Creative Learning in Problem Solving and Development of Computational Thinking

Tatyane S. C. da Silva[1(✉)], Jeane C. B. de Melo[2], and Patricia C. A. R. Tedesco[1]

[1] Centro de Informática, Universidade Federal de Pernambuco, Av. Jorn. Aníbal Fernandes, Cidade Universitária, Recife, Pernambuco, Brazil
{tscs,pcart}@cin.ufpe.br

[2] Departamento de Computação, Universidade Federal Rural de Pernambuco, Rua Dom Manuel de Medeiros, s/n, Dois Irmãos, Brazil
jeane.bmelo@ufrpe.br

Abstract. Society is increasingly facing complex problems and creativity is one of the skills that help in solving these problems in addition to being one of the skills of professionals of the future. Additionally, Computational Thinking involves solving problems using models, abstractions, organization, and decomposition of these elements in an algorithmic way and thus can contribute to the development of an individual's ability to be creative. Therefore, this study aims at understanding the relationship between creative learning in problem-solving and the development of Computational Thinking, to assist the teaching and learning of programming. To this end, a Conceptual Model was elaborated, relating the pillars of Computational Thinking to the problem solving process and later applied in a Computational Thinking subject carried out in a Higher Education Computing course This model is an adaptation of its previous version. It encompasses both the pillars of Computational Thinking and CPS (Creative Problem Solving), as well as techniques of creativity to assist in Creative Thinking. The results of the research show that there is relevance in the insertion of creativity in the problem-solving process through Computational Thinking. The quasi-experiment well received by the teacher. The students reported that the activity helped in solving problems because through it it is possible to have a model to follow with similar problems.

Keywords: Higher education · Problem solving · Creative learning · Computational thinking · Programming education

1 Introduction

The Programming discipline is part of the basic training in Computer Science courses. Its content is focused on teaching concepts, computational models and programming language [2].

This work was carried out with support from the National Council for Scientific and Technological Development (CNPq).

H. C. Lane et al. (Eds.): CSEDU 2020, CCIS 1473, pp. 199–215, 2021.
https://doi.org/10.1007/978-3-030-86439-2_11

The literature presents some challenges faced by teachers in the programming teaching process, among which we can highlight: to present problem-solving techniques and to work on the student's abstraction capacity, in search of possible solutions [2]. In addition to overcoming these challenges, it is necessary to provide an environment that stimulates creativity, thus being able to favor the student's autonomous learning in solving problems and working with the capacity for abstraction.

These difficulties inevitably cause high failure or dropout rates for students in the Computing course. When analyzing the Computing area, it is observed that the programming discipline is pointed out as responsible, or contributes effectively, to evasion and failure in the first periods of the course [2]. The high rate of student dropout corroborates the low demand for careers related to Exact Sciences.

The traditional teaching methodology, commonly used in programming classes, which are usually divided into theoretical, theoretical-practical and/or laboratory classes, has not been satisfactory [2]. These resources are relevant for presenting the results of a process, but they do not show the development process in itself.

Another factor to be considered in the programming teaching-learning process is Creative Thinking. According to Young [9], we live in a knowledge society, characterized by changes that require innovative individuals. At the same time, the the importance of Computational Thinking [5] stands out, since it is included in the list of Skills and Competencies required for professionals in the 21st century. According to National Research, Computational Thinking encompasses problem-solving using models, abstractions, grouping, and decomposition of these elements in an algorithmic manner. Although cognitive processes are commonly used by computer science professionals, formal training in this area of knowledge is not necessarily needed since in many of these activities there will be, at least, the use of information technology related to computational (algorithmic) reasoning [5].

Although cognitive processes are commonly used by computer science professionals, formal training in this area of knowledge is not necessarily needed since in many of these activities there will be, at least, the use of information technology related to computational (algorithmic) reasoning [5].

In turn, [3] points to the fundamental role of the development of creativity in students. Traditionally used approaches do not always favor creative solutions; thus, changes in educational practices, training programs, and creativity stimulation are necessary, to develop an engaging and innovative educational environment, therefore favorable to the development of Computational Thinking and programming learning.

In this perspective, Computational and Creative Thinking is seen as cognitive tools that expand the knowledge and skills that can be applied in obtaining a solution to a given problem. That is computational tools when used creatively, lead to the development of new approaches to both old and new problems, observing different stimuli and perspectives that may be relevant in their solution [1].

Aiming to propose an approach to the problem of applying programming concepts to solve real-world problems using the elements of Computational Thinking, this paper presents a research question: QP1 - Is creative learning a factor that influences the development of computational thinking?

