UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ISABELLE MARIA LIMA DE SOUZA

AN EDUCATIONAL PROCESS FOR TEACHING-LEARNING PROGRAMMING WITH EDUCATIONAL ROBOTICS IN HIGH SCHOOL

CAMPINA GRANDE – PB

2024

ISABELLE MARIA LIMA DE SOUZA

AN EDUCATIONAL PROCESS FOR TEACHING-LEARNING PROGRAMMING WITH EDUCATIONAL ROBOTICS IN HIGH SCHOOL

Tese apresentada ao Programa do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I, pertencente à linha de pesquisa Educação em Ciência da Computação e área de concentração Ciência da Computação, como requisito para obtenção do grau de Doutor em Ciência da Computação.

Orientadores: Prof. Dr. Wilkerson de Lucena Andrade | Prof. Dra. Lívia Maria Rodrigues Sampaio Campos

CAMPINA GRANDE – PB

2024

S729e Souza, Isabelle Maria Lima.
An Educational process for teaching-learning programming with educacional robotics in high school / Isabelle Maria Lima de Souza – Campina Grande, 2024.
178 f. : il. color.

Tese (Doutorado em Ciência da Computação) - Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2024.
"Orientação: Prof. Dr. Wilkerson de Lucena Andrade, Profa. Dra. Lívia Maria Rodrigues Sampaio Campos."
Referências.
1. Teaching-Learning Programming. 2. Computacional Thinking. 3. Educacional Robotics. 4. Metodologia e Ciências da Computação. I. Andrade, Wilkerson de Lucena. II. Campos, Lívia Maria Rodrigues Sampaio. III. Título.

CDU 004.42:37(043)
FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA ITAPUANA SOARES DIAS GONÇALVES CRB-15/093

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO EM CIENCIA DA COMPUTACAO
Rua Aprígio Veloso, 882, Edifício Telmo Silva de Araújo, Bloco CG1, - Bairro Universitário, Campina Grande/PB, CEP 58429-900
Telefone: 2101-1122 - (83) 2101-1123 - (83) 2101-1124
Site: http://computacao.ufcg.edu.br - E-mail: secretaria-copin@computacao.ufcg.edu.br / copin@copin.ufcg.edu.br

FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

ISABELLE MARIA LIMA DE SOUZA

AN EDUCATIONAL PROCESS FOR TEACHING PROGRAMMING WITH EDUCATIONAL ROBOTICS IN HIGH SCHOOL

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação como pré-requisito para obtenção do título de Doutor em Ciência da Computação.

Aprovada em: 05/02/2024

Prof. Dr. WILKERSON DE LUCENA ANDRADE, UFCG, Orientador

Prof. Dr. LÍVIA MARIA RODRIGUES SAMPAIO CAMPOS, UFCG, Orientadora

Prof. Dr. JORGE CESAR ABRANTES DE FIGUEIREDO, UFCG, Examinador Interno

Profa. Dra. ELIANE CRISTINA DE ARAÚJO, UFCG, Examinadora Interna

Prof. Dr. LUIZ MARCOS GARCIA GONCALVES, UFRN, Examinador Externo

Prof. Dr. ROBERTO ALMEIDA BITTENCOURT, UEFS, Examinador Externo

Documento assinado eletronicamente por **WILKERSON DE LUCENA ANDRADE**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 06/02/2024, às 10:21, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **JORGE CESAR ABRANTES DE FIGUEIREDO**, **PROFESSOR 3 GRAU**, em 06/02/2024, às 11:06, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **ELIANE CRISTINA DE ARAUJO**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 06/02/2024, às 14:32, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Roberto Almeida Bittencourt**, **Usuário Externo**, em 06/02/2024, às 15:58, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Luiz Marcos Garcia Gonçalves**, **Usuário Externo**, em 08/02/2024, às 09:15, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **4155407** e o código CRC **96366484**.

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

An Educational Process for Teaching-Learning Programming with Educational Robotics in High School

Isabelle Maria Lima de Souza

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Metodologia e Técnicas da Computação

Wilkerson de Lucena Andrade
(Orientador)
Lívia Maria Rodrigues Sampaio Campos
(Co-orientadora)

Campina Grande, Paraíba, Brasil

Resumo

Este estudo propõe e investiga o impacto de um processo educacional validado para ensino-aprendizagem de programação com Robótica Educacional (RE) no desenvolvimento do Pensamento Computacional (PC) em estudantes do Ensino Médio. A abordagem proposta integra ER na programação de ensino-aprendizagem, com o objetivo de melhorar a eficácia do ensino e o desenvolvimento de habilidades de PC. Enquadrado na Pesquisa em Design Educacional, o estudo aborda três aspectos principais: 1) A facilitação da aquisição de habilidades de PC por meio da programação com ER, 2) A conformidade do processo educacional na perspectiva de um especialista, e 3) O impacto do ensino-aprendizagem de programação com ER nas habilidades de CT dos alunos. A pesquisa culmina em insights sobre o processo educacional proposto, denominado CTProgER. Apoiado na Teoria Antropológica do Didático e validado através do método Delphi e de um estudo de intervenção, o CTProgER integra o Teste de PC de Román-Gonzalez para avaliar o impacto das competências de PC. O CTProgER conceitua seis momentos didáticos e operacionaliza uma Instituição [I] como espaço de ensino, vinculando o Objeto de entrada (O) representativo do conhecimento a ser ensinado aos indivíduos (X) com o desenvolvimento das relações R(X,O). Além disso, instâncias específicas (O Robô Faxineiro, O Robô Contador e O Robô Motorista) foram desenvolvidas em alinhamento com as diretrizes do CTProgER, abordando aspectos de programação e robótica. Um experimento de duas rodadas do método Delphi envolveu especialistas para validar o CTProgER e suas instâncias, refinando estratégias de validação e garantindo o alinhamento com as diretrizes do CTProgER. Finalmente, um estudo de intervenção avaliou a eficácia do CTProgER entre alunos do Ensino Médio Técnico e Profissional em Informática, demonstrando melhorias significativas nas habilidades de PC. A principal contribuição desta tese é o desenvolvimento e disponibilização de um processo educativo validado especificamente concebido para o ensino-aprendizagem de robótica, com foco central na melhoria da PC. Alunos e professores do ensino secundário podem beneficiar da implementação da abordagem CTProgER, promovendo a compreensão da promoção da CT através da programação de ensino-aprendizagem com robótica.

Abstract

This thesis investigates the impact of the educational process for teaching-learning programming with Educational Robotics (ER) on the development of computational thinking (CT) among High School students. Toward this direction, it proposes an approach integrating ER into teaching-learning programming to enhance teaching effectiveness and CT skill development. Framed within Educational Design Research, the study addresses three key aspects: 1) The facilitation of CT skill acquisition through programming with ER, 2) The conformity of the educational process from an expert's perspective, and 3) The impact of teaching-learning programming with ER on students' CT skills. The research culminates in insights into the proposed educational process, named CTProgER. Supported by the Anthropological Theory of the Didactic and validated through the Delphi method and an intervention study, the CTProgER integrates the Román-Gonzalez CT Test to assess CT skills impact. The CTProgER conceptualizes six didactic moments and operationalizes an Institution [I] as the teaching space, linking input Object (O) representing knowledge to be taught to individuals (X) with the development of relations R(X,O). Additionally, specific instances ("The Cleaning Robot", "The Accountant Robot", and "The Driver Robot") were developed in alignment with CTProgER guidelines, addressing programming and robotics aspects. A two-round Delphi method experiment engaged experts to validate CTProgER and its instances, refining strategies for validation and ensuring alignment with CTProgER guidelines. Finally, an intervention study evaluated CTProgER effectiveness among Computer Technical and Vocational High School students, demonstrating significant improvements in CT skills. The main contribution of this thesis is the development and availability of the validated educational process specifically designed for robotics teaching-learning, with a central focus on enhancing CT. High School students and teachers can benefit from implementing the CTProgER approach, fostering an understanding of promoting CT through teaching-learning programming with robotics.

Agradecimentos

Antes de tudo, desejo expressar minha sincera gratidão a Deus por conceder-me a oportunidade de concretizar um sonho pessoal, bem como por me dotar da força e coragem necessárias para superar as adversidades ao longo da jornada desta pesquisa. Embora tenham sido muitas as batalhas travadas, todas foram vencidas.

À minha família, e em especial aos meus dedicados pais, Elza e Josival (Cia), manifesto minha profunda gratidão pela inabalável dedicação à minha educação ao longo dos anos. A presença constante, compreensão, auxílio e torcida pelo meu sucesso são pilares fundamentais. Aos meus irmãos, Igor e Ianne, verdadeiros companheiros em cada passo, e aos meus sobrinhos, Thifany, Miguel e Luna, luzes que iluminam minha vida. Foi pensando no futuro de vocês que diariamente encontrei motivação para prosseguir com este estudo desafiador. Aos meus cunhados, Felipe e Francinete, agradeço do fundo do coraccão por sempre torcerem por mim e se alegrarem com minhas vitórias. Ao meu querido anjo, minha mãe de coração Magna, agradeço profundamente pelos cuidados diários, amor incondicional e apoio constante em minha jornada de crescimento. Sua presença é a maior bênção em minha vida.

Aos meus respeitados orientadores, Wilkerson Andrade e Lívia Sampaio, expresso minha profunda gratidão pela paciência, pelos valiosos ensinamentos e pelos incentivos constantes ao longo de nossa colaboração. Sinto-me privilegiada por ter trilhado o caminho da pós-graduação sob a orientação de líderes tão inspiradores.

Aos meus estimados colegas Ana Liz, Carolina, Elaine, Erick, Jucélio e Raul, expresso minha sincera gratidão pelo apoio, carinho e pela torcida constante, tanto por mim quanto pela minha pesquisa.

À professora Ana Paula, meu sincero agradecimento pelo apoio integral durante a implementação da minha proposta na escola. Seu auxílio foi fundamental para a concretização deste trabalho. A você, dedico minha eterna gratidão.

Aos professores, funcionários e amigos da Universidade Federal de Campina Grande (UFCG/COPIN), e em especial a todos que integram o Laboratório de Práticas de Software (SPLab) e o Laboratório de Sistemas Distribuídos (LSD), expresso meus mais sinceros agradecimentos por toda colaboração e apoio proporcionados ao longo desta jornada.

Contents

List of Symbols

CT - *Computational Thinking*

CTPRogER - *Educational Process for Teaching-Learning Programming with Educational Robotics*

ER - *Educational Robotics*

STEAM - *Science, Technology, Engineering, Arts, and Mathematics*

TV High School - *Technical and Vocational High School*

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides a comprehensive overview, establishing a logical and rational context supported by arguments and scientific reasoning that substantiate the undertaking of the study presented in this thesis. Section 1.1 offers a succinct contextualization, while Section 1.2 delves into the specifics of the research problem. The objectives and research questions are elucidated in Section 1.3, followed by a discussion of the scientific methodology in Section 1.4. The study's contributions are outlined in Section 1.5, and the ethical considerations throughout the research are explicated in Section 1.6. Finally, Section 1.7 presents an overview of the organizational structure of this thesis.

1.1 Contextualization

The social revolution of the last decades has changed how human beings do their tasks. In the 21st century, many jobs no longer exist or have been attributed to technologies generating more complex problems that humans need to solve. Therefore, it is essential to improve the problem-solving skills of the next young generation to respond to new social challenges quickly and efficiently [109].

Computational Thinking (CT) is a modern process to improving problem-solving skills [22]. CT can be defined as a problem-solving process exploring common skills in Computer Science [33; 107; 153]. Besides CT being part of computer scientists' lives, it can also be essential for anyone who wants to adapt to the future and, therefore, should be introduced in school early [193].

The Computer Science Teachers Association (CSTA) has introduced CT in its Model Curriculum for K-12 Computer Science [178] (Basic Education stages in Brazil) to improve students' problem-solving skills. CSTA defends the CT potential for training students in computing education and other sciences in Basic Education stages such as Kindergarten, Elementary School, and High School.

Teaching-learning programming has been explored worldwide to stimulate CT in High School students, becoming no longer an exclusive practice in Computing and Engineering programs [192]. Studies indicate that teaching-learning programming meets difficulties when learning typical concepts [42; 110; 164], which can be a consequence of low self-efficacy for most students [143; 190]. Students have various difficulties in programming, a complex process demanding multiple skills, and can exist several reasons for these difficulties: absence of student preparation, lack of adequate didactics and computational tools that help teachers and students in teaching-learning problems [81].

Research in education in computer science indicates that students often encounter difficulties in programming due to an emphasis on code syntax rather than prioritizing programming logic and CT [39]. This student behavior may be due, in turn, to an insufficient understanding of basic programming concepts and a poor ability to transform programming skills into practice [42]. Some scholars have also suggested improving the education programming environment [164]. Consequently, computational instruments and methodologies guidelines must be considered to facilitate education programming and CT development.

Robotics is one of these computational instruments. In the past decade, robotics has grown as a field that has the prospect of significantly influencing engineering and science education at several educational stages and helping to learn in Science, Mathematics, Technology, Informatics, and other school subjects [5; 125; 40]. Robots became significant in education when Papert [144] introduced the programming language LOGO and the floor turtle, which can execute commands (forward, backward, right, and left) by computer [146].

Educational Robotics (ER) has emerged to define learning environments incorporating scrap materials or assembly kits. These kits consist of motors and sensors that computers and software can control through programming [72]. ER is becoming increasingly popular in classrooms to support education at different educational stages that favor developing teamwork, logical reasoning, and creativity. ER is an instrument praised for fostering essential skills for the 21st century, such as collaboration [84] and creativity [137; 161; 7]. 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 [28; 41; 116]. The ER use in teaching-learning programming allows abstract concepts to be implemented visually and physically to understand Computer Science concepts more easily [139]. Several studies have shown that teaching-learning programming and ER foster students' CT skills [8; 18; 71; 98; 102; 55].

Educational companies commonly propose education processes to encourage Science, Technology, Engineering, Arts, and Mathematics (STEAM) teaching through ER [44; 139; 138]. The STEAM education processes by LEGO® Education, LEGO® ZOOM, Modelix Robotics, Fischer Technik GmbH® are the most popular in the Basic Education stages. Above all, these educational processes are intended to teach STEAM, especially Physics and Robotics concepts, with the assembly and programming of robots.

ER is a tool that can effectively help in teaching programming, contributing to CT development [192]. Providing teaching-learning programming through High School could help beginning programming students understand computer concepts playfully [139]. Concerning this, methodologies are proposed for STEAM teaching through robotics in Basic Education stages [26; 196; 80], as the LEGO® Education, LEGO® ZOOM, Modelix Robotics, and Fischer Technik GmbH®. However, these do not support the computing concepts involved in robotics, such as algorithms, conditionals, and loops. These concepts are taught singly when the robot's programming needs it [65; 66; 103].

1.2 Problem Statement

The central problem of this thesis can be defined as the need to develop an effective educational process to facilitate the teaching and learning of programming with ER, focusing on the development of CT. Specifically, the problem is needing a validated and comprehensive method to instruct students using ER to promote CT skills through teaching-learning programming. This highlights the existing gap in the availability of an efficient educational framework that effectively integrates programming with ER and fosters CT development among students.

Chevalier [44] opposed an operational framework for ER activities that aim to foster two CT skills (computational and creativity) for students from primary school. Nonetheless, the proposed methodology's validation process must be clarified, and the necessary instructional materials for the study replication must be available. In addition, with the operational framework for ER activities, it is impossible to claim a better way to teach programming through ER. There are other proposed frameworks for ER teaching or CT fostering [37; 139]; however, they only contemplate the development of CT through the learning of programming using the ER.

Patiño-Escarcina [148] proposed a new set of methods with guidelines and strategies for applying the educational robotics standard curriculum for kids, named EDUROSC-Kids. The proposed curriculum organizes robotics contents into five disciplines: Robotics and Society, Mechanics, Electronics, Programming, and Control Theory. However, it's worth noting that their focus is not specifically on High School teaching-learning of programming but instead on K-12 education. Additionally, the validation process involved the application with students without including expert analysis or statistical methods to demonstrate its effectiveness in student development.

Loannou and Makridou [102] suggest a growing demand for a practical framework to foster CT through robotics. Such a framework would enable instructional designers and educators to implement CT development consistently and on a large scale. The study of Ching and Hsu [99] underscores the importance of further research that explores the effective design and integration of embodied learning activities that support and enhance the development of computational thinking through robotics.

Therefore, studies that discuss guidelines for implementing ER classes with this objective need ER to be used appropriately for teaching programming and promoting PC skills. Educational programming teaching-learning processes that promote PC skills in high school are necessary. However, designing and validating educational processes of this nature is a challenge because, as far as we know, there are no guidelines for developing and validating an educational process for classes of any nature.

1.3 Research Delimitation

The **thesis** of this research posits that the proposition of a validated educational process for teaching-learning programming with robotics, with a specific focus on the development of CT skills, can significantly optimize the learning of High School students. By adopting this educational process, students are expected to develop strong CT skills, empowering them to tackle complex challenges and promoting a deeper understanding of the fundamental principles of programming. This thesis is based on the premise that a structured and validated educational process can provide a solid foundation for student's academic and professional development, preparing them for the challenges of the 21st century.

The central **hypothesis** of this thesis is that the proposition and validation of an educational process for teaching-learning programming with robotics, focused on the development of CT, will result in a significant increase in High School students' ability to understand and apply programming concepts, as well as more substantial development of problem-solving skills inherent to CT skills compared to traditional teaching methods.

The **aim** of this thesis is to enable the teaching-learning of programming with ER focusing on CT through an educational process proposal. Related to this aim, was defined three general **research questions**:

- **RQ1:** How is it possible to teach-learning programming with ER to facilitate students acquiring CT skills?

  This RQ was addressed through a theoretical study involving a literature review and analysis of previous research conducted by the author of this thesis [65; 66; 103]. The investigation began with exploring ER's impacts on the CT skills of High School students. The primary objective of this RQ was to identify practical approaches for teaching-learning programming with ER to enhance students' acquisition of CT skills and to formulate an educational process conducive to this goal. The Anthropological Theory of Didactics [46] served as the foundation for proposing the educational process, drawing insights from prior research findings. This involved integrating educational theories and pedagogical practices in structuring the educational process, named CTProgER. The CTProgER was grounded in the Anthropological Theory of Didactics, which suggests that teaching and learning are complex processes that can be *understood by analyzing different "didactic moments" occurring during the interaction between the teacher, the student, and the content. The Anthropological Theory of Didactics proposes that every human-created entity, known as an Object (O), is associated with at least one individual, denoted as Person (X). Objects can be abstract concepts or tangible items resulting from intentional human actions. This relationship between the Object and the Person, expressed as R(X,O), signifies a dynamic connection that evolves and can be influenced by various social contexts represented by the Institution [I] where the Person is situated. The CTProgER educational process revolves around the notion of an Institution [I] as the locus of teaching practice, where didactic moments in programming through educational robotics lead to the development of a relational output R(X,O), with Object (O) representing tangible outcomes. The CTProgER structure encompasses six didactic moments: Problematic, Analysis, Discovery, Implementation, Validation, and Assessment. These moments provide a framework for planning and conducting programming classes with robotics, focusing on developing computational thinking. Each moment has a specific purpose and can occur in different orders, depending on the needs of the educational process. Finally, the instantiation of CTProgER was carried out for some programming topics used with high school students. This involved creating specific instances of CTProgER that correspond to different programming themes or units, following the structure of the six didactic moments. Each instance was carefully planned to provide an engaging and meaningful learning experience, with practical and theoretical activities aimed at developing computational thinking skills in students. Materials for teachers and students, including instructions, multimedia resources, and practical exercises, accompany the instances. These instances are concrete examples of how CTProgER can be implemented in practice and support educators wishing to adopt this approach in their classrooms.*

- **RQ2:** How much conformity is there in the educational process of teaching-learning programming with ER from an expert's perspective?

  This RQ was settled through an experiment following the Delphi method [136] to validate objects through human judgment. This RQ aimed to determine how realistic and *conformity the CTProgER educational process is considering the human judgment experts (programming teachers) that analyzed the CTProgER instances with its guidelines and its alignment with real programming classes. Expert programming teachers analyzed CTProgER instances to evaluate their adherence to guidelines and alignment with programming classes. The Delphi method demonstrated alignment between instances and guidelines, marking the initial phase of CTProgER validation. Subsequent rounds of validation refined instances based on expert evaluation, enhancing their alignment with CTProgER guidelines.*

- **RQ3:** What is the impact of the educational process of teaching-learning programming with ER on students' CT skills in High School?

  This RQ was settled through an intervention study with computer Technical and Vocational High School students for teaching-learning programming using the CTProgER educational processes proposed in this thesis. This RQ aimed to obtain evidence that helped validate CTProgER effectiveness considering the CT skill students and understand how to learn programming through ER better in the High School context. The overarching goal was to generate empirical evidence that illuminates the influence of teaching and learning programming through CTProgER on the CT skills of High School students specializing in Computer Technical and Vocational studies. Furthermore, the study endeavored to validate CTProgER's effectiveness from the student's perspective, considering student performance in the Román-Gonzalez CT Test. In 3rd grade, the mean difference is that the results highlight that the educational process built to teach programming through ER is capable of helping the student develop his CT skills better because a significant difference was observed between the participants who had contact with the CTProgER. The statistical data presented in this study constitutes a set of procedures that can be used, in a quantitative way, in the process of validating the effectiveness of the educational process.

To achieve the above-cited aim, it defined the following objectives:

- **OBJ1:** Mapping how ER has been used to stimulate CT skills in students;

- **OBJ2:** Identifying ways to design the educational process for teaching-learning programming in High School with ER, focusing on developing CT;

- **OBJ3:** Instantiating the educational process for some programming topics to be used with High School students;
- **OBJ4:** Validating the instance from experts point of view;
- **OBJ5:** Conducting an intervention experiment in High School to apply the educational process and instance together;
- **OBJ6:** Analyzing the educational process's effectiveness to High School students.

1.4 Methodology

This thesis is a Computer Science Education Research that aims to address complex problems in educational practice with no clear and specific literature for solutions.

Any research must follow a formal and systematic procedure that provides the data and security about the information clarified. This formal and systematic procedure is called the scientific method. The scientific method helps the Academic Society trust the results produced and presented in specific research, considering that repeating the procedure under the same conditions would lead to obtaining the same results [85].

There is no unique classification of scientific research in the literature. However, the most traditional categories for the scientific method are: nature, objectives, approach, and method [83]. Table 1.1 shows the research type according to each research category [83; 85].

Scientific research can be classified according to each research category and research type. Concerning the nature category, this thesis is **Applied Research** because it aims to generate knowledge for practical application. Applied Research is used to create knowledge for the solution of specific problems. It aims to search for the truth for a particular practical application in a specific situation [85]. This thesis intends to propose an educational process that helps teach-learning programming with ER focusing on CT and wants to apply the educational process in a High School.

Concerning the objective category, this thesis is **Explanatory Research** because it aims to identify attributes or factors that determine the occurrence of phenomena. Explanatory Research employs the experimental research method and intends to propose and validate

Table 1.1: Scientific Research Rating

Research Category	**Research Type**
Nature	Basic Research
	Applied Research
Objectives	Exploratory
	Descriptive
	Explanatory
	Normative
Approach	Quantitative
	Qualitative
	Combined
Method	Experiment
	Survey
	Modelling and simulation
	Case study
	Action-research
	Soft System Methodology

the educational process that can better help students from High School in teaching-learning programming with ER focusing on CT.

Concerning the approach category, this thesis is **Combined Research** because it aims to blend Qualitative and Quantitative aspects to propose an instrument that can better help students from High School in programming learning [83; 85]. This thesis intends to consider the dynamic relation between the real world and the subject since the natural environment is a direct source for collecting data intuitively. In other words, this thesis needs to observe qualitative aspects related to students from High School in teaching-learning programming with ER ER focusing on CT during contact with the educational process. Moreover, this thesis intends to analyze data related to students' programming and CT performance to observe the impact caused by the educational process, requiring the use of resources and statistical techniques.

Finally, this thesis could be categorized as Experimental, Survey, and Action-research concerning the method category. It could be Experimental Research because it has a study object to clarify that intends to select influence variables and define how to control and observe the effects [83; 85]. This thesis wants to verify if the procedures proposed in the educational process predict teaching-learning programming with ER focusing on CT. It could be Survey Research because it involves immediately questioning people whose behavior one wants to know and analyze [83; 85]. This thesis intends to obtain data from programming learning and CT skills to investigate the impact of the educational process. Lastly, this thesis could be categorized as Action-research because it is empirical research in which participants and researchers work together to solve the problem [83; 85]. This thesis involves the participants through an intervention experiment to observe the educational process's impact on these people. In other words, this thesis wants to foster an involvement between the participants and the researchers to obtain data that indicates the educational process's effectiveness.

Another research method not presented in Table 1.1 but still relevant is the Design research type [68]. The Design research is used to design/develop an intervention (such as programs, teaching-learning strategies, materials, products, and systems) to solve a complicated educational problem and advance our knowledge about the features of these interventions and the procedures to design and develop them [127; 151]. Concerning the method category, this thesis is **Educational Design Research**, a Design-Based Research variant indicated to the educational world.

Educational Design Research "is the systematic study of designing, developing, and evaluating educational interventions (such as programs, teaching-learning strategies, and materials, products, and systems) as solutions for complex problems in educational practice, which also aims at advancing our knowledge about the characteristics of these interventions and the processes of designing and developing them" [151].

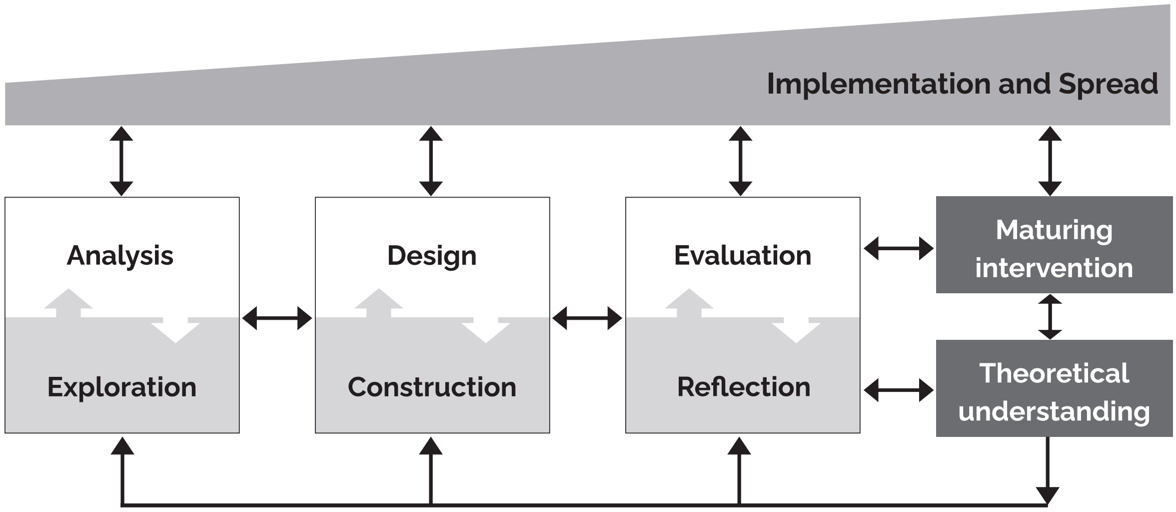


Figure 1.1: A generic model for conducting Educational Design Research, according to McKenney and Reeves (2021)

The Educational Design Research generic model [127] was created based on Educational Design Research models, instruction design, and curriculum development [23; 75; 182]. Figure 1.1 shows the generic model for Educational Design Research with a single, integrated research and development endeavor. It is organized into three main interrelated stages (analysis-exploration, design-construction, and evaluation-reflection) in a flexible process involving interaction with practice and yielding the dual outputs of knowledge and intervention. Educational Design Research results will describe, explain, predict, or manipulate education phenomena that can be spread to the academic society [127].

This thesis desires to develop a systematic and cyclical analysis process, design, development, and evaluation of a teaching-learning programming experience with ER to generate a possible solution to our problem. For this reason, this thesis focuses on research methods with a specialized foundation in educational practices. Therefore, this thesis is **Applied**, **Explanatory**, **Combined**, and **Educational Design Research**. The work process of this thesis is based on the generic model for Educational Design Research conducting (see Figure 1.1) [127].

Figure 1.2 briefly this thesis regarding the generic model for Educational Design Research. Also, there are activities (literature review, field-based investigation, conferences, books, and site visits) to understand how the ER has been used to stimulate CT in High

School, possible existing problems, and obtaining theoretical foundations to help develop this thesis proposal.

This thesis proposal should be designed and built based on the literature and empirical studies during the **design and construction stage**. During the **evaluation and reflection stage**, appropriate strategies were used to generate data associated with the validation process. These stages resulted in a designed and validated version of an educational process and published papers.

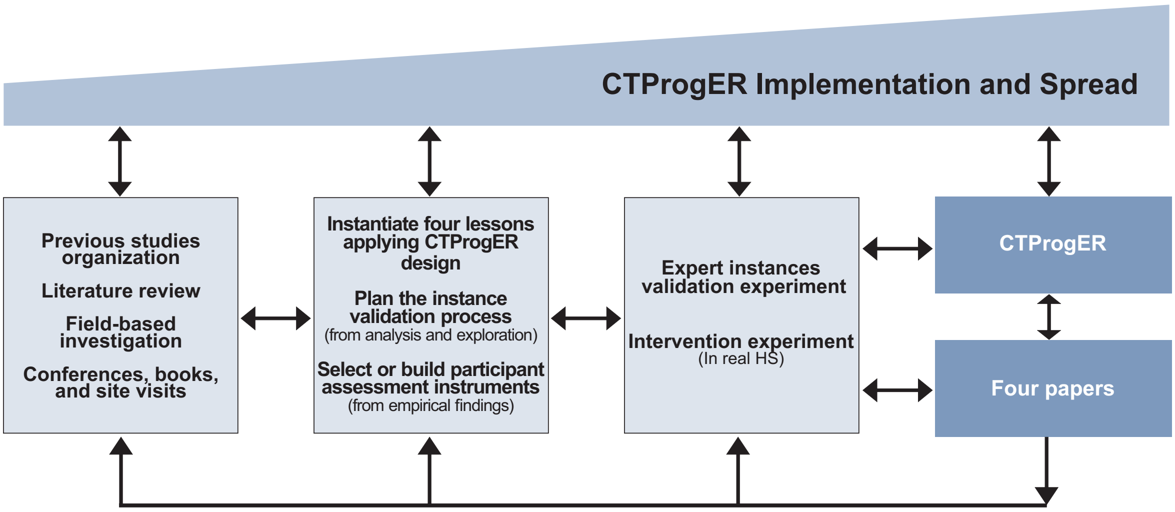


Figure 1.2: Overview of this thesis in light of the generic model for Educational Design Research conducting

Considering Figure 1.2, it is essential to emphasize that the trapezoid is significant. The Educational Design Research of this thesis indicates attention to practical use through the trapezoid, which represents implementation and spread. The trapezoid suggests that the practice is present from the beginning of the process and gradually increasing, not as an afterthought made from observation. The bi-directional arrows in all processes imply that practice impacts the constant core processes, the final outputs, and vice versa. Whereas, by the nature of this thesis, the practical application is essential to improve a methodological instrument that fosters CT skills by teaching programming with ER in High School. This thesis is an educational process, following an in-the-wild approach to understanding the participant's reactions and their respective impact on their learning.

As presented in Section 1.3, this thesis has three research questions and seven objectives.

In light of the generic model for Educational Design Research, this thesis was organized to achieve the objectives and answer the defined research questions. Accordingly, each stage of the educational process model was related to these objectives, as shown in Table 1.4.

Table 1.2: Distribution of objectives according to educational process Educational Design Research model stages

Objectives	Analysis and Exploration	Design and Construction	Evaluation and Reflection
OBJ 1	X		
OBJ 2	X		
OBJ 3	X	X	X
OBJ 4		X	
OBJ 5			X
OBJ 6			X
OBJ 7			X

1.5 Contributions

The main contribution of this thesis lies in the development and availability of a validated educational process tailored explicitly for teaching and learning programming with ER, with a central emphasis on enhancing CT. This process represents an efficient approach to integrating robotics into the academic curriculum, aiming to foster essential CT skills among students. The careful validation of this process confirms its effectiveness and relevance in the educational context, providing a solid and proven educational process to enhance teaching-learning programming with robotics. By specifically targeting the development of CT, this research not only equips students with technical skills in programming and robotics but also empowers them with a crucial analytical and critical mindset to tackle the complex challenges of the 21st century. Furthermore, by providing a validated model for programming teaching-learning with robotics centered on CT, this thesis has the potential to positively influence educational practices worldwide, promoting a more holistic and practical approach to computer science and technology education.

1.6 Ethics Code

Because it involved human beings, this study complied with the procedures established by Resolution 196/96 of the National Health Council. It registered the study proposal on Plataforma Brasil under CAAE 90723918.5.0000.5182. Before starting the activities with the experts and intervention at the school, the participants were clarified about the actions that would be carried out. The School Director read and signed the Consent at school Term, which can be consulted in Appendix A. Likewise, before starting the intervention and data collection, for students under 18 years of age, each legal guardian read and signed the Assent Term, which can be seen in the Appendix B. Students over 18 years of age and judges participating in this study read and signed the Informed Consent Term, which can be seen in Appendix C. To safeguard ethical criteria, it treats all data anonymously. As will be verified, it built the text of this thesis with nominative elements that make it impossible to identify the experts, the school, the classes, and the students, and, in this way, it intended to ensure the information's confidentiality.

1.7 Thesis' Outline

The outline of this thesis document is as follows:

- **Chapter 2** presents the background. It describes the definitions of Computational Thinking, its skills classification, and the approaches to disseminating CT, mainly High School students. Furthermore, it presented the Roman Gonzales CT test used to obtain the student's performance in CT skills. Also, it explains the Educational Robotics to foster CT skills in students from several educational levels. Besides, it describes the Anthropological Theory of Didactics used as a base to propose the educational process. Finally, it explains the Delphi method to validate objects through human judgment (expert) used in educational process validations.

- **Chapter 3** presents the related words. It highlights the main quantity results of the conducted systematic mapping study. Besides, it cites studies about CT fostered through teaching-learning programming and methodological propositions.

- **Chapter 4** presents the educational process, CTProgER, to systematize, guide, and facilitate the teaching-learning of programming with ER focusing on CT that answers the RQ1.

- **Chapter 5** presents the experiment to determine whether the CTProgER is realistic and conformity through the experts' point of view using the Delphi method that answers the RQ2.

- **Chapter 6** presents the intervention experiment to evaluate the CTProgER's effectiveness in developing computer Technical and Vocational High School students' CT skills that answer the RQ3.

- **Chapter 7** presents final considerations on the results of this thesis and the possibilities for future works.

Chapter 2

Background

This chapter presents the fundamental concepts of this thesis. Computational Thinking is discussed in Section 2.1, precisely the definition, the involved abilities, and the approaches that promote CT. Román-Gonzalez' CT Test, an instrument to measure the CT skills used in this study, is explained in Section 2.2. Educational Robotics, the tool for teaching-learning programming, is presented in Section 2.3. The Anthropological Theory of Didactics that supports the CTProgER educational process is detailed in Section 2.4. Finally, the Delphi method is described in Section 2.5.

2.1 Computational Thinking

The CT idea began in 1980 with Seymour Papert [144; 146; 147; 145] studies about constructionist learning that defends developing students' thinking skills through computer science concepts. It gained visibility in 2006 when Jeannette Wing [107] defended it as a "fundamental, not rote skill" for problem-solving in his paper published in Communications of the ACM magazine. In this paper, Jeannette mentioned that CT designates a universally feasible attitude and skill that computer scientists and anyone willing to learn can use [175].

Nowadays, CT has been explored in Computer Science education research, and its definition is still in progress, but there is a trend related to the problem-solving cognitive process. CT idea is related to thinking a computer scientist when facing problems to be solved [107]. However, CT is fundamental not only to computer scientists, but it can be applied in daily life, and is needed to adjust to the future and should be taught at early ages [193]. In reading, writing, and arithmetic, should include CT in the analytical skills of all children [107; 185]. In addition, CT is a composition of engineering concepts of mathematics that help human beings think, giving them better abilities to solve problems. These concepts, fundamental to computer science, encompass mental skills for designing and solving problems that can be observed in typical procedures of computer scientists, such as when they decompose, abstract, use recursion, and schematize algorithms to solve their problems.

Researchers aim to classify these concepts and skills in the literature to delimit a CT formal definition. To Hu [100], CT symbolizes a cognitive process. Consequently, it should be seen as a hybrid paradigm that adjusts different thinking models such as logical, algorithmic, analytic, mathematical, engineering, and creative thinking. According to Aho [4], CT is the set of constructed thoughts organized in the instructions for solving problems. In this sense, the computation is a process of a formal model. At the same time, CT is a mental process applied in this model's formalization, determining CT as the primary provider of problem-solving ability.

Bar and Stephenson [24] define CT as a problem-solving skill in such a way that posteriors can perform this skill by a computer. In this way, technology is a means of learning by doing, changing recipient students into technology and knowledge builders through the CT foster.

There are several other CT definitions. It is interesting to note that many converge using the skills and problem-solving words in their definition. Some researchers classify CT into several skills [24; 53; 185]. The most classification is proposed by Bar and Stephenson [24], such as:

- **Data collection:** identifying a data source for an issue;
- **Data analysis:** writing a program to perform basic statistical calculations;
- **Data representation:** using data structures such as an array, list, etc.;
- **Decomposition:** defining objects and methods, define main functions;
- **Abstraction:** using of procedures to encapsulate a repeating set of functions;
- **Algorithms and Procedures:** studying classical algorithms and running an algorithm for a problem;

- **Automation**: creating automated solutions using logic, algorithms, and programming;
- **Parallelism:** splitting data or tasks to be processed in parallel;
- **Simulation:** animating of an algorithm, scanning parameter.

That way, many researchers propose studies aimed at fostering CT skills or some of its skills. Some of these studies indicate incorporating CT into the curriculum at all educational levels, from kindergarten to university [134; 181; 188]. In partnership with the National Science Foundation and the Computer Science Teachers Association in 2011, the International Society for Technology in Education launched the Model Curriculum for K-12 Computer Science [178]. This partnership resulted in an operational definition to insert the teaching of computing in schools, specifying in their guidelines the CT category by Barr and Stephenson [24]. The working definition helped the scientific community to recognize the computer science potential for school students. In this sense, the CT introduction from the first school years can stimulate the youngsters' problem-solving skills.

Several researchers introduce CT in High School surrounding computing, programming, and the use of technology. Specifically, teaching-learning programming has been explored worldwide to stimulate CT in High School students, becoming no longer an exclusive practice in Computer and Engineering courses [192]. Some researchers used visual language like Scratch to work with computational concepts, procedures, and perspectives [37]. Other researchers used script language like python [1] for programming and coding, which is helpful for complex real-world and scientific problems. Also, others defended unplugged programming as a hands-on and practical approach to teaching and learning that emphasizes embodied and distributed cognition [12].

The scientific community has contributed to CT's democratization in education. Whether theoretical, practical, and proposals for tools and technologies, many initiatives strengthen the discussion on the importance of CT in developing essential problem-solving skills for young students. These skills can be worked on in education in any area, not exclusive to Computer Science. Particularly in High School, the CT introduction in daily educational practices has been explored as concepts, skills, and practices that can be applied both in everyday situations and in science, diluting these skills in living. As well as reading, writing, and calculating skills, as endorsed by Wing [185; 107].

2.2 Román-Gonzalez' CT Test

The Román-Gonzalez' CT Test is an assessment instrument proposed by Román-Gonzalez *et al* in 2015 [87; 159] that tries to identify the ability to form and solve problems based on the fundamental concepts of Computing in an unplugged approach. The Román-Gonzalez' CT Test was created based on a quantitative and aptitudinal approach. In addition, it was subjected to a rigorous validation process, which proved its effectiveness in three aspects, namely: content validity [159], the validity of the criteria [160] and convergent validity [160]. Psychometric studies conducted with the Román-Gonzalez' CT Test demonstrated that the test is reliable ($\alpha < .80$) and compatible for assessing CT in students between 10 and 16 years old [159].

The Román-Gonzalez' CT Test consists of 28 multiple-choice questions, each with four answer alternatives (A, B, C, and D), of which only one is correct. The questions use logical syntaxes derived from programming languages and comprise concepts from the following CT skills: abstraction, decomposition, pattern recognition, and algorithms [36].

The Román-Gonzalez' CT Test is organized into three segments. The first focuses on using direction arrows (right, left, up, and down) to move the character. The second is based on movement relative to the character's position and direction using block codes, and the last uses pencils to make drawings with block codes.

From the beginning of the Román-Gonzalez' CT Test, students have up to 45 minutes to do their best. He doesn't need to answer all the questions. However, the more questions answered, the greater the possibility of obtaining better scores. Before starting the Román-Gonzalez' CT Test, there are 3 example questions so that the student becomes familiar with the type of questions they will encounter, in which the characters involved throughout the 28 questions will appear.

Pac-Man is the character that always moves along the marked path towards the Ghost character, i.e., the beginning is always Pac-Man, and the end is Ghost. In turn, the Artist is the one who will make drawings with the pencil. The characters can be seen in Figure 2.1.

In the first example (see Figure 2.2), it is asked which commands take 'Pac-Man' to the Ghost along the indicated path. In other words, take 'Pac-Man' precisely to the box where the spirit is (without passing or stopping), and strictly follow the path marked in yellow (without

Figure 2.1: Pac-Man, Ghost and Artist characters from the Román-Gonzalez' CT Test

leaving or touching the walls, represented by the orange squares). The correct alternative in this example is B.

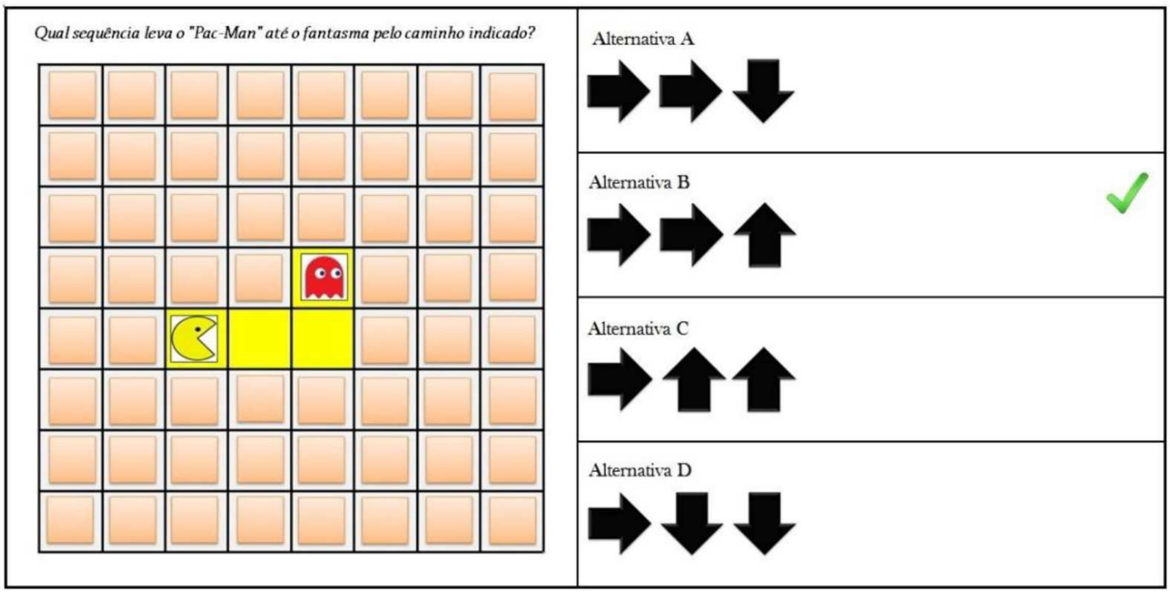


Figure 2.2: A generic model for conducting Educational Design Research, according to McKenney and Reeves (2021)

In the second example (see Figure 2.3), it is asked which commands take 'Pac-Man' to the Ghost along the marked path. But in this case, the answer options use blocks that fit together instead of arrows. Remember that the question asks you to take 'Pac-Man' EXACTLY to the house where the Ghost is located (without passing or stopping) and strictly follow the path marked in yellow (without leaving or touching the walls, represented by the orange squares). The correct alternative in this example is C.

