - Partially; 3 - Yes
Could you please briefly explain your answer regarding criterion Q05?	
Q06: Is it possible to identify the programming structures that must be implemented to solve the problem proposed in this lesson?	0 - No; 1 - Very little; 2 - Partially; 3 - Yes
Could you please briefly explain your answer regarding criterion Q06?	
Q07: Is it possible to identify ways to test the solution to be implemented in this lesson?	0 - No; 1 - Very little; 2 - Partially; 3 - Yes
Could you please briefly explain your answer regarding criterion Q07?	
Q08: Is it possible to identify how the student is assessed in this lesson?	0 - No; 1 - Very little; 2 - Partially; 3 - Yes
Could you please briefly explain your answer regarding criterion Q08?	
Q09: Is it possible to identify the knowledge that the student must obtain in this lesson?	0 - No; 1 - Very little; 2 - Partially; 3 - Yes
Could you please briefly explain your answer regarding criterion Q09?	

A. *CTProgER Framework*

The CTProgER Framework for teaching programming through ER focused on the CT skills is based on the methodologies for teaching with commercial ER consolidated like LEGO® Education with the concepts proposed by Anthropological Theory of the Didactic and our study [14], [15], [23], [36], [35].

In this sense, our CTProgER Framework considers an Institution [I] as the space for teaching practice where, given as input (input) an Object (O), it is possible to develop a relation R(X,O) as output (see Figure 2).

The **input** involves identifying one or more programming concepts that the teacher wants to teach to help students develop specific knowledge. This development process comprises conceptual, procedural, and attitudinal aspects related to concepts, facts, and principles; learning how to do and; learning how to be. The **output** represents the relation between the individual and the object (O). The individual is the student, and the object concerns the CT worked through programming concepts. In Anthropological Theory of the Didactic, any work from human activity is considered an object (O), so an individual (X) can establish a relation with the object, an action represented by R(X,O). In this sense, it is possible to state that a given object exists for an individual that the relation R(X,O) is not empty, R(X,O) $\neq 0$. We understood it as learning the concepts given as input when a positive relation is observed.

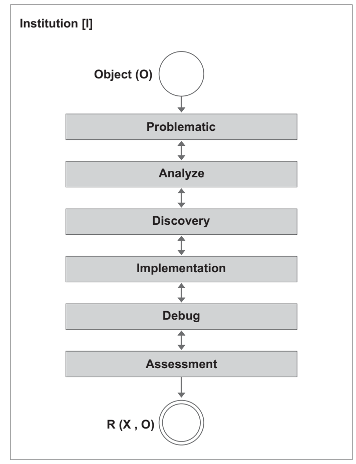

Fig. 2. CTProgER Framework Process

To support this proposition, we consider six didactic moments from Anthropological Theory of the Didactic, and the lessons learned from our previous studies are:

To support this proposition, we consider six didactic moments from Anthropological Theory of the Didactic, and the lessons learned from our previous studies are they:

1) **Problematic:** Involves presenting a problem situation that favors the 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 to be the protagonist of their proposed solutions.
3) **Discovery:** Involves the understanding of the concepts of work given as input. At this didactic moment, the teacher must lead the students to establish a connection between 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 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 student must take the lead in their practices, and the teacher should facilitate applying the new concepts studied.
5) **Debugging:** 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 of 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.

B. CTProgER Framework Instances

To apply the CTProgER Framework for teaching programming through ER focusing on developing CT skills, we planned four programming content lessons: Concept of algorithms, data input and output, data types, variables, constants, condition control structure, and repeating control structures. Also, we introduced robotics contents in the lesson plan: Touch sensor, Ultrasonic sensor, Color sensor, and Light sensor. These lessons will be applied during a month, each lasting 1h30min with students organized in groups of five people.

We built two instances following each aspect present in the CTProgER Framework. We chose the lessons "The Cleaning Robot", which introduces algorithms and starts with the domestic cleaning robots, and "The Driver Robot" introduces a condition control structure with the autonomous vehicles for instantiating. Therefore, we developed the teacher's and student's material for each lesson. Below we will detail the main points of each instance[2].