From the QP1 inquiry, it is possible to analyze if creative learning assists students a problem-solving. To answer the research question the following hypothesis, H1 was formulated: H1: Creative learning helps in the development of computational thinking and problem-solving.

The phases involved in the development of such research are described in the present work, which is organized as follows: Sect. 2 presents the theoretical reference. Section 3 presents Related Works, Sect. 4 presents the Conceptual Model. Finally, Sect. 5 regards the final considerations of the paper, highlighting the contributions of the study.

2 Theoretical Background

The definitions that support this work are presented in this section. The domains covered include Computational Thinking, Programming Teaching, and Creativity.

2.1 Computational Thinking

The concept of Computational Thinking (CT) was proposed in 2006 by Jeannette Wing [10] and is related to problem solving and the perception of human behavior, both guided by definitions of the fundamentals of Computer Science [5]. The CT addresses a set of definitions, skills, and practices of computing that can be applied both in everyday activities and in other areas of knowledge [5,8].

According to Barr and Stephenson [14] Computational Thinking has 9 concepts that define it: (i) Data collection: process of collecting data or information about a problem; (ii) Data analysis: finding patterns resulting in conclusions; (iii) Data representation: represent and organize data in graphs, tables, text or figures; (iv) Decomposition of problems: dividing a complex problem into smaller and manageable tasks; v) Abstraction: reduce complexity to define main ideas; (vi) Algorithms and Procedures: a sequence of steps to solve a problem or reach an end; (vii) Automation: using computers to perform repetitive tasks; (viii) Parallelization: organizing resources to simultaneously perform tasks to achieve a common goal; (ix) Simulation: representing or creating a model of a process.

According to the BBC - Computational Thinking [3], Computational Thinking has four pillars that help solve complex problems: Decomposition, Recognition, Abstraction, and Algorithms.

- Decomposition - consists of breaking down a problem or complex system into smaller, more manageable parts.
- Pattern Recognition - characterized by looking for similarities between problems and subproblems.

– Abstraction - has the purpose of focusing only on important information searching for the solution, ignoring irrelevant details.
– Algorithms - intended to develop a systematic solution to the problem, or the rules to follow to solve it.

The use of the four pillars assists in programming and solving complex problems which are those that, at first sight, one does not know how to solve easily. Finally, these simple steps or rules are used in programming to help solve the problem in the best way [3].

For Proctor and Blikstein [15], Computational Thinking uses the computer as a way to implement thinking, distinguishing it from a tool, and considering Computer Science as an element to develop students' problem-solving skills. In this perspective, the PC aims to explain thinking in the design of computing, using, for example, the "divide to conquer" approach to decompose the complex task into some simple tasks; employing the idea of recursion to translate code into data; applying division thinking to abstract the real problem in a few steps to solving it; using iteration to perform the same or similar processes.

In this way, when talking about Computing, as a science, one must take into account its fundamental principles, such as abstraction and logical reasoning, which can be applied in the solution of problems and the development of knowledge [15]. The study of computer programming is a way to understand the central ideas related to Computational Thinking since it can assist in the development of skills such as logical reasoning, problem-solving and algorithmic thinking.

In this research the four pillars of Computational Thinking [3] will be used, corroborating with the objectives of the current proposal.

2.2 Programmıng Teachıng and Learnıng

The literature presents a set of difficulties associated with programming learning and teaching [7]. Considering the difficulties presented by the students, these were divided into three categories: teaching strategies, student attitudes, study methods, and natural programming difficulties [7]. Many students are accustomed to the memorization strategies (read, see solved exercises), which are not enough to learn to program. It was necessary to engage in intensive problem solving practice, facing the difficulties related to it and trying to resolve them. This should be based on generic problem-solving skills previously acquired that students generally do not have [7].

Another issue to be observed in programming learning is the increase in students' lack of interest, which can be related to teaching methods, which are sometimes still based on expository classes, as described [16]. The difficulties in learning programming are related to the use of means that can motivate and engage students in the process, as well as to the adjustment of different emotions that are associated with the learning process, such as frustration.

For [17], the study on teaching and learning programming presents evidence that:

- A methodology for teaching programming concepts based solely on syntax is negative for the learning process of beginning students.
- Students are discouraged due to difficulties with introductory programming materials, as they are unable to create the correct mental models about the knowledge being learned.
- There is a need for new proposals for instructional methodologies, which are not based purely on syntax.