In the third example (see Figure 2.4), it is asked which commands the Artist should follow to draw the figure that appears on the screen. In other words, you should MOVE the pencil to draw the figure. The MOVE command pushes the pencil while drawing, while the

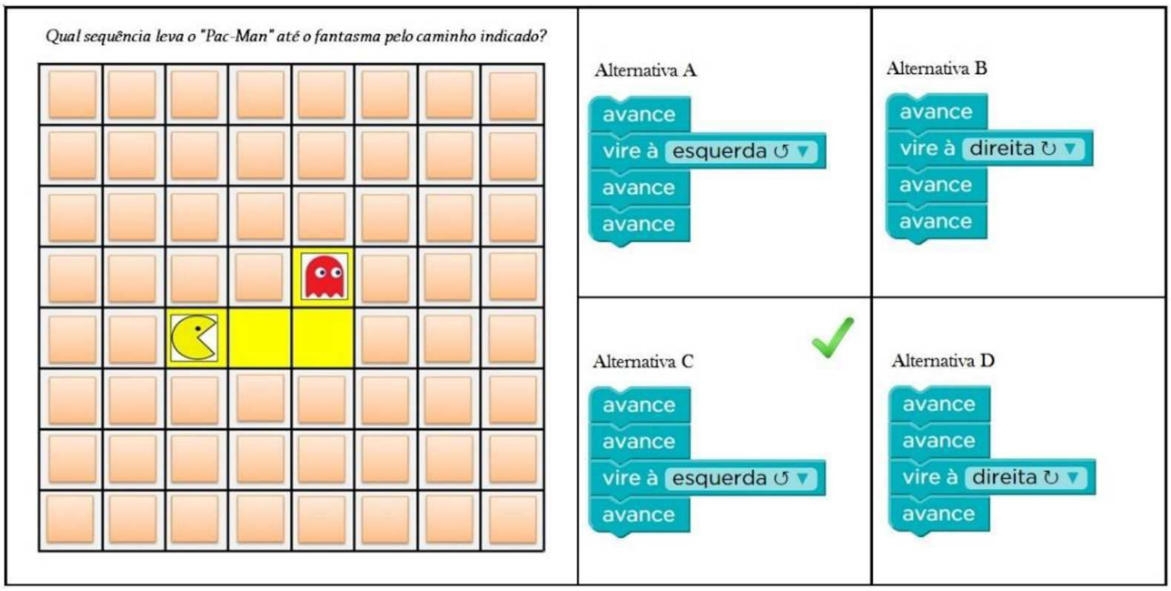


Figure 2.3: A generic model for conducting Educational Design Research, according to McKenney and Reeves (2021)

SKIP command makes the Artist stand up without drawing. The gray arrow indicates the direction of the pen's first movement. The correct alternative in this example is A.

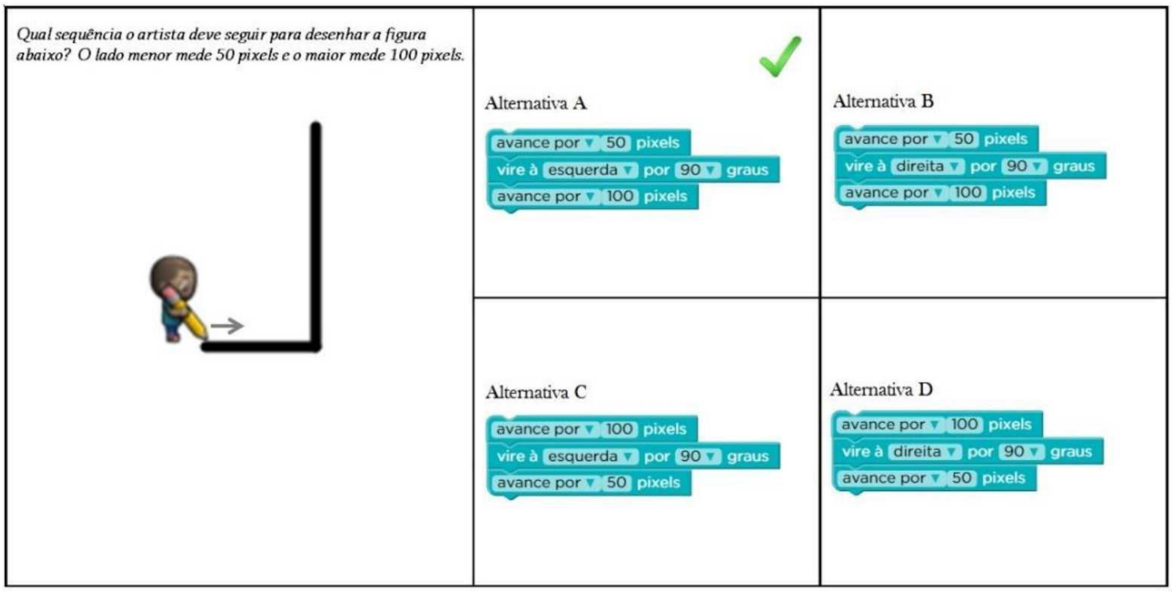


Figure 2.4: A generic model for conducting Educational Design Research, according to McKenney and Reeves (2021)

The original version of the Román-Gonzalez' CT Test was written in Spanish (European) and translated into Portuguese (Brazil) by Rafael Marimon Boucinha and Christian Puhlmann Brackmann with authorization from the creator. The Román-Gonzalez' CT Test underwent a review and validation process conducted by a researcher from the Federal University of Rio Grande do Norte (UFRN). However, it's important to note that the specific details of this process have not yet been published. It is awaiting the release of this information to provide a more comprehensive insight into the validation and the results obtained with Román-Gonzalez' CT Test.

The Román-Gonzalez' CT Test can be applied in any browser (e.g., Chrome, Firefox, Edge) and can be accessed virtually from any device (e.g., computer, tablet, smartphone). However, as an instrument that can be applied unplugged, it was also created in physical format, printed on paper. Rafael and Christian converted all questions in the browser to a printed document, and an Answer Sheet was also developed so that students could mark the correct option. The Román-Gonzalez' CT Test and answer sheet used in this thesis can be viewed in Appendices D and E.

2.3 Educational Robotics

Robotics is a science that involves concepts from Computing, Engineering, Physics, Neuroscience, and Artificial Intelligence, intending to create and use [70] robots. Robotics is the study of robots' ability to feel and act in the physical world autonomously and intentionally [122]. Reports indicate that Isaac Asimov foreshadowed the word robotics, a science fiction writer inspired by the science fiction play Rossumovi Univerzální Roboti written by Karel Capek in 1920, which introduced the term “robot" in various languages and science fiction.

Nowadays, traditional teaching methodologies focusing on the teacher as a transmitter of knowledge make up complex reflections since they place the student as a simple decorator. These reflections highlight the need for new teaching and learning practices that allow students to win and maintain knowledge, actions that require an active posture from them [155]. The student must perceive himself as the protagonist of his learning. This need justifies educational technologies such as robotics applications, games, and materials.

RE was established with the beginning of the demand for practices and tools for teaching and learning. The ER use is becoming increasingly popular in classrooms to foster 21st-century skills such as creativity [73; 137; 161]. However, the idea of using robots in classrooms has since Seymour Papert first promoted the belief in the late 1970s [146].

As an ascension of Papert's LOGO [144], ER refers to the discipline that integrates robotics concepts and education, aiming to promote learning through the construction, programming, and interaction with robots. Its main objective is to provide students with a practical and engaging experience that stimulates the development of skills such as computational thinking, problem-solving, teamwork, and creativity while exploring STEAM concepts [156]. It is still possible to define ER as the construction of the mechanism that a computer can control for an educational purpose [180]. These concepts allow us to consider the ER as an instrument for learning STEAM subjects and Computer Science concepts.

ER research tries to understand how relations with robots can foster and support learning in humans (from young children to adults) [6; 194]. The ER strategies can be categorized into two goals: 1) Learning about Robots (Education in Robotics or Robotics as a Science); 2) Learning with Robots (Robotics for Education) [61]. Also, the ER could be offered as intra-curricular or extra-curricular, according to the educational proposed goal [135]. Intra-curricular activities are included in the school curriculum as an official syllabus. In this case, all student activities can be done by ER. Extra-curricular activities are provided after school hours and are not considered in the school curriculum. These extra-curricular activities are usually proved through experiments from several researchers or training courses that do not impact a considerable student number.

Through ER, promote student interaction with robots for some educational purposes. When considering Basic Education, ER's use involves several purposes: teaching programming, teaching robotics as a science, interdisciplinary science teaching, CT development, participation in tournaments, and Olympics [171]; that are offered in an intra-curricular or extracurricular way. Although the purpose of educational practices with ER is diverse, it is possible to identify a practical nature involving problem situations, whether to teach curriculum science or programming logic [171]. The student is engaged in "learning by doing", commonly called hands-on. In this context, students are meant to instruct the robot to perform a distinct task. Regardless of the purpose and learning strategies, there is a trend to plan, design, or implement an algorithm to control the robot's behavior to perform a specific task [44]. This idea of the plan, design, or implement an algorithm to control the robot's behavior is directly related to solving problems from the CT Skills, such as abstraction, decomposition, simulations, data collection, and analysis [150].

In literature about this idea, there is a consensus that ER can improve several CT-related skills. According to Zilli [195], the ER can foster in students the following skills:

- Logical reasoning;
- Manual and aesthetic skills;
- Interpersonal and interpersonal relationships;
- Application of theoretical concepts in project development;
- Research and understanding;
- Representation and communication;
- Conducting research;
- Problem-solving considering errors and successes;
- Application of theories in concrete ways;
- Creativity;
- Critically.

This thesis considered ER as Robotics for Education for teaching programming, aiming at developing CT through intra-curricular activities. As defended by Vygotsky [118], learning is a dynamic action based on human experience with the real world and social life. In this sense, considering studies by literature, this study believed that in an educational space with RE, students work collaboratively to understand problems and plan, implement, and present their robotics solutions [132; 60]. In this conjecture, teacher and student play different roles, which are far from the parts commonly observed in traditional teaching methodologies, which drive the engagement of students and teachers with the learning process. Skinner *et al.* [76] refer to engagement as the degree of student connection or involvement with their school effort and, consequently, with people, activities, goals, values, and environments.

Therefore, it is understood that ER can favor teaching-learning programming with ER, and this study used ER as a teaching tool.

2.4 Anthropological Theory of Didactics

The Anthropological Theory of Didactics is a program of research in Mathematics education formed by Yves Chevallard in the 1980s studying didactic transposition processes [35; 46; 48; 47; 49]. For the Anthropological Theory of Didactics, Mathematics in its various dimensions is about human activities. The Anthropological Theory of Didactics comprises a theoretical framework that allows analyzing the teaching and learning scenario both from an educational perspective and the point of view of social organizations and their multiple realities or situations explored in the contextualization.

Bosch *et al.* [34] present and defend the use and application of this theory outside the specific domain of Mathematics, which Nunes [167] carried out in the context of teaching Physics, a science of an empirical nature. In the context of programming, Mathematical reasoning is easily observed, in which problems are solved through logic and well-formulated rules, which aligns with the general purpose of the applicability of the Anthropological Theory of Didactics, a theory designed for the specific teaching of Math objects. However, it is necessary to consider the other epistemological specificity of programming, as it is an object with empirical, i.e., an object exposed to the subject's experience that can offer different knowledge.

In software development, it can identify two types of problem complexity: essential and accidental. Essential complexity is related to transforming real problems into coding that interacts with complex environments in a way that can be easily modified. In contrast, accidental complexity is related to the state of the tools and processes that are used in coding [38]. Both complexities are closely related to the programming action in that no single solution to a problem, method, technique, or tool leads to the final objective. However, the subject's experience in its empirical origin can influence and shorten this path.

Acknowledging that a learner's experiences are inherently tied to human activities, it becomes evident that fostering these experiences can be achieved by leveraging educational tools that place the human aspect at the core of the learning process. Educators can enhance the teaching-learning programming journey by utilizing educational strategies prioritizing the human element, recognizing the learner's individuality and unique perspective. This approach aligns with the principles of the Anthropological Theory of Didactics. It emphasizes the importance of tailoring educational methods to the human experience, ultimately enriching the understanding and application of programming concepts. In this context, educational tools and methodologies that embrace the human-centric approach can be pivotal in creating a more engaging and effective learning environment for aspiring programmers.

Hence, by considering all fields of human activity, the Anthropological Theory of Didactics can be found in programming as an area of knowledge, an opportunity to prospect tasks and theories that justify its applicability in teaching-learning programming, both by enabling the development of Mathematical reasoning as well by involving the subject in social organizations and their multiple realities or situations explored in contextualization, an empirical case. In this theory, the object is a human activity resulting from knowledge. This knowledge can be known, taught, or taught as long as different technical tasks and technologies are applied to work with them [191]. The Anthropological Theory of Didactics comprises structural (praxeologies) and functional (didactic moments).

It is important to emphasize that the Anthropological Theory of Didactics and the concept of Object-Oriented Programming represent distinct approaches in education and computing, differing primarily in conceiving and applying the term “object".While Anthropological Theory of Didactics focuses on the analysis of didactic processes and the understanding of the object of knowledge, Object-Oriented Programming, as a programming paradigm, employs the term “object” to represent concrete instances of entities in the code [157].

In the Anthropological Theory of Didactics, the object is regarded as a human activity resulting from knowledge. According to Chevallard, the object of knowledge can be known, taught, or learned, and its understanding is built through the application of various technical tasks and technologies. The Anthropological Theory of Didactics focuses on praxeology (structural elements) and didactic moments (functional components) that constitute learning and teaching dynamics, transcending the physical representation of objects. On the other hand, in Object-Oriented Programming, an “object” refers to a specific instance of a class, representing a real-world entity in the programming code. In this paradigm, objects have properties (attributes) and behaviors (methods), and the interaction between these objects is crucial for designing modular and reusable software systems.

Therefore, the main difference lies in the "object". In the Anthropological Theory of Didactics, the object is an abstract construction representing a human activity related to knowledge, while in Object-Oriented Programming, the object is a concrete entity encapsulating data and functionalities in a software system. While the Anthropological Theory of Didactics focuses on learning and teaching, Object-Oriented Programming is a practical approach to organizing and structuring code in software development.

2.4.1 Praxeology

The Anthropological Theory of Didactics presupposes that any activity associated with the production, distribution, or acquisition of knowledge should be understood as a human activity and thus offers a general model of human activities built on the praxeology idea. In other words, praxeology is the theory of human action based on the notion that humans engage in intended behavior instead of reflexive behavior and other unintentional behavior. According to Chevallard [50]:

> "A praxeology is, in some way, the basic unit into which one can analyse human action at large. [...] What exactly is a praxeology? We can rely on etymology to guide us here - one can analyse any human doing into two main, interrelated components: praxis, i.e. the practical part, on the one hand, and logos, on the other hand. "Logos" is a Greek word which, from pre-Socratic times, has been used steadily to refer to human thinking and reasoning - particularly about the cosmos. [...] [According to] one fundamental principle of the ATD - the Anthropological Theory of Didactics - no human action can exist without being, at least partially, "explained", made "intelligible", "justified", "accounted for", in whatever style of "reasoning" such an explanation or justification may be cast. Praxis thus entails logos, which, in turn, backs up praxis. For praxis needs support just because, in the long run, no human doing goes unquestioned. Of course, a praxeology may be a bad one, with its "praxis" part being made of an inefficient technique - "technique" is here the official word for a "way of doing" - and its "logos" component consisting almost entirely of sheer nonsense - at least from

the praxeologist's point of view!"

In the Anthropological Theory of Didactics, a Praxeology, also known as Praxeological Organization (OP), is represented by the four-tuple [T, τ, θ, Θ], where:

- T: type of task that can be subdivided into several tasks [t];
- τ: a technique used to task perform;
- θ: a technology applied to task perform;
- Θ: a theory involved in task performance.

According to Chevallard *et al.* [51], a Praxeological Organization is constituted by a practice block (practical-technical) represented by the two-tuple [T, τ] referring to "knowhow" and a theory block (technological-theoretical) one [θ, Θ] related to "a knowledge."

Type of task [T] is everything a person requests through a verb. It refers to soliciting someone to act. When a task [t] falls into a type of task [T], it is possible to consider that $t \in T$ (t belongs to T). A task has a parent Task Type [T] when a verb is used, e.g., clean the house, compute the quadratic equation roots, greet a neighbor, and read a book. It is necessary to emphasize that the task idea is a relatively precise sense. Cleaning the house is one type of task, but just cleaning is not one; computing the quadratic equation roots is a task, but just calculating is not. Thus, it is observed that the task idea is related to a type of task that requires a determinant to be considered a type of task.

Technique [τ] is related to the way to solve a task [T] or the set of subtasks ($t \in T$) necessary for its resolution. The technique is related to "know-how" and has meaning only when it is connected to a task [T].

To understand the technique concept, one must remember that human actions will be physically performed in a school, university, or classroom. This physical location is called an **Institution** [I]. When a type of task [T] is constituted in an Institution [I], it is possible to establish one or more techniques [τ] that can be accepted and recognized institutionally. This technique has a direct relationship with the subjects of this Institution [I], that is, with the individuals of the Institution [I] involved in solving a specific task. The techniques or set of them may not necessarily exist or be accepted by other Institutions and subjects.

In this context, performing a task or type of task requires the application of a specific technique, which is understood as the practical-technical block [T, τ] of praxeology. Some human action is understood as encouraged by a task that the activity is carried out to achieve; in other words, every human action is considered intentional or motivated. What it does to carry out the task is called a technique. The fact that tasks originate in types amounts to nothing else than the ordinary and crucial human experience of being able to use the same technique for a whole collection of similar tasks; inversely, the technique is recognized and defined by the type of tasks to which it applies. The two-tuple [T, τ] is called a practice block because it is the most fundamental way to define an organized practice. In brief, the most elementary form of learning can be represented as the perception of tasks related by a standard technique [186].

Technology [θ] into the Anthropological Theory of Didactics moves away from common sense technology. According to Chevallard *et al.* [51] a technology formalizes the tripod “Justification - Explanation - Production of new techniques". The technology aims to ensure that a task or set of tasks can be accomplished. Technology is responsible for making the technique understandable in presenting the correct technique for the context. Finally, new techniques must offer new ways of solving the task, considering pre-existing technologies related to few or no techniques.

Finally, **Theory** [Θ] refers to a more elevated level of “Justification - Explanation - Production", presenting an explanation and justification of the technology related to the technology applied in the task context. The theory justifies this and explains the technology placed.

Technology and theory determine the technological-theoretical block [θ, Θ] directly related to the worked knowledge. One of the characteristics of human routine and learning, not least in school contexts, is that techniques are often subject to explanations, reasoning, and other forms of discourse, termed technology. Ultimately, the technology may be explained and justified, and this super discourse about the technology is called a theory [186].

2.4.2 The Personal Relation of a Person (X) and an Object (O)

The Object (O) idea is a particular aspect of the Anthropological Theory of Didactics that can exist by at least one Person (X). Objects can be abstract, such as equations, ideas, concepts, poems, etc., or concrete, such as tools, physical structures, plants, people, etc. Every work or product resulting from intentional human action is an Object [34].

While an Object (O) comes into existence for a Person (X), it is stated that a relation between the Object and the Person, defined by R(X,O) has been established. In other words, the personal relations R(X,O) is not empty, that is, R(X,O) $\neq \emptyset$. This relation is a dynamic action that, over time, comes into existence, discontinues to exist, or changes. This relationship R(X,O) is a human action originating. Thus, its change can be caused by multiple social spaces, represented in the Anthropological Theory of Didactics by Institution [I], where the Person is inserted.

With this relation concept in mind, it is possible to observe the educational places (schools, universities, classrooms) as an Institution [I], the student as a Person (X), and the knowledge established by science and defined for learning as Object (O). Thus, within the educational process, it is possible to promote relations between students and knowledge as presented in the personal relation idea and broaden the understanding and perception of a given Object, consequently encouraging new relations with other objects, which is called learning.

2.4.3 Didactic Organizations and their Didactic Moments

The Anthropological Theory of Didactics considers that the Praxeological Organization can be applied to any human activity and not exclusively to Mathematics. In particular, the Praxeological Organization can describe a teacher and student practice in didactic praxeologies or didactic organizations.

This particular didactic organization is a set of tasks, techniques, technologies, and theories mobilized to study specific knowledge in an Institution. Therefore, these organizations can be called Didactic Praxeological Organizations. In other words, the Didactic Praxeological Organization can be related to the teaching and learning process in a typical educational space since the four-tuple characteristics [T, τ, θ, Θ] are included in the methodological processes.

According to Chevallard [52], constructing a Praxeological Organization does not follow a single chronological way. However, whatever the study path followed, certain situations, whether qualitative or quantitative, are always present, even with particularities. These situations are called by Chevallard as didactic moments since, regardless of the study path, an appropriation of the studied concepts may occur as determined in the Praxeological Organization. Chevallard names this act of appropriation as institutionalizing knowledge, i.e., the act of learning.

The didactic moments are instruments for analyzing the immense Didactic Praxeological Organizations in different media. It is possible to glimpse how the praxeological elements are presented within the Didactic Praxiological Organization. The didactic moments make it possible to understand the contextualization process in teaching a specific Object, whether from a science or a day-to-day human action. In other words, didactic moments are steps to follow in a learning process. The Anthropological Theory of Didactics presents six didactic moments, they are:

1. First Meeting.
2. Moment of Exploration.
3. Constitution and Development of the Technological-Theoretical Block,.
4. Moment of the Technique Work.
5. Moment of Institutionalization.
6. Moment of Evaluation.

It is important to emphasize that these didactic moments do not necessarily have to follow a chronological order for their realization through the Anthropological Theory of Didactics. They can perform simultaneously and at different times, starting in one class and ending in others.

The **First Meeting** refers to the first contact a person has with the Object. This encounter can take place in different ways, and a type of task can be rediscovered as concepts already considered known are rediscovered. In the First Meeting, it is time to explore possibilities related to the Object, ideas already institutionalized, questions, problems, etc., that will serve as initial subsidies that will help institutionalize a new Object, that is, of new knowledge. It is a moment destined to rescue the student's previous knowledge.

The **Moment of Exploration** involves a problem that will support planning a type of task [T] and a technique [τ]. Studying and solving a specific problem involves the constitution of an initial technique so that a technique itself can be developed. Analyzing problems represents a way of creating and forming at least one technique so that it can be applied to problems of a similar nature.

The **Constitution and Development of the Technological-Theoretical Block** involve the first two moments and evolve gradually as the moments become effective. The technological theoretical block aims to base the techniques worked at the Moment of Exploration that respond to the tasks explored at the First Meeting. This moment is the initial step of traditional teaching methodologies, as it involves exploring concepts or theories related to the technique and type of task. Many methodologies start from theoretical work without a previous presentation of a related problem. That is, solutions to at least known problems are first taught.

The **Moment of the Technique Work** applies the technique previously proposed at the Moment of Exploration. The Moment of the Technique Work aims to evolve the technique, making it more efficient and reliable (usually requires retouching the technology developed so far). In addition, this moment can favor updating and the emergence of new techniques and technologies in a justified way. In other words, it is the moment to rescue previous knowledge related to the Object and promote reflection on possible techniques consistent with the problem.

The **Institutionalization Moment** aims to promote evolution from the informal to the formal stage, emphasizing the practical-theoretical and technological-theoretical blocks. It is the moment to smell and institutionalize based on defined knowledge patterns about the Object studied. In other words, it is the moment when the personal institutionalizes the object; that is, it is the moment when the learning takes place.

Finally, the **Moment of Evaluation** deals with understanding the object's institutionalization process. The Moment of Evaluation aims to ascertain the domain or understanding obtained after all didactic moments. In practice, it is a reflection moment (by every individual involved in the learning process), following different criteria, what has been learned, and which didactic moments should be reviewed or improved.

2.5 Delphi Method

Designed by RAND Corporation in the 1950s, the Delphi method assists judgmental prediction and decision-making in various research domains. Delphi was initially devised to help experts (judges) achieve better forecasts than they might through a traditional group meeting [163].

The Delphi method is indicated when a problem can result from cooperative, subjective judgments or decisions and when the expert group does not favor effective communication. This situation is observed when the experts have time differences, distance, and personality conflicts. Physical expert group meetings may be too expensive. However, a prediction effort may benefit from a Delphi process even when face-to-face group meetings are possible [32; 67; 163; 179].

The Delphi method allows for a progressive consensus on a study object. This method occurs through questioning rounds to an expert group, in which the answers are analyzed in search of a consensus among them. The consensus criteria depend on the researcher's study. However, the strictness of the consensus criteria must be respected [184].

The reason to use the Delphi method is due to its capacity to establish consensus on best policy [25]. In addition, the Delphi viability is ideal in areas where consensus is missing, process protocols, and other best methods where agreement is essential [92]. It is helpful when little knowledge or uncertainty surrounds the area being investigated [97; 128].

According to Rowe and Wright [163], the Delphi method can be used under the following conditions:

- When expert judgment is necessary because statistical methods are inappropriate;
- When a number of experts is available;
- When the alternative is to average the forecasts of several individuals;
- When a choice is a traditional group.

2.5.1 The Delphi Investigator

The investigator can be a researcher and is very important to the Delphi method [94]. The investigator's responsibility is to organize the project that will be applied to Delphi, recruit the experts that best fit the project, gather knowledge and opinions from the experts, collect, compose, synthesize, and redistribute their varied responses qualitatively or quantitative [179]. Therefore, the investigator's skill in the Delphi method is as essential as all other aspects of the technique itself. The investigator's performance can promote problems from cognitive heuristics and biases if the investigator gets too involved with the project's desired outcome.

The investigator bias most common within the Delphi method comes from inserting his opinions into the analysis process. Using standardized checklists or rubrics to grade responses in quantitative evaluations into Delphi projects can help counter investigator bias [30]. The Delphi rounds response may also involve the qualitative feedback data from the expert's assessment. In these cases, the investigator bias risk can be more significant. However, the Delphi method has no standardization for qualitative data analyses. Mixing qualitative and standardized quantitative measures may reduce potential bias in the synthesis requirement of the investigator's efforts [94].

2.5.2 The Delphi Experts

A dictionary definition of an expert is someone with special knowledge, skill, or training in something [69]. However, there remains little consensus on who is an expert in the Delphi method. This lack of consensus may not be about who experts are but what attributes they possess. An expert should represent their professional group, with sufficient expertise not to be disputed or the power demanded to discover new knowledge [79].

Experts provide information that can quickly produce an opinion about a study object. They can provide knowledge when traditional research has not been accomplished. This possibility guarantees high content, face-to-face, and concurrent validity [25]. There are several suggestions for expert selection, but the expert groups should be unique for each Delphi project. Rowe and Wright [163] outline the following guides for using expert opinion in the Delphi project:

- Use experts with relevant domain knowledge;
- Use heterogeneous experts;
- Use between 5 and 20 experts;

- For Delphi feedback, provide the mean or median estimate of the panel plus the rationales from all panelists for their assessments;
- Continue Delphi polling until the responses show stability. Generally, two or three structured rounds are enough;
- Obtain the final prediction by weighting all the experts' estimates equally and aggregating them.

2.5.3 The Delphi Design

The Delphi method uses structured knowledge, experience, and expert creativity. If organized correctly, it provides better information than a single expert and some people who do not have specialized expertise in the field [119]. The main characteristics of the Delphi method are the specialist's anonymity, the statistical representation of the distribution of results, and feedback from the expert's responses for re-evaluation in subsequent rounds [187].

The Delphi method should be designed to obtain the expert's reasoning in individual responses through a set of questions to perform the rounds. The questions are related to the study object that the investigator needs to understand. Obtaining information from experts through questions will assist in this understanding. Further information can be gathered to get the best estimates of these factors for eliciting the responding expert assessment confidence [63].

First-round estimates are the most divergent of all rounds [63]. The investigator should compile, edit, and synthesize the expert's responses and return a report with the aggregate of the group's. After obtaining the feedback, the experts are given a chance to reevaluate their initial responses. Each group will determine how many iterations they go through before reaching a stopping point. The round goal is to obtain a strong consensus or a continuing lack of consensus. The latter indicates significant differences between expert's opinions. It takes at least two rounds to characterize processes as a Delphi method. However, only some studies have a round number greater than three rounds [86], as there is a trend to carry out the fourth and later rounds without modifying the expert's opinion.

The round anonymity allows participants (investigator and experts) to avoid socialization that induces bias. When an expert's thinking is made public, the possibility of influencing the thinking of other specialists can be established. The Delphi method allows previous rounds' prior opinions to be reexamined in later rounds, in conjunction with the new aggregate of data and reasoning [63].

2.6 Summary of Chapter

This chapter has elucidated the fundamental concepts essential for comprehending the research undertaken and explained in this thesis document. Initially, it delved into the concepts surrounding CT and its associated competencies. Following that, the CT Test, employed to assess CT skills within the scope of this thesis, was introduced. Subsequently, Educational Robotics was examined as a means to foster CT and, consequently, as a tool capable of facilitating the teaching and learning of programming. Later, the Anthropological Theory of Didactics was expounded upon, serving as the theoretical foundation underpinning the proposed educational process in this thesis. Lastly, the Delphi Method was introduced and chosen for its application in validating the educational process presented herein. The subsequent chapter will explore work related to the research context expounded upon in this presentation.

Chapter 3

Related Work

This chapter presents the outcomes of a Systematic Literature Mapping (SLM) conducted within the research context outlined in this work. In Section 3.1, the objectives guiding the mapping process will be discussed, including the methodology employed for selecting and excluding works and the resulting findings. Furthermore, it will highlight related works that incorporate CT in programming teaching (Section 3.2) and those proposing educational processes (Section 3.2). This chapter is intricately linked to **RQ1** (*How can the teaching and learning of programming with Educational Robotics facilitate students in acquiring CT skills?*) and addresses the following objectives:

- **OBJ1:** Mapping how ER has been used to stimulate CT skills in students.

3.1 Systematic Literature Mapping

This study employs SLM to comprehensively understand theme gaps by categorizing primary studies. The primary goal is to enhance comprehension regarding utilizing ER within the scope of CT. The systematic mapping protocol for Software Engineering, proposed by Petersen *et al.* [149], is adopted to achieve this. The detailed procedures are outlined in the subsequent subsections. This SLM adheres to the CT definition by Wing (2006), which characterizes it as problem-solving skills based on Computer Science concepts and emphasizes the necessity of incorporating CT skills into early-age learning contexts [108].

Initially, the research questions and search strings for databases in the English language context were defined. The databases consulted were Scopus, IEEEXplorer, ScienceDirect, SpringerLink, and ACM. The works published between 2009 and 2020 were used as a reference. Initially, it excluded duplicate and secondary studies, resulting in 3.722 studies. Then, the selection was performed in two steps. In the first step, it read the titles, abstracts, and keywords when available, obtaining a set of 406 studies at the end. In the second step, read the introduction, methodology, and conclusion sections of the studies selected in the first step and apply the inclusion and exclusion criteria. Studies related to the mechatronic industry and medicine learning through robotics have been removed. Ultimately, it selected 38 studies that could answer the RQs, which were read in full.

Souza *et al.* [171] can be consulted for additional details on the review protocol.

3.1.1 Results

This SLM was conducted from July 2019 to October 2020. This section shows the results of the selected studies' analysis. Initially, it will present the general data and answer the proposed research questions.

General Results

The distribution of countries in which the studies were conducted is illustrated in Figure 3.1. The countries that show the highest prevalence in publications are Cyprus (2 works), Brazil (3 works), Spain (5 works), the United States (6 works), and Greece (8 works). Other countries are represented with a single published work.

This outcome is no surprise, as Greece has emerged as a hub for research and innovation in Computational Thinking (CT) and Robotics, driven by a combination of factors that foster this growth. To begin with, Greece boasts a rich tradition in education and research rooted in its historical legacy of Philosophy and Mathematics. This knowledge foundation naturally translates into an increasing interest in applying CT skills to address complex problems. Moreover, the country has fostered collaborations with renowned institutions in the United States, Europe, and other regions worldwide, known for their strong research traditions in these fields. Notable collaborators include institutions like the Massachusetts Institute of Technology (MIT) in the USA, universities in Germany, France, the United Kingdom, Spain, and other prominent centers of robotics and computer science research.

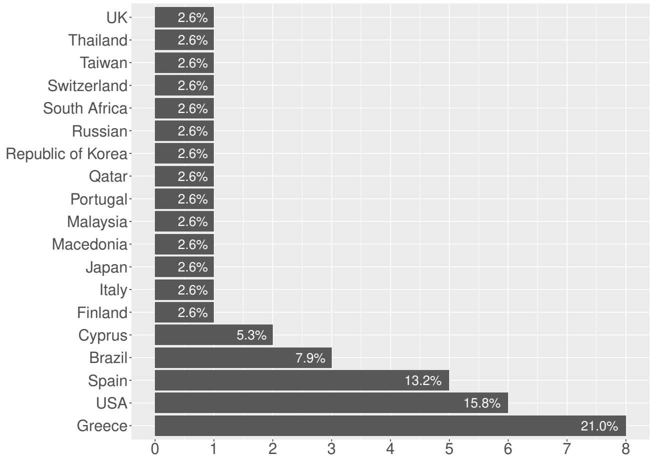


Figure 3.1: Origin of studies authors

This evidence coincides with the USA's second-place ranking. American organizations have played a pivotal role in disseminating strategies for CT skill development and Computer Science, with organizations like the Computer Science Teachers Association (CSTA) [178] leading the way. Furthermore, American institutions, such as the Computer Science Framework - K12 and the Creative Computing Curriculum (CCC), propose the primary curricula discussed in the literature.

On the other hand, the presence of Spain and Brazil in the research and innovation landscape for CT and robotics underscores the growing global significance of these fields and the pursuit of technological solutions to contemporary challenges. Both countries aim to reap the economic and social advantages of these advancing technologies.

Regarding the yearly publication trends of selected works, it can observe the data presented in Figure 3.2. The distribution of these works highlights an increasing interest in CT and ER starting in 2016, with 3 works in 2016, 5 in 2017, 11 in 2018, 11 in 2019, and 4 in 2020. In contrast, there was only one work each in 2010, 2012, 2014, and 2015.

The surge in interest in CT and Robotics ER since 2016 may signal a noteworthy transformation in the scientific and technological landscape. There is a growing emphasis on innovation, interdisciplinary collaboration, and practical implementation. This upswing holds the promise of substantial advancements in these domains and the potential to positively impact various facets of society.

The studies were also categorized based on the research methodology employed. In

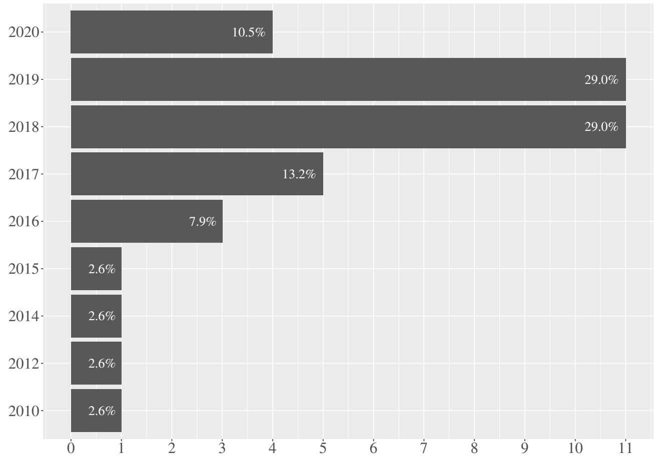


Figure 3.2: The absolute number of studies by year of publication

most cases, the studies showcased experiences related to developing Computational CT skills through ER. Nevertheless, a few were confined to illustrating the applied procedures without incorporating any data analysis, whether qualitative or quantitative. Validation was observed in only a subset of the studies, and this validation typically took the form of a comparative case study. The types of research methodologies identified within the selected studies encompassed action research (1 work), quasi-experiment (4 works), case study (14 works), and experience report (19 works). Figure 3.3 shows the distribution by research methodology employed.

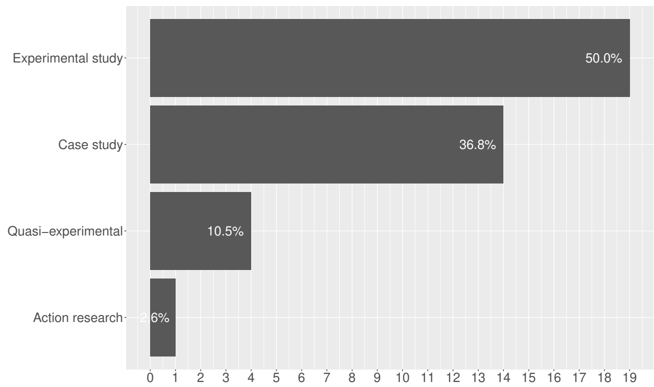


Figure 3.3: Classification of Research Methodology Employed

Educational Strategies for Fostering Computational Thinking Through Educational Robotics

Regarding the educational goals for teaching ER focusing on CT development, the most common objective identified was programming teaching, appearing in 16 works [142; 176; 106; 101; 90; 17; 96; 43; 165; 112; 140; 113; 10; 177; 29; 131], while CT was emphasized in 13 works [101; 90; 59; 89; 21; 19; 45; 16; 133; 104; 9; 20; 54]. Notably, many works that highlighted CT as an objective also incorporated programming instruction or logical reasoning activities, hence categorizing them as CT, in line with the descriptions provided by the authors and the definition of CT by Wing (2011) [185]. Additionally, Robotics teaching was noted in 9 works [31; 27; 130; 129; 91; 19; 165; 112; 104], with 6 works [27; 129; 56; 165; 112; 117] concentrating on Science, Technology, Engineering, and Mathematics (STEM) education. Mathematics [176; 56; 165; 112], computer science [31; 130; 96; 11], and logical reasoning [162; 91; 21; 152] were featured in 4 works each, and algorithmic thinking was addressed in 1 work [126].

Attention was given to the employed educational strategies to gain insight into how these educational goals were implemented in the classroom. The selection of educational strategy was based on the information provided by the authors regarding their didactic methodologies in school and their intended student outcomes.

Collaborative teamwork [162; 27; 176; 106; 56; 59; 89; 17; 21; 96; 19; 43; 112; 117; 45; 16; 140; 133; 104; 20; 11] and problem-solving [130; 106; 56; 17; 21; 19; 43; 165; 112; 117; 45; 16; 140; 133; 104; 126; 9; 20; 54; 131; 11] emerged as the most frequently used strategies, appearing in 21 works. Works mentioning any aspect of teamwork or collaboration were included within this category. Learning by doing, or hands-on learning, was identified in 11 works, while challenges were incorporated into 6 works. Project-based learning was found in 3 works, and multimedia activities were observed in 2 works. Multimedia activities encompassed various media such as paper, pen, images, videos, and online resources. The use of storytelling was noted in 2 works, while interdisciplinary motivation, theoretical classes, and workshops were each reported in just 1 study. It's important to emphasize that a single work may encompass one or more educational strategies.

This evidence indicates that the pedagogical strategy that has emerged as an effective means of promoting CT is teaching-learning programming. This approach offers numerous significant advantages that justify its selection as the primary focus of this thesis. Firstly, instructing programming provides a practical and tangible experience applying CT principles. Students are allowed to create, experiment, and witness the outcomes of their efforts, thereby rendering CT more tangible and pertinent. Furthermore, programming challenges students to address intricate issues, stimulating logical thinking, creativity, and innovation skills pivotal for success and active social engagement.

Secondly, the SLM has revealed that the choice of teaching-learning programming garners substantial support from scientists, educators, and academic communities worldwide. The burgeoning availability of curricula, resources, and educational initiatives that advocate programming as a tool for fostering CT is compelling evidence of this trend. This approach demonstrates its applicability across a broad spectrum of educational settings, from elementary schools to advanced levels of higher education.

Challenges in Classroom Settings for Promoting Computational Thinking Through Educational Robotics

The SLM revealed that a significant number of works (26) do not mention encountering difficulties during ER teaching, while a smaller portion (12) highlight one or more difficulties. The most frequently identified challenge is the limited availability of teaching resources, as observed in four studies [27; 130; 20; 29].

Other difficulties that were identified include issues with programming language (3 works) [31; 130; 101] and limitations in students' prior knowledge (2 works) [130; 59]. Challenges with programming language typically relate to difficulties comprehending syntax and constraints imposed by the available programming resources. Furthermore, a single work identified unique challenges, such as the absence of tools or technologies for evaluating Computational Thinking (CT) [104], a lack of teacher methodology [29], limited usability of robotics kits [27], restrictions on the number of robotics kits available [96], limited functionality of robotics kits [117], low usability of programming language IDE [45], parental non-consent [162], a lack of assembly manuals [27], and issues with frequent student absences [162]

Based on these results, it becomes evident that ER has emerged as a potent tool for fostering CT, creativity, and problem-solving skills among students across various educational levels. Nonetheless, within this research survey, several studies underscore the substantial challenges and limitations educators encounter when integrating robotics into their classrooms. A consensus is becoming increasingly apparent regarding the primary obstacles that impact the effectiveness of these educational sessions.

First and foremost, class time constraints are one of the most prominent challenges. Robotics classes often contend for a limited time within the context of comprehensive, standard curricula, which can curtail the opportunity for in-depth exploration of robotics activities. Consequently, educators may be pressured to expedite topic coverage, potentially impeding students' understanding and appreciation of the intricacies of programming and robotics.

Furthermore, hurdles to proficiency in programming languages' syntax represent another significant challenge. Programming robots typically involves the utilization of specific programming languages, which can be intricate for novice students and educators. Grasping syntax intricacies, debugging code, and addressing errors can be demanding, especially for those without prior programming experience, such as high school students and teachers.

Nevertheless, it is pivotal to underscore that, despite these constraints, ER remains a valuable instrument for nurturing 21st-century skills. The ER approach furnishes a hands-on, immersive learning environment that fosters critical thinking, creativity, and collaboration. Overcoming these obstacles necessitates implementing effective teaching-learning strategies, sustained support for teachers, and adapting curricula to meet students' diverse needs. This evidence underscores the imperative of establishing a dedicated educational process for introducing programming with robotics aimed at stimulating Computational Thinking in High School.

Computational Thinking Skills in Educational Robotics Classes

Regarding the CT skills observed during ER classes, although the studies analyzed are supposed to develop CT through ER, the SLM demonstrated that only half of the studies (19) mentioned developing one or more CT skills during classes. When considering the studies that say the skills, they are mainly related to programming concepts and practices. However, it identified no consensus among researchers about CT skills. Table 3.1 summarizes the CT skills shown in the SLM.

Some authors do not use definitions proposed in the literature and create their own. This SLM classified the CT skills based on descriptions from two works: Barr and Stephenson [24] and CSTA [178]. Studies mentioning computer subjects as CT skills were categorized as Computer Science Concepts.

Table 3.1: CT Skills identify

CT Skills	Study	Frequency
Automation	[140]	1
Simulation	[16]	1
Pattern recognition	[59]	1
Data representation	[54]	1
Parallelism	[130; 129; 165]	3
Data collection	[130; 129; 140; 54]	4
Generalization	[21; 19; 117; 140; 20]	5
Abstraction	[59; 21; 19; 117; 20]	5
Data analysis	[112; 117; 140; 54; 56; 16]	6
Decomposition	[59; 21; 19; 140; 9; 20]	6
Computer Science Concepts	[162; 27; 91; 16; 130; 129; 27; 165; 162; 29]	10
Algorithms and Procedures	[162; 27; 29; 56; 59; 21; 19; 112; 117; 140; 126; 9; 20; 54; 130; 129]	16
Not specified	[31; 142; 176; 106; 101; 90; 89; 17; 96; 43; 45; 113; 10; 133; 177; 104; 152; 131; 11]	19

The evaluation of CT skills can take place at different levels of granularity. High granularity is achieved when the final performance in a test is weighted, whereas fine granularity involves the observation and assessment of individual skills separately [13]. Within the SLM, it was observed that just 50% of the included studies mentioned CT skills, albeit without assessing them individually. In instances where CT was assessed in the context of ER teaching, it predominantly employed a high-granularity approach.

This prevailing practice may be attributed to the inherent complexity of evaluating CT skills in isolation, as many are often interrelated [14]. However, while skill assessment presents a challenge, there is a valuable opportunity to identify and further nurture specific CT skills during ER teaching. Consequently, future studies must investigate which CT skills can be effectively developed and enhanced by integrating ER into educational curricula.

Assessment Tools and Instruments for Evaluating Computational Thinking Skills in Educational Robotics Classes

Regarding the assessment tools and instruments for evaluating CT skills in ER classes, the SLM revealed no standardized research instrument commonly utilized for CT assessment in ER classes. The most frequently identified methods were tests and questionnaires that involved logical reasoning, created by the authors (20 works). Additionally, semi-structured assessments (17 works), qualitative observational evaluations (17 works), and the Bebras Challenge (3 works) were employed. Other assessment instruments were mentioned in only one study. Tables 3.2 provide an overview of the tools and instruments for CT evaluation outlined in this SLM.

The SLM categorized the Bebras, Román-Gonzalez's CT Test, Lippman tests, and Brennan's and Resnick's frameworks separately because these were utilized by institutions and researchers beyond the original creators' contexts. Furthermore, these tests consider methodological and validation aspects. It is important to note that there is no universally agreed-upon approach to assessing CT skills. The absence of widely accepted instruments and tools is evident, as several authors propose unique assessment methods. Therefore, this SLM categorizes the tools and instruments employed to evaluate CT skills based on the approaches mentioned by the authors in their respective studies.