The instances were developed for teaching any programming language that could be worked with robotics. Consequently, they do not have code in a specific programming language but the concepts involved. The teacher needs to perform the programming syntax during the lesson according to the guidelines from the teacher material.

The instances were organized by CTProgER Framework didactic moments, so the lessons have six moments that were transformed into topics, they are: **Problematic** (knowing the problem); **Analysis** (analyzing the problem); **Discovery** (discovering knowledge); **Implementation** (implementing a solution); **Debugging** (testing the solution); **Assessment** (questions set must answer at the lesson end).

C. Experts' Profile

The experts involved in the instance validation are six professors and researchers who have already done research or worked with CT, teaching programming, or ER. Among the six experts, 50% (3) are female and 50% (3) male. Besides, 6,7% (1) graduated from Higher Education with a bachelor's degree, 66,6% (4) with a licentiate degree, and 16,7% (1) with a technologist course. The training area in Higher Education of the experts was computer science in 50% (3), computing in 33,3% (2), and systems for the internet in 16,7% (1). The percentage of the experts with a master's degree was 83,3% (5), and 16,7% (1) have a Ph.D. The highest training area (master's or Ph.D.) of experts is in Computer Science in 83,3% (5) and Mathematics and Technological Education in 16,7% (1). Finally, 66,7% (4) of the experts have teaching experience in Technical and Vocational Education and 33,3% (2) in Higher Education.

[2]The instances are available online: https://bit.ly/3mIT10j.

D. Instances' Validation Results

According to data presented in Figure 3, it is possible to observe the majority of experts voting in The Cleaning Robot Instance and The Driver Robot Instance with positive answers $> 50\%$ plus one in each criteria from the Barema.

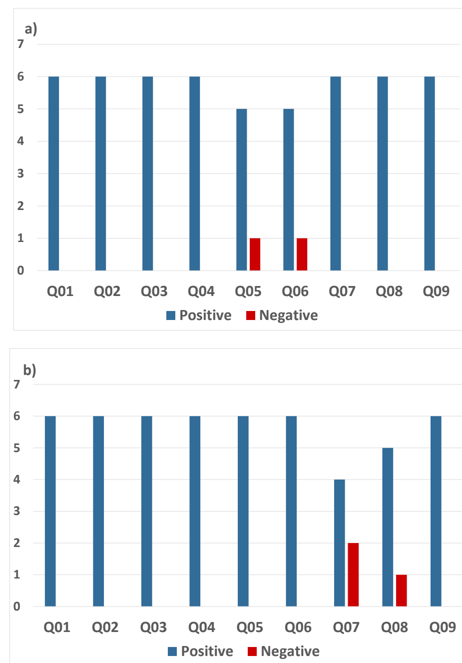

Fig. 3. a) Experts' validation data by version 1 to The Cleaning Robot Instance. b) Experts' validation data by version 1 to The Driver Robot Instance.

In addition, we calculated the percentage of absolute agreement (PAA) between de experts. Table II shows PAA by instance. We observed that both instances have a PPA $> 70\%$, and as the majority voting is positive, we considered that our instances are validated in the first round of experts' validation.

According to our consensus criteria, both instances were validated. However, as we applied the Delphi method, it is necessary to execute one more expert round to minimize possible inconsistencies of instantiation of our CTProgER

TABLE II
PERCENTAGE OF ABSOLUTE AGREEMENT IN ROUND 1

Instance	PAA (%)
The Cleaning Robot	77.8
The Driver Robot	77.8

Framework, precisely adjusting the Q5, Q6, Q7, and Q08 criteria.

The Q05 criteria refer to prior knowledge students need for The Cleaning Robot Instance, and the Q06 criteria refer to programming structures that must be implemented during The Cleaning Robot Instance. As the subject of this instance was algorithms introduction, identifying which prior knowledge was necessary to start this subject study could be difficult. The same was observed concerning the difficulty to recognize the programming structures in Q06 criteria because this lesson did not use a specific programming language but pseudo-code.