Considering this scenario, some creative strategies can be used in programming teaching, among which we can cite: diversifying the proposed tasks through methods of education, transformation, simulation, among others; usage of Computational Thinking to solve problems; creating a space for the dissemination of student work; sharing personal experiences related to the studied topic; guiding the student to seek additional information on topics of interest to them [1].

2.3 Creativity

There are several definitions of the term creativity, and there is no consensus as to its exact meaning. Among the authors, this being an issue addressed from multiple points of view, we have, for example, Sternberg's Theory of Creativity [4] and MacKinnon's Theory [8]. This is related to the fact that creativity, like intelligence, is a complex construct, with diverse aspects, such as the characteristics of the individual, the creative process, items present in the creative product, or aspects of the environment where people are inserted, which may influence one's creative expression [1].

According to Sternberg's Theory of Investment in Creativity [4], six interrelated elements will be creative: intellectual skills, knowledge, thinking styles, personality, motivation, and appropriate environment. These multiple perspectives of creativity are related to combinations of aspects inherent to the individual, depending on cognitive, emotional and environmental factors.

For MacKinnon [8,18], three basic conditions are necessary for creativity: the response must be new or at least statistically infrequent; the response must adapt to reality and serve to solve a problem or achieve a recognizable goal and must include the evaluation, design, and development of the original insight.

For this paper, we will use the definition of MacKinnon [8,18], since it aligns with the research objective of using creativity as a tool to help solve problems.

2.4 Creative Learning

Creative learning can be defined as a personal transformation based on obtaining new skills and knowledge, which occur through direct engagement in carrying out projects that are meaningful to the individual, making them creative and capable of developing products in any context.

Creative learning, according to [3], proposes that the student uses processes of personalization of information, comparison with data based on interpretation, and own points of view and generation of new ideas that go beyond what was initially stated. In this way, the student becomes the subject of their process to learn by incorporating and producing knowledge in a personalized, active, and creative way

Creative learning has three striking characteristics, namely: (1) the personalization of information, referring to how the information is integrated with the subjectivity of the learner; (2) the confrontation with the data, related to the questioning, to the non-acceptance of the data as a single truth, which allows the learner to identify incongruities, gaps, and contradictions; and, (3) the production of own and "new" ideas, related to the transcendence of the data, to go beyond what is set and build new relationships.

Studies on creative learning have highlighted the low frequency of this type of learning in the institutions of our educational system, which by the way it was historically constituted, was characterized and characterized by a homogenizing education, oriented towards the assimilative-reproductive conception of knowledge [3].

Despite little encouragement in educational institutions, creative learning should be prioritized for at least two reasons: (1) the stability of what has been learned and its possibilities of transfer to new contexts; (2) it is potential as a unit of development of the condition of the subject in the process of learning or in any activity that the apprentice develops. In this perspective, the next subsection presents strategies that corroborate to promote creativity in students [3].

2.5 Creative Problem Solving (CPS)

Creative Problem Solving (CPS) was created by Osborn [6]. It is a methodological paradigm composed of methods and techniques to analyze, identify and solve problems.

Research, Discovery of Ideas and Discovery of Solutions. This model's strategy is to obtain a clear and precise definition of the problem and generate several solutions.

The problem is delimited at the Investigation stage. According to Osborn, the definition of the problem is fundamental to propose new questions and possibilities.

The generation and development of ideas happen in the Discovery of Ideas stage. The most promising ideas are then selected and developed in the Project activity.

The Solution Discovery phase encompasses the evaluation of the provisional ideas, the choice of the final solution and its subsequent implementation. The evaluation of ideas highlights critical intelligence, analytical thinking, and convergent thinking.

These phases drive the Creative Thinking process. The CPS method structures the entire process, in sequential steps, so that people understand what to do, step by step, to produce one or more creative and meaningful solutions. In this process, divergent thinking and creativity techniques are used to generate many ideas and opportunities that, depending on the phase, can be freely given or information, problem definitions, ideas, or implementation strategies, allowing the thought to flow. In this phase of thought wandering, judgment is suspended. Convergent thinking is also used when evaluating and making choices between the various ideas, or possibilities originated, in the divergent phase. In the convergent phase, data and information choices are made most important, the most promising ideas and the most appropriate strategies [6].

3 Related Work

Some research considers that it is important for students to first develop computational thinking skills and then have contact with coding [20]. In this perspective, one can learn how to solve problems computationally through game-based learning [20]. One approach that can be used to learn computational thinking is the unplugged method, through which students develop thinking skills without using technology.

The use of Creativity to promote Computational Thinking in programming learning is present in the scientific literature, however, when we consider these elements in teacher training, studies are still incipient [12].