Table 3.2: Tools or instruments CT evaluate

Evaluate tools	Study	Frequency
Brennan's and Resnick's framework	[129]	1
Lippman test	[126]	1
Román-Gonzalez' CT Test	[59]	1
Scoring protocol	[29; 54]	1
Academic performance	[142; 165; 54]	1
Bebras Challenge	[140; 104; 54]	3
Qualitative observational assessment	[27; 91; 56; 101; 89; 17; 21; 19; 43; 112; 45; 16; 9; 29; 54; 11]	17
Semi-Structured Interviews	[162; 31; 27; 142; 130; 91; 56; 90; 89; 21; 19; 113; 10; 133; 20; 152; 11]	17
Author's test	[162; 31; 27; 130; 91; 176; 106; 56; 101; 89; 21; 96; 19; 165; 117; 177; 9; 20; 131; 129]	20

These findings reveal no standardized instrument for assessing Computational Thinking (CT) in students during Educational Robotics (ER) classes. It's common for authors of these studies to propose new assessment instruments. Among the 32 studies, only 6.2% (2) utilized tools referenced in the literature and supported by the scientific community.

The first referenced instrument is Bebras [1], which, despite featuring questions related to CT skills, does not necessitate a deep understanding of computing concepts for successful completion [62]. The second is the Computational Thinking Test by Román-Gonzalez [87; 88], which follows a similar approach to Bebras. Both these instruments provide application guidelines and offer participants scoring criteria. Nevertheless, it's worth noting that, to knowledge, there is no scientific evidence to suggest that Bebras can thoroughly assess CT and its associated skills [15]. On the other hand, the Román-Gonzalez CT Test was specifically developed and validated for this purpose, making it a recommended instrument for measuring students' CT skill performance in ER classes [87; 88].

Furthermore, it's essential to acknowledge that the assessment mechanism SLM proposed by authors in these studies may not genuinely "measure" CT in students during ER classes. The complexity of evaluating CT skills is a subject of ongoing research and development in the field.

The Audience Most Common in Educational Robotics Classes to Foster Computational Thinking

Regarding characterizing the profiles of students who participated in Educational Robotics (ER) classes, only 58% (22) of the studies mentioned the students' ages. Table 3.3 summarizes the age ranges identified in these works. Among the 32 works considered, 5% (2) did not specify the educational levels to which their research was applied.

Identifying the age of students is a relevant factor in understanding how ER is applied in different age groups. This data can provide insights into the appropriateness of pedagogical approaches, the types of robotics used, and the results achieved in specific age groups. For example, teaching strategies and proposed challenges can vary considerably between elementary and high school students due to differences in cognitive abilities and developmental levels.

The educational levels that have seen the most substantial number of initiatives aimed at fostering CT skills through ER are primarily middle schools, as evident in 13 works conducted across various countries, including Brazil ([104]), Japan ([142]), Macedonia ([113]),

[1] Bebras: https://www.bebras.org

Table 3.3: Age range

Study	Age range	Study	Age range
[31; 152]	From 10 to 12	[112]	From 8 to 13
[130; 129]	From 14 to 15	[45]	From 9 to 10
[176; 177]	From 14 to 17	[133]	From 6 to 10
[19; 20]	From 15 to 18	[104]	From 14 to 28
[162]	From 4.5 to 6	[9]	From 5 to 6
[27]	From 3 to 5	[29]	From 4 to 6
[106]	From 13 to 18	[54]	From 8 to 10
[59]	From 5 to 14	[11]	From 18 to 40
[17]	From 9 to 12	[142; 91; 56; 101; 90; 89; 21; 96; 43; 165; 117; 16; 140; 113; 10; 126]	Not specified

Malaysia ([106]), Qatar ([152]), Thailand ([56]), the USA ([16; 133]), and Greece ([130; 129; 19; 20]). It's worth noting that the same work may encompass one or more distinct educational levels. Table 3.4 summarizes the educational levels identified in these works.

The fact that 5% of works did not specify the level of education is a significant gap, as the educational context plays a crucial role in the application of educational robotics. Pedagogical objectives and methods vary widely between primary, secondary, and higher education. Therefore, understanding the context in which ER is being used is fundamental to interpreting the results and the effectiveness of pedagogical approaches.

The lack of detailed information about student age and educational level in ER works can make analysis and generalization of findings difficult. These factors are crucial for a complete understanding of the impact of ER on the development of computational thinking skills and the educational context in general. Therefore, future research may benefit from providing more comprehensive information about the profile of students involved in their investigations.

Robotics Kits Used in the Classroom to Foster Computational Thinking

Regarding using robotics kits in the classroom to promote CT, the SLM revealed that LEGO® stands out as the most frequently employed, being observed in 19 [142; 130; 129; 101; 17; 21; 19; 43; 112; 117; 16; 113; 126; 20; 152; 29; 54; 131; 11] of the surveyed works. Following this, Arduino was utilized in 7 works [31; 176; 106; 56; 165; 10; 177], Bee-Bot in 5 [91; 90; 89; 17; 9], and Thymio in 3 works [59; 96; 45]. The remaining robotics kits were each noted

Table 3.4: Teaching Levels

Teaching Levels	Study	Frequency
Primary School	[54]	1
Vocational	[177]	1
Others	[176; 17]	2
Postgraduate	[10; 11]	2
Undergraduate	[96; 43; 10; 20]	4
Preschool	[162; 27; 90; 89; 9; 29]	6
Elementary school	[31; 91; 101; 59; 21; 165; 112; 45; 140]	9
Middle school	[142; 130; 129; 106; 56; 19; 16; 113; 133; 104; 20; 152; 131]	13
Not specified	[117; 126]	2

in just one of the analyzed studies. It's important to note that one or more robotics kits may be used in each study. Table 3.5 details the robotics kits.

Table 3.5: Robotics Kits

Robotics Kits	Study	Frequency
Code & Go Robot Mouse	[162]	1
Dash	[101]	1
Fischertechnik	[104]	1
Hamster robot	[140]	1
KIBO	[27]	1
Micro Robots Citizen	[142]	
Robot arm (Free Robotics)	[133]	1
Thymio	[59; 96; 45]	3
Bee-Bot	[91; 90; 89; 17; 9]	5
Arduino	[31; 176; 106; 56; 165; 10; 177]	7
LEGO®	[142; 130; 129; 101; 17; 21; 19; 43; 112; 117; 16; 113; 126; 20; 152; 29; 54; 131; 11]	19

The popularity of LEGO® robotics kits can be justified by several factors that make them an exceptionally advantageous choice for education. One of the primary factors contributing to the preference for LEGO® is its accessibility. Although these kits can be relatively expensive, they are widely available and affordable for schools and educational institutions with various budget constraints.

A common consensus among authors justifying the selection of LEGO® kits is their user-friendly interface, particularly well-suited for beginners. The components are designed with intuitiveness in mind, enabling students to embark on building and programming robots quickly. Additionally, LEGO® provides detailed manuals and online learning resources, ensuring accessibility even for teachers with no prior experience in robotics.

The versatility and customization possibilities stemming from the variety of pieces allow a single LEGO® kit to create a diverse range of projects. This flexibility enables teachers to tailor their activities and lesson plans to meet their students' needs and educational goals. Another significant point emphasized by these authors is that due to the global recognition of LEGO® kits, there exists an active community of educators and LEGO® enthusiasts who share ideas, resources, and best practices. This community fosters a supportive environment for teachers aiming to enhance their skills and exchange knowledge regarding integrating robotics into their classes.

The combination of affordability, ease of use, flexibility, educational features, and community support positions LEGO® as a popular and practical choice for teachers looking to introduce robotics and computational thinking into their classrooms. Its appeal is underscored by its positive impact on student skills development and its role in promoting interactive and engaging learning. It is a valuable instrument for incorporating into the educational process of teaching-learning programming with ER.

3.2 Computational Thinking Through Programming Teaching

Over the past decade, computer programming has become pervasive in education, integrated into the curriculum, or offered as an extracurricular activity. A primary objective of this integration is to nurture CT skills in students, enabling them to tackle complex real-world problems effectively. Consequently, there is an increasing need for a more profound understanding of how much learners have developed CT skills [114]. CT extends beyond programming or coding; it is the bedrock for robust teaching-learning programming [82]. Therefore, leveraging teaching-learning programming becomes paramount in enhancing CT skills across a diverse spectrum of students.

Nouri *et al.* [141] set out to ascertain which skills are nurtured in students as they engage with programming in schools. To achieve this, they interviewed teachers with experience in teaching programming over several years. The interviews aimed to shed light on the skills that teachers perceive students developing through programming. The CT skills identified align with the CT framework proposed by Brennan and Resnick [37], encompassing computational concepts, computational practices, and computational perspectives. Their analysis also identified general skills associated with digital literacy and 21st-century skills, including cognitive skills and attitudes, language skills, collaborative skills and attitudes, and creative problem-solving skills and attitudes.

In High School, teaching-learning programming often revolves around block-based programming languages rather than traditional or object-oriented programming languages [82; 121; 123; 124; 166; 189], visual programming tools [64; 74; 115; 123], or robotic coding [111; 154]. For instance, Malizia *et al.* [123] proposed block-oriented programming based on the idea that CT skills, such as abstraction, can be developed through the manipulation of tangible objects. This approach assists students in grasping abstract concepts and nurturing their CT skills. They introduced a framework for implementing block-oriented programmable objects to aid novice programming students in fostering CT skills.

Noh and Lee [139] suggested teaching programming to elementary school students through robotics and examined its effectiveness. They analyzed the impact of students' prior skills and gender on the outcomes and discussed suitable teaching and learning methods in robotics programming. The results indicated that teaching programming through robotics can enhance computational thinking and creativity. This study suggests that designing and implementing a robotics programming course in classrooms while tailoring the approach to students' prior skills is the most effective strategy.

The research of Aparicio *et al.* aimed to analyze if robotics facilitated programming learning. A robotic kit and a miniature course were developed to solve the problem. The robotic kit used the Arduino platform and was presented in the context of this course. The course consisted of programming concepts. This course was taught to Higher Education students who needed to gain previous programming knowledge. In the end, the impact of the course was analyzed. Results demonstrated that students showed interest in using robotics to learn computer programming. Results also showed the main adoption determinants of the robotic kit. Enjoyment was the most relevant to the robotic kit's intention to use. However, perceived ease was the dimension that had more impact on robotic kit usage.

Alegre *et al.* [3] assert that CT expresses thoughts formally, similar to the processes carried out by automated systems. Although programming languages are not the exclusive medium for expressing CT, they are amenable to automation. Hence, employing programming as a conduit for fostering CT is a natural choice. Accordingly, the authors designed and offered an elective course titled 'Introduction to Computational Thinking' to teach programming and bolster mathematical problem-solving skills in high school students and teachers. They utilized Codeworld, an educational tool for teaching mathematics and programming using a programming language akin to Haskell. While the course was well-received by students, it was considered somewhat challenging. Statistical analysis, including pre-tests and post-tests, demonstrated that students acquired language proficiency and a solid grasp of programming fundamentals.

StoffovÃ¡ and Zboran's [174] study discusses adapting Slovak primary school education during the COVID-19 pandemic, where practical lessons were often omitted due to distance teaching. Focusing on programming education, the authors share their experience utilizing modern teaching aids and digital technologies, including virtual and augmented reality, remote laboratories, and simulation models. They emphasize teaching programming through programmable toys, robots, and microcontrollers, using simple tools and interactive environments for block or icon programming. Despite the challenges, this approach fosters enjoyable learning experiences for elementary school students, resulting in rapid progress and a sense of achievement in programming skills.

The study on adapting primary education in Slovakia during the COVID-19 pandemic highlights the use of modern educational resources and digital technologies for teaching programming, emphasizing programmable toys, robots, and microcontrollers in interactive environments [174]. Despite the challenges, this approach provides enjoyable learning experiences for elementary school students, resulting in rapid progress and a sense of achievement in programming skills. In contrast, this thesis proposes and investigates the impact of a validated educational process for teaching-learning programming with ER, aiming to develop CT skills among High School students. Integrating ER into teaching-learning programming, the study focuses on teaching effectiveness and CT skill development, culminating in developing a specific validated educational process, CTProgER, for teaching-learning programming with robotics. Anchored in the Anthropological Theory of Didactics and validated through the Delphi method and intervention study, this research evaluates the effectiveness of CT-ProgER among Technical and Vocational High School students, demonstrating significant improvements in CT skills.

In the existing literature, one can find various methodological proposals for teaching. This section presents a selection of recommendations related to the thesis at hand.

Schiavani [167] employed the Anthropological Theory of Didactics [52] as a framework to identify critical components during the application of didactic moments. The author conducted interventions in high school physics education using Educational Robotics (ER). These interventions featured student lesson plans created by LEGO® Education, aligning with LEGO® Education's methodology. Schiavani's study aimed to provide insights and solutions for developing and structuring didactic activities that enhance the contextualization of ER education. Additionally, it sought to uncover the limits and possibilities of ER in Physics education. Schiavani's work reveals that ER in physics teaching offers a comprehensive approach, addressing the practical-technical and technological-theoretical aspects, as outlined in the Anthropological Theory of Didactics. This proposal can be regarded as a methodological approach to teaching Physics with ER, providing a foundational framework and practical guidelines for its implementation.

The approach taken in this thesis shares similarities with that proposed by Schivani [167], as it applies an educational process grounded in the Anthropological Theory of Didactics to teach programming to high school students using ER. However, a distinction lies in the indirect application of the Anthropological Theory of Didactics in executing the educational process. In a prior work [172], the author demonstrated an educational process derived from the Anthropological Theory of Didactics and ER methodologies, utilized in teaching sciences, akin to LEGO® Education [196], complemented by the previous experiences of the author of this thesis in ER [65; 66; 103; 170; 171].

This thesis unveils the outcomes of an intervention that assesses how Computer Technical and Vocational High School students develop CT skills, both with and without exposure to the educational process. The results indicate a more pronounced impact on students who engage with the educational process, thereby providing insights that serve as guidelines for validating educational processes, particularly quantitative aspects.

Brennan and Resnick's [37] study introduces a framework for teaching and assessing

Computational Thinking (CT) development. They adopt a design-based learning approach to teach programming using Scratch, a programming environment enabling young learners to create interactive stories, games, and simulations. Their Computational Thinking Framework comprises three dimensions: computational concepts, computational practices, and computational perspectives. They describe methods for evaluating these dimensions, including project portfolio analysis, artifact-based interviews, and design scenarios. Additionally, they offer a set of recommendations for assessing student learning when they engage in programming.

Distinguishing itself from Brennan and Resnick's [37] study, this thesis employs a distinct educational approach to CT development. While Brennan and Resnick's work introduces a comprehensive framework for teaching and evaluating CT through Scratch and design-based learning, the current study focuses on implementing an educational process rooted in the Anthropological Theory of Didactics and ER. The proposed educational process is designed to teach programming to high school students, explicitly emphasizing cultivating CT skills. In contrast to Brennan and Resnick's three-dimensional framework, this study incorporates the Anthropological Theory of Didactics principles, indirectly utilizing them during the educational intervention. Doing so introduces a unique perspective on CT skill development and assessment in the context of High School education, contributing to the broader landscape of effective programming pedagogy.

Azman and Mohamed [22] aim to integrate CT into a higher education curriculum and propose a framework incorporating CT components into the creativity and innovation process. The Azman and Mohamed framework is structured around four integrated phases of creativity: 1) Identifying the Main Challenge, 2) Recognizing Patterns, 3) Data Compression, and 4) Step-By-Step Planning. These integrated ideation phases can be viewed as class stages or didactic moments. The authors have designed their framework using straightforward terminology and keywords for improved comprehension. The integration of CT into the innovation process also enhances student interaction, fostering active collaboration during discussions and presentations within the classroom.

This thesis shares similarities with the work presented by Azman and Mohamed [22], as both studies utilize ER to foster CT. However, this thesis focuses on quantitatively understanding the impact of the educational process on CT skills. The goal is to validate the educational process introduced in a previous work by the author [172] and, consequently, contribute to the literature by providing guidelines applicable to the validation of educational processes. Furthermore, this study corroborates prior research by the same author [103; 170; 168], highlighting ER as an effective tool for enhancing CT skills in Technical and Vocational High School students.

Chevalier *et al.* [44] presents a model designed to assist teachers in identifying CT concepts within ER activities and to aid them in effectively planning ER activities. This model was applied to two groups of primary school students participating in an ER activity using the educational robot Thymio. To validate the model, the authors designed and analyzed an ER activity to address the trial-and-error loop problem. The results indicate, among other findings, that a non-instructional approach to ER activities encourages trial-and-error behavior. In other words, the students participating in the ER activity engaged in activities with greater confidence and assertiveness. These results underscore the need for precise instructional interventions within ER activities and exemplify how teachers can utilize the proposed model to design ER activities that facilitate the development of CT skills.

Distinguishing from Chevalier *et al.*'s study [44], this thesis takes a distinct approach to CT development through ER. While Chevalier *et al.* present a model aimed at assisting teachers in identifying CT concepts within ER activities and planning effective ER activities, the current study concentrates on quantitatively assessing the impact of an educational process grounded in the Anthropological Theory of Didactics and ER on CT skills development in high school students. Unlike Chevalier *et al.*'s focus on primary school students, this thesis explicitly addresses the context of Technical and Vocational High School students. By quantifying the impact on CT skills, this study aims to validate the proposed educational process and contribute guidelines for the effective teaching and learning of programming with ER.

The Patiño-Escarcina *et al.* [148] study delves into the challenges faced in children's learning, particularly addressing issues like frustration and inattention. Robotics emerges as a valuable resource in tackling these challenges, empowering students, and enhancing their learning experiences. The research distinguishes between two paradigms of using robotics: as the primary focus and as a secondary tool. While most approaches emphasize teaching robotics as the main focus, this study highlights the significance of integrating robotics into various disciplines to support the learning process. The main contribution lies in a comprehensive three-step methodology for Robotics in Education, offering guidance for implementing robotics projects independently or in conjunction with other subjects. The methodology, termed EDUROSC-Kids, introduces tools for organizing robotics learning topics, defining desired learning outcomes, and evaluating student progress. Divided into three phases setting up the environment, describing the project, and performing evaluation the proposed curriculum categorizes robotics content into five disciplines. It outlines recommended levels of knowledge attainment for different age groups. This research paves the way for enhanced educational experiences by linking general learning processes with robotics education. Additionally, the validation process involved application with students without including expert analyses or statistical methods to demonstrate its effectiveness in student development based on the Bloom taxonomy.

On the other hand, this thesis proposes and investigates the impact of a validated educational process for teaching-learning programming with ER on the development of CT skills among Technical and Vocational High School students. In contrast to the research by Patiño-Escarcina *et al.* [148], the main focus here is on teaching-learning programming, without delving into concepts of mechatronics and electronics, as the robots are pre-assembled to allow students to focus solely on programming concepts. This approach highlights the importance of promoting CT through teaching-learning programming with robotics, utilizing the Anthropological Theory of Didactics and validated through the Delphi method and an intervention study with high school students. It integrates the Román-Gonzalez CT Test to assess the impact of CT skills. This thesis culminates in insights into the proposed educational process, and its effectiveness is evaluated through intervention studies, demonstrating significant improvements in CT skills among Technical and Vocational High School students. The main contribution of the thesis lies in providing a validated educational process centered on teaching and learning programming with robotics that benefits High School students and teachers. It is important to note that these two systems cannot be compared numerically, as they follow different validation processes and have distinct methodological structures.

3.3 Summary of Chapter

This chapter delineates the research phase involving SLM. The objective was to provide an in-depth overview of the characteristics of works related to the development of CT skills through ER. The integration of ER in educational settings has witnessed widespread adoption, focusing on enhancing educational practices. Notably, specific initiatives aim to foster CT skills through ER, and this chapter presents the results of the SLM, which aspires to comprehend the utilization of ER in the educational context for the promotion of CT skills among students.

The findings indicate that the most prevalent approaches include programming, language instruction, and the application of CT, with educational strategies emphasizing collaboration and problem-solving. A significant challenge identified is related to the allocation of class time. The study identifies a range of CT skills primarily linked to programming, with algorithms and sequencing being the most frequently observed. Regarding the research instruments for CT assessment, it was observed that tests devised by the studies' authors are the most commonly used, followed by semi-structured interviews. Additionally, it was noted that the primary audience for ER-based lessons is students in high school and elementary school, typically in the age range of 0 to 17 years old. Finally, LEGO® technology emerges as the most widely employed tool.

The subsequent chapter will introduce the theoretical and practical definitions of the proposed educational process for teaching-learning programming with ER, explicitly focusing on the CT skills referred to as CTProgER.

Chapter 4

CTProgER Educational Process Design

This chapter delineates the educational process, named CTProgER, proposed in this thesis, offering a comprehensive definition. Additionally, it presents the theoretical and practical foundations that informed the design of this educational process, along with practical guidelines provided to empower current and future teachers in integrating it into their robotics programming classes. Finally, three instances of CTProgER are presented. This chapter is intricately linked to **RQ1** (*How can the teaching and learning of programming with ER facilitate students in acquiring CT skills?*) and addresses the following objectives:

- **OBJ2:** Identifying ways to design an educational process for teaching-learning programming in High School with ER, focusing on developing CT;
- **OBJ3:** Instantiating the educational process for some programming topics to be used with High School students.

4.1 CTProgER Educational Process Model

To fulfill **OBJ2**, a theoretical Systematic Mapping Study, as detailed in Chapter 3, and an analysis of previous research conducted by the authors of this thesis was undertaken. The investigation revealed that numerous educational companies propose various methodologies to support teaching with ER. In the state of Paraíba, the most commonly utilized methodologies include LEGO® Education, LEGO® ZOOM [196], Modelix Robotics [158], and Fischertechnik GmbH® [80]. These methodologies are currently accessible to educational institutions across all segments. However, they are predominantly adopted in Brazil at the Basic Education stages for STEAM (Science, Technology, Engineering, Arts, and Mathematics) teaching.

In this context, it was deemed necessary to consider the reformulation of these commercial methodologies, even if they are intended for STEAM teaching. Furthermore, there was a need to scrutinize the validity of these methodologies and enhance their processes, incorporating educational theories or methods supported by well-established viability studies in the literature. The Anthropological Theory of Didactics [58] was identified through a bibliographic survey aimed at exploring methodological proposals for teaching and learning that have scientific documentation or are employed by educational organizations. This investigation has focused on methodological recommendations from various knowledge areas, particularly Human Sciences and Computer Science Education.

The CTProgER was conceived with two primary focuses. The first is **organizational focus**, outlining how the educational process should be structured, considering the scope of instruction. It details how classes should be organized and which artifacts should be constructed to achieve the proposed educational objectives. The second is **validation focus**, encompassing which artifacts should undergo validation processes throughout the development of the thesis work. This dimension aims to ensure the robustness and effectiveness of CTProgER, validating criteria and essential elements for the success of the proposed educational process. Figure 4.1 illustrates the general aspects of the CTProgER.

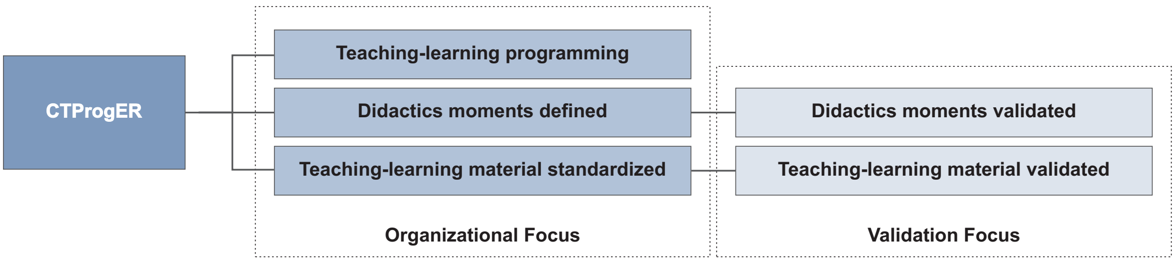


Figure 4.1: The scheme of the CTProgER educational process

To propose the CTProgER educational process for teaching-learning programming in High School through ER, focusing on developing CT skills, it considered the Anthropological Theory of Didactics and the lessons learned from the previous research [65; 66; 103; 168;

170].

The Anthropological Theory of Didactics determines that within an Institution [I] (schools, universities, workshops, study groups...), the teaching of an Object (O) (Physics, Chemistry, Mathematics, Programming, Artificial Intelligence) may require didactic moments. The didactic moments' execution involves the constitution of praxeology and its elements of the four-tuple [T, τ, θ, Θ] (T is the type of task that can be subdivided into several tasks [t]; τ is the technique used to task perform; θ is the technology applied to task perform; Θ: is the theory involved in task performance). When passing through the didactic moments, using the [T, τ, θ, Θ] elements, it is possible to observe a relation between the Individual (X) (student and teachers) and the Object (O) [52; 58] (see Chapter 2, Section 2.4). It understands this successful relation as learning.

In the Anthropological Theory of Didactics, the learning process of a concrete object within an institution is not given in one way, nor does it follow a single chronological path. However, whatever the study path followed, it is possible to perceive that certain situations are always present, even if they have a specificity of qualitative or quantitative origins. These case types are called didactic moments by Chevallard [52; 58]. In other words, didactic moments are steps to be followed within a learning process.

The Anthropological Theory of Didactics comprises six didactic moments, which are:

1. First Meeting.

2. Moment of Exploration.

3. Constitution and Development of the technological-theoretical block.

4. Moment of the Technique Work.

5. Moment of Institutionalization.

6. Moment of Evaluation.

It is important to emphasize that these didactic moments do not necessarily have to follow a chronological order for their execution through the Anthropological Theory of Didactics. They can perform simultaneously and at different times, starting in one class and ending in others.

In this sense, the CTProgER educational process for teaching-learning of programming with ER focusing on CT skills was proposed. As well as the Anthropological Theory of Didactics, the CTProgER considers an Institution [I] as the space for teaching-learning practice were given as input an Object (O), it is possible to develop a relation R(X,O) as output (see Figure 4.2).

The **input** involves identifying one or more programming concepts that want to be teaching-learning and 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 (X) and the object (O). The individual is the student, and the object concerns the CT working through programming concepts. In the Anthropological Theory of Didactics, 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 whose relation R(X,O) is not empty, R(X,O) $\neq$ 0. It is understood that when the concepts given as input are learned, then a positive relation is observed.

To support this proposition, it was proposed six didactic moments based on the Anthropological Theory of Didactics and the lessons learned from previous research are:

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

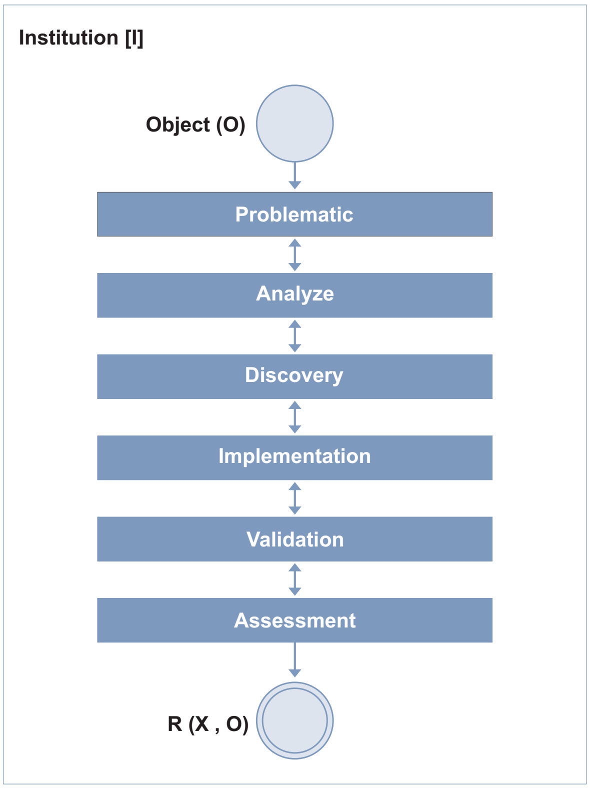


Figure 4.2: CTProgER Educational Process Design

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

4.1.1 CTProgER Educational Process Instances

An organization of the lesson plan and creation of the CTProgER instances were performed to meet the OBJ3. Each lesson corresponds to a 1h30min programming class that applied following the CTProgER guidelines and using the created and validated instances, as is detailed in intervention design in Chapter 5.

Three programming content lessons were designed to implement the CTProgER for teaching programming with ER, focusing on CT skills. Each lesson is intended to last 1 hour and 30 minutes, with students organized into groups of five individuals. The comprehensive lesson plan is outlined in Table 4.1.

Table 4.1: Lessons Plan

Lesson	**Duration**	**Contents**
The Cleaner Robot (O Robô Faxineiro)	1 Week 1h30min	Concept of algorithms; Problems in the construction and execution of algorithms; Construction of algorithms in natural language.
The Accountant Robot (O Robô Contador)	1 Week 1h30min	Data input and output; Data types; Variables Constants; Touch sensor.
The Driver Robot (O Robô Motorista)	1 Week 1h30min	Data input; Condition Control Structure; Ultrasonic sensor.

It created three instances following each aspect presented in the CTProgER educational process, which are "The Cleaning Robot", The Accountant Robot, and "The Driver Robot", as described in Table 4.1. Therefore, it developed the teacher's and student's material for each lesson. Below, it details the main points of each instance.

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 but the concepts involved. The teacher needs to perform the programming syntax during the lesson according to the guidelines from the teacher's material.

The instances were organized by CTProgER didactic moments, so the lessons had six moments transformed into topics. They are:

1. **Problematic:** knowing the problem (Conhecendo o Problema).
2. **Analysis:** Analyzing the problem (Analizando o Problema).
3. **Discovery:** Discovering knowledge (Descobrindo Conhecimentos)
4. **Implementation:** Implementing a solution (Implementando uma Solução).
5. **Validation:** Testing the solution (Testando a Solução).
6. **Assessment:** Questions set must be answered at the lesson end.

The following Subsection details the main points of each instance.

4.1.2 The Cleaning Robot Instance

The Cleaning Robot Instance introduces algorithms and starts with the context of the domestic cleaning robots, presenting news from the Olhar Digital website[1] to promote the content contextualized with real life. In this instance, the character Miguel has a problem that must be solved throughout the lesson. Figure 4.3 presents the student version of the Cleaning Robot instance graphic.

Problem/situation: "*Student Miguel, from the 1st Grade of Integral High School at Nossa Senhora das Graças State School, decided to teach his robot, built with LEGO®, to help him in cleaning his house. Initially, he wants the robot to clean only his room corners like a square. Unfortunately, the robot does not have sensors, i.e., Miguel will not be able to make the robot autonomous. He needs to pass the exact coordinates to perform robot work efficiently. His task is to propose a solution that helps Miguel teach his robot*".

[1] `https://olhardigital.com.br`

Figure 4.3: The student version of The Cleaning Robot Instance

After the problem presentation, the material discusses logical thinking and its relation to algorithms, ending the **problematic didactic moment**. Due to the problem/situation involving the context of square shapes, the **analysis didactic moment** will take place on the construction of geometric shapes. This association aims to help students understand the rules for building complex geometric shapes like triangles. Later, he can adapt to the context of the square of the problem/situation.

Then, in the **discovery didactic moment**, the concept and construction of the algorithm in natural language are presented, taking as an example a day-to-day situation. These concepts will lead to a natural language algorithm that makes the robot move in a triangle shape. These concepts will support the **implementation didactic moment**. The student must propose a solution in natural language that favors the robot to move in a square shape.

The solution created will be tested at the **validation didactic moment**. If it does not solve the problem, the student can make the necessary adjustments. To close the lesson, during the **assessment didactic moment**, the student must answer questions set to reflect on the concepts of algorithms learned.

The teacher and student versions of The Cleaning Robot instance are available online[2].

4.1.3 The Accountant Robot Instance

The Accountant Robot Instance introduces primitive programming instructions, such as data input and output, type of data, variables, and constants, with the accountant robot presenting news from the ISTOÉ Dinheiro Magazine[3] to promote the content contextualized with real life. In this instance, the character Luna Yasmin has a problem that must be solved throughout the lesson. Figure 4.4 presents the student version of the Accountant Robot instance graphic.

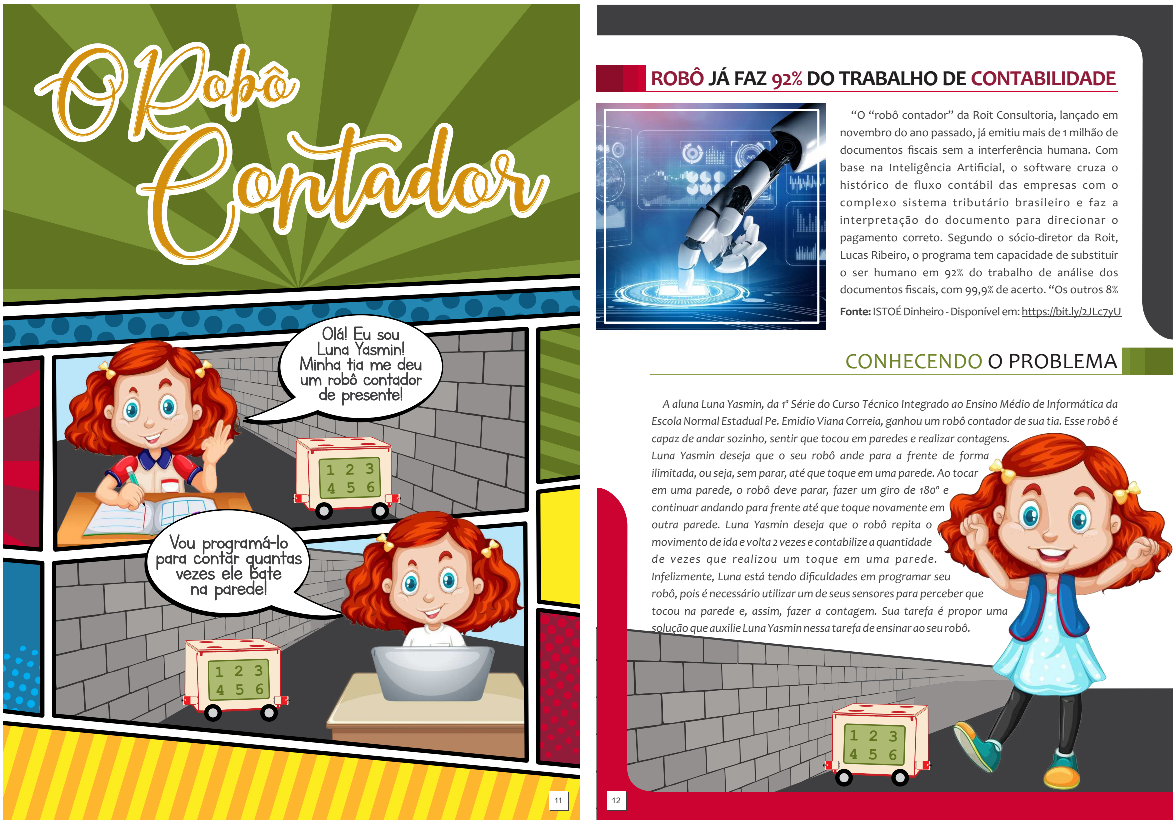


Figure 4.4: The student version of The Accountant Robot Instance

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

[3] https://istoedinheiro.com.br/robo-ja-faz-92-do-trabalho-de-contabilidade

Problem/situation: "*Student Luna Yasmin, from the 1st Grade of the computer TV High School at Escola Normal Estadual Pe. Emídio Viana Correia received an accounting robot from her aunt. This robot can move alone, feel that it has touched walls, and perform counting. Luna Yasmin wants her robot to move forward unlimitedly, i.e., without stopping, until it touches a wall. When touching a wall, the robot must stop, turn left 180Âº, and continue walking forward until it touches another wall again. Luna Yasmin wants the robot to repeat the back-and-forth movement two times and count the number of times it touches a wall. Unfortunately, Luna is having difficulty programming her robot, as it is necessary to use one sensor to realize that she has touched the wall and thus count. Her task is to propose a solution that helps Luna Yasmin teach her robot.*".

After the problem presentation, the material discusses the primitive instructions and sensors, ending the **problematic didactic moment**. Due to the problem/situation involving the context of the way robots sense the world around them, the **analysis didactic moment** will raise questions about movements such as walking forward, making turns, and feeling like it touched a wall. This association aims to understand the rules for manipulating data input and output in a robotic context to solve the problem/situation later.

Then, in the **discovery didactic moment**, the concept of For this, there is a deepening of knowledge in data types, variables, constants, and touch sensors. Also, it analyzes the algorithm in natural language that he can later adapt to the context of a programming language to solve the problem/situation. These concepts will support the **implementation didactic moment**. The student must propose a solution in a programming language that favors the robot to walk, touch shapes, and make accounts.

The solution created will be tested at the **validation didactic moment**. If it does not solve the problem, the student can make the necessary adjustments. To close the lesson, during the **assessment didactic moment**, the student must answer questions set to reflect on the concepts of data types, variables, constants, and touch sensors.

The teacher and student versions of The Accountant Robot instance are available online[4].

[4] `https://github.com/isabellelimasouza/CTProgER.git`

4.1.4 The Driver Robot Instance

The Driver Robot Instance introduces a condition control structure. It starts with the context of autonomous vehicles, presenting news from the Olhar Digital website[5] to promote the content contextualized with real life. In this instance, the character Thifany Laís has a problem that must be solved throughout the lesson. Figure 4.5 presents the student version of The Driver Robot Instance graphic.

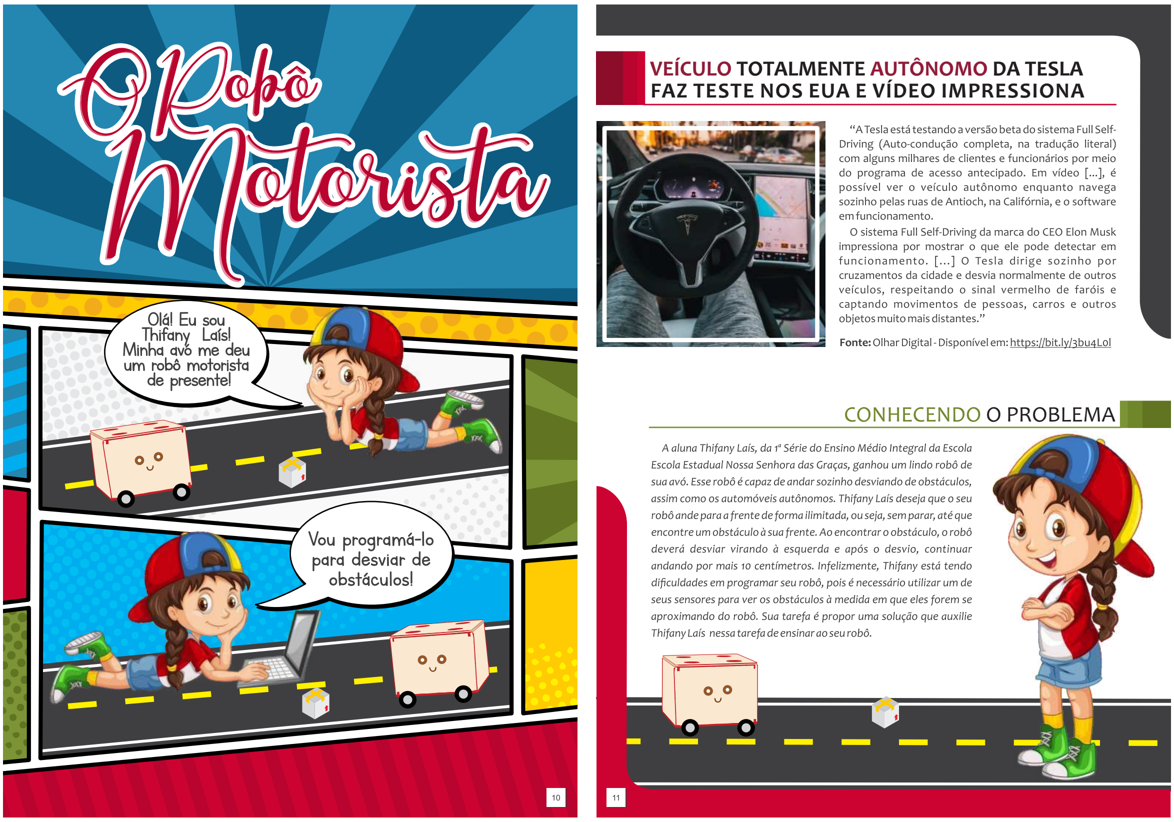


Figure 4.5: The student version of The Driver Robot Instance

Problem/situation: "*Student Thifany Laís, from the 1st Grade of Integral High School at Nossa Senhora das Graças State School, received a beautiful robot from her grandmother. This robot can move alone, avoiding obstacles, just like autonomous vehicles. Thifany Laís wants her robot to move forward unlimited, that is, without stopping until she encounters an obstacle in front of her. When finding the obstacle, the robot must deviate by turning to the left and continuing walking for another 10 centimeters. Unfortunately, Thifany is having*

[5] https://olhardigital.com.br

difficulties programming her robot, as it is necessary to use one of its sensors to see obstacles as they approach the robot. Her task is to propose a solution that helps Thifany Laís teach her robot".

After the problem presentation, the material discusses the possibility of analyzing and deciding on the decision instructions concept, ending the **problematic didactic moment**. Due to the problem involving movement based on data, the **analysis didactic moment** will raise questions about movements such as walking forward and making turns. In addition, it will discuss the need for instruments that allow the robot to feel the world around it.

Then, in the **discovery didactic moment**, the concept of data inputs, echolocation system with ultrasonic sensors, and conditional control structures. The concepts of conditional control structures are treated in natural language and will support the **implementation didactic moment**. The student must propose a solution in a programming language that favors the robot to move and avoid obstacles.

The solution created will be tested at the **validation didactic moment**. If it does not solve the problem, the student can make the necessary adjustments. To close the lesson, during the **didactic assessment moment**, the student must answer questions about concepts of the echolocation system with ultrasonic sensors and conditional control structures learned.

The teacher and student versions of The Driver Robot instance are available online[6].

4.2 Summary of Chapter

In this chapter, the endeavors undertaken throughout this thesis to address **RQ1** (*How can programming be taught with Educational Robotics to facilitate students in acquiring Computational Thinking (CT) skills?*) were outlined. The response to RQ1 involved proposing an educational process for teaching programming with robotics, emphasizing CT skills. Thus, the theoretical concepts supporting the proposed educational process, CTProgER, were introduced. Additionally, the conceptual definition of CTProgER and practical guidelines were presented to empower current and future educators to implement it in their robotics programming classes, achieving OBJ2. Lastly, three instances of CTProgER were provided and explained, serving as illustrative examples of classes based on CTProgER. These instances

[6]`https://github.com/isabellelimasouza/CTProgER.git`

are intended to aid educators in the instantiation process, and they are made available for free use in educational or scientific research, fulfilling OBJ3. The following chapter will unveil the outcomes of the CTProgER validation experiment, considering the perspective of expert teachers.

Chapter 5

Assessing the CTProgER Validity from Experts' Perspective

This chapter presents an expert assessment as part of Educational Design Research conducted to assess the effectiveness of CTProgER. The expert assessment utilized the Delphi method [136] to validate objects through human judgment. Its primary objective was to ascertain the conformity of the CTProgER educational process and its instances. This assessment involved expert human judgment from programming teachers who analyzed CTProgER instances, evaluating their adherence to guidelines and alignment with usual programming classes. This chapter is tied explicitly to **RQ2:** How much conformity is there in the educational process of teaching-learning programming with ER from an expert's perspective? The expert assessment pursued the following objectives:

- **OBJ4:** Validating the instance from an expert point of view.

5.1 Research Design

In this Educational Design Research expert assessment, researcher strategies are employed to validate objects through human judgment. Since the validation from the students' perspective stage is intended to take place in the natural environment of a local High School, it is crucial to minimize any divergence between the instances and the educational process guidelines outlined in this thesis. To accomplish this goal, the research question and hypotheses outlined below were formulated to steer the development of this expert assessment:

RQ2: How much conformity is there in the educational process of teaching-learning programming with ER from an expert's perspective?

- *H2.0:* There is no evidence that the educational process of teaching and learning programming with ER can conform to an expert's perspective.
- *H2.1:* The educational process of teaching and learning programming with RE can conform to an expert's perspective.

In this context, the Delphi method was chosen to attain a progressive consensus on a study object. This method involves iterative rounds of questioning posed to a group of experts, with the responses analyzed to identify a consensus among them. The criteria for consensus depend on the researcher's study, but it is essential to adhere to the strictness of these criteria, as emphasized by Williams [184]. The Delphi method leverages structured knowledge, experience, and the creativity of experts. When organized effectively, it yields more robust information than relying solely on a single expert or individuals lacking specialized expertise in the field, as noted by Linstone [119].

The Delphi method is characterized by critical features, including specialist (expert) anonymity, the statistical representation of result distributions, and feedback from experts' responses for reevaluation in subsequent rounds [187]. The Delphi process typically requires a minimum of two rounds to be identified as such. Nevertheless, only a few studies incorporate more than three rounds [86], as there is a tendency to conduct fourth and subsequent rounds without altering the experts' opinions.

Figure 5.1 illustrates the workflow of validating an instance through multiple validation rounds with experts. The adopted workflow is structured into four steps:

- **Step 1:** Methodology Definition / Adjustments;
- **Step 2:** Methodology Presentation;
- **Step 3:** Assessment;
- **Step 4:** Analyzis.

These steps were categorized based on the individuals involved, including the researcher and experts, represented by horizontal lines (see Figure 5.1).