The Q07 criteria identify ways to test the solution implemented during the lessons. Although our CTProgER Framework has a debug didactic moment, it is possible that it may not be apparent, In the guidelines we indicated the student instance that the student should follow to test his solution in The Driver Robot Instance. The Q08 criteria refer to how the student is assessed during the lesson. Our CTProgER Framework has a didactic assessment moment. The student instances has a question group and lesson plans orientation on how the teacher can assess the student. However, it is possible that it may not be clear how the student is assessed during The Driver Robot Instance.

We summarized (see Tables III and IV) and analyzed the expert's qualitative considerations to update the instances of the second round. These considerations are necessary to improve our instances and minimize future problems during interventions when we test the efficiency of our CTProgER Framework. We considered the experts' review to adjust both instances, and we carried out the second round of instances' validation.

To analyze the second round of instances' validation, initially, we observed a majority vote of experts, considering the 50% positive answers plus one in each question to validate. We analyzed the votes by instance. According to data presented in Figure 4, it is possible to observe the majority of experts voting in The Cleaning Robot Instance and The Driver Robot Instance with positive answers $>$ 50% plus one in each criteria from the Barema.

We also observed a PPA = 100% and an average improvement in agreement compared to the first round of 22.2% in The Cleaning Robot Instance (see Table V). In The Driver Robot Instance, we observed a PPA = 88.9% and an average improvement in agreement compared to the first round of 11.1%. In both instances, the majority voting is positive, and a PAA is above 70%. For that reason, we considered that our instances

TABLE III
EXPERT CONSIDERATIONS (QUALITATIVE DATA) ON INSTANCE "THE CLEANING ROBOT" (OUR TRANSCRIPT).

Criteria	Observation
Q2	I had to go back in the document to understand that the objective favors perceptions that algorithms are present in our daily lives. As I understand it, this observation is related to the teacher material.
Q03	As it focuses on High School students, I think the description of the activity "Implementing a solution" should have a little more detail, reinforcing the issue of creating the algorithm even in natural language.
	The development objective is to build a sequence of steps to make the robot walk under a square. However, the role of the triangle (still specifically equilateral) in the textbook example was not clear.
Q05	It is unclear what prior knowledge the student needs to have to progress in this class.
	We identify square, equilateral triangle geometric shapes and angles about the contents. However, the function of the triangle (and even specifically equilateral) was not clear.
Q06	It is unclear what the structures would be. After using natural language, I could make the association with the idea of input - processing - output.
Q07	To make it easier for the student to test the solution, he could use a sequence of images for each instruction. As if it were a sequential simulation of the robot's movement.
Q08	Through the answers to the questions, the first question is in the form of a "yes" or "no" answer. Is that all you want the student to answer at this point? If you're expecting any more information, it's unclear in the question. On the third question: are you repeating the initial question? To student build the code again? (I didn't understand the purpose). Or do you want the student to write code for the rectangle on the side?
Q09	I was in doubt if knowledge revolves around the explicit concepts presented about algorithms and triangles or the ability to draw a parallel on how technological solutions can perform tasks.

are validated in the second round of experts' validation.

E. Threats of Validity

This study has threats to validity. The CTProgER Framework didactic moments cannot suit the High School target audience, which can disqualify the instances' validation process presented. The criteria from the Barema cannot be clear. The experts' answers do not reflect the response that the author of this study needs, which can also disqualify the instances' validation process. Finally, it is possible that the work process adopted was extensive, especially the instances' validation step, which may have influenced the experts' answers. We will conduct an experiment in High School to apply the methodological framework and instance and thus re-evaluate the framework guidelines proposition and the content of validated instances in future works.

VI. CONCLUDING REMARKS AND FUTURE WORKS

We introduce a methodological framework for teaching programming in High School with ER, named CTProgER Framework, and validate two CTProgER Framework instances. The framework was defined following an Educational Design Research conducted using a Delphi technique. The CTProgER

TABLE IV
EXPERT CONSIDERATIONS (QUALITATIVE DATA) ON INSTANCE "THE DRIVER ROBOT" (OUR TRANSCRIPT).