Among the approaches to learning programming, Shell *et al.* [21] use Computational Creativity Exercises as a way to integrate Computational Thinking with Creative Thinking as a way to improve the performance of students in Computer Science courses. In this sense, Epstein's Generation Theory [21] is used as a basis for the definition of Creative Thinking, dividing it into four competencies: Capture, Challenge, Expand, and Engage.

Basically, the Capture competency refers to the ability to recognize and write down unique ideas as they occur. Challenging relates to the ability to generate new approaches to problems, inciting thinking to bypass established patterns of behavior. The competence to extend or expand seeks to take knowledge beyond the discipline, thus allowing the innovative integration of ideas. The last competence considers the stimulus, which can be social or environmental, as a way to obtain new experiences and ideas.

Additionally, the work of Shell *et al.* [21] establishes the principles of Computational Creativity Exercises (CCEs) that should promote (1) a balance of attributes between Computational and Creative Thinking and (2) mapping between computational and creative concepts and skills, as manifested in different disciplines. The exercises have a set of creative, computational, and collaborative goals considering problem-solving.

Promoting the integration of computational creativity exercises based on Epstein's creative skills [21], the study points to improvements in the learning of Computational Thinking in Computer Science courses, however, this does not clearly show how problem-solving is associated with pillars of Computational Thinking.

Studies that relate Computational Thinking to programming learning, recommend that students first develop computational thinking skills, and then have contact with coding [22]. In this perspective, one can learn how to solve problems computationally through Game-Based Learning [23]. One approach that can be used in this regard is the unplugged method, through which students develop computational thinking skills without using technology [24].

3.1 Computer Unplugged

Kuo and Hsu [20], developed research that aimed to identify the computational concepts and learning behaviors of students who participate in the unplugged activities of the computational thinking board game [20]. This research claims that the board game played a beneficial role in promoting students' logical thinking, achieving the development of computational thinking without the use of computers. This study does not clearly show which elements or pillars of computational thinking the study addressed. The next subsection presents a work that uses the pillars of computational thinking for problem-solving.

4 Conceptual Model

The Conceptual Model for this study is a modified version of the model by Silva *et al.* [18]. The Conceptual Model by Silva *et al.* [18] in addition to having the pillars of Computational Thinking and CPS (Creative Problem Solving) has creative techniques to assist in the development of Creative Thinking. The next subtopic presents the model by Silva *et al.* [18].

4.1 Conceptual Model Creative Process and Computational Thinking

Seeking to relate the Creative Process to Computational Thinking, Silva *et al.* [18] proposes a Conceptual Model (Fig. 1) and applies it to a class of games programming, taught by the author.

During the research, creativity techniques were used with the students of this class, applying them to solve problems that, besides, promote the development of computational thinking.

The conceptual model proposed by Silva et al. [18] was used in the present work, which has its guidelines that differ from the initial work. similar to the model in this study, but with some differences: creativity techniques were used in problem solving activity. The students used a C # programming language and the Unity engine. The teacher and researcher were the same person. Comparison of two classes: one used and one not used. The pillars of computing and their relationship with creativity techniques: The Decomposition phase of the pillars of Computational Thinking is related to the six-hat technique, as this technique helps to divide the problem and observe it from different perspectives, and can be used in the phase definition of the CPS problem. The Pattern Recognition stage of the Computational Thinking pillars is correlated with the Dominate to Destroy (D2D) technique. This technique aims to recognize the patterns to create something new or innovative and concerns the generation phase of the CPS, which is the stage selected for this function. The Abstraction phase of the

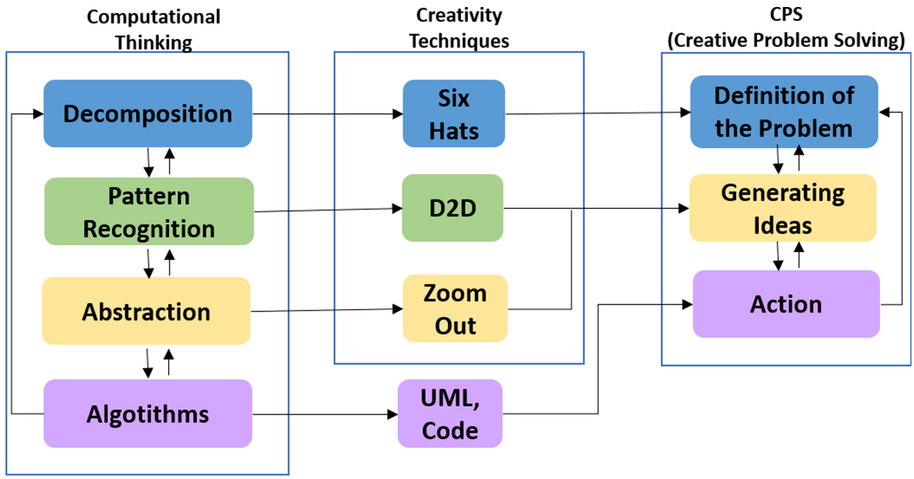

Fig. 1. Conceptual Model proposed in [20].