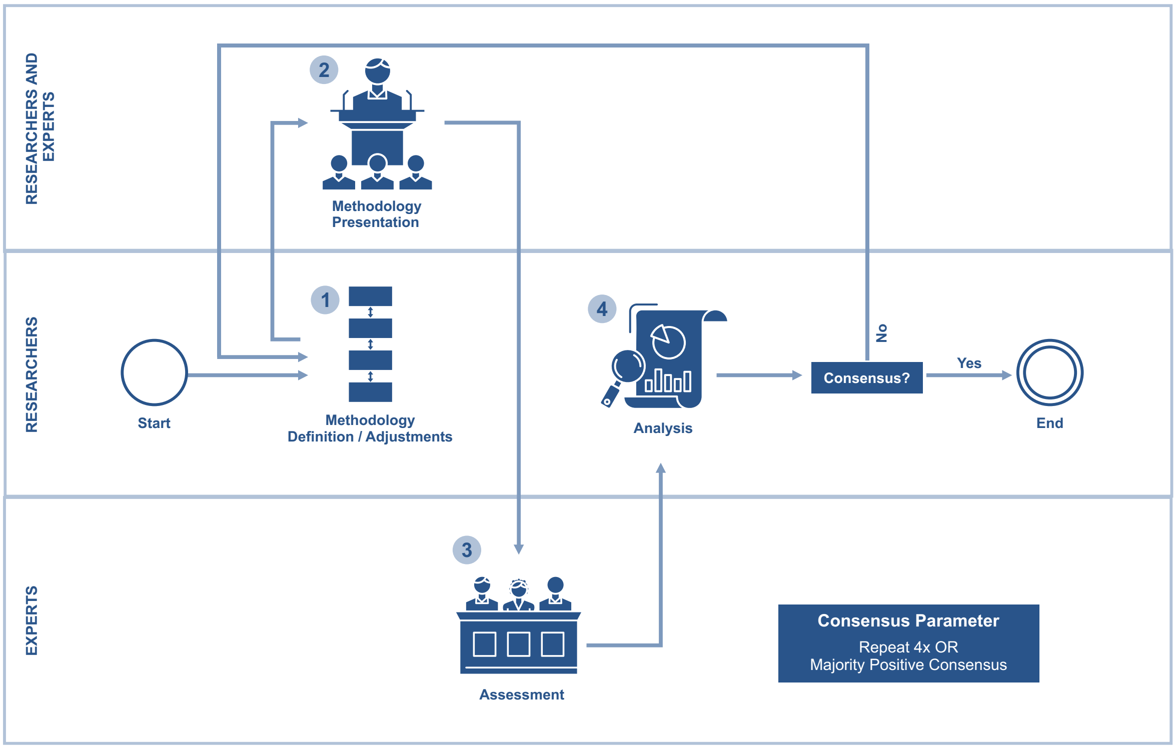


Figure 5.1: Expert Assessment Workflow

Step 1 is integrated into the instance validation workflow because, according to expert infusion, it may be necessary to update instances created during the definition of the CT-ProgER after each round.

Step 2 supplies all the essential information for experts to validate. An online website[1], which includes a research presentation video and key concepts supporting the execution of the validation process, was accessible. Furthermore, the website was used to disseminate documents related to instance validation.

Step 3 intends to evaluate the compatibility of instances created with the CTProgER guidelines through the anonymous judgment of experts. The validation process may uncover potential weaknesses in the CTProgER proposal. Validation rounds were carried out with experts who documented their impressions in the defined Barema instance validation. This Barema must be prepared following the CTProgER guidelines.

Step 4 intends to tabulate the results generated by experts, categorizing responses as positive or negative and identifying potential consensus among expert opinions regarding

[1]Website: `https://sites.google.com/view/isabellelimadoutorado`

the CTProgER guidelines. Consensus is established through a majority vote, requiring at least 50% plus one of positive answer in the Barema. Furthermore, achieving consensus involves a percentage of absolute agreement (PAA) exceeding 75% [173].

5.1.1 Data Collection Procedures

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

Google Forms collected data anonymously, and Google Sheets tabulated the experts' responses, identifying the majority vote. The percentage of absolute agreement for each instance during validation was calculated separately. R programming was applied, precisely the IRR (Internal Rate of Return) package and the AGREE function.

During the process, two validation rounds of the instances were conducted with experts who recorded their impressions in the Barema (refer to Table 5.1), considering nine criteria related to CTProgER educational process guidelines. Experts could respond to each criterion with the following options: 0 - No, 1 - Very little, 2 - Partially, 3 - Yes.

The experts' responses were categorized into positive (2 - Partially and 3 - Yes) and negative (0 - No and 1 - Minimal). This categorization aimed to assess potential agreement among the experts' opinions regarding the nine criteria in the Barema. An agreement was determined through majority voting, requiring at least 50% plus one of positive answer for each Barema criterion to validate the instance. In addition to majority voting, the agreement also required a percentage of absolute agreement (PAA) exceeding 75% [173].

Calculating the PAA involves determining the frequency of raters' unanimous ratings and dividing it by the total number of ratings. This metric typically ranges from 0 to 100%. It is also known as the percentage of exact agreement or specific agreement. Additionally, one may find it beneficial to assess the percentage of ratings within one performance level of each other, referred to as exact and adjacent agreement. This approach is particularly relevant when multiple rating levels exist beyond 4 or 5 [93].

The PAA is obtained through Equation 5.1, where NC is the number of times experts agree on a rating, and TA is the total number of ratings.

$$PAA = \frac{NC}{TA} * 100 \tag{5.1}$$

Table 5.1: Barema Criteria

Criteria	**Judgment**
Q01: Is it possible to identify the content of this lesson?	0 - No, 1 - Minimal, 2 - Partially, 3 - Yes
Q02: Is it possible to identify the objective of this lesson (what should the student learn in this lesson)?	0 - No, 1 - Minimal, 2 - Partially, 3 - Yes
Q03: Is it possible to identify the development lesson objective (what the teacher wants the student to foster)?	0 - No, 1 - Minimal, 2 - Partially, 3 - Yes
Q04: Is it possible to identify the problem/situation in this lesson?	0 - No, 1 - Minimal, 2 - Partially, 3 - Yes
Q05: Is it possible to identify prior knowledge needed in this lesson (student-centered)?	0 - No, 1 - Minimal, 2 - Partially, 3 - Yes
Q06: Is it possible to identify the programming structures that must be implemented to solve the problem proposed in this lesson?	0 - No, 1 - Minimal, 2 - Partially, 3 - Yes
Q07: Is it possible to identify ways to test the solution to be implemented in this lesson?	0 - No, 1 - Minimal, 2 - Partially, 3 - Yes
Q08: Is it possible to identify how the student is assessed in this lesson?	0 - No, 1 - Minimal, 2 - Partially, 3 - Yes
Q09: Is it possible to identify the knowledge the student must obtain in this lesson?	0 - No, 1 - Minimal, 2 - Partially, 3 - Yes

5.1.2 Instruments' Validation

Several documents supported the validation process to facilitate the experts' activities. The documents designed to assist in expert validation included:

- **Supporting Material for Experts (video):** presentation of the main concepts underlying the study and workflow.
- **Supporting Material for Experts (text):** detailed proposal document with theoretical concepts and procedures to be followed.
- **Student Material:** two documents will be made available to the student during programming lessons with robotics as a teaching tool. The instances should be two programming lessons for 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 for High School.
- **Validation Barema:** anonymous form for recording the experts' answers to questions concerning the nine criteria for evaluating the methodology instances. Each criterion can be answered with 0 - No, 1 - Minimal, 2 - Partially, or 3 - Yes.

The documents were provided to the experts via the website[2] and are available online[3].

5.1.3 Experts' Profile

Six professors and researchers with prior experience on researching or working with CT, teaching programming, or ER were engaged as experts for the instance validation.

Among the six experts, 50% (3) are female, and 50% (3) are male. Furthermore, 16.7% (1) hold a bachelor's degree, 66.6% (4) hold a licentiate degree, and 16.7% (1) hold an associate's degree (equivalent to a technologist degree in Brazil). In terms of their field of study in Higher Education, 50% (3) specializes in computer science, 33.3% (2) in computing, and 16.7% (1) in systems for the Internet. Most experts, 83.3% (5), have a master's degree,

[2]Website: `https://sites.google.com/view/isabellelimadoutorado`
[3]Repository: `https://github.com/isabellelimasouza/CTProgER.git`

while 16.7% (1) have a Ph.D. The predominant field of expertise for those with a master's or Ph.D. is Computer Science, accounting for 83.3% (5), and Mathematics and Technological Education for 16.7% (1). Lastly, 66.7% (4) of the experts possess teaching experience in Technical and Vocational Education, while 33.3% (2) have experience in Higher Education. Refer to Table 5.2 for a detailed breakdown of the experts' profiles

Table 5.2: Details of Experts' Profile

Expert	**Gender**	**Higher Education**	**Higher Education Area**	**Highest level of training**	**Highest level of training Area**	**Training level in progress**
1	Female	Bachelor's degree	Computer Science	PhD	Computer Science	-
2	Male	Degree (Licentiate)	Computing	Master	Computer Science	PhD in Computer Science
3	Male	Degree (Licentiate)	Computer Science	Master	Computer Science	PhD in Computer Science
4	Female	Associate's degree	Sytems for internet	Master	Computer Science	PhD in Computer Science
5	Female	Degree (Licentiate)	Computing	Master	Mathematics and Technological Education	-
6	Male	Degree (Licenciate)	Computer Science	Master	Computer Science	PhD in Decision Models and Health

5.2 Results

Two rounds of the Delphi method were undertaken to investigate Research Question **(RQ2)**, aiming to validate the conformity of CTProgER. To validate, experts' evaluations were concentrated on aligning CTProgER guidelines, taking into account didactic moments, their descriptions, and the approach to content within the instances. This section will present the quantitative outcomes of the first Delphi round, followed by the procedures for implementing improvements in instances according to the experts' qualitative considerations. Lastly, the results of the second round, which ultimately provide insights into answering RQ2, will be elucidated.

5.2.1 Delphi's First Round for Validating CTProgER Instances

The first round involved the CTProgER Instances (Materials for Students) and the CTProgER Instances (Materials for Teachers). To observe the validation of instances in the first round, the analysis began by observing a majority vote from the experts, requiring at least 50% plus one of positive answer for each criterion. The votes were examined on a per-instance basis.

As depicted in the data presented in Figure 5.2, a majority of experts voted positively for The Cleaning Robot Instance and The Driver Robot Instance, surpassing 50% plus one of positive answer for each criterion in the Barema.

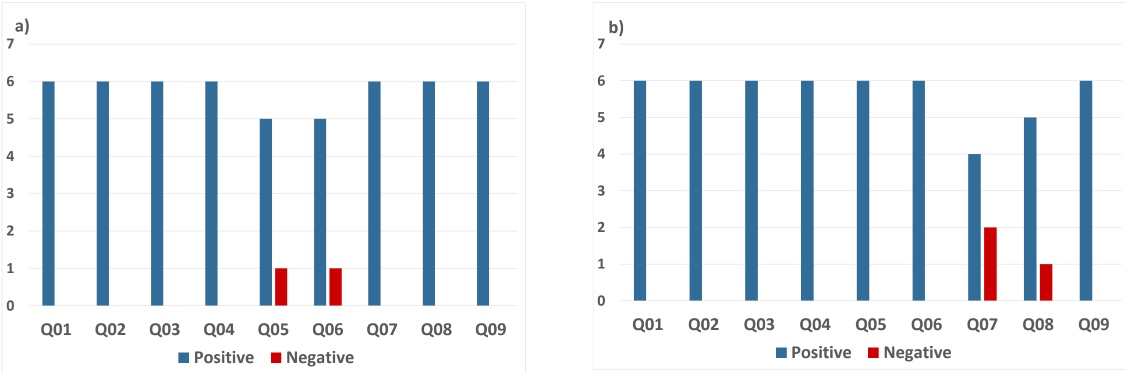


Figure 5.2: a) Experts' validation data by version 1 to The Cleaning Robot Instance. b) Experts' validation data by version 1 to The Driver Robot Instance.

According to Figure 5.2 (a), there is a 100% positive majority vote (5) for criteria Q01, Q02, Q03, Q04, Q07, Q08, and Q09, and 80% (4) for Q05 and Q06 in The Cleaning Robot Instance. Similarly, as shown in Figure 5.2 (b), the positive majority vote is 100% (5) for Q01, Q02, Q03, Q04, Q05, Q06, and Q09 criteria, and 80% (4) for Q07 and Q08 in The Driver Robot Instance.

Additionally, it calculated the Percentage of Absolute Agreement (PAA) between the experts. Table 5.3 displays the PAA by instance. Both instances were observed to have a PPA greater than 70%, and considering the positive majority voting, it can be concluded that the instances were validated in the first round of experts' validation.

Table 5.3: Percentage of absolute agreement in round 1

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

According to the consensus criteria, both instances were validated. However, since it employed the Delphi method, it is imperative to conduct an additional round to mitigate potential nonconformity in the validation process, specifically refining the Q5, Q6, Q7, and Q08 criteria.

Based on the experts ’ evaluation, the fundamental changes made to the instances were primarily related to how the content was presented in the student materials. At times, the concepts were described using less appropriate terminology and did not align with the didactic sequence proposed by CTProgER. Furthermore, there was a significant improvement in the clarity of instructions provided to students during the validation didactic moment. These revisions aimed to enhance the understanding and practical application of the content.

In the subsequent subsection, all modifications made to each instance will be thoroughly described, considering specific evaluation criteria. This detailed approach allows for a meticulous analysis of the adaptations made in response to expert evaluations, providing a comprehensive insight into the improvements implemented in the instructional material within the context of a doctoral thesis in computer science.

5.2.2 Refining Instances Based on Qualitative Insights from Experts

The improvements in the instances were made based on the considerations provided by experts through text fields. **Criterion 01** is related to identifying the content intended to be taught in this lesson. This criterion aimed to ascertain whether the materials were sufficient for experts to identify the content designed for teaching-learning. According to CTProgER guidelines, the content is the input given as the entry point of the process. Table 5.4 summarizes the experts’ considerations of criterion 01.

Table 5.4: Expert Insights Transcription - Criteria 01

Instance	Expert	Translated Freely
The Cleaning Robot	1	No considerations.
	2	Evidently, the lesson’s purpose is to understand how to give instructions to a robot, a concept consistently linked to algorithmic content throughout the material.
	3	Algorithm, sequential structure, square, and equilateral triangle.

	4	Yes. The content (algorithmic concepts) to be taught is evident in both materials (student and teacher).
	5	No considerations.
	6	No considerations.
The Driver Robot	1	No considerations.
	2	Yes, however, it could be clearer if you use the terms 'decision structure' or 'choice' instead of 'verification'.
	3	Decision structure and sensors.
	4	Conditionals, input and output, and sensors.
	5	No considerations.
	6	No considerations.

Three experts considered that the contents of the instance "The Cleaning Robot" are related to the concept of algorithms, and in "The Driver Robot" instance, the idea of the control structure (condition), highlighting alignment in their perspectives and with the actual content considered during the instantiation process. However, among the three experts who evaluated the instance "The Driver Robot", one (Expert 2) suggested changing the terminology from "verification" to decision or choice structure. This suggestion was accepted, and the instance material was modified (see Figures 5.3 and 5.4).

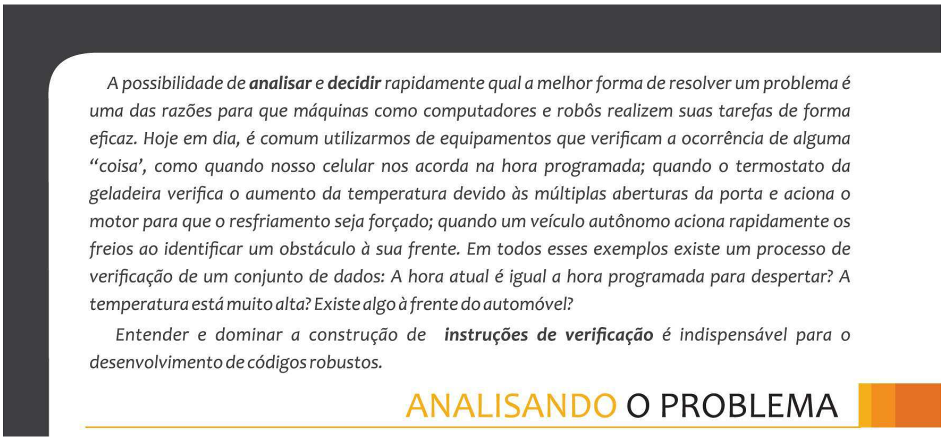
Criterion 02 deals with identifying the objective of the class, precisely what the class intends to encourage in the student. This criterion identifies the class's objective to develop in the students. It extends beyond the input content to cover other actions and knowledge that the student must do and have throughout the class. Table 5.5 summarizes the experts' considerations of criterion 02.

Table 5.5: Expert Insights Transcription - Criteria 02

Instance	Expert	Translated Freely
The Cleaning	1	No considerations.

Robot	2	I can comprehend that the lesson's objective is to teach the very fundamental concept of algorithms, incorporating an association with robotics. The class is designed to guide me in learning how to solve the robot's problem.
	3	Motivate him to build a robot similar to the one in the initial report. And he will be capable of contemplating algorithms.
	4	I had to revisit the document to comprehend that the objective emphasizes the perception that algorithms are a part of our daily lives. From my understanding, this observation is connected to the teacher's material.
	5	No considerations.
	6	No considerations.
The Driver Robot	1	No considerations.
	2	Yes, I can identify the objective; however, the second script introduces pseudocode algorithms with a specific syntax. This form of writing the algorithm needs to be explained to the student in this script, including its elements, etc.
	3	It prompts the student to consider getting around obstacles with the robot.
	4	This script focuses more on the input and output of data, conditionals, and sensors.
	5	No considerations.
	6	No considerations.

Three experts considered that the objective of the class of the "The Cleaning Robot" instance is related to the necessary foundations for introducing algorithms. These assessments align with the content considered during the instantiation process. Regarding the "The Cleaning Robot" instance, Expert 4 noted the need to consult the teacher's material to abstract the class objective. This need arises because Criterion 02 aligns with what the class aims to stimulate in the student. In other words, to accurately discern this objective, the

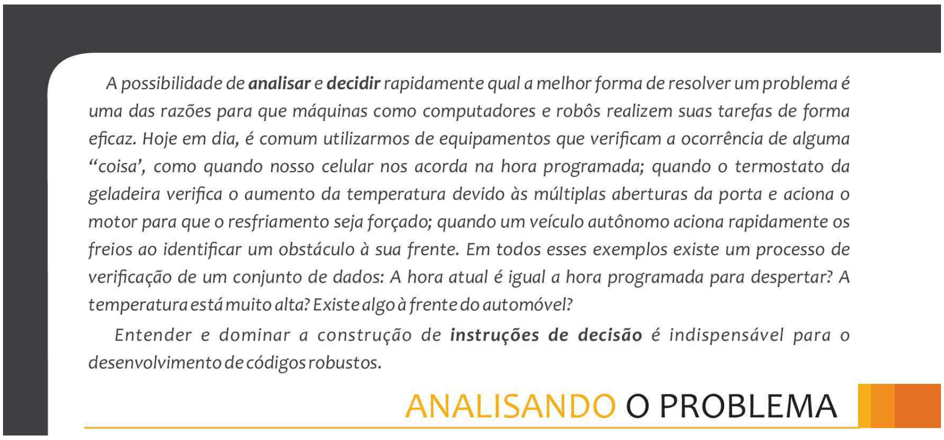


A possibilidade de **analisar** e **decidir** rapidamente qual a melhor forma de resolver um problema é uma das razões para que máquinas como computadores e robôs realizem suas tarefas de forma eficaz. Hoje em dia, é comum utilizarmos de equipamentos que verificam a ocorrência de alguma "coisa', como quando nosso celular nos acorda na hora programada; quando o termostato da geladeira verifica o aumento da temperatura devido às múltiplas aberturas da porta e aciona o motor para que o resfriamento seja forçado; quando um veículo autônomo aciona rapidamente os freios ao identificar um obstáculo à sua frente. Em todos esses exemplos existe um processo de verificação de um conjunto de dados: A hora atual é igual a hora programada para despertar? A temperatura está muito alta? Existe algo à frente do automóvel?

Entender e dominar a construção de **instruções de verificação** é indispensável para o desenvolvimento de códigos robustos.

ANALISANDO O PROBLEMA

Figure 5.3: The Driver Robot instance content in the first round.



A possibilidade de **analisar** e **decidir** rapidamente qual a melhor forma de resolver um problema é uma das razões para que máquinas como computadores e robôs realizem suas tarefas de forma eficaz. Hoje em dia, é comum utilizarmos de equipamentos que verificam a ocorrência de alguma "coisa', como quando nosso celular nos acorda na hora programada; quando o termostato da geladeira verifica o aumento da temperatura devido às múltiplas aberturas da porta e aciona o motor para que o resfriamento seja forçado; quando um veículo autônomo aciona rapidamente os freios ao identificar um obstáculo à sua frente. Em todos esses exemplos existe um processo de verificação de um conjunto de dados: A hora atual é igual a hora programada para despertar? A temperatura está muito alta? Existe algo à frente do automóvel?

Entender e dominar a construção de **instruções de decisão** é indispensável para o desenvolvimento de códigos robustos.

ANALISANDO O PROBLEMA

Figure 5.4: The Driver Robot instance content after modification.

teacher needs to assume a student's perspective rather than that of a teacher as it pertains to the student's experience during the class. It is crucial to emphasize that the validation of the instance encompasses two materials: the student and the teacher. Therefore, the experts may not have fully grasped the objective of the class.

In turn, the considerations of three experts on Criterion 02 of the "The Driver Robot" instance presented distinct objective classes related to the planned goals during the instantiation process. Expert 2 indicated that the instance introduced pseudocode that had yet to be covered in previous classes. It is essential to highlight that the two lessons are not sequential; therefore, the contents of the two instances are not expected to complement each other. An additional instance between "The Cleaning Robot" and "The Driver Robot" is necessary.

Moreover, the students' material was not designed to eliminate the teacher's presence in the classroom but rather to be guided by the teacher, who will select a programming language for the implementation process. Thus, all algorithmic structures in the materials should be built in pseudocode, with the teacher guiding the teaching of the programming language syntax. Therefore, the teacher must be trained using the teacher's material to supplement the lesson successfully following the CTProgER guidelines.

The experts' considerations on criterion 02 kept the students' material the same; however, the teachers' material had more detailed instructions on how to work with the programming language.

Criterion 03 pertains to identifying the development objective, i.e., what the teacher wants the student to develop. This criterion identifies the developmental goal within the teacher's plans, ensuring students progress through the proposed activity. It extends beyond the input content to encompass other skills and abilities that may evolve throughout the class. Table 5.6 summarizes the experts' considerations of criterion 03.

Table 5.6: Expert Insights Transcription - Criterion 03

Instance	**Expert**	**Translated Freely**
The Cleaning Robot	1	No considerations.
	2	As it is aimed at High School students, I believe the description of the activity 'Implementing a Solution' should provide more detail, emphasizing the importance of creating the algorithm, even in natural language.
	3	The development objective is to construct a sequence of steps to make the robot walk within a square. However, the role of the triangle (specifically an equilateral one) in the textbook example must be clarified.
	4	A solution for the cleaning robot that lacks sensors. For this purpose, essential concepts (algorithms and triangles) are introduced. From what I understand, this observation is related to the student's material.

	5	No considerations.
	6	No considerations.
The Driver Robot	1	No considerations.
	2	Reiterating that pseudocode is presented, which has a well-defined structure, and the student needs to understand how it works, along with essential concepts at this point, such as the concept of variables.
	3	Utilizing the sensor and decision structure.
	4	It works with a solution for the robot to navigate around an obstacle, employing programming concepts (conditionals) and utilizing the robot's equipment that allows it to receive environmental information (sensor).
	5	No considerations.
	6	No considerations.

Three experts considered that the development objective of the "The Cleaning Robot" instance is related to broader concepts, such as mathematical concepts like squares, rectangles, and triangles, as well as the operation of sensors. Indeed, the class addresses these understandings and requires students to reflect on them, as they will form the foundation for solving the problem. Being an introductory algorithm class, the instance "The Cleaning Robot" enables the teacher to reflect more deeply on the unfolding of developmental objectives; each student's experience may influence this reflection. However, all considerations align with what was envisaged during the instantiation process.

Expert 4 emphasized that, in his perception, the developmental objective is associated with the student's material. However, the developmental aim is tied to the teacher's vision of what the class should instill in the student. This vision encompasses not only content but also attitudinal aspects. Therefore, validating this criterion must be conducted, considering the teacher's material. This interpretation may have posed challenges in validating this criterion, not just for expert 4 but also for others. Consequently, additional clarification was provided in the experts' guidelines for conducting the second round of validation.

In turn, the considerations of three experts on criterion 03 of the “The Driver Robot” instance presented objectives related to concepts of conditional structures connected to the planned goals during the instantiation process. Expert 2 indicated that the instance introduced pseudocode that had yet to be covered in previous classes. It is essential to highlight that the two lessons are not sequential; therefore, the contents of the two instances are not expected to complement each other. An additional instance between ”The Cleaning Robot” and ”The Driver Robot” is necessary. Moreover, the students’ material was not designed to eliminate the teacher’s presence in the classroom but rather to be guided by the teacher, who will select a programming language for the implementation process. Thus, all algorithmic structures in the materials should be built in pseudocode, with the teacher guiding the teaching of the programming language syntax. Therefore, the teacher must be trained using the teacher’s material to supplement the lesson successfully following the CTProgER guidelines.

Criterion 04 aspires to identify the problem of the class. It believes this criterion is easy to observe in both instances, so a section with the “Conhecendo o Problema” name exists. Table 5.7 summarizes the experts’ considerations of criterion 04.

Table 5.7: Expert Insights Transcription - Criterion 04

Instance	**Expert**	**Translated Freely**
The Cleaning Robot	1	No considerations.
	2	Yes, the association between the real-world problem and what one aims to teach is evident.
	3	It aims to construct a sequence of steps to make the robot move along a square.
	4	I understood that the issue revolves around room cleaning.
	5	No considerations.
	6	No considerations.
The Driver Robot	1	No considerations.
	2	Yes, it is possible to comprehend the issue presented in the lesson.
	3	Making the robot navigate around obstacles.

	4	After the initial lesson, I understood that the issue revolves around constructing a robot that can navigate around obstacles. My inquiry pertains to the precise nature of the problem and the implicit concepts involved whether it applies solely to the construction aspect or extends to a comprehensive understanding of the concepts taught in the class.
	5	No considerations.
	6	No considerations.

Three experts considered that the problem of the "The Cleaning Robot" instance is related to the movement of the robot in the shape of squares, a consideration aligned with what was envisaged during the instantiation process. By manipulating the robot in a space with a specified size and shape, it is possible to introduce the concept of a sequence of well-defined steps to achieve a goal, that is, algorithms. Additionally, the problem scenario allows for the introduction.

Based on the considerations of three experts on criterion 04 of the "The Driver Robot" instance, there is a consensus that the problem revolves around robots that move and navigate around obstacles. However, Expert 4 raises a question about what the problem is. There is doubt about whether the problem initializes the question or if it extends to the knowledge that will be developed throughout the class. In this context, it is necessary to emphasize that the problem refers to the challenges, questions, or issues that will be addressed during the teaching session. The problem often serves as the central focus or starting point for learning in a class. It may involve presenting a real-world problem, a theoretical question, or a practical situation that students need to understand, analyze, or solve. Thus, in the "The Driver Robot" instance, the problem is to make the robot move and navigate obstacles, simulating an autonomous vehicle.

The experts' considerations on criterion 04 kept the students' material the same; however, additional information about the issue was included in the teacher's material and the validation guidelines for the experts.

Criterion 05 aspires to identify the prerequisite knowledge for this class, i.e., the knowledge the student should have acquired through their past experiences, which will be necessary to solve the proposed problem. Table 5.8 summarizes the experts' considerations of criterion 05.

Table 5.8: Expert Insights Transcription - Criterion 05

Instance	Expert	Translated Freely
The Cleaning Robot	1	No considerations.
	2	It is unclear what prior knowledge the student needs to possess to advance in this class.
	3	We identified geometric shapes such as squares and equilateral triangles and discussed angles in the content. However, the role of the triangle, particularly the equilateral one, needed to be clarified
	4	Algorithms and triangles.
	5	No considerations.
	6	No considerations.
The Driver Robot	1	No considerations.
	2	It is not possible to identify the previous concepts. A paragraph could be included to provide clarification on this matter.
	3	Concept of unlimited, echolocation.
	4	Algorithms.
	5	No considerations.
	6	I didn't find it straightforward. In the second paragraph of the problem statement, you reinforce the idea of prior knowledge, but you don't specify it.

The experts demonstrated difficulty visualizing the prerequisite knowledge required for students to advance in the lessons. This challenge is apparent both in the agreement analysis (see Graph 5.2 a)) and in the considerations made by the experts. In the instance "The

Cleaning Robot, " Expert 2 considered that it is unclear what this knowledge entails. Experts 3 and 4 understood that it is related to geometric figures, aligning with what was deliberated during the instantiation process. However, Expert 4 signaled that algorithms would also be prerequisite knowledge, which does not align with reality, as algorithms are the knowledge provided as input, intended to be taught and learned in this instance.

Figure 5.5: The Cleaning Robot instance prerequisite knowledge in the first round.

This discrepancy in understanding may be a consequence of the nature of this criterion, as visualizing prerequisite knowledge requires the expert to think about the role of a teacher and understand what they need to carry out the activities. Possibly, the experts expected explicit terminology when alluding to this knowledge; however, this information becomes more evident when considering the teacher's material. Although the student's material encompasses it, there is no explicit nomenclature. Therefore, a slight adjustment was made to the text of the "The Cleaning Robot" instance (see Figures 5.5 and 5.6).

Figure 5.6: The Cleaning Robot instance prerequisite knowledge after modification.

The same difficulty was demonstrated in the “The Driver Robot” instance. Hence, a modification was made to the student’s material to facilitate visualization and class conduct (see Figures 5.7 and 5.8). This modification is considered relevant because if the experts cannot visualize the prerequisite knowledge, it is possible that the class’s conduction may not correctly align with the CTProgER guidelines. Therefore, these considerations helped shape how the instances should address prerequisite knowledge to minimize discrepancies between the CTProgER guidelines and what is executable in the classroom.

Criterion 06 aspires to identify the programming structures that must be implemented to solve the proposed problem. Table 5.9 summarizes the experts’ considerations of criterion 06.

Some experts demonstrated difficulty visualizing the programming structure in “The Cleaning Robot” instance. This challenge is apparent both in the agreement analysis (see

Figure 5.7: The Driver Robot instance prerequisite knowledge in the first round.

Figure 5.8: The Driver Robot instance prerequisite knowledge after modification.

Graph 5.2 a)) and in the considerations made by the experts. In the “The Cleaning Robot” instance, the problem revolves around an introduction to algorithms, pinpointing the necessary prior knowledge for this content is not straightforward. Similarly, recognizing the programming structures in the C06 criteria poses a challenge as this lesson does not employ a specific programming language but rather pseudo-code. Although the experts explicitly highlighted difficulties in writing, they specified aspects such as input-processing-output, moving forward, and rotations. While moving forward and rotations may not be common programming structures, robots must perform movements in robotics, which are considered fundamental structures in robot programming. All attempts by the experts to indicate these structures are in conformity with what was considered during the instantiation process.

Table 5.9: Expert Insights Transcription - Criterion 06

Instance	Expert	Translated Freely
The Cleaning Robot	1	No considerations.
	2	It is unclear what the structures would entail. After employing natural language, I could associate with the input-processing-output concept.
	3	The structures are forward and rotate.
	4	Walk X steps and turn Y degrees.
	5	I did not identify the dimensions of Miguel’s room in the text.
	6	No considerations.
The Driver Robot	1	No considerations.
	2	Yes, but I emphasize the importance of explaining how pseudo-code works or indicating that this would be necessary prior knowledge.
	3	Decision structure, loop (in unlimited forward).
	4	Conditionals and data input and output.
	5	No considerations.
	6	I couldn’t identify it.

Conversely, the experts demonstrate greater ease in identifying the programming structures that the "The Driver Robot" instance encompasses. Control structures (condition and repetition) were considered; however, the focus of this instance is necessarily on conditions. Expert 3 highlighted the repetition control structure, associating it with the robot walking indefinitely forward. A valid consideration given the nature of "walking indefinitely"; however, in robotics, walking indefinitely, especially in block-based programming languages, is a repetition that remains transparent to the programmer. It is treated uniquely with a block that automatically repeats itself, so the repetition structure may not necessarily be visualized through walking indefinitely but rather through loops that allow repeating various blocks. Yet, the expert's perspective does not invalidate the purpose of the class, as the student invisibly deals with repetitions.

Criterion 07 aspires to identify the ways to test the solution to be implemented. Despite the CTProgER incorporating a validation didactic moment, the guidelines it provided in the student instance for testing their solution may need to be clarified. Table 5.10 summarizes the experts' considerations of criterion 07.

Table 5.10: Expert Insights Transcription - Criterion 07

Instance	Expert	Translated Freely
The Cleaning Robot	1	No considerations.
	2	To facilitate the testing of the solution for the student, they could use a sequence of images for each instruction, creating a sequential simulation of the robot's movement.
	3	Yes, a straightforward way to test the solution is evident. It is important to emphasize that algorithms are not unique; there can be more than one way to solve a problem and more than one way to test it.
	4	No considerations.
	5	In the testing environment for robots.
	6	No considerations.
The Driver	1	No considerations.

Robot	2	The activities section could have been more straightforward for me, as it required transforming pseudo-code (which was not clearly taught) into a programming language (which is also not explained in the script).
	3	No considerations.
	4	In the testing environment for robots.
	5	No considerations.
	6	No considerations.

In the "The Cleaning Robot" instance, the steps the robot should follow to achieve the goal were visually inserted. Figure 5.9 illustrates how the instance was initially conceived and how it was structured after the modification, while Figure 5.10 demonstrates the structure of the instance after the change.

Meanwhile, in the context of "The Driver Robot" instance, additional information has been incorporated to provide enhanced guidance to the students throughout the testing phase of their solution. Figure 5.11 illustrates how the instance was initially conceived and structured after the modification, while Figure 5.12 demonstrates the structure of the instance after the change. Also, Expert 2 emphasized the challenge of comprehending the transformation of pseudocode into a programming script. It is crucial to underscore that the instance was crafted to convey the logical concept, allowing the instructor the flexibility to select the programming language and supplement the instruction of its syntax. Alongside the adjustments made to the student material, instructions were included for adjudicators to conduct the second round of validation.

Criterion 08 aspires to identify how the student is evaluated in this class. Despite the CTProgER incorporating a validation didactic moment, the guidelines it provided in the student instance for testing their solution may need to be clarified. Table 5.11 summarizes the experts' considerations of criterion 08.

According to expert considerations, it became evident that students should respond to a set of questions, particularly when considering the students' material. Additionally, in a complementary manner, the instructor should conduct observations following the guide-

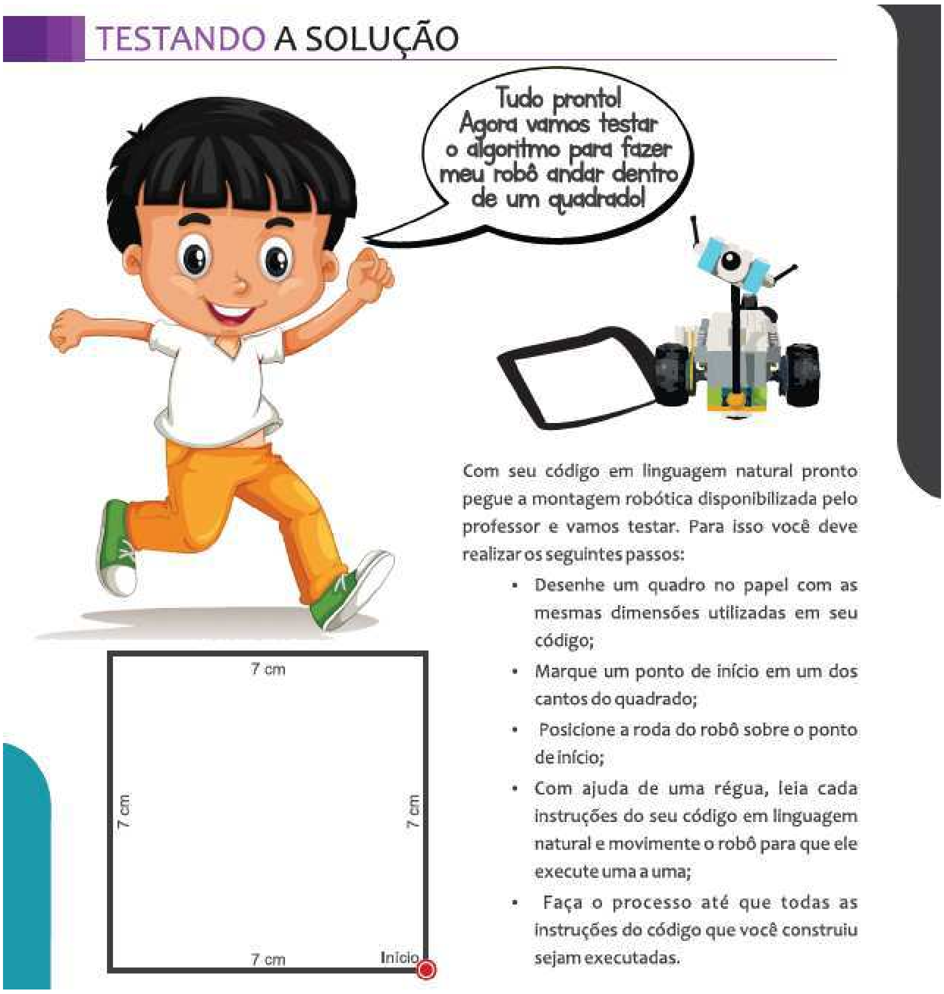


Figure 5.9: The Cleaning Robot instance prerequisite knowledge in the first round.

lines in the instructor's material. However, the insights from expert 3 in both instances prompt a reflection on the formulation of evaluative questions. Merely posing questions with binary answers such as "yes" or "no" is deemed insufficient, especially considering that the CTPRogER is grounded in learning theories to foster logical and critical reasoning development. Consequently, modifications were implemented in the question structures, encouraging students to contemplate actions and provide a more in-depth explanation of the assimilated knowledge.

In the instance "The Cleaning Robot," the question was posed: "Did your natural language code enable the robot to move within a square as Miguel required?" This question could easily be answered with a "yes" or "no"; however, the intention is to prompt the student to indicate the shape of the robot's trail, which could signify whether or not it is progressing

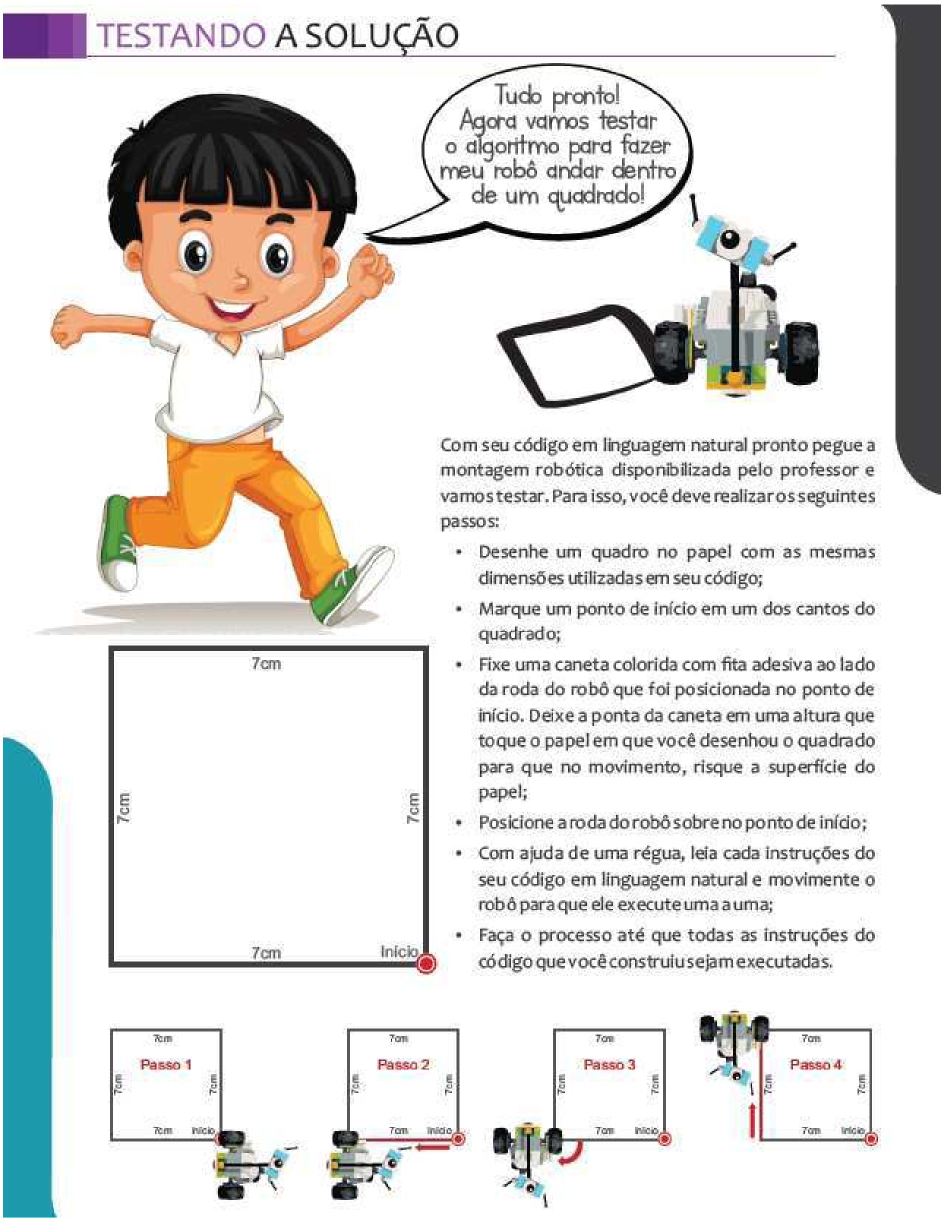


Figure 5.10: The Cleaning Robot instance prerequisite knowledge after modification.

toward solving the problem. Therefore, the question was modified to: “What shape did the pen attached to the robot draw on the paper after the execution of your test? Did the drawn shape align with what Miguel needed?”

Table 5.11: Expert Insights Transcription - Criterion 08

Instance	Expert	Translated Freely

Robot The Cleaning	1	No considerations.
	2	No considerations.
	3	Regarding the responses to the questions, the first one seeks a simple 'yes' or 'no' answer. Is this the only information you want from the student at this stage? If you anticipate additional details, the question needs to be clarified. Regarding the third question: is the intention to have the student rebuild the code similar to the initial question? (I didn't grasp the purpose.) Or is the objective for the student to write code for the rectangle on the side?
	4	Building on the assimilation of theory (algorithmic concepts and triangles), practice (creating the algorithm that cleans the room based on the concept of moving and turning), and the testing of the solution.
	5	No considerations.
	6	No considerations.
The Driver Robot	1	No considerations.
	2	I continue to emphasize the need for refinement in the pseudocode and the robot's programming language.
	3	In the first question, is your objective for the student to write 'Walk without limit'? In question 2, calculating the distance may be challenging; perhaps the teacher has to put in more effort. In the 'testing the solution' section, for question 1, do you want the student to respond with only 'yes' or 'no'?
	4	Assimilation of theory, implementation, testing and validation.
	5	No considerations.
	6	No considerations.

Finally, criterion 09 identifies the knowledge the student should obtain at the end of

Figure 5.11: The Driver Robot instance prerequisite knowledge in the first round.

this class. Table 5.12 summarizes the experts' considerations of criterion 09. The experts' considerations on this criterion reinforced the quantitative assessment of positive agreement presented in Graph 5.2. In both instances, the considerations aligned with the inputs provided at the outset of the CTProgER process, demonstrating that the inputs and outputs of the educational process were conformity addressed in the instantiation process.

Table 5.12: Expert Insights Transcription - Criterion 09

Instance	Expert	Translated Freely
The Cleaning	1	No considerations.
	2	I can identify that the knowledge of "initial concepts of algorithms" can be acquired by the end of this class.

Robot	3	No considerations.
	4	I was curious whether the knowledge revolved around the explicit concepts presented regarding algorithms and triangles or the ability to draw parallels on how technological solutions can perform tasks.
	5	No considerations.
	6	No considerations.
The Driver Robot	1	No considerations.
	2	Yes, it is possible to perceive that the student should comprehend decision structures.
	3	No considerations.
	4	Understand how machines read the world and make decisions.
	5	No considerations.
	6	No considerations.

These considerations play a pivotal role in refining the instances and mitigating potential issues during interventions while evaluating the effectiveness and conformity of CTProgER. Incorporating expert feedback, adjustments have been made to both instances, followed by a subsequent round of validation. Moreover, these considerations are imperative for enhancing instances and preempting challenges in future interventions, aligning with the observed efficiency of CTProgER from a learning perspective.