Criteria	Observation
Q01	Yes, but it could be more evident if you use the terms "decision structure" or "choice" instead of "verification".
Q02	Yes, I can identify the objective, but the second script presents algorithms in pseudo-code with a specific syntax. It was not explained to the student in this script how this way of writing the algorithm is done, which elements, etc.
Q04	After the first lesson, I understand that the problem revolves around the robot's building that avoids obstacles. My question revolves around the exact problem and the implicit concepts, that is, understanding the concepts passed on in class. My problem is to know if the problem is just the construction or if it expands to the understanding of the concepts as well.
Q05	It is not possible to identify the previous concepts. There could be a paragraph that could clarify this.
Q06	Yes, but I reinforce the importance of explaining how pseudo-code works or signaling that this would be necessary prior knowledge.
Q07	The activities part was confusing for me, where you are asked to transform the pseudo-code (which was not clearly taught) into a programming language (which is also not explained in the script).
Q08	I continue to reinforce the need to adjust the pseudo-code issue and the robot's programming language.
	In the first question, is your goal for the student to write "Walk without limit"? In question 2, it may not be so straightforward to calculate the distance. Maybe the teacher has to work harder. In the "testing the solution" section, in question 1, you want the student to answer only "yes" or "no".

TABLE V
PERCENTAGE OF ABSOLUTE AGREEMENT IN ROUND 2

Instance	PAA (%)	Average Difference (%)
The Cleaning Robot	100	22.2
The Driver Robot	88.9	11.1

Framework was proposed regarding consolidated methodologies for teaching with ER, combined with the Anthropological Theory of Didactics and evidence from our previous ER studies [14], [15], [23], [36], [35].

In the CTProgER Framework, an Institution [I] is the space for teaching practice that, when given as input (input), an Object (O) is possible through the experience of didactic moments embedded in the teaching of programming through ER, develop a relation R(X,O) as output. The didactic moments are: problematic, analyzes, discovery, implementation, debug, and assessment.

As far as we know, methodology frameworks related to our proposal do not follow an adequate validation process. We planned to validate our methodology framework through face-to-face studies in a typical Institution [I]. However, due to the COVID-19 pandemic, we could not get field validation yet. In this sense, we validate the CTProgER Framework instances through the experts' validation.

We validated the CTProgER Framework instances through the Delphi method that showed the concordance between instantiating and guidelines through experts' evaluation. The

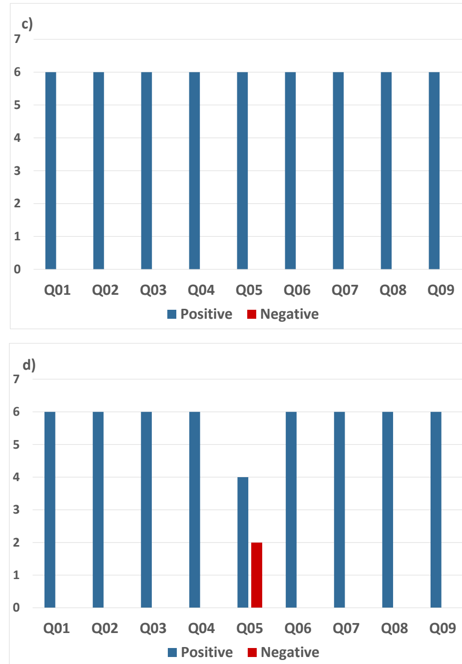

Fig. 4. c) Experts' validation data by version 1 to The Cleaning Robot Instance. d) Experts' validation data by version 1 to The Driver Robot Instance.

instances' validation was the first step to CTProgER Framework validation that allowed the aligning of the concordance of the instances with the CTProgER Framework guidelines. The first round provided necessary data to improve the concordance of our instances with the CTProgER Framework guidelines. This data was used to adjust the validated instances in the second round. The instances' validation process improved our instances and minimized future face-to-face studies problems, where we will test our framework validity in our future studies.

We intend to investigate instruments to observe the R(X,O) relation in future works. Also, we plan to apply the CTProgER Framework and the instances validated in this study in a typical High School to observe its impact on teaching programming. Ultimately, we will investigate the possibility of the CTProgER Framework being extensible to other Institutions [I].

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] C. Angeli and N. Valanides. Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, 105, 2020.

[2] S. Atmatzidou and S. Demetriadis. Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 2016.

[3] 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.

[4] B. Beech. The delphi approach: recent applications in health care. *Nurse Researcher (through 2013)*, 2001.