5.2.3 Delphi's Second Round for Validating CTProgER Instances

On analyzing the second round of instance validation, an initial step involved the examination of a majority vote among experts, adhering to a criterion wherein validation required $>$ 50% plus one of positive answer for each question. The analysis was conducted on a per-instance basis. As depicted in Figure 5.13, the data reveals a predominant trend of experts casting positive votes exceeding 50% plus one of positive answer for each criterion within the Barema, particularly evident in "The Cleaning Robot" Instance and "The Driver Robot" Instance.

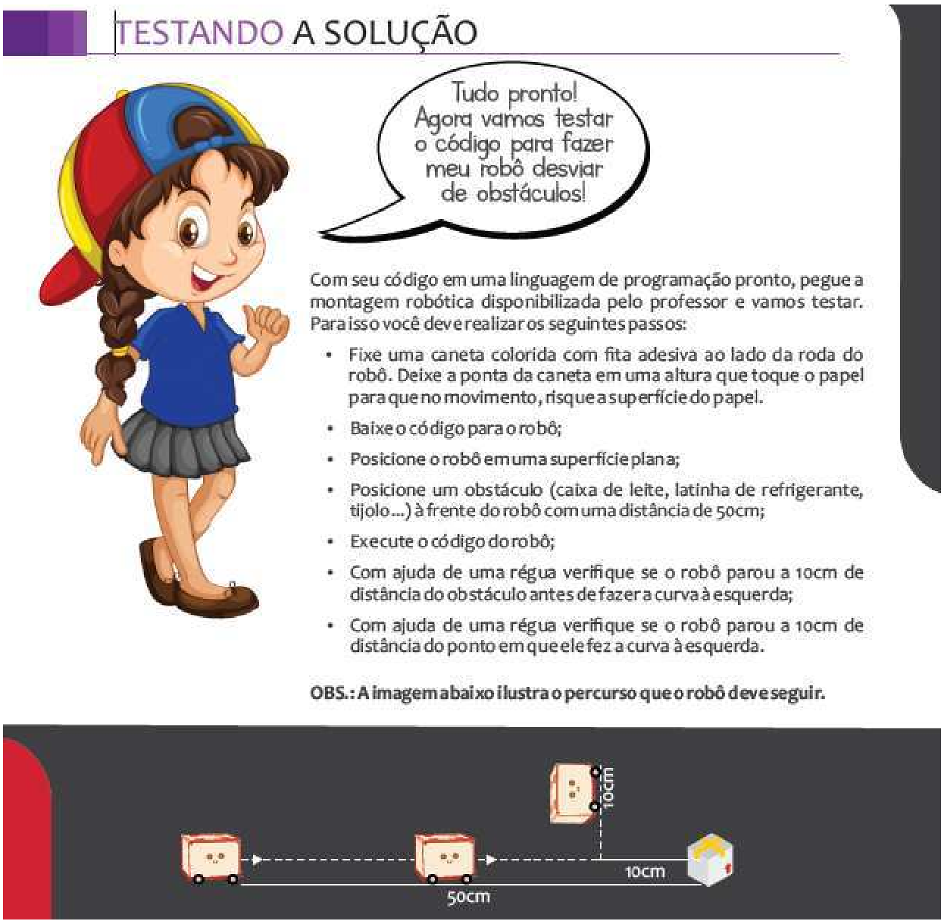


Figure 5.12: The Driver Robot instance prerequisite knowledge after modification.

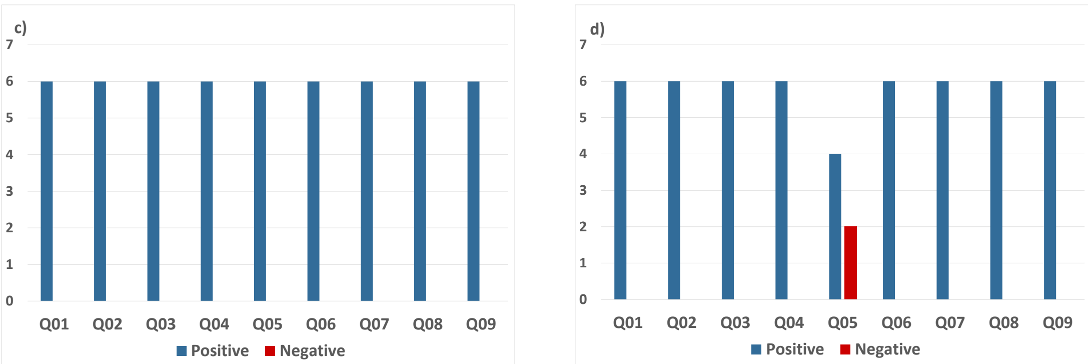


Figure 5.13: c) Experts' validation data by version 2 to The Cleaning Robot Instance. d) Experts' validation data by version 2 to The Driver Robot Instance.

According to Figure 5.13 (c), the positive majority voting is 100% from in all criteria "The Cleaning Robot" Instance. According to Figure 5.13 (d), the positive majority voting is 100% (5) in Q01, Q02, Q03, Q04, Q06, Q07, Q08 and Q09 criteria, and 66.7% (4) in Q05 from "The Driver Robot" Instance.

Table 5.13: Percentage of absolute agreement in round 2

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

Furthermore, it computed the Percentage of Absolute Agreement (PAA) among the experts, and Table 5.13 presents the PAA scores per instance. Notably, it observed a PPA of 100% and an average improvement in agreement of 22.2% in "The Cleaning Robot" Instance compared to the initial round. In "The Driver Robot" Instance, a PPA of 88.9% was noted, with an average improvement in agreement of 11.1% from the first round. Positive majority voting was evident in both instances, and the PAA exceeded 70%. Consequently, it asserts that the instances received validation during the second round of expert assessment.

In the first round of expert evaluation, criterion 05 was analyzed, and there was no disagreement regarding the opinions presented (see Figures 5.2 and 5.13). However, a discrepancy in the assessments was observed in the second round. Despite this difference, it is crucial to emphasize that the PAA continues to be met, and thus, the result is considered positively validated.

The discrepancy between the evaluation rounds can be attributed to various factors. Firstly, interpretation and subjective judgment issues may influence the experts' opinions. Additionally, it is possible that the experts in the second phase of the evaluation process did not fully understand modifications made to the instances during the rounds.

It is essential to acknowledge the inherent complexity of expert evaluations and the potential for interpretation variations, even when objective criteria are established. Despite the discrepancy, the ongoing compliance with the PAA underscores the robustness of the result obtained in the first round and the importance of considering various factors that can influence expert assessments.

Hence, upon examining the outcomes derived from the instances' validation process, as assessed through the lens of experts specializing in teaching programming, it addresses RQ2. The findings led to rejecting the null hypothesis, denoted as *H2.0:* There is no evidence that the educational process of teaching and learning programming with ER can conform to an expert's perspective, denoted as *H2.1:* The educational process of teaching and learning programming with RE can conform to an expert's perspective.

Based on expert evaluations, these results suggest that an educational process centered around teaching and learning programming with ER may be in conformity with the educational process guidelines. Conversely, the findings indicate that an ER educational process will likely demonstrate more conformity, as experts perceive. This conclusion underscores the significance of incorporating expert perspectives in evaluating educational methodologies and informs potential avenues for optimizing the teaching and learning of programming within the context of ER.

5.3 Threats of Validity

This study is subject to specific threats to validity that warrant careful consideration. One significant concern pertains to the potential lack of suitability of the CTProgER didactic moments for the High School target audience. This incongruence could undermine the validity of the instances' validation process. Another point of concern is the potential need for more clarity in the criteria outlined by Barema, raising questions about the interpretability of these criteria. The experts' responses, therefore, may not necessarily align with the specific insights sought by the author of this study, introducing a potential source of bias that could compromise the validity of the instances' validation process.

The expert assessment may threaten validity because all selected experts come from the same post-graduate program. This concentration may introduce bias or limit the diversity of perspectives in the evaluation process. It is important to note that the experts were chosen in this context due to the lack of volunteers for this specific task, which led to the need to select them from the available members of the graduate program.

Furthermore, it is worth noting that the adopted workflow, particularly the instances' validation step, may have been extensive. This extensive nature could have influenced the responses provided by the experts, possibly impacting the overall robustness of the validation outcomes. Future investigations and refinements to the methodology may be necessary to address these validity concerns and enhance the generalizability of the findings. Despite these challenges, it is essential to acknowledge and address these potential threats to pursue a more nuanced and comprehensive understanding of the study's outcomes.

5.4 Summary of Chapter

This Chapter delineates an expert assessment as part of Educational Design Research conducted to evaluate the efficacy of CTProgER from an expert perspective (teacher). The expert assessment employed the Delphi method [136] to validate artifacts through expert judgment. The primary aim was to assess the conformity of the CTProgER educational process. Expert (programming teachers) judgment was sought, who, utilizing the Delphi method, analyzed CTProgER instances to evaluate their adherence to guidelines and alignment with actual programming classes. The CTProgER educational process was conceptualized based on established methodologies for teaching with ER, incorporating insights from the Anthropological Theory of Didactics and drawing upon evidence from prior ER studies [65; 66; 103; 170; 171].

Within the CTProgER educational process, an Institution [I] is the teaching practice locus. Given an input (input), the experience of didactic moments in the teaching of programming through ER enables the development of a relation R(X,O) as the output, with Object (O) being a feasible outcome. The didactic moments encompass problematic, analysis, discovery, implementation, validation, and assessment. The proposed educational process must undergo thorough validation. The plan involves face-to-face studies conducted in a representative Institution [I] to validate CTProgER from the experts' perspective, i.e., the teaching standpoint. The instances' validation, carried out through the Delphi method, contributed to gauging the instances' alignment with CTProgER guidelines. Thus, the concordance between instantiation and guidelines was validated through expert evaluation.

The second round of validation played a crucial role in refining the instances and preempting potential issues in future interventions. This iterative process is integral to substantiating the CTProgER validity and effectiveness in subsequent studies. Validating CTProgER instances through the Delphi method demonstrated the alignment between instantiation and guidelines based on expert evaluation. The instances' validation marked the initial phase of CTProgER validation, aligning the instances with CTProgER guidelines. The data from the first round informed necessary adjustments to enhance the concordance of the instances with the CTProgER educational process guidelines in the subsequent round.

The subsequent Chapter will present the outcomes of an intervention experiment conducted in a typical high school setting, aiming to validate CTProgER from the student learning perspective.

Chapter 6

Analyzing the CTProgER Effectiveness Through a High School Intervention

This chapter introduces an intervention study, a pivotal component of Educational Design Research, within the basis of this doctoral thesis. Its purpose is to evaluate the effectiveness of CTProgER as perceived by students. The intervention study specifically targeted Computer Technical and Vocational High School students, focusing on teaching-learning programming through CTProgER. The primary goal of this study is to enhance the teaching-learning of programming with an emphasis on CT skills. Additionally, the intervention study seeks to generate empirical evidence shedding light on the impact of teaching-learning programming through CTProgER on the CT skills of High School students specializing in Computer Technical and Vocational studies. This investigation also seeks to validate the effectiveness of CTProgER from the students' standpoint. This Chapter refers to the third research question (**RQ3:**) posed in this thesis: "How effective is the impact of the educational process of teaching-learning programming with ER on students' CT skills in High School?" The intervention study was designed with the following objectives in mind:

- **OBJ5:** Conducting an intervention experiment in High School to apply the educational process and instance together;
- **OBJ6:** Analyzing the educational process's effectiveness to High School students.

6.1 Research Design

This intervention study aims to assess the effectiveness of CTProgER from the student's perspective. The design of this intervention was elaborated through a two-step process: (1) Executing an intervention experiment within a High School setting to apply the CTProgER guidelines and instance concurrently; (2) Evaluating the effectiveness of the CTProgER for High School students. The formulation of the following research questions and hypotheses guided the progression of this intervention study:

(RQ3) How effective is the impact of the educational process of teaching-learning programming with ER on students' CT skills in High School?

- *H.0:* There is no evidence that the educational process of teaching-learning programming with educational robotics positively impacts the students' CT skills in High School.

- *H.1:* The educational process of teaching-learning programming with educational robotics impacts the students' CT skills in High School.

In an intervention study, various variables require observation and control, including height, age, population rate, or exam scores. Typically, the goal is to investigate the impact of one variable on another. For instance, a study may examine whether students who spend more time studying achieve higher exam scores. Independent and dependent variables are crucial in delineating a cause-and-effect relationship within a study. The independent variable serves as the cause, and its value remains unaffected by other variables in the study. On the other hand, the dependent variable represents the effect, and its value is contingent upon modifications in the independent variable.

Table 6.1 illustrates the design employed in this intervention study conducted by the principal investigator. This design encompasses two distinct student groups, namely the control and experimental groups, characterized by comparable profiles. Both groups engaged with identical programming and robotics content during the intervention; however, the experimental group's intervention adhered to the CTProgER guidelines for teaching-learning programming through educational robotics.

In contrast, the control group's interventions involved conventional teaching-learning programming, wherein the teacher presented the content in an expository manner. Subsequently, students attempted to program the robots based on the provided content. Several variables are integral components of this design:

- High School (HS): This independent variable represents the actions performed during the school year from High School;

- Educational Robotics (ER): This independent variable represents the ER use as a teacher instrument;

- Programming Teaching (PT): This independent variable represents the programming teaching;

- Educational Process: This independent variable represents the use of all guidelines and lessons proposed in this thesis proposal;

- CT Skills Performance (CT): This dependent variable represents students performance in CT tests. The CT test is from the tool or instrument to measure CT skills.

The control and experimental groups were exposed to programming and robotics education, sharing a common foundation of knowledge. The sole difference between these two groups is that the experimental group had the additional CTProgER, whereas the control group was not exposed to this specific intervention. This distinctiveness provides an opportunity to assess and compare the impacts of CTProgER on the group that received conventional programming and robotics education alone.

Table 6.1: Intervention Experiment Design

Groups	Independent Variable	Dependent Variable
Experimental	HS + ER + PT + CTProgER	CT1
Control	HS + ER + PT	CT2

Initially, a profile survey must be administered to obtain data that facilitates understanding the composition of the participating students' profiles. After, a CT test was applied to both groups before (pre-test) and after (post-test) the three lessons. This test assessed the students' CT skills in both groups before and after the programming lessons. Hence, the aim was to observe and analyze the impact of CTProgER on enhancing students' CT skills.

Each lesson was delivered simultaneously to both groups at different times by the same instructor, with the only distinction being the teaching methodology. The experimental group adhered to the CTProgER guidelines, while the control group followed traditional instructions, as detailed earlier in Chapter 4. Before each lesson, a programming pre-test was administered, consisting of pseudocode questions related to the lesson content, specifically, the input object in CTProgER. These questions did not require the students to possess specific knowledge of programming language syntax but rather a logical understanding of computation-related problems, considered inherent concepts within CT.

Following the completion of each lesson, a programming post-test with the same questions as the programming pre-test was conducted to assess any changes in the understanding of the lesson content. Each lesson was designed for 1 hour and 30 minutes; however, additional time may be necessary depending on the student group. Nonetheless, by CTProgER guidelines, a lesson should encompass didactic moments within a time frame that can start and finish on the same day or on different days. In other words, a lesson may span one or more classrooms, provided that, upon conclusion, the necessary elements are in place to observe the existence of a relation (equal or different from zero) between the subject (student) and the object (content), denoted as R(X,O), as described in the CTProgER guidelines in Chapter 4.

The execution of each lesson is documented through audio recording in the classrooms, a necessary artifact for a qualitative analysis of the behavior of student groups and a deeper understanding of the variables influencing teaching and learning within the scope of the lessons. The intervention design is depicted in Figure 6.1.

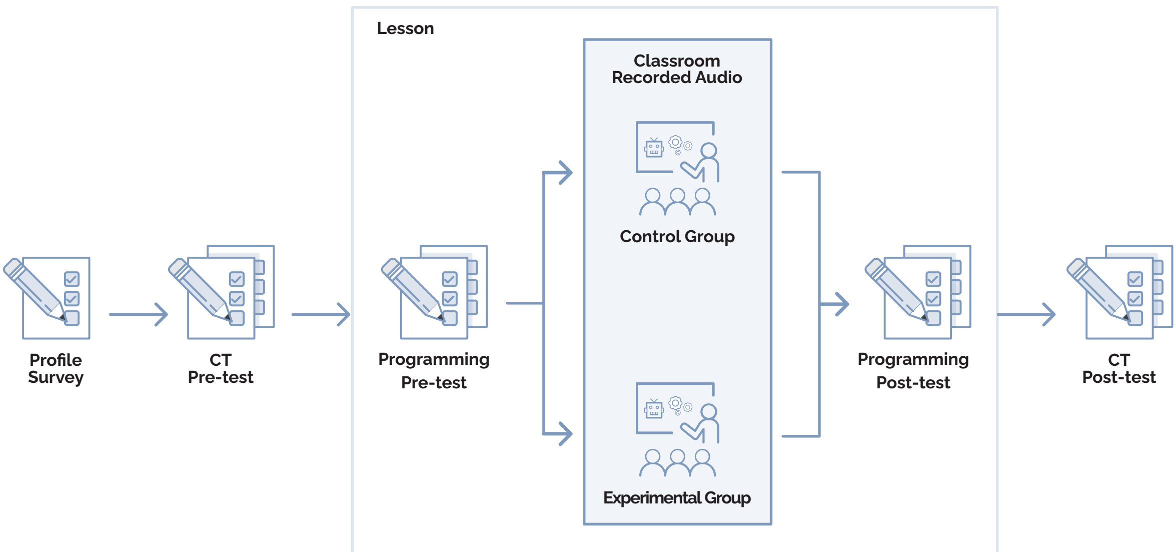


Figure 6.1: Intervention Lesson Design

The author of this thesis taught the intervention classes, a decision made due to limitations in finding available teachers to appropriate the methodology within the context of the selected school for intervention implementation. This approach was adopted to ensure consistent implementation of the educational process proposed, allowing for more direct control over the teaching and learning process during the intervention. While it was a pragmatic choice, this approach also provided the author with a unique opportunity to be deeply involved in the development and execution of the experiment, enabling a deeper understanding of the challenges and opportunities encountered along the way.

6.1.1 Instruments

The student **profile survey** aspires to comprehend the composition of the participating students' profiles through 10 subjective and objective questions about aspects such as age, nature of primary education, academic repetitions, subjects of interest, and computer science and robotics exposure. This survey included a question regarding the students' names because, in this study, the student's progress throughout the intervention was analyzed. However, as Section 1.6 of Chapter 1 outlined, each student was assessed anonymously, adhering to all rules stipulated by the university's code of ethics.

The **Román-Gonzalez Computational Thinking Test**, as outlined by Román-Gonzalez *et al.* [87], served as the instrument for collecting CT skills data post-intervention. The CT test comprised 28 questions distributed across 12 pages, with an average of three questions per page. Each question presented four answer alternatives (A, B, C, and D), with only one correct option. Participants were allotted up to 45 minutes from the commencement of the test to respond to the questions; however, there was no stipulation that all questions must be answered.

It is worth noting that the CT test employed in this study was developed by Professor Doctor Román-Gonzalez from the Universidad Nacional de Educación a Distancia (UNED). The translation and adaptation of the test were carried out by researchers Rafael Marimon Boucinha and Christian Puhlmann Brackmann, with due authorization for its use in this study granted by them.

The **programming pre-test and post-test** comprise pseudocode questions that can be logically solved by applying the concepts covered in each lesson. Example questions in pseudocode are illustrated in Code 6.1. 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. As a result, this evaluation instrument can be readily utilized by teachers or researchers for replication purposes.

Código Fonte 6.1: 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
```

All instruments employed in this study are available online[1] on the repository of this thesis.

6.1.2 Robotics Kit

The LEGO® Mindstorms NXT Robotics Kit (see Figure 6.2) was employed in the intervention as an educational tool designed to impart fundamental principles of programming, engineering, and automation engagingly and practically. The LEGO® Mindstorms NXT kit comprises various components, including programmable bricks, sensors, and motors, all crafted to enable the creation of fully functional robots.

The programmable brick of the robot, commonly called the "intelligent" or "brain", is

[1]thesis Repository: https://github.com/isabellelimasouza/CTProgER_Tese_Doutorado.

Figure 6.2: LEGO® Mindstorms NXT Robotics Kit

programmed through a user-friendly interface utilizing block-based programming. This feature allows users, including beginners, to develop code for controlling the movements and interactions of the robot.

The sensors integrated into the kit are crucial for the robot's interaction with its environment, potentially including touch, color, light, and ultrasonic sensors. The motors, in turn, provide the robot with the capability for movement, enabling the execution of various tasks. The versatility of the LEGO® Mindstorms NXT kit makes it an exceptional choice for exploring concepts in STEAM, providing a practical and playful experience in developing fundamental skills.

For robot programming, a block-based programming language - NXT - was employed using the LEGO® MINDSTORMS Edu NXT 2.0 software (see Figure 6.3) in this doctoral thesis in computer science. This software, developed by National Instruments LabVIEW, is highly intuitive, with programming carried out through configurable blocks. Programs created using the LEGO® MINDSTORMS Edu NXT 2.0 can be transferred to the NXT via USB or Bluetooth.

6.2 Data Collection Procedures and Data Clippings

The data collection for the intervention took place from August to November 2022, encompassing quantitative and qualitative data on students' cognitive development before, during,

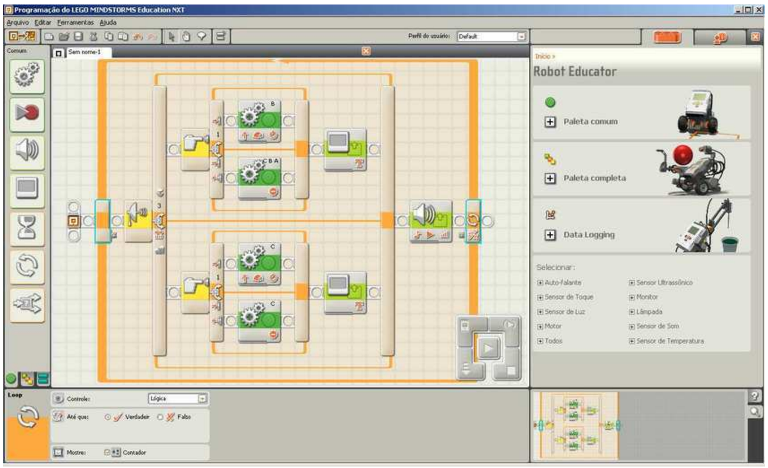


Figure 6.3: LEGO® MINDSTORMS Edu NXT 2.0 Software Overview

and after the intervention. This information was acquired through the administration of a profile survey, the CT Test developed by Román-Gonzalez *et al.* [87], and programming tests.

Student data was organized for statistical analysis to provide a comprehensive overview of High School, considering students from all years and individual years. Thus, four data clippings of students were considered, as specified in Table 6.2.

Table 6.2: Classification of Student Clippings Based on High School Grade for Data Analysis

Data Clippings	High School Student Grade
Clipping 1	1st grade
Clipping 2	3rd grade
Clipping 3	1st and 3rd grade
Clipping 4	1st, 2nd, and 3rd grade

Due to the pandemic, which hindered schools where this study was applied from effectively delivering computer science content in 2020 and 2021, students from different High School years (2nd and 3rd grade) could be comparably assessed to those in the 1st year regarding CT skills. This feasibility arose because students in the 1st, 2nd, and 3rd grades operated at an equivalent knowledge level. The limitations imposed by the situation created a scenario where the content was not effectively covered.

While school reports and classroom observations are valid, the distribution graphic of pre-test performance across the three High School grades (1st, 2nd, and 3rd grades) involved in the intervention was analyzed. Figure 6.4 reveals that the frequency of students with above-average performance (highlighted by red lines) tends to be higher in the 1st grade, indicating beginner-level computing students. In contrast, the averages for the other classes are slightly lower. However, the density chart shows a similarity in the distribution of students across all three grades, as their areas are similar and largely overlap.

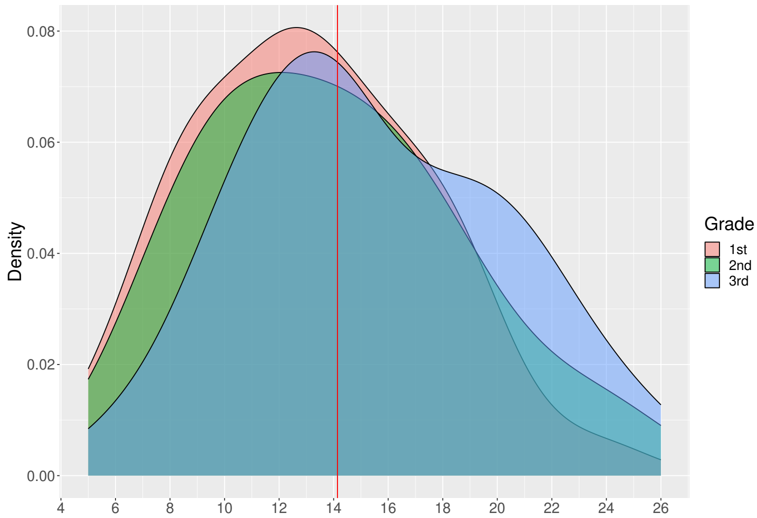


Figure 6.4: Density Distribution of Student Performance in the CT Pre-test Across Grades

The descriptive statistical analysis of the pre-test scores for each grade, considering the mean, median, and standard deviation, revealed that the 3rd grade has an average score of 15.49. In contrast, the 1st grade has 13.14, and the 2nd grade has 13.93. On the other hand, the median shows minimal variation among the grades. However, the 3rd grade has a higher mean and median and exhibits a larger standard deviation (see Table 6.2).

A larger standard deviation in descriptive statistical analysis may indicate more significant variability or dispersion of the data about the mean. In simpler terms, when the standard deviation is large, individual scores deviate further from the mean, suggesting greater heterogeneity or dispersion in the data. This information about the standard deviation may imply that the data is more diverse or that there is a wide range of scores within the sample, potentially impacting the interpretation and consistency of results.

In contrast, a smaller standard deviation suggests that the scores are closer to the mean, indicating less dispersion and greater consistency in the data. Hence, it is observed that the

3rd and 2nd grades may exhibit more heterogeneity, while the 1st grades show less heterogeneity.

Table 6.3: Mean, Median, and Standard Deviation of Grades: Descriptive Statistics Analysis

Grades	Mean	Median	SD
1st	13.14	13.00	4.30
2nd	13.93	13.50	4.76
3rd	15.49	14.00	4.83

Additionally, a comparison was conducted between the mean and median differences across each grade. Specifically, the 1st grade was compared with the 2nd, the 1st with the 3rd, and the 2nd with the 3rd. The corresponding data is presented in Table 6.2. The occurrence of negative values in the data is attributed to the orientation of the grades in the calculation. Consequently, it is evident that in all comparisons, the grade in the second position in the difference operation exhibits both higher mean and median values than the grade positioned in the first place.

Table 6.4: Difference Comparison Between Mean and Median Across Series

Grades Comparison	Mean Difference	Median Difference
1st - 2nd	-5.69%	-3.70%
1st - 3rd	-15.17%	-7.14%
2nd - 3rd	-10.06%	-3.57%

In the CT context, it is crucial to recognize that minimal differences in mean and median scores between student groups do not necessarily imply that one group is superior. When assessing computational skills, particularly in educational settings, it's essential to consider the nuanced nature of these abilities.

A small difference in mean and median scores may indicate a statistically relevant variance. Still, the practical significance of such differences in skill development is a critical aspect to ponder. CT is a multifaceted skill, and minor variations in test scores may not unequivocally translate to one group being better or worse than another.

In cognitive abilities, especially when dealing with skills like CT, the significance of minor differences should be interpreted cautiously. Skills development is a gradual process, and a marginal disparity in test scores may not indicate substantial disparities in overall proficiency. Hence, it was deemed that students across the grades exhibit similarities in their CT skills and can be collectively grouped according to clippings da Tabela 6.2.

The data collection presented in this section does not differentiate between the control and experimental groups. These exploratory analyses were conducted solely to understand the similarity between the groups and to explore the possibility of grouping them in clippings. Soon, data specific to the experimental and control groups will be presented.

6.3 Data Analysis Procedures

The statistical analysis of the data employed the R programming language and the RStudio software, facilitating the application of hypothesis tests to evaluate the normality and homoscedasticity of the data. These are standard criteria for selecting appropriate statistical tests for the analysis of each scenario. Statistics support data analysis with parametric and non-parametric hypothesis tests based on the parametrization. Parametric hypothesis tests require the fulfillment of assumptions, including normality (the sample should come from a population with a normal distribution), homoscedasticity or homogeneity (the sample should exhibit equal variance among the evaluated groups), and continuity and equality of intervals. On the other hand, non-parametric tests demand fewer assumptions for their application, serving as an alternative when parametric assumptions are not met. Additionally, non-parametric tests are recommended for tiny samples and analyses involving ordinal measures.

When assessing the statistical assumptions of normality and homoscedasticity using the Shapiro-Wilk and Levene hypothesis tests, respectively, with a significance level $\alpha = 0.05$ (see Table 6.5), it was possible to reject, with 95% confidence, the null hypothesis that the data follows a normal distribution in 1st, 2nd, and 3rd grades. These grades exhibited a *p-value* in the Shapiro-Wilk test greater than α in some data groups' CT pre-test and post-test. The results indicated that not all data from the analysis cuts met the normality and homoscedasticity assumptions (CT pre-test and CT post-test). Thus, it identified the need for independent statistical tests for the data parametrization.

Therefore, to assess whether there is a significant difference in the performance in CT be-

Table 6.5: Statistical Assumptions (*p-Values* from Normality and Homoscedasticity Tests for Computational Thinking Test Data Analysis)

Data Clippings	CT Pre-Test		CT Post-Test	
	Shapiro	Levene	Shapiro	Levene
Clipping 1	0.6300	0.7717	<0.05	0.9572
Clipping 2	0.5538	0.8502	0.0523	0.5362
Clipping 3	0.3030	0.9003	<0.05	0.9449
Clipping 4	0.1488	0.9608	0.0071	0.9594

tween the experimental and control groups, considering the data groups, the non-parametric hypothesis test, the paired and unpaired Mann-Whitney (U) test was chosen based on the CT Test results. To calculate and analyze the effect of CTProgER on CT skills, Cohen's effect size index (d) was used [120]. The tests were conducted with a confidence level 95% and statistical significance set at $\alpha = 0.05$. Finally, audio recordings of the classes were transcribed and qualitatively analyzed to analyze behavioral aspects.

6.3.1 Sample and Participants' Profile

This study considered a sample of 93 students from the Computing Technical and Vocational High School in the Paraíba state. The students in each grade were divided into experimental and control groups according to the school's organizational structure, as the lessons took place within the High School curriculum. The school had two classes for the 1st grade, one class for the 2nd grade, and two classes for the 3rd grade. In this context, for the grades composed of two classes, one class was randomly chosen as the control group and the other as the experimental group. Meanwhile, the 2nd grade was randomly selected as the experimental group. The number of students in the experimental and control groups based on class allocation is displayed in Table 6.6.

To assess the representativeness of the student sample (93 students), it utilized the sample size calculation outlined in Equation 6.1, where n represents the calculated sample size, N is the population size, Z denotes the standardized normal variable associated with the confidence level, p signifies the true probability of the event, and e denotes the sampling error.

Table 6.6: Quantity of Students in the Experimental and Control Groups Based on Class Allocation

Class	Group	Universe	Sample
1st A	Control	24	23
1st B	Experimental	21	21
2nd	Experimental	19	14
3rd A	Experimental	21	20
3rd B	Control	16	15
Total		**101**	**93**

$$n = \frac{N \cdot Z^2 \cdot p \cdot (1-p)}{Z^2 \cdot p \cdot (1-p) + e^2 \cdot (N-1)} \tag{6.1}$$

In all clipping sets, the sampling error was less than 4% (see Table 6.7); however, clipping 4 was considered the reference parameter for the sample of 93 students. Assuming the student population size of 101 students, the sample size calculation results suggest that the selection of 93 students is representative of an error (the difference between estimated and actual numbers) of 2.87% at a 95% confidence level (the probability that the useful sampling error is less than the accepted sampling error).

Table 6.7: Clippings Sample Detail

Data Clippings	Universe	Sample	Group		Sampling Error	Confidence Level
			Experimental	Control		
Clipping 1	45	44	21	23	2.23%	95%
Clipping 2	37	35	20	15	3.90%	95%
Clipping 3	82	79	41	38	2%	95%
Clipping 4	101	93	55	38	2.87%	95%

From the sample of 93 students, 45.16% (42) are male, and 54.85% (51) are female (see Figure 6.5). Considering clipping 4, the control group consists of 44.7% (17) male students and 55.3% (21) female students, values that closely align with the distribution of the experimental group, which comprises 45.5% (25) male students and 54.5% (30) female students. The quantities of students in each group and subset, categorized by gender, can be visualized in Table 6.5.

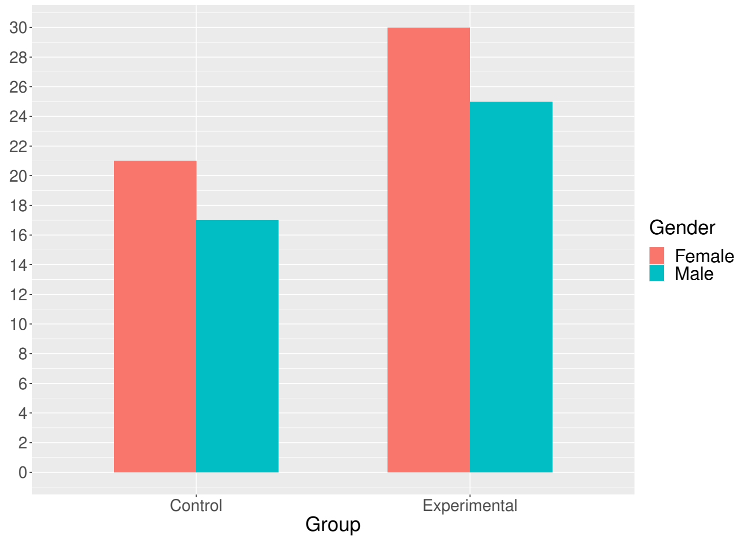


Figure 6.5: Gender Distribution Across Student Groups

Table 6.8: Gender Distribution Across Student Groups and Classes

Group	**Class**	**Female**	**Male**	**Total by Class**
Control	1st A	13	10	**23**
Control	3rd B	8	7	**15**
Experimetal	1st B	14	7	**21**
Experimetal	2nd	3	11	**14**
Experimetal	3rd A	13	7	**20**

As depicted in Figure 6.6, the average age of the student sample is 16.34. The control group has an average age of 16.24, while the experimental group has an average age of 16.42. These values demonstrate that, despite the sample comprising students from different grades, there is a tendency for the average age of both the experimental and control groups to be similar. Specific differences between the ages of each class can be observed in Figure 6.7.

From the student sample, 100% (93) are first-time attendees in the high school series, meaning they are not repeating the grade. Additionally, 28% (26) completed most of their elementary education in private schools, while 72% (67) attended public schools. Regard-

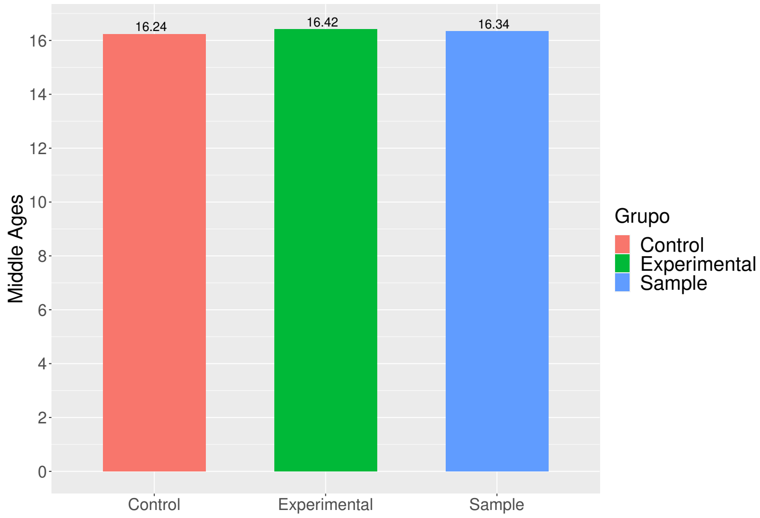


Figure 6.6: Analyzing Student Middle Ages

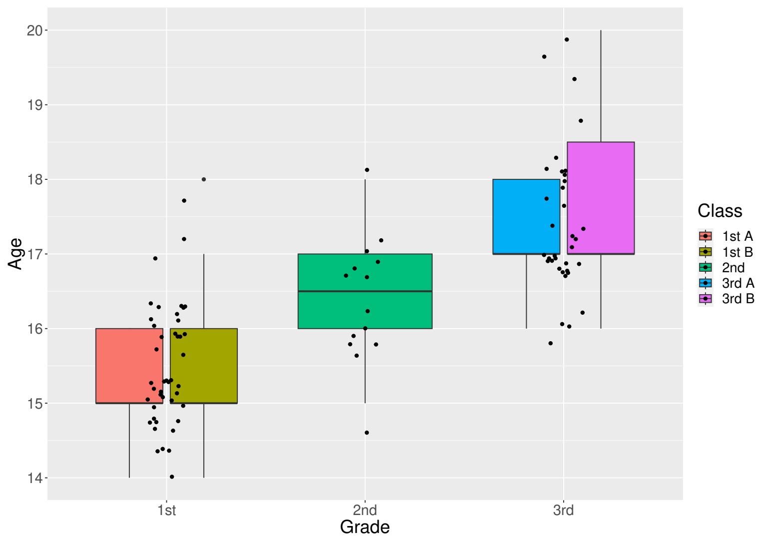


Figure 6.7: Analyzing Student Ages Based on Grades and Classes

ing the students' profile, 84.9% (79) state that they had no exposure to robotics before the programming and robotics classes offered within the intervention study. However, they expressed an interest in learning about the subject. The breakdown of students' exposure to and interest in robotics can be observed in Figure 6.8.

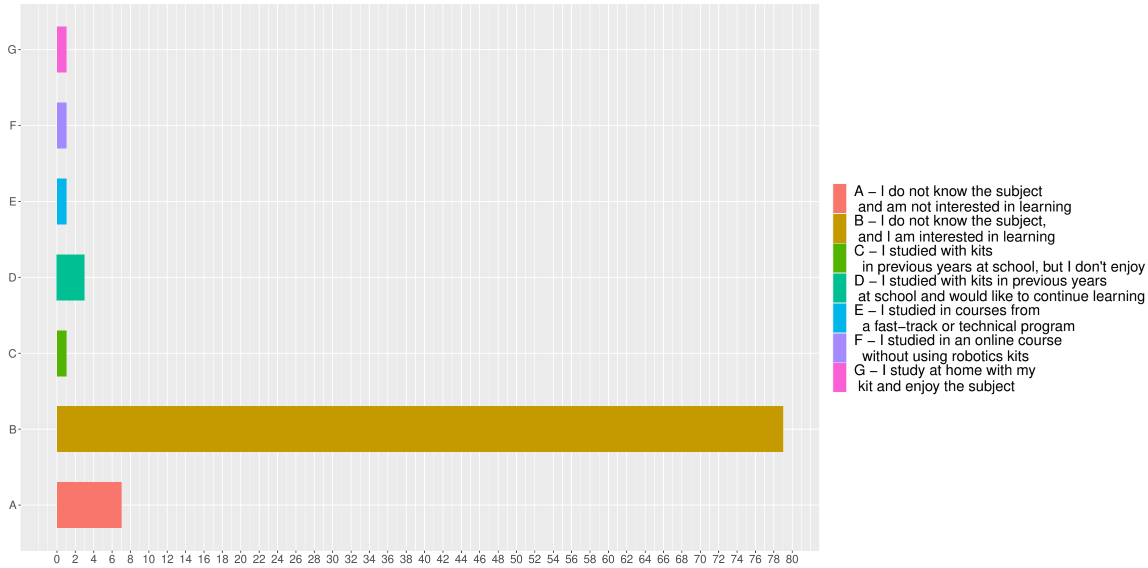


Figure 6.8: Pre-Intervention Exposure to Robotics: Students' Prior Experiences

6.4 Results

Its purpose is to evaluate the effectiveness of CTProgER, as perceived by students. The intervention study specifically targeted Computer Technical and Vocational High School students, focusing on teaching-learning programming through CTProgER. The primary goal of this study is to enhance the teaching-learning of programming with an emphasis on CT skills. This section presents the results of quantitative analyses of student data collected during the intervention, serving as a foundation for addressing RQ3.

6.4.1 How effective is the impact of the educational process of teaching-learning programming with ER on students' CT skills in High School?

To address the third research question (RQ3), it considered the student clipping 4, representing the entire high school, i.e., 1st, 2nd, and 3rd grades. Initially, a descriptive statistical analysis was conducted through a graphical examination of students' performance in the CT test before and after the intervention. The performance distribution graph illustrates that the frequency of students with above-average performance (highlighted by red lines) indicates no difference between the experimental and control groups in the pre-test, demonstrating that both groups exhibit similar performance (see Figure 6.9).

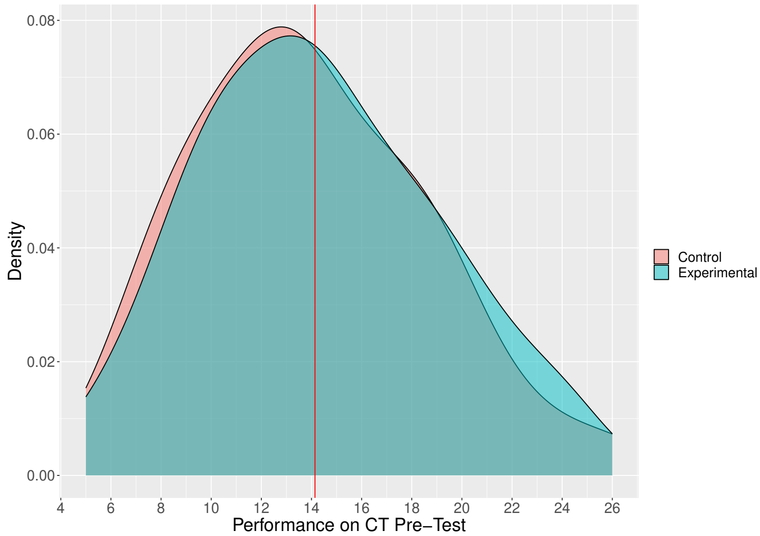


Figure 6.9: Distribution of Student Groups Performance CT Pre-test

On the other hand, the performance distribution graph in the post-test reveals a higher frequency of students with above-average performance (highlighted by red lines) in the experimental group compared to the control group (see Figure 6.10). Thus, CTProgER may positively impact PC skills through robotics-enhanced programming instruction.

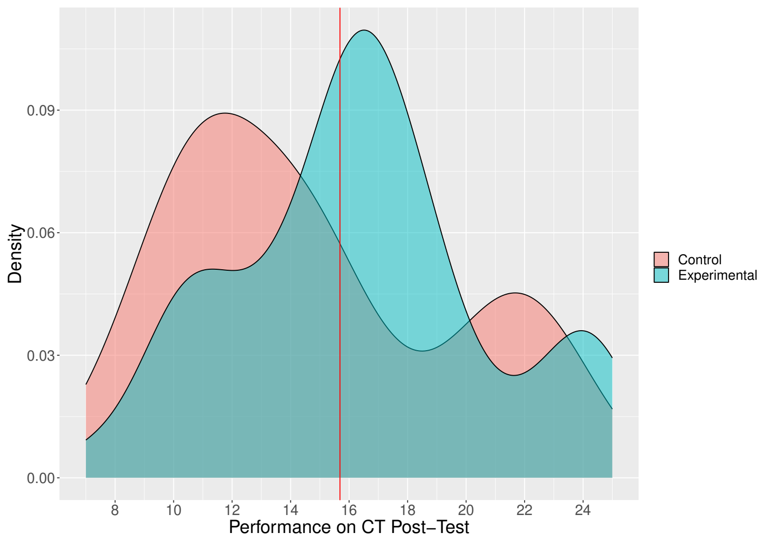


Figure 6.10: Distribution of Student Groups Performance CT Post-test

Complementary to this, considering that 7 and 25 were the minimum and maximum scores obtained by the experimental group in the post-test, and the control group received 8 and 23, it analyzed the mean and standard deviation of the control and experimental groups. According to the data presented in Table 6.9, in the post-test, the performance of students in the experimental group showed a higher mean compared to the control group by 10.59%, a higher median of 12.50% and a standard deviation (SD) with less variation (4.36), demonstrating that the experimental group has a better performance than the control group.

Table 6.9: Groups Performance From Román-Gonzalez CT Test

Test	**Experimental**			**Control**			**Mean Difference**	**Median Difference**
	Mean	**Median**	**SD**	**Mean**	**Median**	**SD**		
Pre-Test	14.27	14.00	4.70	13.95	13.50	4.65	2.33%	3.57%
Post-Test	16.33	16.00	4.36	14.76	14.00	4.69	10.59%	12.50%

This fact can be verified through the boxplot graph in Figure 6.11, indicating that the distribution of post-test data in the control group is asymmetric, and most values are concentrated in the upper half of the box (interquartile). If there is no data in the first quartile, it suggests that the lower 25% of the data is missing or has shallow values compared to the upper part of the distribution. It may be indicative of right-skewed data, where most of the variation occurs at higher values. The experimental group shows a homogeneous distribution and a higher median than the control group. It provides more substantial evidence that the experimental group performed better than the control group in the post-test, i.e., after the intervention.

This evaluation indicates a real difference in the performance of the groups and a positive effect attributed to CTProgER, given that students in the experimental group tend to show superior performance in the CT test compared to the control group. Thus, this exploratory analysis provides evidence that CTProgER promotes the development of CT in High School students through the teaching-learning of programming with robotics. In this regard, to refute the null hypothesis H1.0: There is no evidence that the educational process of teaching-learning programming with educational robotics positively impacts the students' CT skills in High School., the Mann-Whitney *U Test* (unpaired and paired) was applied with a confidence level of 95%, and significance level α = 0.05.

The Unpaired U Test data demonstrates a significant difference between the experimental

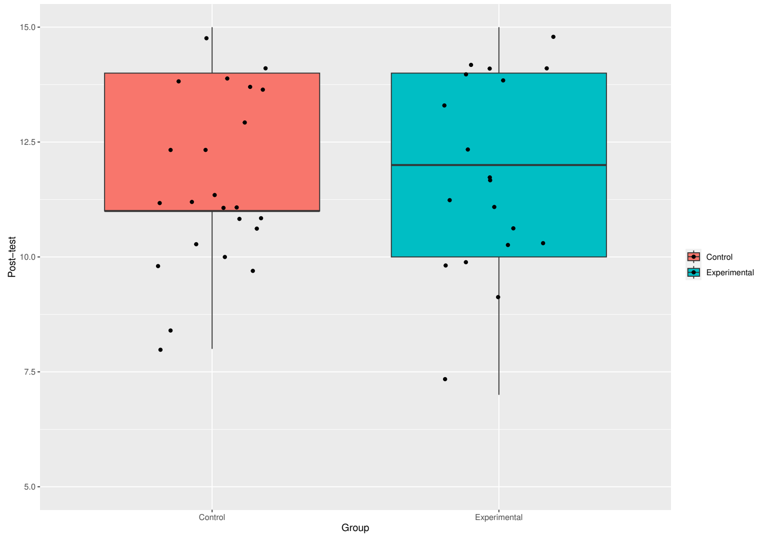


Figure 6.11: Distribution of Student Groups Performance CT Post-test

and control groups, presenting a *p-value* of 0.04852, a value lower than α (see Table 6.10). The Unpaired *U Test* is a statistical analysis that compares the medians of two independent samples to determine if there is a significant difference between them. Since the median of the experimental group is higher than that of the control group (see Table 6.9), it is possible to attribute this difference to the improvement in the experimental group's performance.

Table 6.10: Hypothesis Unpaired Test From Román-Gonzalez CT Test

Test	**U Test**	***p-value***	**Confidence Level**	
			Min	**Max**
Post-Test	12973	<0.05	0.00001539296	3.999984

In addition to the unpaired Mann-Whitney *U Test*, the paired Mann-Whitney *U Test* was applied to identify if there was a significant difference between the pre-test and post-test of each group. The paired Mann-Whitney *U Test* is a version of the Mann-Whitney *U Test* designed to compare two independent groups in a paired sample. Through it, it is possible to observe a significant difference between the pre-test and post-test of the experimental group, as indicated by the *p-value* of 0.01924, value less than α (see Table 6.11). On the other hand, it is impossible to identify the significance of the exact data for the control group, which demonstrates that there is no significant difference between students' performance in this group before and after the intervention. Thus, it can be said that the experimental group improved significantly when comparing students' performance before and after the interventions and that in the control group, it cannot be statistically stated that the same difference exists.

Table 6.11: Hypothesis Paired Test From CT Test

Groups	U Test	*p-value*	Confidence Level	
			Min	Max
Experimental	1903.0	<0.05	0.0000553709	0.0000553709
Control	786.5	0.5013	-1.9999610000	2.9999780000

Therefore, based on the paired and unpaired Mann-Whitney *U Test*, it is possible to reject the null hypothesis H1.0: There is no evidence that the educational process of teaching-learning programming with educational robotics positively impacts the students' CT skills in High School. It is possible to assume the alternative hypothesis H1.0: The educational process of teaching-learning programming with educational robotics impacts the students' CT skills in High School.

Upon further examination of Table 6.12, it is noteworthy that Cohen's d is precisely equivalent to the tabulated Z-score from a standard normal distribution [57]. Referencing the standard normal distribution table allows us to discern that Cohen's d-effect size was small, indicating a subtle impact. Specifically, 63% of students in the experimental group exhibited a mean superior to that of the control group.

When comparing the pre-test to the post-test within the experimental group, the Cohen's d effect size remained small, with 63% of students demonstrating an enhanced mean in the post-test compared to the pre-test. Conversely, in the control group, Cohen's d for the pre-test to post-test comparison was deemed insignificant, and 56% of students displayed a superior mean in the post-test relative to the pre-test.

Table 6.12: Effect Size in CT From Román-Gonzalez CT Test

z X y	Effect (d)	Effect Size	(y > z)%
Experimental Post-Test (z) x Control Post-Test (y)	0.35	Small	63%
Experimental Pre-Test (z) x Experimental Post-Test (y)	0.45	Small	67%
Control Pre-Test (z) x Control Post-Test (y)	0.17	Insignificant	56%

Despite the small effect observed in the experimental group, the notable aspect is that the control group exhibited no statistically significant effect. This analysis suggests that the experimental group, which underwent the CTProgER educational process alongside programming and robotics instruction, outperformed the control group, which received the same teaching-learning programming and robotics exposure but lacked the CTProgER intervention. The absence of a significant effect in the control group underscores the distinctive impact of the CTProgER educational approach. It is essential to recognize the positive aspects of the control group, as their exposure to programming and robotics alone contributed to improved performance, albeit without the additional CTProgER intervention. Given that it is known that Computational Thinking can be fostered through the teaching of programming and exposure to robotics [170]. This nuanced comparison highlights the specific benefits introduced by the CTProgER, emphasizing its unique and positive influence on developing CT skills among High School students.

6.4.2 Complementary Statistical Analyses: Unfolding Implications of CTProgER Impact

The RQ3 of this study was addressed based on data clipping 4, providing valuable insights into the impact of CTProgER on CT skills and how it is possible to teach and learn programming with robotics. However, considering the other data clippings, a more comprehensive examination reveals additional crucial information that enriches the overall understanding of the CTProgER's impact on students' skills.

Upon scrutinizing the various subsets, it becomes evident that CTProgER positively influenced CT skills in different contexts and grade levels (included in the experimental group). A detailed analysis of these aspects provides a more comprehensive view of the program and its effects on developing students' CT skills.

This subsection thoroughly explores the data from each clipping, highlighting nuances that may not have been fully captured in the initial analysis. These crucial aspects include variations in student performance, specific grade-level trends, and group dynamics. By delving into this more profound analysis, the aim is to present a holistic perspective of the impact of CTProgER on students' CT skills. Each subset represents a piece of the puzzle, and a complete understanding of these nuances is essential for drawing robust conclusions and valuable insights for the academic and educational community. The holistic perspective refers to an approach that considers a system as an integrated whole rather than analyzing its parts individually. In a broader context, the holistic view desires to comprehend the entirety of a phenomenon, taking into account its interconnections and interdependencies rather than focusing solely on isolated parts; i.e., the holistic approach seeks a global and integrated understanding [77].

Initially, a descriptive statistical analysis was conducted through a graphical examination of students' performance in the CT test before and after the intervention. The performance distribution graph illustrates that the frequency of students with above-average performance (highlighted by red lines) indicates no difference between the experimental and control groups in the pre-test, demonstrating that both groups (experimental and control) of all clipping exhibit similar performance (see Figure 6.12).

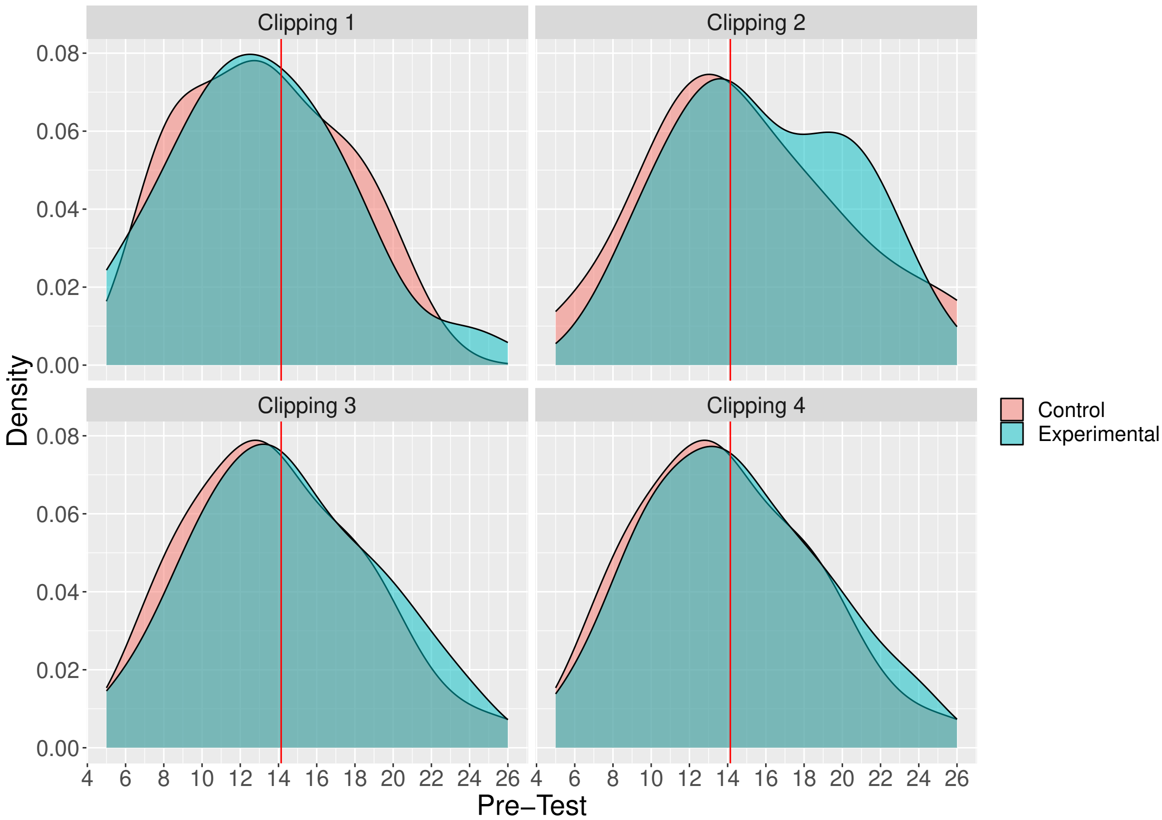


Figure 6.12: Distribution of Student Clipping by Groups of Performance CT Pre-test

This consistency in density patterns suggests that before the intervention or exposure to CTProgER, the experimental and control groups exhibited similar knowledge or CT skills. Hence, the density plots of the pre-test provide a visual foundation for the initial comparability between the groups, essential for interpreting the changes and impacts resulting from the intervention over time.

The analysis of these visual patterns reinforces the robustness of the methodological approach, highlighting the initial equivalence of the groups before the introduction of CTProgER. This observation is crucial for accurately interpreting differences identified in the post-tests, enabling a more reliable analysis of the intervention's specific impact on improving students' CT skills.

The analysis of the post-test density graphs in the four data clippings suggests a consistent pattern of superior performance in the experimental group (see Figure 6.13). This observation is evidenced by the behavior of the density curves, which indicate that the experimental group tends to exhibit a higher average performance (highlighted by red lines) than the control group.

The constancy of this pattern across different clippings strengthens the evidence that the intervention with CTProgER positively impacted the students' CT skills. This consistency in results underscores the program's effectiveness in promoting measurable improvements. The observed benefits are not specific to a particular grade or context but a general trend across various analyses. This conclusion reinforces the relevance and potential impact of CTProgER on the development of CT skills among participating students.

The data reveals intriguing results that indicate the need for a more detailed analysis of the impact of CTProgER on different groups and grades. Consequently, an analysis of the experimental and control groups' means, medians, and standard deviations was conducted, as well as their mean and median differences in each clipping (see Table 6.13).

In clipping 1 (1st grade), considering the pre-test, the experimental and control groups presented very close means, suggesting an initial equivalence in CT skills. The mean difference is -1.28%, a negative value indicating that the control group performs 1.28% better than the experimental group. However, there is no median difference between the two groups. The median is crucial as the hypothesis tests used in this study are based on it. After the intervention, considering the post-test, a significant increase in the mean of the experimental

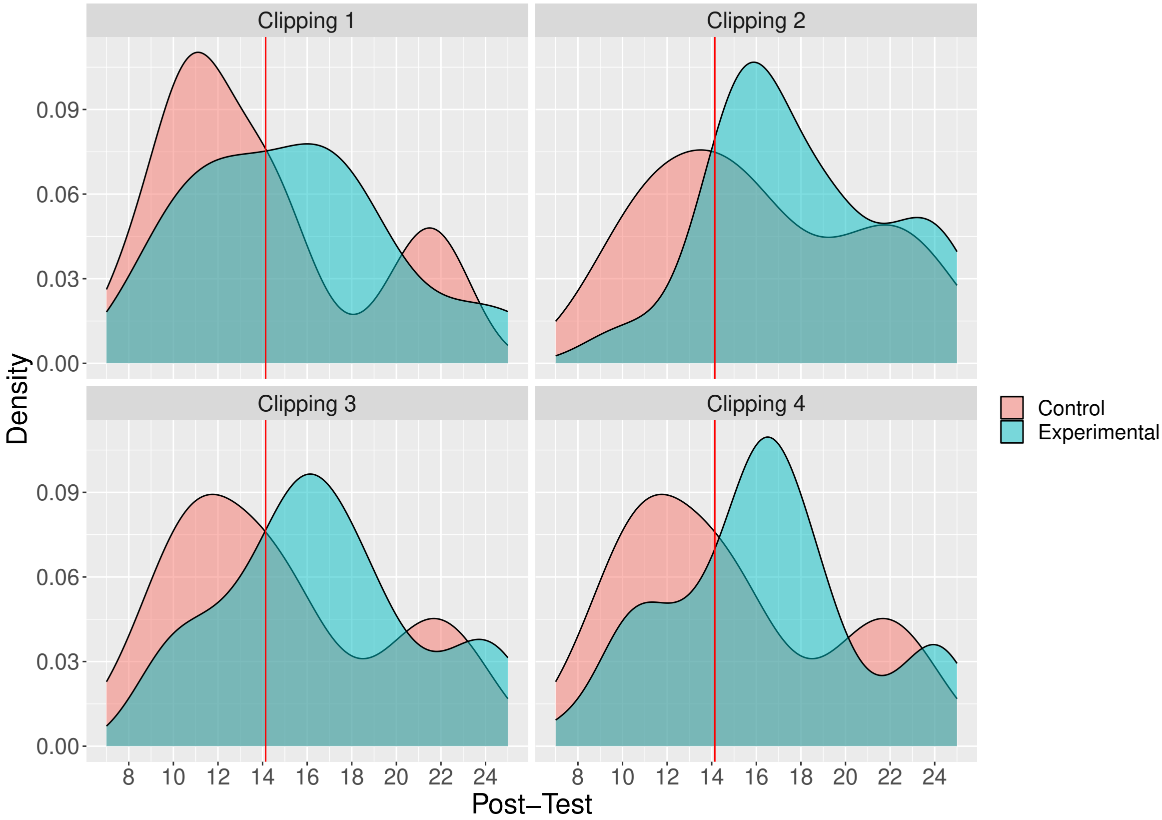


Figure 6.13: Distribution of Student Clipping by Groups of Performance CT Post-test

Table 6.13: Groups Performance From Román-Gonzalez CT Test by Clipping

Clippings	Test	Experimental			Control			Mean	Median
		Mean	Median	SD	Mean	Median	SD	Difference	Difference
Clipping 1	Pre-Test	13.05	4.60	13.00	13.22	4.11	13.00	-1.28%	0.00%
	Post_Test	15.38	4.48	16.00	13.87	4.55	12.00	10.90%	25.00%
Clipping 2	Pre-Test	15.80	4.55	14.50	15.07	5.32	14.00	4.87%	3.45%
	Post_Test	18.00	4.00	16.50	16.13	4.73	15.00	11.57%	9.09%
Clipping 3	Pre-Test	14.39	4.73	14.00	13.95	4.65	13.50	3.18%	3.57%
	Post_Test	16.33	4.40	16.66	14.76	4.69	14.00	10.59%	15.96%
Clipping 4	Pre-Test	14.27	4.70	14.00	13.95	4.65	13.50	2.33%	3.57%
	Post_Test	16.33	4.36	16.00	14.76	4.69	14.00	10.59%	12.50%

group compared to the control group is observed. The mean difference of 10.90% and the median difference of 25.00% indicate a positive impact after the intervention.

In clipping 2 (3rd grade), considering the pre-test, the mean scores of the experimental and control groups are again similar. The mean difference is 4.87%, indicating that the experimental group outperforms the control group by 4.87%. However, the median difference between the groups is 3.45%, which is attributed to the better performance of the experimental group. Although these data demonstrate the superiority of the experimental group, it considers this difference minimal in the assessment of competencies and skills, as discussed in Subsection 6.2. After the intervention, assessing the post-test, the experimental group showed a considerable increase in the mean compared to the control group, with a mean difference of 11.57% and a median difference of 9.09%, suggesting a positive impact.

In clipping 3 (1st and 3rd grades); considering the pre-test, the initial means are close, indicating equivalence in the initial CT skills. The mean difference is 3.18%, signifying that the experimental group is 3.57% better than the control group. After the intervention, the experimental group showed a considerable improvement compared to the control group, with a mean difference of 10.59% and a median difference of 15.96%.

The data from clipping 4 (1st, 2nd, and 3rd grades) has been discussed in Subsection 6.4.1, demonstrating a similar performance trend between the experimental and control groups as observed in the previous clippings.

Paired and unpaired *U Test* were applied to assess whether there is a significant difference between the clippings, as suggested by the mean and median differences (see Table 6.14). The non-parametric *U Test* results indicate significant differences in clippings 3 and 4, as the *p-values* are less than 0.05. These results suggest a statistically significant difference between the experimental and control groups in Clippings 3 and 4 regarding the impact of CTProgER on CT skills. However, for clippings 1 and 2, the *p-values* are more significant than 0.05, indicating insufficient statistical evidence to reject the null hypothesis that there is no significant difference between the groups in these clippings. Confidence intervals provide an estimate of the magnitude of the observed differences, offering additional information about the variability of the results.

In addition to the unpaired Mann-Whitney *U Test*, the paired Mann-Whitney *U Test* was applied to identify if there was a significant difference between the pre-test and post-test of each group by clipping. Through it, it is possible to observe a significant difference between the pre-test and post-test of the experimental group, as indicated by the *p-value* being less than 0.05 (see Table 6.15).

Table 6.14: Hypothesis Unpaired Test From Román-Gonzalez CT Test by Clippings

Clippings	**U Test**	**p-value**	**Confidence Level**	
			Min	**Max**
Clipping 1	293.5	0.224	-1.00000600	4.00000900
Clipping 2	191.5	0.169	-0.99996890	5.00001160
Clipping 3	988.5	<0.05	0.00001007	4.00002525
Clipping 4	12973	<0.05	0.00001539	3.99998400

The paired *U Test* results reveal that students in the experimental group of clippings 3 and 4 show a positive and significant difference when comparing the pre-test with the post-test, i.e., they have a *p-value* less than 0.05. In contrast, the control group of students in these clippings shows no significant difference. No significant difference is observed in the experimental or control groups in the remaining clippings. The confidence intervals for each group estimate the magnitude of the observed differences, contributing to a more comprehensive interpretation of the results.

Table 6.15: Hypothesis Paired Test From CT Test by Clippings

Clippings	**Group**	**test-U**	***p-value***	**Confidence Level**	
				Min	**Max**
Clipping 1	Experimental	282.0	0.1203	-0.9999713000	5.0000445000
	Control	281.0	0.7159	-2.0000430000	2.9999400000
Clipping 2	Experimental	259.0	0.1092	-0.9999978000	5.0000024000
	Control	127.5	0.532	-2.9999500000	5.0000380000
Clipping 3	Experimental	1072.0	<0.05	0.0000489582	4.0000240000
	Control	786.5	0.5013	-1.9999610000	2.9999780000
Clipping 4	Experimental	1903.0	<0.05	0.0000553709	0.0000553709
	Control	786.5	0.5013	-1.9999610000	2.9999780000

Understanding statistical nuances is essential for an accurate interpretation of research results. When analyzing clipping data, which sometimes involves separate grades and other times grouped grades, the choice between examining series individually or in groups can have significant implications for identifying statistically significant differences. There are underlying reasons for variability in statistical significance when comparing these two approaches in the *U Test*, providing valuable insights into result interpretation and its implications for research. Several factors contribute to the variation in statistical significance when analyzing series data individually versus grouped in the *U Test*: **variability in individual data, sample size, cumulative effect, and variable control** [78].

Regarding **variability in individual data**, when analyzing grades individually (1st and 3rd), data variability may be higher, making it more challenging to identify significant differences. Both clippings 1 and 2 did not yield a considerable hypothesis test result despite evidence through favorable mean and median differences in favor of the experimental group. Therefore, grouping the data can reduce this variability and highlight more consistent patterns, as identified in clippings 3 and 4.

The **sample size** can influence statistical significance. In individual series, the sample size is smaller, which may result in less statistical power to detect differences. Although the *U Test* has been designed for small and different sample sizes, it is essential to recognize that the reduced sample size can impact the statistical significance of the data. The *U Test* is robust in handling small samples. Still, the inherent limitation of sample size can affect the test's ability to identify statistically significant differences, especially when sample sizes are very small. Therefore, it is crucial to consider this limitation when interpreting the results, acknowledging that the restricted sample size may impact statistical accuracy. Grouping the grades makes the total sample size larger, increasing statistical power. However, the lack of a significant p-value in clippings 1 and 2 may not necessarily indicate that the experimental and control groups are not distinct regarding CT skills after interventions, as the mean and median differences suggest otherwise.

When grouping the grades, there is the possibility of an **accumulated effect**, where individual differences in each grade add up, resulting in a more pronounced and statistically significant disparity. This information means that even if the differences in each grade individually are not statistically significant, when grouped, these small variations can accumulate, amplifying the overall impact and leading to more statistically significant results. In this specific research context, this approach can have crucial implications for understanding how interventions, when analyzed collectively across various grades, can have a more robust and measurable impact on the development of CT. The aggregated analysis can reveal patterns and trends that may not be readily apparent when observing grades individually, providing a more comprehensive view of the program's impact on students' CT skills. This consideration is vital when interpreting the results, offering valuable insights into the effectiveness of interventions and CTProgER across different educational contexts.

The lack of significant difference between the experimental and control groups can also be attributed to the fact that both groups had exposure to programming and robotics education. Literature suggests that these experiences can foster the development of CT. Therefore, both groups could have enhanced their CT skills naturally due to this common exposure. This data can be verified through the mean and median data of the control group in Table 6.13, as even subtly, an improvement in the CT skills of this group of students can be observed. However, the experimental group, which had exposure to CTProgER, showed a more significant increase than the control group, highlighting that teaching-learning programming with robotics following an educational process designed to develop CT can potentially influence the development of these skills. The similarity in the educational interventions received by the groups may have contributed to a common foundation of knowledge and CT skills, resulting in a lack of statistically significant difference between them after the interventions. However, the improvement in the experimental group positively demonstrates the value of CTProgER. Hence, observing the intervention's effect on these groups is also necessary according to each clipping.

Upon further examination of Table 6.16 considering all clippings observing the effect by comparing the post-test of the experimental group with the control group, it also analyzes the effect under the pre-test and post-test of both groups separately. It is noteworthy that Cohen's d is precisely equivalent to the tabulated Z-score from a standard normal distribution [57].

In the specific analysis of **clipping 1**, when comparing the post-test results between the experimental and control groups, a small Cohen's effect is observed. This measure, expressing the practical magnitude of the difference in terms of standard deviation, indicates a modest influence of CTProgER on CT skills. Additionally, it is essential to highlight that 62% of students in the experimental group outperformed the control group. This observation suggests that a significant portion of participants subjected to the CTProgER intervention experienced measurable improvements in their CT skills compared to peers not exposed to CTProgER.

Table 6.16: Effect Size in CT From Román-Gonzalez CT Test by Clipping

Clippings	**z X y**	**Effect (d)**	**Effect Size**	**(y >z)%)**
Clipping 1	Experimental Post-Test x Control Post-Test	0.33	Small	62%
	Experimental Pre-Test x Experimental Post-Test	0.51	Medium	69%
	Control Pre-Test x Control Post-Test	0.15	Insignificant	55%
Clipping 2	Experimental Post-Test x Control Post-Test	0.43	Small	66%
	Experimental Pre-Test x Experimental Post-Test	0.51	Medium	69%
	Control Pre-Test x Control Post-Test	0.21	Small	58%
Clipping 3	Experimental Post-Test x Control Post-Test	1.42	Small	92%
	Experimental Pre-Test x Experimental Post-Test	0.50	Small	69%
	Control Pre-Test x Control Post-Test	0.17	Insignificant	56%
Clipping 4	Experimental Post-Test x Control Post-Test	0.35	Small	63%
	Experimental Pre-Test x Experimental Post-Test	0.45	Small	67%
	Control Pre-Test x Control Post-Test	0.17	Insignificant	56%

When analyzing the overall effect, considering the experimental group's post-test compared to the control group, a small effect was observed. However, examining the effect within each group separately, considering a comparison between pre-and post-test, reveals exciting nuances. In the experimental group, the effect is of medium magnitude, indicating a more substantial influence of CTProgER on CT skills. Notably, 69% of students in this group showed higher averages in the post-test than in the pre-test. On the other hand, in the control group, the effect is practically insignificant, showing minimal changes in averages between the pre-and post-tests. Only 55% of students in this group demonstrated improvement in averages in the post-test compared to the pre-test. This differentiation in effects highlights the potential efficacy of CTProgER in promoting specific improvements in CT skills, emphasizing the importance of examining the impact within each experimental and control group for a comprehensive understanding.

These results underscore the relevance of the intervention, even though the difference could be more expressive in terms of Cohen's effect. Considering both the magnitude of the difference and the proportion of benefited students may enrich the interpretation, indicating that, while the impact is modest statistically, a considerable portion can benefit from the implemented approach.

In the specific context of **clipping 2**, a small Cohen's effect is observed when comparing the experimental and control groups' post-test results. This measure, which quantifies the practical magnitude of the difference in terms of standard deviation, also suggests a modest influence of CTProgER on CT skills. Additionally, it is crucial to highlight that 66% of students in the experimental group outperformed the control group. This observation indicates that a considerable portion of participants in this clipping, subjected to the CTProgER intervention, also experienced measurable improvements in their CT skills compared to those not exposed to CTProgER.

A small effect is identified when examining the overall effect considering the experimental group's post-test compared to the control group. However, when analyzing the effect within each group separately and comparing the pre-and post-test, the behavior is similar to clipping 1. In the experimental group, the effect is of medium magnitude, indicating a more substantial influence of CTProgER on CT skills. Notably, 69% of students in this group showed higher averages in the post-test than in the pre-test. On the other hand, in the control group, the effect is small, demonstrating minimal change in averages between the pre-test and post-test. Only 58% of students in this group showed improvement in averages in the post-test compared to the pre-test.

In the context of clipping 3, a small Cohen's effect is observed when comparing the experimental and control groups' post-test results. Furthermore, it is noteworthy that 92% of students in the experimental group outperformed the control group. This observation indicates that a substantial majority of participants in this clipping, subjected to the CTProgER intervention, experienced measurable improvements in their CT skills compared to those not exposed to CTProgER.

A small effect is identified when examining the overall effect considering the experimental group's post-test compared to the control group. However, when analyzing the effect within each group separately and comparing pre-and post-tests, the behavior is again similar to previous clippings. In the experimental group, the effect is of medium magnitude, indicating a more substantial influence of CTProgER on CT skills. Notably, 69% of students in this group showed higher averages in the post-test than in the pre-test. On the other hand, in the control group, the effect is insignificant, showing minimal change in averages between the pre-test and post-test. Only 56% of students in this group demonstrated improvement in averages in the post-test compared to the pre-test.

The effect data from **clipping 4** has been discussed in Subsection 6.4.1, demonstrating a similar effect analysis observed in the previous clippings.