[5] M. U. Bers, L. Flannery, E. R. Kazakoff, and A. Sullivan. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 2014.

[6] F. Bolger and G. Wright. Improving the delphi process: Lessons from social psychological research. *Technological forecasting and social change*, 2011.

[7] A. Bordini, C. Avila, M. Marques, L. Foss, and S. Cavalheiro. Pensamento computacional nos ensinos fundamental e médio: uma revisão sistemática. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, 2017.

[8] C. Chalmers. Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, 2018.

[9] 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.

[10] 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.

[11] Y. Chevallard, M. Bosch, and J. Gascón. *Estudiar matemáticas: el eslabón perdido entre la enseñanza y el aprendizaje*. Biblioteca del Normalista de la SEP, 1998.

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

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

[14] 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.

[15] 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.

[16] A. L. Delbecq, A. H. Van de Ven, and D. H. Gustafson. *Group techniques for program planning: A guide to nominal group and Delphi processes*. Scott, Foresman,, 1975.

[17] A. C. M. Faro. Técnica delphi na validação das intervenções de enfermagem. *Revista da Escola de Enfermagem da USP*, 1997.

[18] G. Fischertechnik. Fischertechnik. In *The construction kit for children of all ages Fischertechnik*, 2018.

[19] Q. Fu, L.-W. Zhang, J.-C. Hong, and Y. Dong. Development and verification of the programming resilience scale for university students. *Australasian Journal of Educational Technology*, Jul. 2021.

[20] R. A. Giovinazzo. Modelo de aplicação da metodologia delphi pela internet. *Vantagens e Ressalvas*, 2008.

[21] B. Graham, G. Regehr, and J. G. Wright. Delphi as a method to establish consensus for diagnostic criteria. *Journal of clinical epidemiology*, 2003.

[22] D. J. Hardy, A. P. O'Brien, C. J. Gaskin, A. J. O'Brien, E. Morrison-Ngatai, G. Skews, T. Ryan, and N. McNulty. Practical application of the delphi technique in a bicultural mental health nursing study in new zealand. *Journal of advanced nursing*, 2004.

[23] 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.

[24] W. Jeannette M. Computational thinking. *Commun. ACM*, Mar. 2006.

[25] J. Leonard, A. Buss, R. Gamboa, M. Mitchell, O. S. Fashola, T. Hubert, and S. Almughyirah. Using robotics and game design to enhance children's self-efficacy, stem attitudes, and computational thinking skills. *Journal of Science Education and Technology*, 2016.

[26] H. A. Linstone, M. Turoff, et al. *The delphi method*. Addison-Wesley Reading, MA, 1975.

[27] D. Mead and L. Mosely. The use of the delphi as a research approach. *Nurse researcher*, 2001.

[28] J. W. Murry Jr and J. O. Hammons. Delphi: A versatile methodology for conducting qualitative research. *The review of higher education*, 1995.

[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] T. Plomp et al. Educational design research: An introduction. *Educational design research*, 2013.

[31] A. Raabe, A. L. M. Santana, N. Ellery, and F. Gonçalves. Um instrumento para diagnóstico do pensamento computacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 2017.

[32] M. Robotics. Kit de robótica modelix - programa mais educação 2015 - manual metodológico. *Disponível em: https://goo.gl/JL5k99*, 2015.

[33] G. Rowe and G. Wright. Expert opinions in forecasting: the role of the delphi technique. In *Principles of forecasting*. Springer, 2001.

[34] 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.

[35] 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.

[36] 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.

[37] S. E. Stemler. A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. *Practical Assessment, Research, and Evaluation*, 2004.

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

[39] M. Turoff and H. A. Linstone. The delphi method-techniques and applications. *JMR, Journal of Marketing Research*, 2002.

[40] P. L. Williams and C. Webb. The delphi technique: a methodological discussion. *Journal of advanced nursing*, 1994.

[41] J. T. Wright, R. A. Giovinazzo, et al. Delphi: uma ferramenta de apoio ao planejamento prospectivo. *Caderno de pesquisas em administração*, 2000.

[42] 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.

[43] 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.

[44] E. f. L. ZOOM. Zoom education for life. In *Zoom for Education*, 2010.