The detailed analysis of Cohen's effects across different clippings reveals that the impact of CTProgER goes beyond the findings presented in the U tests. These analyses, extending beyond measures of statistical significance, play a crucial role in verifying the effectiveness of CTProgER, especially when considering students' perspectives and the observed impact on their performances in CT.

The results indicate that, for various clippings, CTProgER positively influenced the CT skills of students in the experimental group. Even when U tests did not reveal statistically significant differences, Cohen's effects highlight that the program had a practically relevant magnitude. This data is particularly evident when examining the average improvement in students' grades in the experimental group compared to the control group.

As expected, it is essential to emphasize that both the experimental and control groups were positively impacted by programming and robotics education. This observation aligns with existing literature [168; 183; 2], suggesting that exposure to such educational approaches can naturally enhance students' CT skills. However, the more pronounced and statistically significant effects in the experimental group indicate that CTProgER played an additional and valuable role in developing these skills.

In summary, the detailed analyses of Cohen's effects provide a more comprehensive and nuanced view of CTProgER's impact, validating its effectiveness beyond the confines of conventional statistical hypothesis tests. The data reflect improvements in CT skills and students' perceptions of this enhancement. This info underscores the importance of considering multiple perspectives when evaluating the effectiveness of educational interventions, and the results presented here contribute to a comprehensive understanding of CTProgER's impact in the context of CT development, providing insights for studies desiring to validate an educational process.

6.5 Threats of Validity

The validity of a scientific study is directly connected to the reliability established through the applied methodological process, indicating how reliable the instruments used in the study are, from theoretical foundations to the conclusions drawn [95] and [105]. Consequently, it is necessary to consider aspects that can affect construct, internal, external, and conclusion validity. This subsection presents the limitations and threats to the validity of this study.

6.5.1 Construct Validity

- **Design of Experiments:** There may be a threat related to content definition concerning the possibility that the topics defined by Computer Science professionals for the intervention may only comprehensively cover some essential aspects of the introduction to programming language. It would be beneficial to conduct an additional review and validation of the content by experts in education and psychometrics, ensuring that the chosen topics genuinely assess the desired skills to mitigate this threat. Regarding the student profile questionnaire, the threat lies in the possibility that the questions formulated may not accurately reflect the desired information about the participants. Lack of clarity or ambiguity in the questions can lead to varied interpretations by participants, compromising the validity of responses and, consequently, impacting the formation of data analysis clipping. One strategy to overcome this threat would be to conduct pre-tests of the questionnaire with a pilot group, adjusting and refining questions based on feedback and results obtained. Another possibility would be to carefully review the questions, involving experts in psychometrics and educational psychology, to ensure proper understanding and relevance of the questions.

- **Human Factors:** The CT test papers by Román-Gonzalez *et al.* [87] and the data collected in the student profile questionnaire were manually corrected and cataloged. Despite double-checking the data, correction errors may occur due to human factors. Using automated tools or sytems for test correction could reduce dependence on human intervention and minimize associated errors.

6.5.2 Internal Validity

- **Student History:** The impossibility of controlling factors related to students' educational and personal experiences in this study may have influenced the analysis results. Variables such as the school's administrative structure, the quality of elementary education, participation in extracurricular activities, access to technological resources, and the prior knowledge of students and teachers were not adjusted. These uncontrolled variables could have impacted the study's outcomes, and their presence should be considered when interpreting and generalizing the results.

- **Maturation:** Students may have become familiar with the CT test by Román-Gonzalez *et al.* [87] throughout its repetitions (pre-test and post-test), which possibly aided in addressing the proposed questions. Furthermore, since the questions were the same, students may have felt demotivated when responding to the post-test and during the intervention classes.

- **Contamination:** Since the experimental group had access to CTProgER and its instances, students received specific materials, unlike the control group. The customized materials for the audience may have sparked the interest of many students. Throughout the classes, it is possible that students from both control and experimental groups shared knowledge, experiences, and materials.

- **Confusion of Factors:** This threat pertains to the possibility of other elements influencing students' performance throughout the academic year. Since they have been exposed to curriculum components related to computing, these additional factors can clarify the specific analysis of the intervention's impact. It would be relevant to carefully analyze external factors that may affect the results, aiming to distinguish between the effects of the intervention and other elements in the educational environment to address this threat. Furthermore, collecting information on additional variables, such as participation in extracurricular activities or prior educational history, could contribute to controlling and better understanding these confounding factors, providing more accurate and detailed insight into the impact of the intervention on the development of students' CT skills.

6.5.3 External Validity

- **Participants:** The sample used in this study is representative of a population of students from three grades (1st, 2nd, and 3rd) from the Computing TV High School in the Paraíba state. Therefore, generalizing the results to other populations of students is not practical.

- **Experiment Setup:** Due to limitations in the school's robotics laboratories, the completion of activities may have been compromised by a lack of materials and even environmental conditions such as air conditioning and physical space constraints.

6.5.4 Conclusion Validity

- **Statistical Power:** A small number of participating students in each analysis subset may threaten the statistical power due to the diminished ability of the study to detect statistically significant effects if they exist. Statistical power is the probability of a statistical test detecting a real difference when it exists. With small samples, data variability can have a more pronounced impact on results, making it more challenging to identify fundamental differences between groups, if any. Although hypothesis tests were used considering data parametrization, which is also suitable for small and distinct sample sizes, when the sample size is small, the study may have lower sensitivity to detect real effects, increasing the risk of Type II errors (failure to reject a false null hypothesis). This context implies that the study may need to identify differences or effects that genuinely exist in the population, resulting in less reliable and generalizable conclusions. To mitigate this threat, increasing the sample size whenever possible can enhance the study's statistical power, increasing the likelihood of detecting true effects if present. However, this becomes a challenge when applying interventions in real schools that have a standardized number of students per class.

6.6 Summary of Chapter

This chapter serves as a foundational introduction to an intervention study, a critical element of Educational Design Research embedded within the overarching of this doctoral thesis. The focal point of this intervention study is the evaluation of CTProgER's effectiveness, as perceived by students. Specifically tailored for Computer Technical and Vocational High School students, the intervention targets teaching-learning programming by applying CT-

ProgER. The primary aim is to enhance the teaching and learning of programming, mainly cultivating CT skills.

In alignment with the third research question (**RQ3:** "How effective is the impact of the educational process of teaching-learning programming with ER on students' CT skills in High School?) of this thesis of this intervention study is intricately connected. The overarching goal is to generate empirical evidence that illuminates the influence of teaching and learning programming through CTProgER on the CT skills of High School students specializing in Computer Technical and Vocational studies. Furthermore, the study endeavors to validate CTProgER's effectiveness from the student's perspective.

In clipping 2 (3rd grade), considering the pre-test, the mean scores of the experimental and control groups are again similar. The mean difference is that the results highlight that the educational process built to teach programming through ER is capable of helping the student develop his CT skills better because a significant difference was observed between the participants who had contact with the CTProgER. However, it does not generalize the results to other student populations since it only considered data from students enrolled in a Brazilian public TV High School. The statistical data presented in this study constitutes a set of procedures that can be used, in a quantitative way, in the process of validating the effectiveness of the educational process [172]. It believes that the procedures presented in [172], together with the result of this study, contribute to the scientific community in the sense of guiding the validation of an educational process.

Chapter 7

Conclusion and Future Work

This thesis constitutes a Computer Science Education Research endeavor aimed at enhancing the teaching and learning of programming with educational robotics, with a specific emphasis on developing computational thinking (CT) skills. The investigation was structured around three primary research questions:

- **RQ1:** How is it possible to teach-learning programming with ER to facilitate students acquiring CT skills?
- **RQ2:** How much conformity is there in the educational process of teaching-learning programming with ER from an expert's perspective?
- **RQ3:** What is the impact of the educational process of teaching-learning programming with ER on students' CT skills in High School?

These research questions guided the pursuit of six overarching objectives:

- **OBJ1:** Mapping how ER has been used to stimulate CT skills in students;
- **OBJ2:** Identifying ways to design an educational process for teaching-learning programming in High School with ER, focusing on developing CT;
- **OBJ3:** Instantiating the educational process for some programming topics to be used with High School students;
- **OBJ4:** Validating the instance from an expert point of view;

- **OBJ5:** Conducting an intervention experiment in High School to apply the educational process and instance together;
- **OBJ6:** Analyzing the educational process's effectiveness to High School students

7.1 Implications of RQ1 Findings

The **RQ1** was tackled by accomplishing objectives 1, 2, and 3. Initially, the theoretical study, consisting of a literature review and analysis of prior research conducted by the authors of this thesis, has provided insights into the application landscape of ER as a tool for teaching-learning programming. Additionally, it has delved into general educational theories, essential for establishing a robust foundation for the proposed educational process aimed at teaching and learning programming with robotics, with a specific focus on CT. The Anthropological Theory of Didactics proposes that every human-created entity, known as an Object (O), is associated with at least one individual, denoted as Person (X). Objects can be abstract concepts or tangible items resulting from intentional human actions. This relationship between the Object and the Person, expressed as R(X,O), signifies a dynamic connection that evolves and can be influenced by various social contexts represented by the Institution [I] where the Person is situated. Hence, the Anthropological Theory of Didactics [46], and the previous research was used as a base to propose the educational process named CTProgER. The CTProgER contemplates six didactic moments (problematic, analysis, discovery, implementation, validation, and assessment). Also, it 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.

In addition to the conceptual design of CTProgER, three specific instances were developed, named "The Cleaning Robot", "The Accountant Robot", and "The Driver Robot", each aligned with the guidelines established in CTProgER. Each instance was designed to address different programming and robotics education aspects, as CTProgER outlined. "The Cleaning Robot" was conceived to cover content related to the introduction to algorithms, utilizing these concepts as a fundamental input. Meanwhile, "The Counter Robot" was crafted with a focus on primitive instructions, encompassing variables, input and output of data, data types, constants, and touch sensors. Finally, "The Driver Robot" primarily focused on conditional control structures, particularly exploring ultrasonic sensors. These instances were carefully planned and implemented to provide a comprehensive learning experience, aligning with the pedagogical objectives of CTProgER.

The design process of CTProgER involved a profound reflection on how knowledge is built and transmitted. The research team explored existing educational theories and ventured into fields such as the anthropology of education, especially the Anthropological Theory of Didactics. This approach allowed for a broader understanding of the dynamics involved in the teaching and learning process, leading to valuable insights into designing a more effective educational approach. Notably, insights into the structure of elements [I], (O), (X), R(X,O) from the Anthropological Theory of Didactics favored CTProgER in various ways:

- **Understanding the Relationship between Person and Object:** The Anthropological Theory of Didactics highlights the dynamic relationship between the individual (Person) and knowledge (Object), represented by the function R(X,O). This understanding allowed CTProgER to consider not only the content to be taught but also how that content is perceived, assimilated, and applied by students.

- **Identification of the Importance of Social Context:** The theory acknowledges the influence of social contexts on knowledge construction. Through the Institution [I], where learning occurs, CTProgER recognizes the significance of the educational environment and the interaction among teachers, students, and content.

- **Consideration of Didactic Moments:** The Anthropological Theory of Didactics emphasizes the importance of different moments in teaching-learning. CTProgER incorporates six didactic moments (problematic, analysis, discovery, implementation, validation, and assessment), providing a robust framework for planning and executing programming classes with robotics.

- **Emphasis on Knowledge Construction:** The theory promotes the idea that the teaching-learning process goes beyond information transmission, focusing instead on the active construction of knowledge by the student. CTProgER adopts a student-centered approach, encouraging discovery, exploration, and practical application of programming and computational thinking concepts.

One of the distinctive features of CTProgER's design process was integrating knowledge from various disciplines. In addition to traditional educational theories, concepts from anthropology, computer science, and pedagogy, among others, were explored. This interdisciplinary approach significantly enriched CTProgER's conception, providing a solid and comprehensive foundation for its development.

A central premise of CTProgER's design was the emphasis on practice and user experience. The method was designed to be highly interactive and engaging, providing students with practical and meaningful learning experiences. Activities were carefully planned to stimulate curiosity, creativity, and critical thinking, ensuring an enriching and memorable learning experience.

CTProgER has demonstrated alignment with active teaching methodologies, which have gained prominence in contemporary education. These approaches place the student at the center of the learning process, promoting active participation, collaboration, and the development of practical skills. By incorporating elements of active methodologies, such as problem-based learning, collaborative learning, and active learning, CTProgER provides a dynamic and practical educational experience.

Active teaching methodologies have been widely recognized for their benefits in education. They stimulate critical thinking, problem-solving, and decision-making, preparing students to face real-world challenges. Additionally, they promote autonomy and responsibility for one's learning, empowering students to become autonomous and self-sufficient learners.

The results obtained from the student intervention demonstrate that CTProgER has a positive impact on students' computational thinking development. This finding further emphasizes CTProgER's effectiveness as an approach aligned with active teaching methodologies, as these methodologies prioritize practice, active student participation, and the development of critical skills for academic and professional success. Thus, CTProgER promotes effective learning of programming with robotics and prepares students to be critical thinkers and competent problem solvers in an increasingly technological and complex world.

7.2 Implications of RQ2 Findings

The **RQ2** was addressed through objective 4. A two-round Delphi method [136] experiment examined the conformity of the instances with CTProgER guidelines, as well as the overall coherence of CTProgER, considering the perspective of experts. Teachers and researchers with prior experience in research or practice with CT, programming teaching, or Educational Robotics were engaged as experts to validate the instances. This experiment aided in devising strategies to validate the effectiveness of educational processes from the perspective of expert teachers. The instances' validation, conducted through the Delphi method, contributed to assessing the alignment of the instances with CTProgER guidelines. The insights gained from this experiment are invaluable for refining and enhancing the educational process, ensuring its adequacy and effectiveness in fostering CT skills among students through teaching-learning programming with ER.

A strong culture of collaboration and partnership characterized CTProgER's design process. Researchers, educators, robotics experts, and other professionals worked together, sharing ideas, experiences, and perspectives to collaborate on the design of an innovative educational process. This multidisciplinary collaboration was essential to ensure that CTProgER was holistic and comprehensive in its approach.

Throughout the design process, the involvement of experts played a crucial role. Their diverse expertise and experiences enriched the discussions and created a robust and compelling educational process. In this context, the importance of a rigorous evaluation of CTProgER was recognized to ensure its validity and effectiveness in the educational environment.

The findings made during the design process, especially those that determined the use of the Delphi method for evaluation, promoted the beginning of the construction of guidelines aimed at validating an educational process. Collaboration among experts during the evaluation phase was crucial to identifying strengths and areas for improvement and ensuring that CTProgER was aligned with best practices and educational theories.

The validation process of CTProgER, as outlined in Objective 4, involved a rigorous two-round Delphi method experiment, as described by Murry (YEAR), to ensure a thorough examination of the instances' conformity with CTProgER guidelines and the overall coherence of the educational process. They were engaging teachers and researchers with extensive experience in CT, programming teaching, and Educational Robotics as experts was crucial to obtaining diverse perspectives and insights.

The Delphi method, known for its structured and iterative approach, facilitated the systematic gathering and consolidation of expert opinions. In the first round, experts were presented with the instances developed under CTProgER and asked to provide feedback regarding their alignment with the established guidelines and coherence within the educational process. This initial feedback served as the foundation for refining and enhancing the instances before proceeding to the second round.

During the second round, experts reviewed the revised instances based on the feedback received and engaged in further discussion to achieve consensus on their adequacy and effectiveness. This iterative process allowed for identifying any remaining discrepancies or areas for improvement, ensuring that the final versions of the instances were robust and aligned with CTProgER objectives.

The insights obtained from the Delphi method experiment provided valuable guidance for refining and enhancing the educational process. By incorporating expert feedback and recommendations, CTProgER was able to address potential challenges and optimize its effectiveness in fostering CT skills among students. Moreover, the validation process helped to establish the credibility and validity of CTProgER, reinforcing its suitability for implementation in educational settings.

Overall, the validation with experts through the Delphi method was instrumental in ensuring the adequacy and effectiveness of CTProgER. The collaborative effort between researchers and expert practitioners facilitated the development a robust and coherent educational process poised to impact students' learning outcomes in programming with Educational Robotics.

Applying the Delphi method also provided a valuable learning experience for researchers and expert participants. Through the iterative nature of the Delphi process, researchers gained more profound insights into the complexities of educational process validation and the nuances of expert judgment in the field of Computer Science Education.

As the Delphi rounds progressed, researchers observed how expert opinions evolved and converged, shedding light on the factors influencing consensus-building and decision-making within the expert community. This observation allowed researchers to understand better the diverse perspectives and considerations that experts bring to evaluate educational processes.

Furthermore, the collaborative nature of the Delphi method encouraged open communication and knowledge exchange among participants. Experts had the opportunity to engage in constructive dialogue, share best practices, and learn from each other's experiences. This collaborative exchange of ideas enriched the validation process and contributed to the professional development of all involved.

Moreover, researchers benefited from the feedback and recommendations provided by experts, gaining valuable insights into potential areas for improvement and refinement of CTProgER. Researchers actively solicited and incorporated expert input to enhance the educational process, ensuring its alignment with best practices and pedagogical principles.

Finding experts for the validation process posed a significant challenge and was one of the foremost hurdles encountered during the research. Identifying individuals with the requisite expertise in Computer Science Education, programming teaching, and Educational Robotics, who were also willing to dedicate their time and effort to participate in the validation process, proved daunting. The recruitment challenge was exacerbated by the need for more experts in these specialized domains, compounded by their busy schedules and professional commitments. Despite concerted efforts to reach potential candidates through various channels, including academic networks and professional organizations, assembling a diverse and representative panel of experts took time and effort. This difficulty underscored the importance of proactive planning and strategic outreach in engaging experts for validation activities, highlighting the need for innovative approaches to overcome such obstacles in future research endeavors.

Overall, applying the Delphi method not only facilitated the validation of CTProgER but also served as a platform for continuous learning and professional growth. The insights gained from this experience will inform future research endeavors and contribute to the ongoing advancement of Computer Science Education.

7.3 Implications of RQ3 Findings

The **RQ3** found its resolution in pursuing objectives 5 and 6. An intervention study evaluated the CTProgER's effectiveness as perceived by students. Specifically tailored for Computer Technical and Vocational High School students, the intervention targets teaching-learning programming by applying CTProgER. The primary aim is to enhance the teaching and learning of programming, mainly cultivating CT skills. The results highlight that the educational process built to teach programming through ER is capable of helping the student develop his CT skills better because a significant difference was observed between the participants who had contact with the CTProgER. The procedures to respond to the RQ2 and the result of the RQ3 contribute to the scientific community in guiding the validation of an educational process.

The intervention study was meticulously crafted to create an immersive learning experience wherein students were exposed to programming concepts and educational robotics within the framework of CTProgER. The primary objective of the intervention was to foster the cultivation of CT skills among students, thereby enhancing their overall programming proficiency and problem-solving abilities.

Throughout the intervention, students actively engaged with CTProgER, participating in structured learning activities and hands-on robotics projects. These activities were carefully designed to promote critical thinking, logical reasoning, algorithmic problem-solving, and creativity, all essential components of CT.

The results of these assessments revealed a significant positive impact of CTProgER on students' CT skills, as evidenced by notable improvements in their problem-solving capabilities, computational reasoning, and overall programming proficiency. Moreover, comparative analyses between students who underwent the CTProgER intervention and those who received conventional programming instruction highlighted the distinct advantages of the CTProgER approach. Students who were exposed to CTProgER demonstrated significantly more significant gains in CT skills, indicating the efficacy of this innovative educational process in fostering meaningful learning outcomes.

While the results of interventions may sometimes appear modest, it's crucial to contextualize them within the broader framework of the study. In the case of the intervention, where the CTProgER was the sole difference between the experimental and control groups, even seemingly small results carry valid and significant implications. When evaluating educational practices, the impact extends beyond mere numerical values.

In educational research, the significance of results often lies in the qualitative changes observed in students' learning experiences and outcomes. While statistical measures provide valuable insights, they may only partially capture the depth of transformation occurring within the classroom. For instance, though numerically modest, the observed improvement in students' CT skills reflects a fundamental shift in their cognitive processes and problem-solving abilities. This qualitative enhancement in students' capabilities can have far-reaching implications for their academic and professional trajectories.

Moreover, interventions like CTProgER often catalyze broader systemic changes within educational institutions. By demonstrating the effectiveness of innovative teaching methods, such interventions pave the way for adopting evidence-based practices and refining educational policies. The ripple effects of such interventions extend beyond individual classrooms, influencing pedagogical approaches at the institutional and policy levels.

Furthermore, the significance of intervention studies lies in their immediate outcomes and their potential to inform future research and practice. Each intervention is a building block in the ongoing quest to improve educational outcomes and enhance student learning experiences. The insights gained from interventions like CTProgER contribute to the cumulative body of knowledge in the field, guiding researchers and practitioners in designing and implementing effective educational interventions.

In conclusion, while the results of intervention studies may seem modest at first glance, their implications are far-reaching and multifaceted. By contextualizing these results within the broader landscape of educational research and practice, it can appreciate the profound impact of interventions like CTProgER on student learning and educational innovation.

It is also important to highlight that performing an intervention study in public schools can pose several significant challenges, reflecting the complexity of the school environment and the multiple variables involved in the educational process. Among these challenges, the lack of adequate infrastructure, the unstable dynamics of public schools, and the need to adhere to network educational guidelines stand out.

Firstly, the need for more infrastructure in public schools can be a significant obstacle to implementing intervention studies. Many schools face issues such as overcrowded classrooms, a lack of suitable technological equipment, and deficiencies in physical infrastructure, which can directly impact the quality and effectiveness of educational interventions.

Additionally, the volatile dynamics of public schools, characterized by frequent changes in staff, class schedules, and educational policies, can hinder the conduct of long-term studies. These unforeseen changes can interfere with the schedule and continuity of interventions, thus compromising data collection and result analysis.

Another challenge when conducting intervention studies in public schools is dealing with administrative issues, such as student transfer requests and adjustments to network educational guidelines. The bureaucracy associated with these processes can lead to delays and logistical complications, impacting the planning and execution of interventions.

In light of these challenges, researchers need to be prepared to face adversity and develop flexible strategies to address the complexities of the school environment. This may include establishing solid partnerships with schools, mobilizing additional resources, and adopting adaptive approaches considering each school context's specific conditions and needs. Overcoming these challenges is crucial to ensuring the quality and relevance of intervention studies in public schools and promoting significant improvements in education.

7.4 Contribution

The main contribution of this thesis is the development and availability of a validated educational process specifically designed for robotics teaching-learning, with a central focus on enhancing CT. This process represents an efficient approach to integrating robotics into the academic curriculum, aiming to foster essential CT skills among students. The careful validation of this process confirms its effectiveness and relevance in the educational context, providing a solid and proven educational process to enhance teaching-learning programming with robotics. By specifically targeting the development of CT, this research not only equips students with technical skills in programming and robotics but also empowers them with a crucial analytical and critical mindset to tackle the complex challenges of the 21st century. Furthermore, by providing a validated model for programming teaching-learning with robotics centered on CT, this thesis has the potential to positively influence educational practices worldwide, promoting a more holistic and practical approach to computer science and technology education.

However, over its four years of development, the results obtained in this thesis have generated other significant contributions disseminated within the scientific community. Several studies from this research have already been published and are well-received, providing valuable support and guidance for other researchers involved in similar investigations. Below, it will list the publications resulting from this work:

1. Educational Robotics Applied to Computational Thinking Development: A Systematic Mapping Study in IEEE Frontiers in Education Conference (FIE) - 2021 [171];

2. A Framework for teaching programming in High School through Educational Robotics in IEEE Frontiers in Education Conference (FIE) - 2022 [172];

3. Analyzing the Effectiveness of an Educational Process for Teaching Programming Through Educational Robotics in a Brazilian Technical and Vocational High School in XXXIV Brazilian Symposium on Informatics in Education - 2023 [169].

These publications highlight the relevance and impact of the current study in the field, showcasing its fundamental role in advancing scientific knowledge and guiding researchers engaged in related topics.

7.5 Future Works

Numerous avenues for advancing this work emerge in light of the accomplished objectives and contributions. Several of these possibilities are outlined below:

- Conducting a qualitative study that leverages observational data from the intervention to provide a more nuanced characterization of the impact and a deeper understanding of the effectiveness of the educational process. This study will involve the analysis of class records.

- Analyzing to determine which specific CT skills are most significantly impacted by the CTProgER. This investigation aims to identify and prioritize the essential abilities that experience notable enhancement through implementing CTProgER, contributing to a more targeted understanding of its educational impact on CT skills.

- Exploring the potential for extending CTProgER to other institutions [I].
- Experimenting with expert teachers to assess the efficiency of CTProgER in its instantiation. This study is crucial for understanding how educators, who are not the authors of this thesis, perceive the ease or difficulty of implementing lessons following CTProgER guidelines. This aspect can directly influence the CTProgER's effectiveness.
- Analyzing the possibility of expanding CTProgER beyond the context of teaching-learning programming.

Given the considerations, this study is expected to foster interest in practices that stimulate CT and propel more in-depth research on the subject in the context of teaching-learning programming with robotics and other disciplines in Basic Education. It is of great importance that programming classes can impact students, improving their problem-solving skills. Additionally, it provides more informed assistance to the scientific community regarding the validation process of educational processes.

Bibliography

[1] Sheikh Iqbal Ahamed, Dennis Brylow, Rong Ge, Praveen Madiraju, Stephen J. Merrill, Craig A. Struble, and James P. Early. Computational thinking for the sciences: A three day workshop for high school science teachers. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education - Association for Computing Machinery*, New York, NY, USA, 2010.

[2] Ashraf 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. IEEE, 2022.

[3] Fernando Alegre, John Underwoood, Juana Moreno, and Mario Alegre. Introduction to computational thinking: A new high school curriculum using codeworld. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education - Association for Computing Machinery*, 2020.

[4] Aho Alfred V. Computation and computational thinking. *The Computer Journal*, 55(7):832–835, 2012.

[5] D Alimisis. Educational robotics: New challenges and trends. *Themes in the Science and Technology Education*, 6:63–71, 2013.

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

[7] Bahjat Altakhayneh. The impact of using the lego education program on mathematics

achievement of different levels of elementary students. *European Journal of Educational Research*, 9(2):603–610, 2020.

[8] Charoula Angeli and Nicos Valanides. Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, 105:105954, 2020.

[9] Charoula Angeli and Nicos Valanides. Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, 105:105954, 2020.

[10] Joao Tiago Aparicio, Sergio Pereira, Manuela Aparicio, and Carlos J Costa. Learning programming using educational robotics. In *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6. IEEE, 2019.

[11] Mikko Apiola, Matti Lattu, and Tomi A Pasanen. Creativity and intrinsic motivation in computer science education: experimenting with robots. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*, pages 199–203, 2010.

[12] George Aranda and Joseph Paul Ferguson. Unplugged programming: The future of teaching computational thinking? *Pedagogika*, 68(3), 2018.

[13] Ana Liz Araujo, Wilkerson Andrade, and Dalton Guerrero. Um mapeamento sistemático sobre a avaliaçao do pensamento computacional no brasil. In *Anais dos do Walgprog / CBIE*, 2016.

[14] Ana Liz Souto O Araujo, Wilkerson Andrade, Dalton Guerrero, Monilly Melo, and Isabelle Maria Lima de Souza. Análise de rede na identificação de habilidades relacionadas ao pensamento computacional. In *SBIE / CBIE*, 2018.

[15] Ana Liz Souto O Araujo, Wilkerson L Andrade, Dalton D Serey Guerrero, and Monilly Ramos Araujo Melo. How many abilities can we measure in computational thinking? a study on bebras challenge. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 545–551, 2019.

[16] Gerald Ardito, Betül Czerkawski, and Lauren Scollins. Learning computational thinking together: Effects of gender differences in collaborative middle school robotics program. *TechTrends*, 64(3):373–387, 2020.

[17] Lito Athanasiou, Paraskevi Topali, and Tassos A Mikropoulos. The use of robotics in introductory programming for elementary students. In *International Conference EduRobotics 2016*, pages 183–192. Springer, 2016.

[18] Soumela Atmatzidou and Stavros Demetriadis. Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75:661—670, 2016.

[19] Soumela Atmatzidou and Stavros Demetriadis. Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75:661–670, 2016.

[20] Soumela Atmatzidou and Stavros Demetriadis. Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75:661–670, 2016.

[21] Soumela Atmatzidou and Stavros Demetriadis. A didactical model for educational robotics activities: A study on improving skills through strong or minimal guidance. In *International Conference EduRobotics 2016*, pages 58–72. Springer, 2016.

[22] Sharifah Maryam Syed Azman, Mahyuddin Arsat, and Hasnah 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)*, pages 61–65. IEEE, 2017.

[23] Brenda Bannan-Ritland. The role of design in research: The integrative learning design framework. *Educational researcher*, 32(1):21–24, 2003.

[24] Valerie Barr and Chris Stephenson. Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community? *Acm Inroads*, 2(1):48–54, 2011.

[25] Bernard Beech. The delphi approach: recent applications in health care. *Nurse Researcher (through 2013)*, 8(4):38, 2001.

[26] Marina U Bers, Carina González-González, and Mª Belén Armas-Torres. Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, 138:130–145, 2019.

[27] Marina U Bers, Carina González-González, and Mª Belén Armas-Torres. Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, 138:130–145, 2019.

[28] Marina Umaschi Bers, Louise Flannery, Elizabeth R Kazakoff, and Amanda Sullivan. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72:145–157, 2014.

[29] Marina Umaschi Bers, Louise Flannery, Elizabeth R Kazakoff, and Amanda Sullivan. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72:145–157, 2014.

[30] Nick Black, Maggie Murphy, Donna Lamping, Martin McKee, Colin Sanderson, Janet Askham, and Theresa Marteau. Consensus development methods: a review of best practice in creating clinical guidelines. *Journal of health services research & policy*, 4(4):236–248, 1999.

[31] Maria Blancas, Cristina Valero, Anna Mura, Vasiliki Vouloutsi, and Paul FMJ Verschure. ”crea”: An inquiry-based methodology to teach robotics to children. In *International Conference on Robotics in Education (RiE)*, pages 45–51. Springer, 2019.

[32] Fergus Bolger and George Wright. Improving the delphi process: Lessons from social psychological research. *Technological forecasting and social change*, 78(9):1500–1513, 2011.

[33] Adriana Bordini, Christiano Avila, Monica Marques, Luciana Foss, and Simone 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)*, page 123, 2017.

[34] Marianna Bosch, Yves Chevallard, and Josep Gascón. Science or magic? the use of models and theories in didactics of mathematics. In *Proceedings of the fourth congress of the European Society for research in mathematics education*, pages 1254–1263. Fundemi IQS–Universitat Barcelona, 2006.

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

[36] Christian Puhlmann Brackmann. *Desenvolvimento do Pensamento Computacional através de atividades desplugadas na Educação Básica. 2017. 226 f.* PhD thesis, Tese (Doutorado em Informática na Educação)–Universidade Federal do Rio de Janeiro, 2017.

[37] Karen Brennan and Mitchel 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.

[38] Frederick P Brooks. The mythical man-month. *Datamation*, 20(12):44–52, 1974.

[39] Paolo Bucci, Timothy J. Long, and Bruce W. Weide. Do we really teach abstraction? *SIGCSE Bull.*, 33(1), feb 2001.

[40] Bülent Çavas, L Özge Güney, Emre Karagöz, and Pinar Çavas. More than playing a toy: The effects of lego mindstorms on the students' perceptions about scientists. *Science Education International*, 31(1):92–103, 2020.

[41] Christina Chalmers. Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, 17:93–100, 2018.

[42] Po-Yao Chao. Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education*, 95:202–215, 2016.

[43] Jacqui Chetty. Lego© mindstorms: Merely a toy or a powerful pedagogical tool for learning computer programming? In *ACSC*, pages 111–118, 2015.

[44] Morgane Chevalier, Christian Giang, Alberto Piatti, and Francesco Mondada. Fostering computational thinking through educational robotics: A model for creative computational problem solving. *International Journal of STEM Education*, 7(1):1–18, 2020.

[45] Morgane Chevalier, Christian Giang, Alberto Piatti, and Francesco Mondada. Fostering computational thinking through educational robotics: a model for creative computational problem solving. *International Journal of STEM Education*, 7(1):1–18, 2020.

[46] Yves 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.

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

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

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

[50] Yves Chevallard. Steps towards a new epistemology in mathematics education. In *Proceedings of the IV congress of the European society for research in mathematics education*, pages 21–30, 2006.

[51] Yves Chevallard, Marianna Bosch, and Josep Gascón. *Estudiar matemáticas: el eslabón perdido entre la enseñanza y el aprendizaje*. Biblioteca del Normalista de la SEP, 1998.

[52] Yves Chevallard, Marianna Bosch, and Josep Gascón. *Estudiar matemáticas: el eslabón perdido entre la enseñanza y el aprendizaje*. Biblioteca del Normalista de la SEP, 1998.

[53] Giuseppe Chiazzese, Marco Arrigo, Antonella Chifari, Violetta Lonati, and Crispino Tosto. Exploring the effect of a robotics laboratory on computational thinking skills in primary school children using the bebras tasks. *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 25–30, 2018.

[54] Giuseppe Chiazzese, Marco Arrigo, Antonella Chifari, Violetta Lonati, and Crispino Tosto. Exploring the effect of a robotics laboratory on computational thinking skills in primary school children using the bebras tasks. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 25–30, 2018.

[55] Yu-Hui Ching and Yu-Chang Hsu. Educational robotics for developing computational thinking in young learners: A systematic review. *TechTrends*, pages 1–12, 2023.

[56] Sasithorn Chookaew, Suppachai Howimanporn, Pornjit Pratumsuwan, Santi Hutamarn, Warin Sootkaneung, and Charoenchai Wongwatkit. Enhancing high-school students' computational thinking with educational robotics learning. In *2018 7th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 204–208. IEEE, 2018.

[57] Robert Coe. It's the effect size, stupid. In *British Educational Research Association Annual Conference*, volume 12, page 14, 2002.

[58] Jacques Colomb. Chevallard (yves). —la transposition didactique: du savoir savant au savoir enseigné. *Revue française de pédagogie*, 76(1):89–91, 1986.

[59] Vaso Constantinou and Andri Ioannou. Development of computational thinking skills through educational robotics. In *EC-TEL (Practitioner Proceedings)*, 2018.

[60] Alzira Ferreira da Silva. *RoboEduc: A Learning Methodology with Eucational Robotics (in Portuguese)*. PhD thesis, Doctorate's Thesis at Graduate Program in Electrical and Computer Engineering, Universidade Federal do Rio Grande do Norte, Natal, Brazil, 2009. available at `https://repositorio.ufrn.br/handle/123456789/15128`.

[61] Lidiane Ottoni da Silva Petini. Conhecimentos matemáticos mobilizados por alunos no desenvolvimento de projetos de robótica. *Semana da Matemática do Instituto de Matemática*, 2018.

[62] Valentina Dagiene, Sue Sentance, and Gabriela Stupuriene. Developing a two-dimensional categorization system for educational tasks in informatics. *Informatica*, 28:23–44, 01 2017.

[63] Norman Dalkey and Olaf Helmer. An experimental application of the delphi method to the use of experts. *Management science*, 9(3):458–467, 1963.

[64] Bruno Henrique de Paula, Andrew Burn, Richard Noss, and José Armando Valente. Playing beowulf: Bridging computational thinking, arts and literature through game-making. *International journal of child-computer interaction*, 16:39–46, 2018.

[65] Isabelle Maria Lima de Souza, Rivanilson da Silva Rodrigues, and Wilkerson 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)*, page 490, 2016.

[66] Isabelle Maria Lima de Souza, Rivanilson da Silva Rodrigues, and Wilkerson Andrade. Introdução do pensamento computacional na formação docente para ensino de robótica educacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, page 1265, 2016.

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

[68] Martyn Denscombe. *EBOOK: The Good Research Guide: For Small-Scale Social Research Projects*. McGraw-Hill Education (UK), 2017.

[69] Oxford English Dictionary. Oxford english dictionary. *Simpson, Ja & Weiner, Esc*, page 3, 1989.

[70] Halfpapa Dulce Maria. Um modelo de consciência para aplicação em artefatos inteligentes. In *Tese de doutorado, Universidade Federal de Santa Catarina*. Centro Tecnológico. Engenharia de Produção e Sistemas., 2005.

[71] Rajeev Dwivedi, Arpan Kumar, Bharathy Babu, Nipun Grandhi, Rishi Meka, and Varun Ahuja. The role of competitive robotics in providing context to classroom learning and technical skill development in school age students: A survey of current avenues, assessment, and path forward with systematic implementation. *Education Sciences*, 11(4):167, 2021.

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

[73] Amy Eguchi. Robotics as a learning tool for educational transformation. In *Proceeding of 4th international workshop teaching robotics, teaching with robotics & 5th international conference robotics in education Padova (Italy)*, pages 27–34, 2014.

[74] Andoni Eguíluz, Pablo Garaizar, and Mariluz Guenaga. An evaluation of open digital gaming platforms for developing computational thinking skills. *Simulation and gaming*, pages 143–168, 2018.

[75] Lisser R Ejersbo, Robin Engelhardt, Lisbeth Frølunde, Thorkild Hanghøj, Rikke Magnussen, and Morten Misfeldt. Balancing product design and theoretical insights. *The handbook of design research methods in education: Innovations in science, technology, engineering and mathematics learning and teaching*, pages 149–164, 2008.

[76] Skinner Ellen A., Kindermann Thomas A., and Furrer Carrie J. A motivational perspective on engagement and disaffection: Conceptualization and assessment of children's behavioral and emotional participation in academic activities in the classroom. *Educational and Psychological Measurement*, 69(3):493–525, 2009.

[77] Rainer Feurer and Kazem Chaharbaghi. Defining competitiveness: a holistic approach. *Management decision*, 32(2):49–58, 1994.

[78] Andy Field. *Discovering statistics using IBM SPSS statistics*. sage, 2013.

[79] Arlene Fink, Jacqueline Kosecoff, Mark Chassin, and Robert H Brook. Consensus methods: characteristics and guidelines for use. *American journal of public health*, 74(9):979–983, 1984.

[80] GmbH® Fischertechnik. Fischertechnik. In *The construction kit for children of all ages Fischertechnik*, 2018.

[81] Qian Fu, Li-Wen Zhang, Jon-Chao Hong, and Yan Dong. Development and verification of the programming resilience scale for university students. *Australasian Journal of Educational Technology*, 37(6), Jul. 2021.

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

[83] Tatiana Engel Gerhardt and Denise Tolfo Silveira. *Métodos de pesquisa*. Plageder, 2009.

[84] Christian Giang, Morgane Chevalier, Lucio Negrini, Ran Peleg, Evgeniia Bonnet, Alberto Piatti, and Francesco Mondada. Exploring escape games as a teaching tool in educational robotics. In *International Conference EduRobotics 2016*, pages 95–106. Springer, 2019.

[85] Antonio Carlos Gil. *Métodos e técnicas de pesquisa social*. 6. ed. Ediitora Atlas SA, 2008.

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

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

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

[89] Yen Air Caballero González and Ana García-Valcárcel Muñoz Repiso. Development of computational thinking skills and collaborative learning in initial education students through educational activities supported by ict resources and programmable ed-

ucational robots. In *Association for Computing Machinery*, TEEM 2017, New York, NY, USA, 2017.

[90] Yen Air Caballero González and Ana García-Valcárcel Muñoz-Repiso. Educational robotics for the formation of programming skills and computational thinking in childish. In *2017 International Symposium on Computers in Education (SIIE)*, pages 1–5. IEEE, 2017.

[91] Yen Air Caballero González and Ana García-Valcárcel Muñoz-Repiso. A robotics-based approach to foster programming skills and computational thinking: Pilot experience in the classroom of early childhood education. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 41–45, 2018.

[92] Brent Graham, Glenn Regehr, and James G Wright. Delphi as a method to establish consensus for diagnostic criteria. *Journal of clinical epidemiology*, 56(12):1150–1156, 2003.

[93] Matthew Graham, Anthony Milanowski, and Jackson Miller. Measuring and promoting inter-rater agreement of teacher and principal performance ratings. *Online Submission*, 2012.

[94] Megan M Grime and George Wright. Delphi method. *Wiley StatsRef: Statistics Reference Online*, pages 1–6, 2016.

[95] Travassos Guilherme Horta, Gurov Dmytro, and EAGG Amaral. Introdução á engenharia de software experimental. *UFRJ*, 2002.

[96] Ernest BB Gyebi, Marc Hanheide, and Grzegorz Cielniak. The effectiveness of integrating educational robotic activities into higher education computer science curricula: a case study in a developing country. In *International Conference EduRobotics 2016*, pages 73–87. Springer, 2016.

[97] Derrylea J Hardy, Anthony P O'Brien, Cadeyrn J Gaskin, Anthony J O'Brien, Erina Morrison-Ngatai, Georgina Skews, Tom Ryan, and Neil McNulty. Practical applica-

tion of the delphi technique in a bicultural mental health nursing study in new zealand. *Journal of advanced nursing*, 46(1):95–109, 2004.

[98] Lan Hong. The impact of educational robots on students' computational thinking: A meta-analysis of k-12. *Education and Information Technologies*, pages 1–26, 2023.

[99] Ting-Chia Hsu, Shao-Chen Chang, and Yu-Ting Hung. How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 2018.

[100] Chenglie Hu. Computational thinking: what it might mean and what we might do about it. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, pages 223–227, 2011.

[101] Wei-Yeh Huang, Chiu-Fan Hu, and Cheng-Chih Wu. The use of different kinds of robots to spark student interest in learning computational thinking. In *2018 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, pages 11–16. IEEE, 2018.

[102] Andri Ioannou and Eria Makridou. Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. *Education and Information Technologies*, 23:2531–2544, 2018.

[103] Souza Isabelle M. L., Wilkerson L. Andrade, and Sampaio Lívia M. R. Analyzing the effect of computational thinking on mathematics through educational robotics. In *2019 IEEE Frontiers in Education Conference (FIE)*, pages 1–7, 2019.

[104] Souza Isabelle M. L., Wilkerson L. Andrade, and Sampaio Lívia M. R. Analyzing the effect of computational thinking on mathematics through educational robotics. In *2019 IEEE Frontiers in Education Conference (FIE)*, pages 1–7, 2019.

[105] Wainer Jacques et al. Métodos de pesquisa quantitativa e qualitativa para a ciência da computação. *Atualização em informática*, 1:221–262, 2007.

[106] Dayang NA Jawawi, Noraini Ibrahim, Shahliza Abdul Halim, Rosbi Mamar, Norhidawani Mohamed, and Rooster Tumeng. Adaptation of project-oriented problem-based framework for teaching computer programming. In *2017 7th World Engineering Education Forum (WEEF)*, pages 844–849. IEEE, 2017.

[107] Wing Jeannette M. Computational thinking. *Commun. ACM*, 49(3):33–35, March 2006.

[108] Wing Jeannette M. Computational thinking. *Commun. ACM*, 49(3):33–35, March 2006.

[109] Hi-Lian Jeng, Li-Wen Liu, and Chung-Nien Chen. Developing a procedural problem-solving-based framework of computational thinking components. In *2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 272–277, 2019.

[110] Filiz Kalelioglu. A new way of teaching programming skills to k-12 students: Code.org. *Computers in Human Behavior*, 52:200–210, 2015.

[111] Elizabeth R Kazakoff, Amanda Sullivan, and Marina U Bers. The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 41(4):245–255, 2013.

[112] P Kevin Keith, Florence R Sullivan, and Duy Pham. Roles, collaboration, and the development of computational thinking in a robotics learning environment. In *Computational Thinking Education*, pages 223–245. Springer, Singapore, 2019.

[113] Bayram Koyuncu, Majlinda Fetaji, Bekim Fetaji, and Anis Sefidanovski. Analyzing the impact of new instruction method using hardware control such as robotics in learning programming. In *2019 8th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–4. IEEE, 2019.

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

[115] Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. Computational thinking for youth in practice. *Acm Inroads*, 2(1):32–37, 2011.

[116] Jacqueline Leonard, Alan Buss, Ruben Gamboa, Monica Mitchell, Olatokunbo S Fashola, Tarcia Hubert, and Sultan Almughyirah. Using robotics and game design to enhance children's self-efficacy, stem attitudes, and computational thinking skills. *Journal of Science Education and Technology*, 25(6):860–876, 2016.

[117] Jacqueline Leonard, Alan Buss, Ruben Gamboa, Monica Mitchell, Olatokunbo S Fashola, Tarcia Hubert, and Sultan Almughyirah. Using robotics and game design to enhance children's self efficacy, stem attitudes, and computational thinking skills. *Journal of Science Education and Technology*, 25, 2016.

[118] Vygotsky Lev Semenovich. *Mind in society: The development of higher psychological processes*. Harvard university press, 1980.

[119] Harold A Linstone, Murray Turoff, et al. *The delphi method*. Addison-Wesley Reading, MA, 1975.

[120] Cohen Louis, Morrison Keith, and Manion Lawrence. Research methods in education. *IEducation, Research methods. Routledge.*, 2011.

[121] Sze Yee Lye and Joyce Hwee Ling 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.

[122] Mataric Maja J. Introdução á robótica. *São Paulo: UNESP/BLUCHER*, 2014.

[123] Alessio Malizia, Tommaso Turchi, and Kai 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.

[124] Maria José Marcelino, Teresa Pessoa, Celeste Vieira, Tatiana Salvador, and António José Mendes. Learning computational thinking and scratch at distance. *Computers in Human Behavior*, 80:470–477, 2018.

[125] Maja J Mataric. Robotics education for all ages. In *Proc. AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education*, pages 22–24, 2004.

[126] Igor Matyushchenko, Elena Zvereva, and Tatiana Lavina. Development of algorithmic thinking by means of lego mindstorms ev3 on robotics. In *2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*, pages 444–447, 2020.

[127] Susan McKenney and Thomas C Reeves. Educational design research: portraying, conducting, and enhancing productive scholarship. *Medical Education*, 55(1):82–92, 2021.

[128] Donna Mead and L Mosely. The use of the delphi as a research approach. *Nurse researcher*, 8(4):4–23, 2001.

[129] Alexandres Merkouris and Konstantinos Chorianopoulos. Programming touch and full-body interaction with a remotely controlled robot in a secondary education stem course. In *Proceedings of the 22nd Pan-Hellenic Conference on Informatics*, pages 225–229, 2018.

[130] Alexandros Merkouris and Konstantinos Chorianopoulos. Programming embodied interactions with a remotely controlled educational robot. *ACM Transactions on Computing Education (TOCE)*, 19(4):1–19, 2019.

[131] Alexandros Merkouris, Konstantinos Chorianopoulos, and Achilles Kameas. Teaching programming in secondary education through embodied computing platforms: Robotics and wearables. *ACM Transactions on Computing Education (TOCE)*, 17(2):1–22, 2017.

[132] Carbonaro Mike, Rex Marion, and Chambers Joan. Using lego robotics in a project-based learning environment. *The Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 6(1), 2004.

[133] Blanca Miller, Adam Kirn, Mercedes Anderson, Justin C Major, David Feil-Seifer, and Melissa Jurkiewicz. Unplugged robotics to increase k-12 students' engineering

interest and attitudes. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–5. IEEE, 2018.

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

[135] Omar Mubin, Catherine J Stevens, Suleman Shahid, Abdullah Al Mahmud, and Jian-Jie Dong. A review of the applicability of robots in education. *Journal of Technology in Education and Learning*, 1(209-0015):13, 2013.

[136] John W Murry Jr and James O Hammons. Delphi: A versatile methodology for conducting qualitative research. *The review of higher education*, 18(4):423–436, 1995.

[137] Lucio Negrini and Christian Giang. How do pupils perceive educational robotics as a tool to improve their 21st century skills? *Journal of e-Learning and Knowledge Society*, 15(2), 2019.

[138] Kristie J Newton, Jacqueline Leonard, Alan Buss, Christopher G Wright, and Joy Barnes-Johnson. Informal stem: Learning with robotics and game design in an urban context. *Journal of Research on Technology in Education*, 52(2):129–147, 2020.

[139] Jiyae Noh and Jeongmin Lee. Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development*, 68(1):463–484, 2020.

[140] Jiyae Noh and Jeongmin Lee. Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development*, 68(1):463–484, 2020.

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

[142] Masahiro Osogami, Kazutomi Sugihara, and Kazumasa Ohkuma. The effects for programming learning using actual robots control with scratch. *International Journal of Mechanical Engineering and Robotics Research*, 8(5):764–770, 2019.

[143] Basile Panoutsopoulos. Introducing science technology engineering and mathematics in robotics outreach programs. *Technology Interface International Journal*, 12(1):47–53, 2011.

[144] S. Papert. Constructionism: A new opportunity for elementary science education, 1986. "A proposal to the National Science Foundation" - NSF Abstract #8751190, available at `http://dailypapert.com/wp-content/uploads/2021/02/Constructionism-NSF-Proposal.pdf`. Accessed on September, 18th, 2021.

[145] S. Papert. *The Children's Machine: Rethinking School in the Age of the Computer*. Basic Books, New York, USA, 1993.

[146] Seymour Papert. Teaching children thinking. *Programmed Learning and Educational Technology*, 9(5):245–255, 1972.

[147] Seymour Papert and Idit Harel. Situating constructionism. *Constructionism*, 36(2):1–11, 1991.

[148] Raquel E. Patino-Escarcina, Dennis Barrios-Aranibar, Liz S. Bernedo-Flores, and Luiz M. G. Alsina, Pablo J. Gonçalves. A methodological approach to the learning of robotics with edurosc-kids. *J Intell Robot Syst*, 102:34, 2021.

[149] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, volume 17, pages 68–77, 2008.

[150] Monica Pivetti, Silvia Di Battista, Francesca Agatolio, Brunilda Simaku, Michele Moro, and Emanuele Menegatti. Educational robotics for children with neurodevelopmental disorders: A systematic review. *Heliyon*, 6(10):e05160, 2020.

[151] Tjeerd Plomp et al. Educational design research: An introduction. *Educational design research*, pages 11–50, 2013.

[152] Uvais Qidwai, Ryan Riley, and Sayed El-Sayed. Attracting students to the computing disciplines: A case study of a robotics contest. *Procedia-Social and Behavioral Sciences*, 102:520–531, 2013.

[153] André Raabe, André Luiz Maciel Santana, Natália Ellery, and Filipe Gonçalves. Um instrumento para diagnóstico do pensamento computacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, page 1172, 2017.

[154] Alexander Repenning, David Webb, and Andri Ioannidou. Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM technical symposium on Computer science education*, pages 265–269, 2010.

[155] Mitchel Resnick. *Jardim de Infância para a vida toda: por uma aprendizagem criativa, mão na massa e relevante para todos*. Penso Editora, 2020.

[156] Mitchel Resnick, Fred Martin, Randy Sargent, and Brian Silverman. Programmable bricks: Toys to think with. *IBM Systems journal*, 35(3.4):443–452, 1996.

[157] Solange Oliveira Rezende. *Sistemas inteligentes: fundamentos e aplicações*. Editora Manole Ltda, 2003.

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

[159] Marcos Román-Gonzalez, Juan Carlos Pérez-González, and Carmen Jiménez-Fernández. Test de pensamiento computacional: diseño y psicometría general. In *Iii congreso internacional sobre aprendizaje, innovación y competitividad (CINAIC 2015)*, pages 1–6, 2015.

[160] Marcos Román-González, Juan-Carlos Pérez-González, and Carmen Jiménez-Fernández. Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Computers in human behavior*, 72:678–691, 2017.

[161] Margarida Romero, Alexandre Lepage, and Benjamin Lille. Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, 14(1):1–15, 2017.

[162] Evgenia Roussou and Maria Rangoussi. On the use of robotics for the development of computational thinking in kindergarten: Educational intervention and evaluation. In *International Conference on Robotics in Education (RiE)*, pages 31–44. Springer, 2019.

[163] Gene Rowe and George Wright. Expert opinions in forecasting: the role of the delphi technique. In *Principles of forecasting*, pages 125–144. Springer, 2001.

[164] José-Manuel Sáez-López, Marcos Román-Gonzalez, and Esteban Vázquez-Cano. Visual programming languages integrated across the curriculum in elementary school: A two year case study using ”scratch” in five schools. *Computers & Education*, 97:129–141, 2016.

[165] José-Manuel Sáez-López, Maria-Luisa Sevillano-García, and Esteban Vazquez-Cano. The effect of programming on primary school students' mathematical and scientific understanding: educational use of mbot. *Educational Technology Research and Development*, 67(6):1405–1425, 2019.

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

[167] Milton Schivani. *Contextualização no ensino de física á 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.

[168] Isabelle Souza, Wilkerson Andrade, and Lívia 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*, pages 44–54, Porto Alegre, RS, Brasil, 2021. SBC.

[169] Isabelle Souza, Lívia Sampaio, and Wilkerson 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 EducaÃ§Ã£o*, pages 389–401, Porto Alegre, RS, Brasil, 2023. SBC.

[170] Isabelle M. L. SOUZA, Wilkerson L. ANDRADE, and Lívia M. R. SAMPAIO. Educational robotics applications for the development of computational thinking in a brazilian technical and vocational high school. *Informatics in Education*, 2021.

[171] Isabelle M. L. Souza, Wilkerson L. Andrade, and Lívia M. R. Sampaio. Educational robotics applied to computational thinking development: A systematic mapping study. In *2021 IEEE Frontiers in Education Conference (FIE)*, pages 1–8, 2021.

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

[173] Steven E Stemler. A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. *Practical Assessment, Research, and Evaluation*, 9(1):4, 2004.

[174] Veronika Stoffová and Martin Zboran. Educational robotics in teaching programming in primary school. In *Proceedings of International Conference on Recent Innovations in Computing: ICRIC 2022, Volume 1*, pages 669–682. Springer, 2023.

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

[176] Wagner Titon and Alejandro Rafael Garcia Ramirez. Teaching programming concepts using educational robotics, supported by the arduino platform: An application in the industrial computer learning course. In *2018 XIII Latin American Conference on Learning Technologies (LACLO)*, pages 100–106. IEEE, 2018.

[177] Wagner Titon and Alejandro Rafael Garcia Ramirez. Teaching programming concepts using educational robotics, supported by the arduino platform: An application in the

industrial computer learning course. In *2018 XIII Latin American Conference on Learning Technologies (LACLO)*, pages 100–106. IEEE, 2018.

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

[179] Murray Turoff and Harold A Linstone. The delphi method-techniques and applications. *JMR, Journal of Marketing Research*, 2002.

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

[181] Joke Voogt, Petra Fisser, Jon Good, Punya Mishra, and Aman Yadav. Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4):715–728, 2015.

[182] Feng Wang and Michael J Hannafin. Design-based research and technology-enhanced learning environments. *Educational technology research and development*, 53(4):5–23, 2005.

[183] Xuefeng Wei, Lin Lin, Nanxi Meng, Wei Tan, Siu-Cheung Kong, et al. The effectiveness of partial pair programming on elementary school students' computational thinking skills and self-efficacy. *Computers & education*, 160:104023, 2021.

[184] Patricia L Williams and Christine Webb. The delphi technique: a methodological discussion. *Journal of advanced nursing*, 19(1):180–186, 1994.

[185] Jeanette Wing. Research notebook: Computational thinking what and why. *The link magazine*, 6, 2011.

[186] Carl Winsløw. Anthropological theory of didactic phenomena: Some examples and principles of its use in the study of mathematics education. *Un Panorama de TAD, CRM Docume*, pages 117–138, 2011.

[187] James TC Wright, Renata A Giovinazzo, et al. Delphi: uma ferramenta de apoio ao planejamento prospectivo. *Caderno de pesquisas em administração*, 1(12):54–65, 2000.

[188] Aman Yadav, Ninger Zhou, Chris Mayfield, Susanne Hambrusch, and John T Korb. Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 465–470, 2011.

[189] Hatice 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.

[190] Erman Yukselturk and Serhat Altiok. An investigation of the effects of programming with scratch on the preservice it teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, 48(3):789–801, 2017.

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

[192] Humberto Zanetti and Claudio 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*, page 1236, 2015.

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

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

[195] Silvana do Rocio Zilli et al. A robótica educacional no ensino fundamental: perspectivas e prática. *Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia de Produção.*, 2004.

[196] Education for Life ZOOM. Zoom education for life. *Available in: hhttp://zoom.education*, 2010. Accessed on July 18, 2023.

Appendix A

Consent at School Term

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
Rua.: Aprígio Veloso, nº 882, Bairro Universitário, Campina Grande, PB.
CEP.: 58429-900 - Tel.: 2101-1429

DECLARAÇÃO DE ANUÊNCIA

Eu, ____,
Diretor(a) Escolar da Escola____
____, autorizo o desenvolvimento da pesquisa intitulada: "***An Educational Methodology for Teaching Programming with Educational Robotics***", que será realizada nesta instituição no segundo semestre letivo de 2022, tendo como orientador o Prof. Dr. Wilkerson de Lucena Andrade, co-orientadora a Profª. Drª. Lívia Maria Rodrigues Sampaio Campos, e orientanda Isabelle Maria Lima de Souza.

Campina Grande, ___ de ________________ de 2022.

____
Diretor(a) Escolar

Appendix B

Assent Term

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
LABORATÓRIO DE PRÁTICAS DE SOFTWARE
Rua: Aprígio Veloso, nº 882,
Bairro Universitário, Campina Grande, PB.
CEP: 58429-900 - Tel.: 2101-1429

TERMO DE ASSENTIMENTO

Caro Responsável/Representante Legal:

Estamos convidando o aluno menor sob sua responsabilidade, ____________________________
____
a participar da pesquisa "Uma Metodologia Educacional para o Ensino de Programação com Robótica Educacional" e que está sob a responsabilidade da pesquisadora Isabelle Maria Lima de Souza.

Queremos entender como construir uma metodologia para se ensinar programação através da robótica.

Pedimos seu consentimento para que este aluno de menor sob sua responsabilidade participe da pesquisa, mas caso não queira é um direito seu e não terá nenhum problema em não consentir. Os alunos que irão participar desta pesquisa têm de 11 a 17 anos de idade.

A pesquisa será feita na Escola (Nome da Escola) onde o aluno estuda, e para isso será necessário:

- Realização de atividades práticas com Robótica Educacional, durante 1 (um) mês, com aplicação de questionários e testes com alunos e professores;
- Acesso ao desempenho escolar (notas) dos alunos antes e durante as atividades práticas com Robótica Educacional.

Nas atividades, usaremos kits de robótica, computadores ou notebooks, papel, lápis, régua, materiais considerados seguros, mas é possível que os alunos não se sintam confortáveis em participar das atividades e em responder aos questionários e testes. Mas há coisas boas que podem acontecer, como a melhora do desempenho dos alunos nas atividades escolares e o despertar para uma futura formação Universitária.

Gostaríamos de deixar claro que a participação é voluntária e que você poderá retirar seu consentimento, se assim achar melhor. Isso NÃO causará penalização ou prejuízo de qualquer natureza para você ou para o aluno sob sua responsabilidade. Não será cobrado nada, não haverá gastos e não estão previstos ressarcimentos ou indenizações.

Ninguém saberá que o aluno estará participando da pesquisa; não falaremos a outras pessoas, nem daremos a estranhos as informações que obtermos. Os resultados da pesquisa irão ser publicados na dissertação de mestrado da pesquisadora responsável, periódicos e revistas científicas, mas sem identificar os alunos que participaram.

Desde já, agradecemos a atenção e a participação e colocamo-nos à disposição para maiores informações. Você ficará com uma via deste Termo e, em caso de dúvida(s) e outros esclarecimentos sobre esta pesquisa, você poderá entrar em contato com a pesquisadora principal: Isabelle Maria Lima de Souza, Rua Rosa Farias Dantas, 124, Casa B, Novo Cruzeiro, Campina Grande (PB), tel. (83) 99624-1326.

Caso sinta que o menor sob sua responsabilidade foi prejudicado por participar desta pesquisa, poderá recorrer ao Comitê de Ética em Pesquisas com Seres Humanos – CEP, do Hospital Universitário Alcides Carneiro - HUAC, situado a Rua: Dr. Carlos Chagas, s/ n, São José, CEP: 58401 – 490, Campina Grande-PB, Tel: 2101 – 5545, E-mail: cep@huac.ufcg.edu.br; Conselho Regional de Medicina da Paraíba e a Delegacia Regional de Campina Grande.

CONSENTIMENTO PÓS INFORMADO

Eu ____ portador do RG nº: ____________________ confirmo que Isabelle Maria Lima de Souza me explicou os objetivos da pesquisa, portanto, concordo em dar meu consentimento para o referido menor participar como voluntário e aceito participar da pesquisa "Uma Metodologia Educacional para o Ensino de Programação com Robótica Educacional".

Entendi as coisas ruins e as coisas boas que podem acontecer. Entendi que posso dizer "sim" e participar, mas que, a qualquer momento, posso dizer "não" e desistir e que ninguém vai ficar com raiva de mim. Os pesquisadores tiraram minhas dúvidas e ficou claro como a pesquisa irá acontecer.

Recebi uma via deste termo de assentimento, li e concordo em participar da pesquisa.

Campina Grande - PB, ____ de ____________de 2022.

____

(Assinatura responsável ou representante legal)

____

Isabelle Maria Lima de Souza

Pesquisadora Responsável

Appendix C

Consent Term

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
LABORATÓRIO DE PRÁTICAS DE SOFTWARE
Rua: Aprígio Veloso, nº 882,
Bairro Universitário, Campina Grande, PB.
CEP: 58429-900 - Tel.: 2101-1429

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Você está sendo convidado(a) a participar como voluntário de uma pesquisa cujo título é "***Uma Metodologia Educacional para o Ensino de Programação com Robótica Educacional***" e que está sob a responsabilidade da pesquisadora Isabelle Maria Lima de Souza, aluna de doutorado do programa de Pós-Graduação em Computação da UFCG.

Após ser esclarecido(a) sobre as informações a seguir, no caso de aceitar fazer parte do estudo, assine ao final deste documento, que está em duas vias. Uma delas é sua e a outra é do pesquisador responsável. Em caso de recusa, você não será penalizado(a).

O objetivo da pesquisa é fomentar habilidades do Pensamento Computacional por meio da programação com Robótica Educacional no Ensino Médio através da proposição e validação de uma metodologia de ensino. Segundo Wing (2006), o Pensamento Computacional é um conjunto de habilidades desenvolvidas ao estudar conteúdos da Ciência da Computação, mas que favorecem todas as pessoas em atividades de resolução de problemas, competência fundamental para formação de estudantes segundo os Parâmetros Curriculares Nacionais. Os resultados auxiliarão na verificação dos efeitos que uma metodologia de ensino pode causar no aprendizado de programação e consequentemente nas habilidades relacionadas à resolução de problemas.

Para a realização da pesquisa na Instituição de Ensino, poderá ser necessário:

- Realização de atividades práticas com Robótica Educacional, com aplicação de questionários e testes com alunos e professores;
- Acesso ao desempenho escolar (notas) dos alunos nas disciplinas, antes e durante as atividades práticas com Robótica Educacional.

Todas as informações fornecidas, obtidas e utilizadas na análise desta pesquisa serão tratadas de forma sigilosa. Os riscos envolvidos nesta pesquisa incluem os alunos se sentirem desconfortáveis em participar das atividades e em responder os questionários e testes. Caso isso ocorra, cada aluno poderá se ausentar, assim como parar de responder os questionário e teste, após receberá auxílio necessário. Os benefícios desta pesquisa envolvem a identificação a partir de um estudo baseado no método científico, de habilidades do Pensamento Computacional desenvolvidas com ajuda da Robótica Educacional e, por consequência, o entendimento dos impactos que ela proporciona na capacidade de resolução de problemas dos alunos da Educação Básica. Podendo auxiliar na identificação de elementos que motive e/ou reconduza práticas de ensino de programação, de tal modo a estimular nos alunos as habilidades do Pensamento Computacional, e assim contribuir com o ensino da Instituição. Os resultados serão publicados na tese de doutorado da pesquisadora responsável, periódicos e revistas científicas, mas sem identificar os alunos que participaram.

O contato com a pesquisadora pode ser realizado por e-mail isabellemaria@copin.ufcg.edu.br, por telefone pelo número (83) 996241326 ou no endereço Rua Rosa farias Dantas, 124B, Novo Cruzeiro, Campina Grande – PB. Caso a pesquisa não esteja sendo realizada da forma esperada ou que prejudique os sujeitos de sua Instituição de alguma forma, você pode entrar em contato com o Comitê de Ética em Pesquisa do Hospital Universitário Alcides Carneiro - CEP-HUAC pelo telefone (83) 2101 - 5545 entre segunda e sexta-feira das 07h00 às 17h00 ou pelo e-mail cep@huac.ufcg.edu.br.

CONSENTIMENTO DE PARTICIPAÇÃO DO SUJEITO

Concordo em participar do presente estudo como sujeito. Fui devidamente informado(a) e esclarecido(a) sobre a pesquisa, os procedimentos nela envolvidos, assim como os possíveis riscos e benefícios decorrentes de minha participação. Foi-me garantido que posso retirar meu consentimento a qualquer momento, sem que isto leve a qualquer penalidade ou interrupção de meu acompanhamento.

Dados do participante da pesquisa	
Nome:	
RG:	CPF:
Telefone:	E-mail:

Campina Grande, _____ de _____________ de 2022.

____
Assinatura do participante da pesquisa

____
Assinatura da pesquisadora

Appendix D

Román-Gonzalez' CT Test

TESTE DE PENSAMENTO COMPUTACIONAL

Este teste faz parte do experimento que investigará formas propor e validar uma metodologia para o ensino de programação com Robótica Educacional no Ensino Médio. O objetivo deste questionário é conhecer o perfil dos alunos da 1ª série do Ensino Médio participantes. As respostas serão utilizadas na pesquisa de Doutorado em Ciência da Computação sob a responsabilidade da pesquisadora Isabelle Maria Lima de Souza. Este teste foi desenvolvido pelo Prof. Dr. Marcos Roman Gonzales da Universidad Nacional de Educación a Distancia (UNED) e traduzido/adaptado pelos pesquisadores Rafael Marimon Boucinha e Christian Puhlmann Brackmann, tendo seu uso nesta pesquisa devidamente autorizado pelos mesmos.

INSTRUÇÕES

O teste é composto por 28 perguntas, distribuídas em 12 páginas com aproximadamente 3 perguntas em cada uma. Todas as perguntas têm 4 alternativas de resposta (A, B, C e D) das quais só uma é correta. A partir do início do teste, você dispõe de até 45 minutos para fazer o melhor que puder. Não é imprescindível que você responda a todas as perguntas. Antes de começar o teste, vamos ver 3 exemplos para que lhe familiarize com o tipo de perguntas que vai encontrar, nas quais aparecerão os personagens que lhe apresentamos.


'Pac-Man'

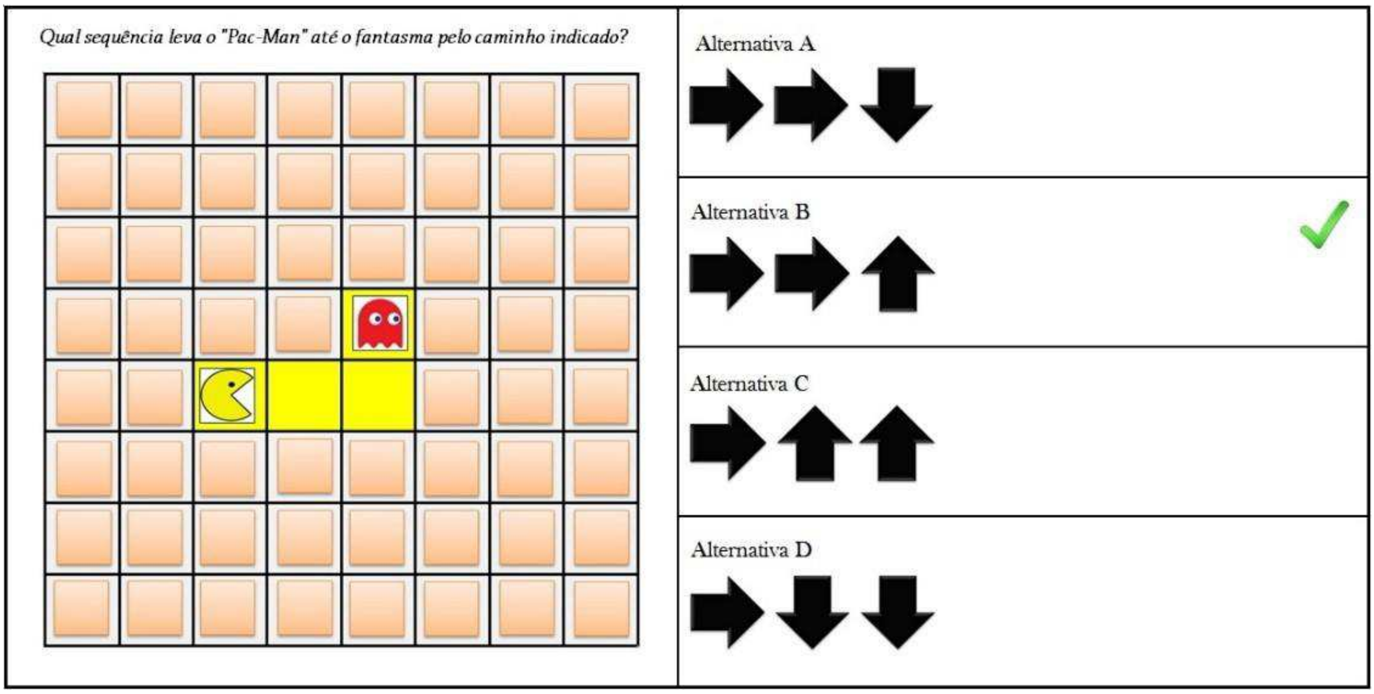

Fantasma


Artista

EXEMPLO I

Neste primeiro exemplo se pergunta quais são os comandos que levam o 'Pac-Man' até o fantasma pelo caminho indicado. Ou seja, levar 'Pac-Man' exatamente à caixa em que o fantasma está (sem passar, nem parar), e seguindo estritamente o caminho marcado em amarelo (sem sair e sem tocar nas paredes, representadas pelos quadrados laranja).

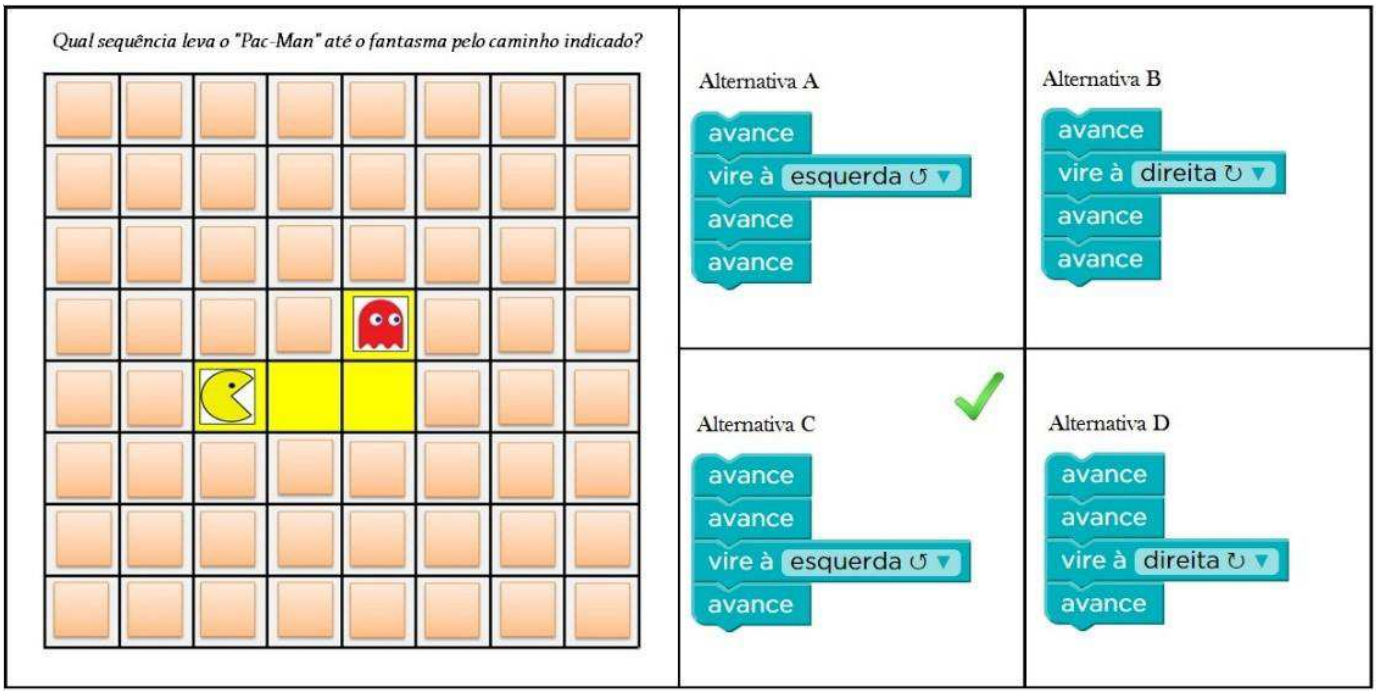
A alternativa correta neste exemplo é a B. Marque a alternativa correspondente, na folha de respostas.



Exemplo I	☐ A	☒ B	C ☐	☐ D

EXEMPLO II

Neste segundo exemplo, se pergunta de novo quais são os comandos que levam o 'Pac-Man' até o fantasma pelo caminho assinalado. Mas neste caso, as opções de resposta, em vez de ser flechas, são blocos que encaixam uns nos outros. Lembramos que a pergunta pede para levar o 'Pac-Man' EXATAMENTE a casa em que se encontra o fantasma (sem passar nem parar), e seguindo estritamente o caminho marcado em amarelo (sem sair e sem tocar nas paredes, representadas pelos quadrados laranja). A alternativa correta neste exemplo é a C. Marque a alternativa correspondente, na folha de respostas.

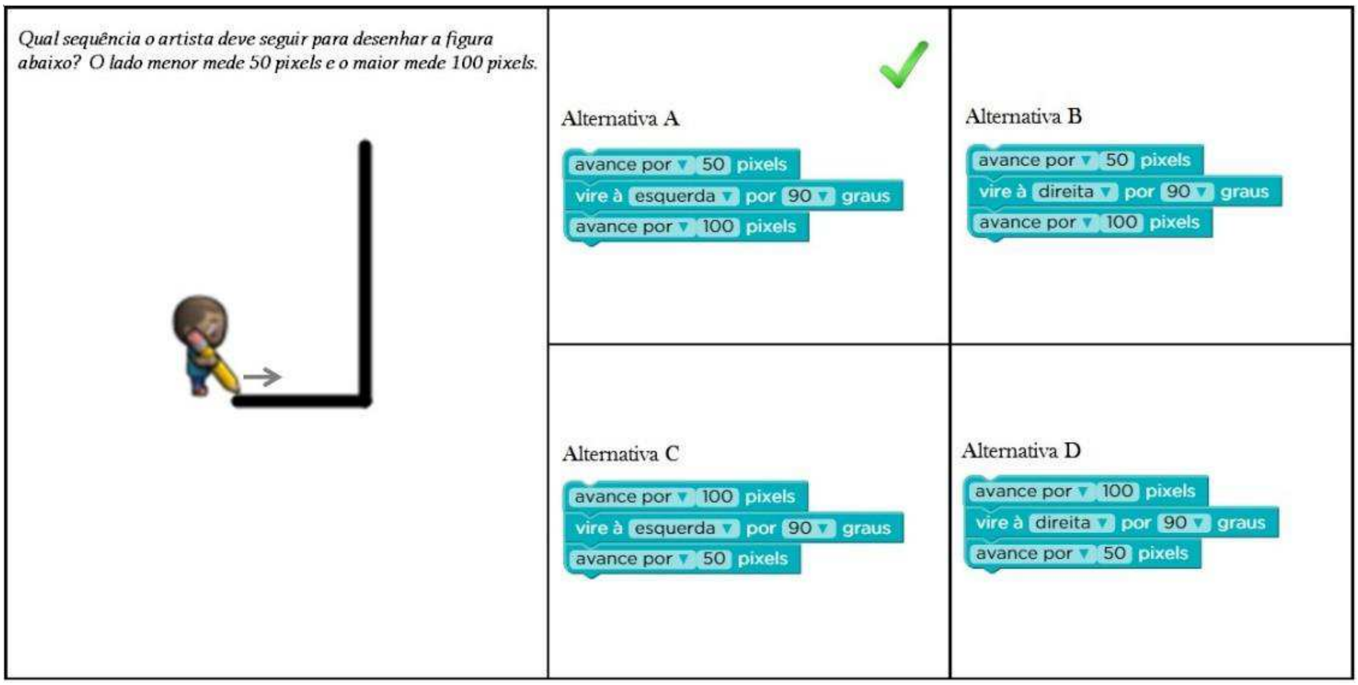


Exemplo II	☐ A	☐ B	☒ C	☐ D

EXEMPLO III

Neste terceiro exemplo se pergunta que comandos deve seguir o artista para desenhar a figura que aparece na tela. Ou seja, como deve MOVER o lápis para que se desenhe a figura. O comando MOVER empurra o lápis desenhando, enquanto que o comando SALTAR faz um alto ao artista sem desenhar. A seta cinza indica a direção do primeiro movimento da caneta.

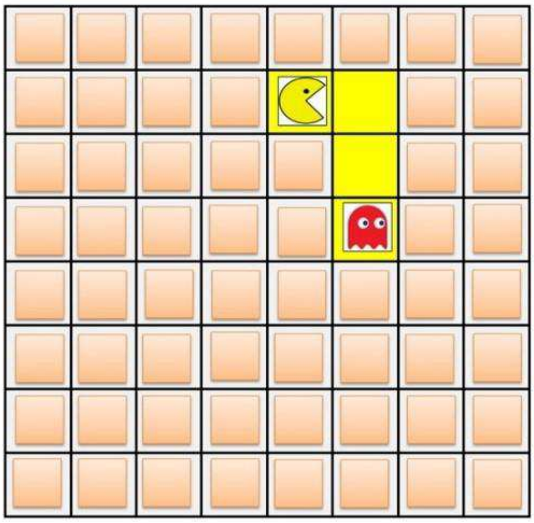
A alternativa correta neste exemplo é A. Marque a alternativa correspondente, na folha de respostas.



Exemplo III	☒ A	☐ B	☐ C	☐ D

QUESTÃO 1

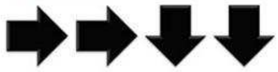
Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?



Alternativa A

Alternativa B

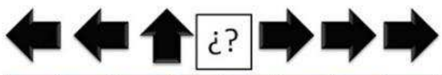
Alternativa C

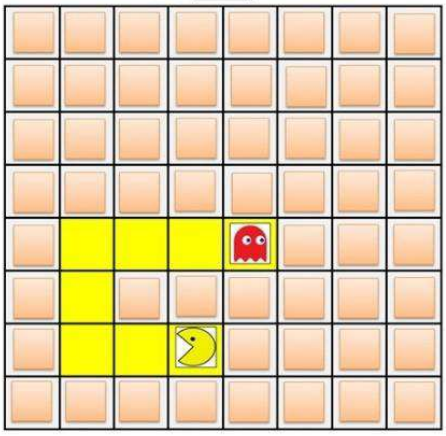


Alternativa D

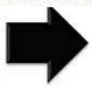
QUESTÃO 2

Qual comando está faltando na sequência para levar o "Pac-Man" até o fantasma pelo caminho indicado?





Alternativa A



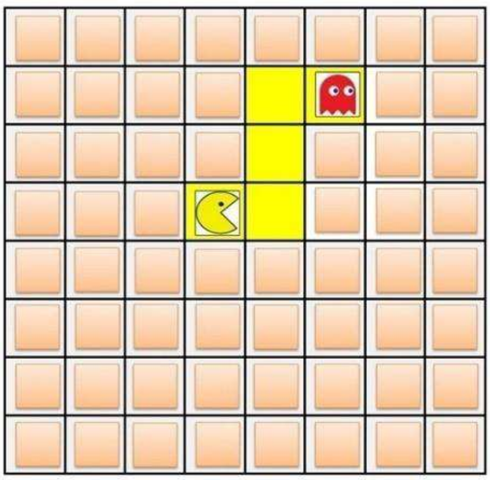
Alternativa B

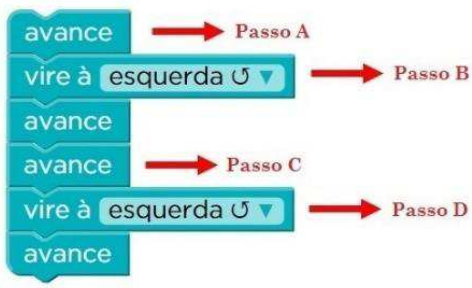
Alternativa C

Alternativa D

QUESTÃO 3

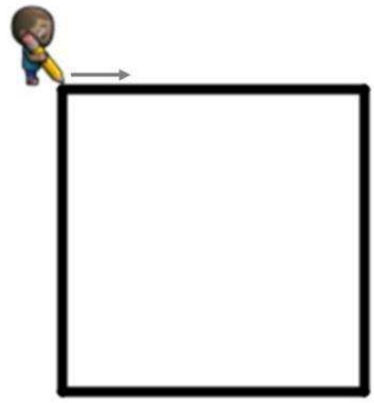
Para levar o "Pac-Man" até o fantasma pelo caminho indicado, qual passo da sequência está incorreto?





QUESTÃO 4

Qual sequência o artista deve seguir para desenhar o quadrado abaixo? Cada um dos lados mede 100 pixels.



Alternativa A

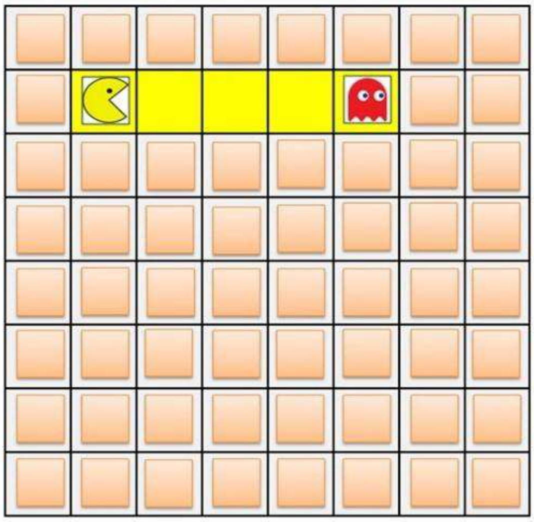
Alternativa B

Alternativa C

Alternativa D

QUESTÃO 5

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?



Alternativa A

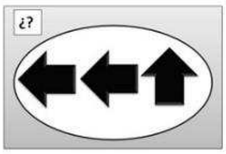
Alternativa B

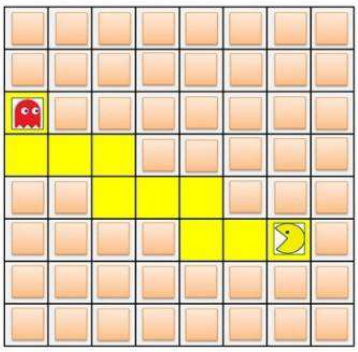
Alternativa C

Alternativa D

QUESTÃO 6

Quantas vezes a sequência abaixo deve ser repetida para levar o "Pac-Man" até o fantasma pelo caminho indicado?





Alternativa A

× 2

Alternativa B

× 1

Alternativa C

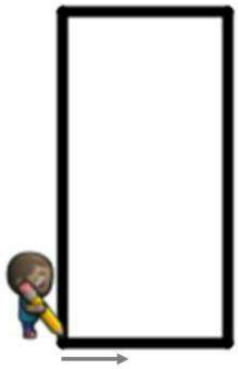
× 4

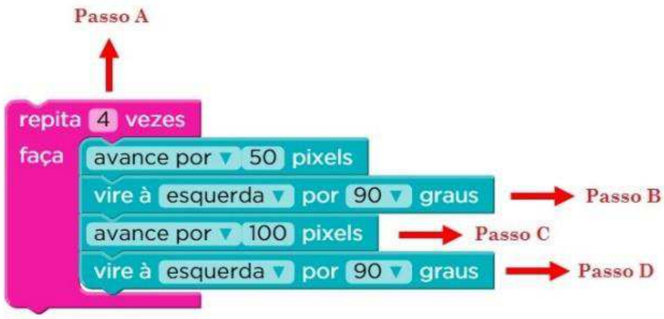
Alternativa D

× 3

QUESTÃO 7

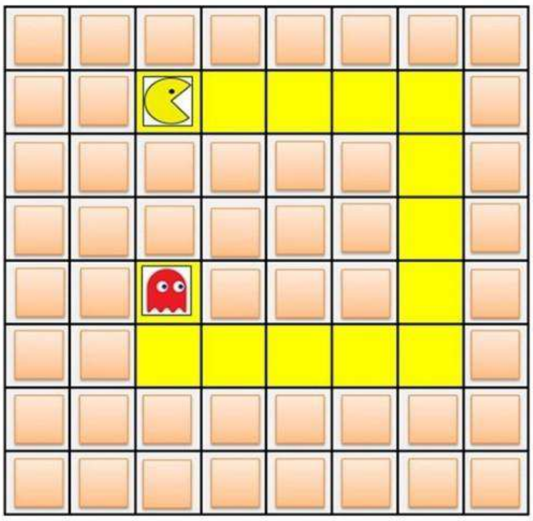
Para que o artista desenha uma vez o seguinte retângulo (50 pixels de largura e 100 pixels de altura), qual passo da sequência está incorreto?





QUESTÃO 8

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?



Alternativa A

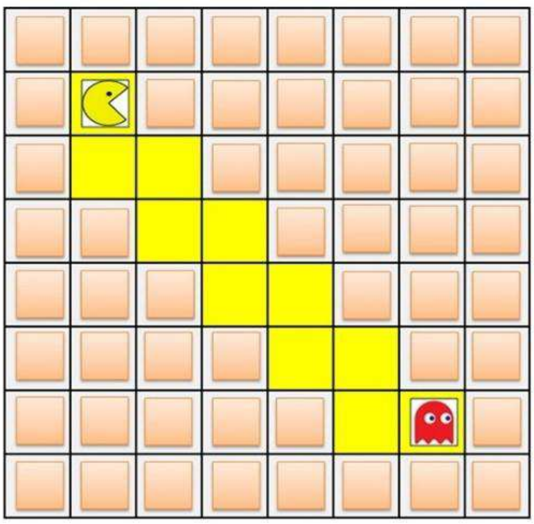
Alternativa B

Alternativa C

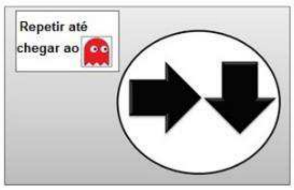
Alternativa D

QUESTÃO 9

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?

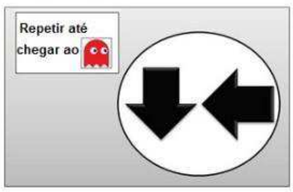


Alternativa A

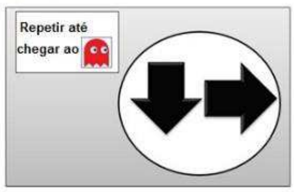


Alternativa B

Alternativa C

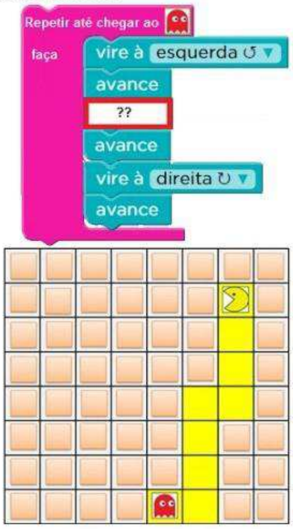


Alternativa D



QUESTÃO 10

Qual bloco está faltando na sequência abaixo para levar o "Pac-Man" até o fantasma pelo caminho indicado?



Alternativa A

Alternativa B

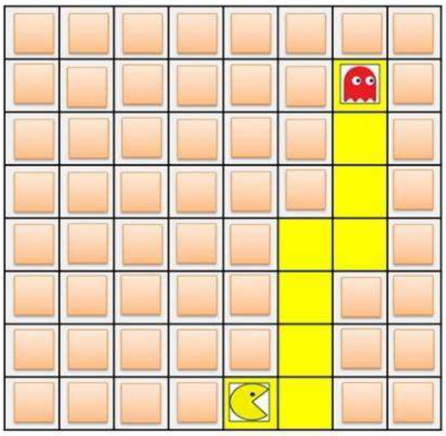
Alternativa C

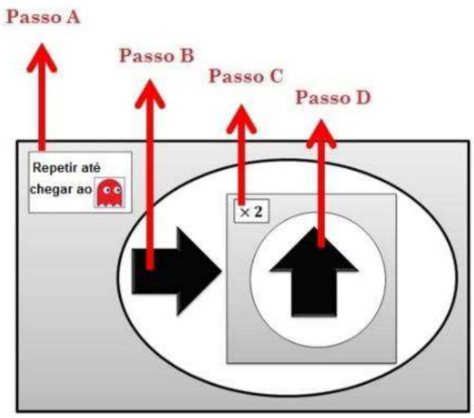
Alternativa D

Não falta nenhum bloco

QUESTÃO 11

Para que o "Pac-Man" chegue até o fantasma pelo caminho indicado, qual passo da sequência está incorreto?





QUESTÃO 12

Qual sequência o artista deve seguir para desenhar a escada que leva até a flor? Cada degrau sobe 30 pixels.

Alternativa A

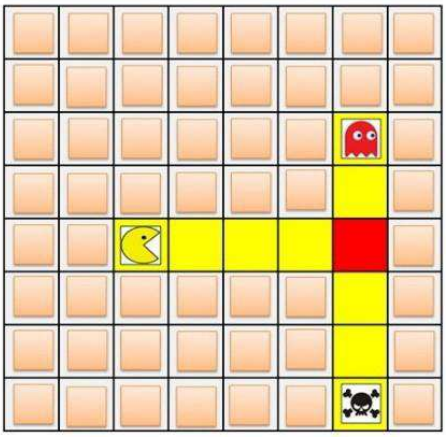
Alternativa B

Alternativa C

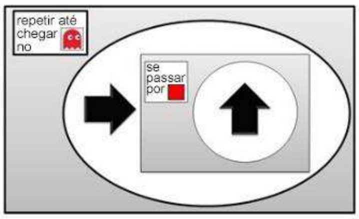
Alternativa D

QUESTÃO 13

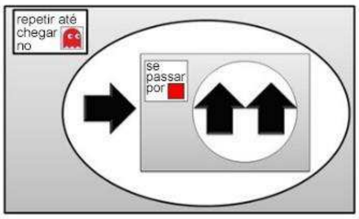
Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?



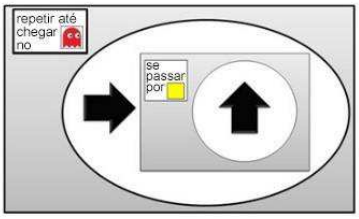
Alternativa A



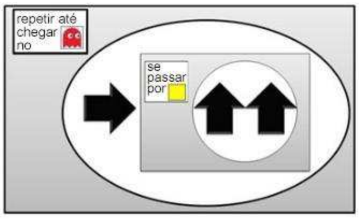
Alternativa B



Alternativa C

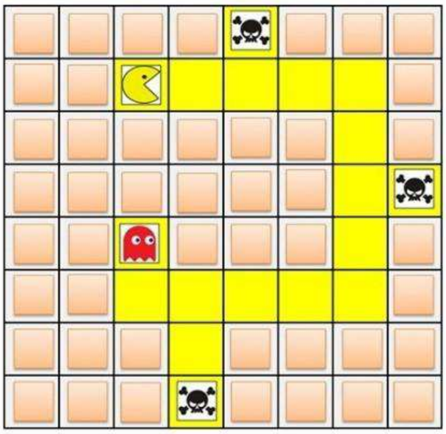


Alternativa D



QUESTÃO 14

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?



Alternativa A

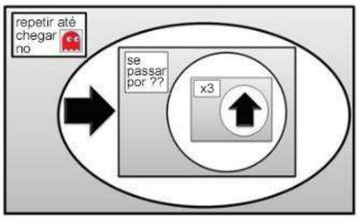
Alternativa B

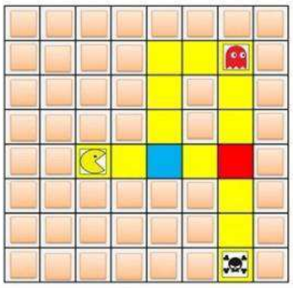
Alternativa C

Alternativa D

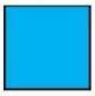
QUESTÃO 15

O que falta na seguinte sequência para levar o "Pac- Man" até o fantasma pelo caminho indicado?

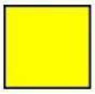




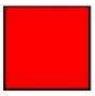
Alternativa A



Alternativa B



Alternativa C

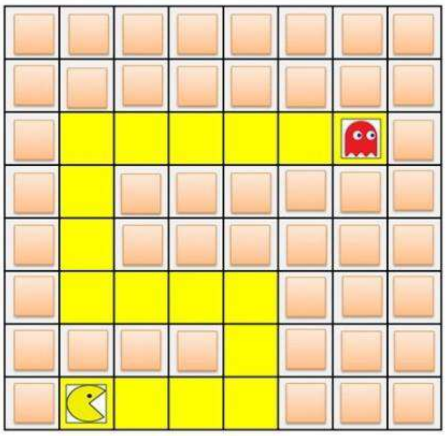


Alternativa D

Tanto a alternativa A como a alternativa C estão corretas

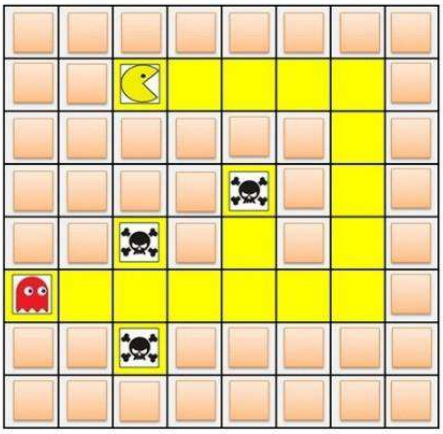
QUESTÃO 16

Para que o "Pac-Man" chegue até o fantasma pelo caminho indicado, qual passo da sequência está incorreto?



QUESTÃO 17

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?



Alternativa A

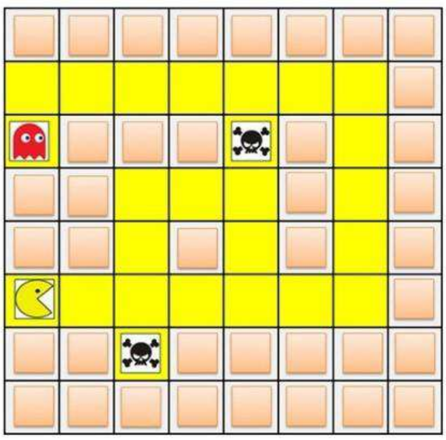
Alternativa B

Alternativa C

Alternativa D

QUESTÃO 18

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?



Alternativa A

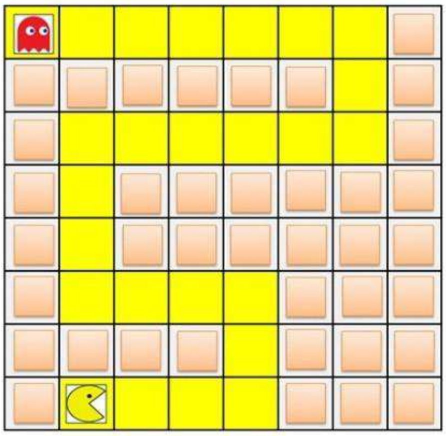
Alternativa B

Alternativa C

Alternativa D

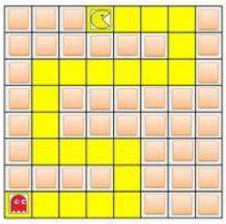
QUESTÃO 19

Para que o "Pac-Man" chegue até o fantasma pelo caminho indicado, qual passo da sequência está incorreto?



QUESTÃO 20

Qual bloco está faltando na sequência abaixo para levar o "Pac-Man" até o fantasma pelo caminho indicado?



Alternativa A

Alternativa B

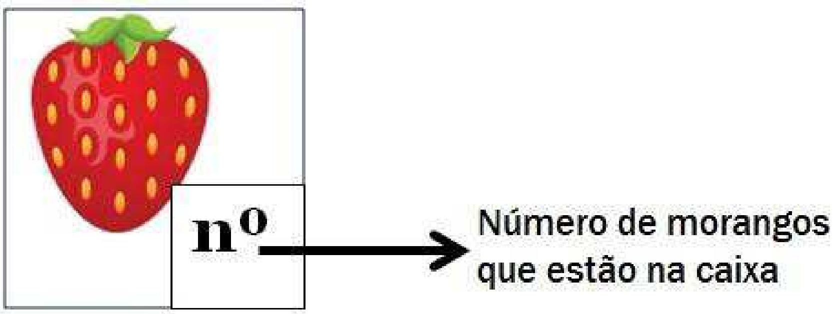
Alternativa C

Alternativa D

Não falta nenhum bloco

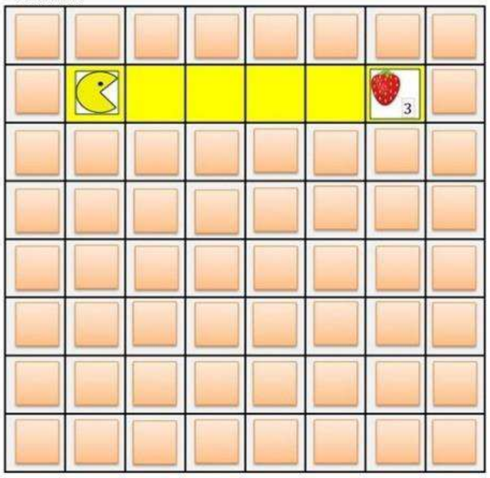
IMPORTANTE: LEIA COM ATENÇÃO

A IMAGEM ABAIXO IRÁ APARECER NAS PRÓXIMAS QUESTÕES. O NÚMERO QUE ESTÁ NO CANTO INFERIOR DIREITO INDICA QUANTOS MORA NGOS EXISTEM NA CAIXA.



QUESTÃO 21

Qual sequência leva o "Pac-Man" pelo caminho indicado até os morangos e faz o "Pac-Man" comer o número de morangos indicado?



Alternativa A

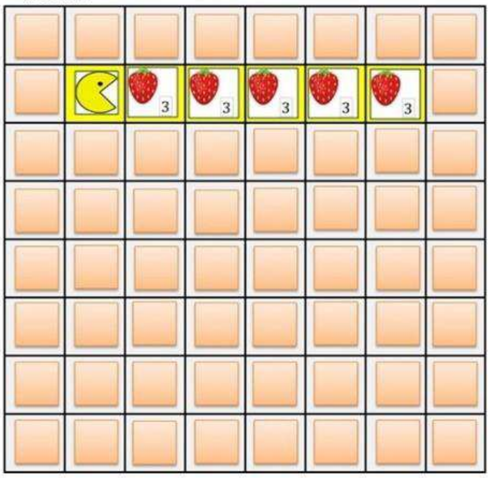
Alternativa B

Alternativa C

Alternativa D

QUESTÃO 22

Qual sequência leva o "Pac-Man" pelo caminho indicado até os morangos e faz o "Pac-Man" comer o número de morangos indicado?



Alternativa A

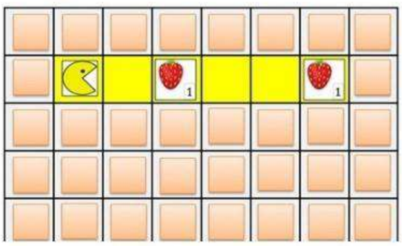
Alternativa B

Alternativa C

Alternativa D

QUESTÃO 23

O que falta na seguinte sequência para que "Pac-Man" avance pelo caminho assinalado comendo o número de morangos indicados?



Alternativa A

$1\ vez$

Alternativa B

$2\ vezes$

Alternativa C

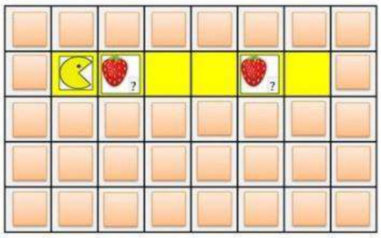
$3\ vezes$

Alternativa D

$5\ vezes$

QUESTÃO 24

O que falta na seguinte sequência para que "Pac-Man" avance pelo caminho assinalado comendo o número de morangos indicados?



Alternativa A

enquanto houver caminho em frente

Alternativa B

enquanto não houver caminho em frente

Alternativa C

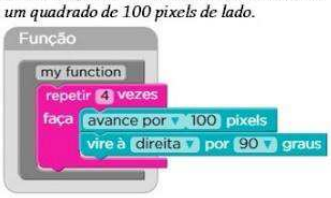
enquanto houver morangos

Alternativa D

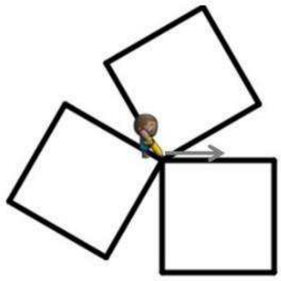
enquanto não houver morangos

QUESTÃO 25

Se temos o seguinte conjunto de instruções, que chamos de "my function", e que desenha um quadrado de 100 pixels de lado.



Qual sequência o artista deve seguir para desenhar a seguinte figura? Cada um dos lados mede 100 pixels.



Alternativa A

Alternativa B

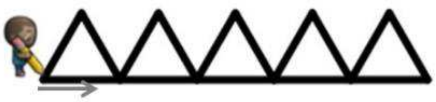
Alternativa C

Alternativa D

QUESTÃO 26

Se temos o seguinte conjunto de instruções, que chamamos de "my function", e que desenha um triângulo de 50 pixels de lado:

O que falta na seguinte sequência para que o artista desenhe a seguinte figura? Cada um dos lados de cada triângulo mede 50 pixels.



Alternativa A

15

Alternativa B

5

Alternativa C

4

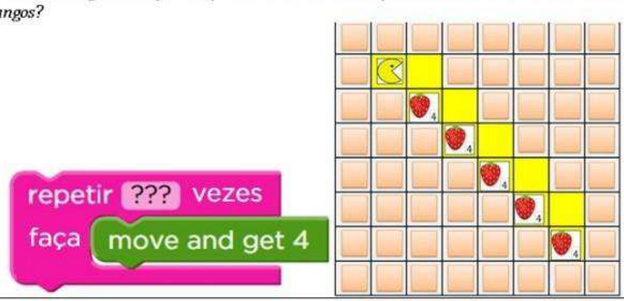
Alternativa D

3

QUESTÃO 27

Se temos a seguinte sequência de instruções que chamamos de "move and get 4":

O que falta na seguinte sequência para levar o "Pac-Man" pelo caminho indicado comendo todos os morangos?



Alternativa A

3

Alternativa B

4

Alternativa C

5

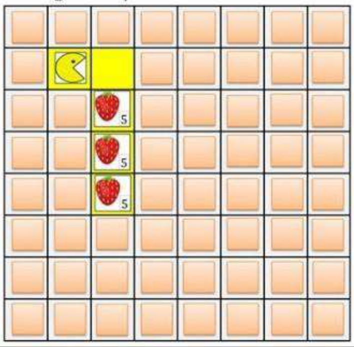
Alternativa D

6

QUESTÃO 28

Se temos o seguinte conjunto de instruções, que chamamos de "get 5":

Qual sequência leva o "Pac-Man" pelo caminho indicado e faz com ele coma o número de morangos correspondentes?



Alternativa A

Alternativa B

Alternativa C

Alternativa D

Appendix E

Román-Gonzalez' CT Test Answer Sheet

UFCG Universidade Federal de Campina Grande

PÓS-TESTE DE PENSAMENTO COMPUTACIONAL

FOLHA DE RESPOSTA

Esta folha de resposta faz parte do experimento que investigará formas propor e validar uma metodologia para o ensino de programação com Robótica Educacional no Ensino Médio. O objetivo deste questionário é conhecer o perfil dos alunos da 1ª série do Ensino Médio participantes. As respostas serão utilizadas na pesquisa de Doutorado em Ciência da Computação sob a responsabilidade da pesquisadora Isabelle Maria Lima de Souza.

DADOS PESSOAIS

Nome Completo:	
Turma:	

RESPOSTAS

Exemplo I	A ☐	B ☒	C ☐	D ☐
Exemplo II	A ☐	B ☐	C ☒	D ☐
Exemplo III	A ☒	B ☐	C ☐	D ☐
Questão 1	A ☐	B ☐	C ☐	D ☐
Questão 2	A ☐	B ☐	C ☐	D ☐
Questão 3	A ☐	B ☐	C ☐	D ☐
Questão 4	A ☐	B ☐	C ☐	D ☐
Questão 5	A ☐	B ☐	C ☐	D ☐
Questão 6	A ☐	B ☐	C ☐	D ☐
Questão 7	A ☐	B ☐	C ☐	D ☐
Questão 8	A ☐	B ☐	C ☐	D ☐
Questão 9	A ☐	B ☐	C ☐	D ☐
Questão 10	A ☐	B ☐	C ☐	D ☐
Questão 11	A ☐	B ☐	C ☐	D ☐
Questão 12	A ☐	B ☐	C ☐	D ☐
Questão 13	A ☐	B ☐	C ☐	D ☐
Questão 14	A ☐	B ☐	C ☐	D ☐
Questão 15	A ☐	B ☐	C ☐	D ☐
Questão 16	A ☐	B ☐	C ☐	D ☐
Questão 17	A ☐	B ☐	C ☐	D ☐
Questão 18	A ☐	B ☐	C ☐	D ☐
Questão 19	A ☐	B ☐	C ☐	D ☐
Questão 20	A ☐	B ☐	C ☐	D ☐
Questão 21	A ☐	B ☐	C ☐	D ☐
Questão 22	A ☐	B ☐	C ☐	D ☐
Questão 23	A ☐	B ☐	C ☐	D ☐
Questão 24	A ☐	B ☐	C ☐	D ☐
Questão 25	A ☐	B ☐	C ☐	D ☐
Questão 26	A ☐	B ☐	C ☐	D ☐
Questão 27	A ☐	B ☐	C ☐	D ☐
Questão 28	A ☐	B ☐	C ☐	D ☐