Computational Thinking pillars is related to the Zoom Out creativity technique, since this technique, like abstraction, aims to train in the individual the ability to observe concepts only in a generic way in the search for the most relevant ones. training. Besides, it is located at the stage of generating ideas, a time of convergence. Finally, in the Algorithm pillar, it is related to the code, UML, or any algorithmic representation of the solution and is located in the Action phase of the CPS model, as it is the solution development phase. In the following subsections, the application of the modified model will be detailed for the Computational Thinking class.

4.2 The Participants

The research has the participation of a group of students from the Computer Science course at a university. All students were from the first period and were taking the Computational Thinking course for the first time. This course predates the Introduction to Programming course. Twenty-nine students participated in the research. Before the activity, the students previously had a lesson on the concepts of Computational Thinking with the teacher. All students and the teacher signed the consent form to conduct the study.

4.3 The Study

The study was divided into a few phases and the activities were based on the Conceptual Model. The research is qualitative, carrying out an analysis of the semi-structured interviews with the students and teacher. The Conceptual Model was based on the pillars of Computational Thinking and Creative Problem Solving (CPS). The model can be seen in Fig. 2.

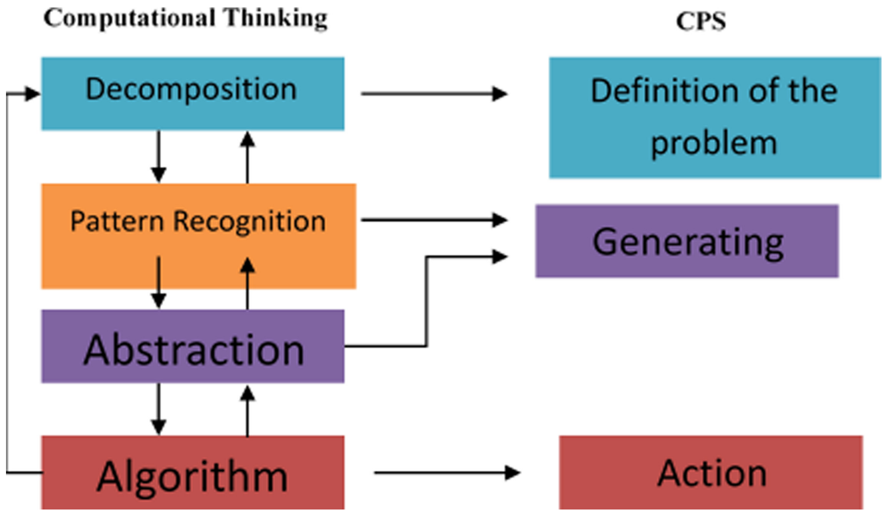

Fig. 2. Conceptual model.

The Conceptual Model is divided into two parts: Computational Thinking and CPS. They are intended to assist in solving problems to facilitate the learning of programming. The modification of the elements of the initial model by Silva *et al.* [18] was made because this study focused on creative learning, which was added implicitly to the proposed activity. The relationship between the phases of the computational thinking and CPS pillars are the same as the model by Silva *et al.* [18].

To apply the Conceptual Model, an activity was carried out in the classroom with all students in the class. The purpose of this activity was to study the impact of creative learning on the development of Computational Thinking to assist in problem-solving.

The intervention was applied by the researcher. The students were divided into five groups and for activity, the researcher created a game called PROGWISTER (prototype). The intervention was performed in a 2-h class. A game is similar to the twister with some changes to be used in the Computational Thinking class.

In PROGWISTER each group randomly takes 5 cards (see Fig. 3) and with them, they will create an algorithm so that the opposing team can follow the step by step instructions on the mat. The cards have the image of the foot (left or right) or hand (left or right) and what color the foot or hand should be on the mat.

Fig. 3. Carpet.

The goal is for the team to create an algorithm with the cards drawn in a way that makes it difficult for the other group to move around on the carpet (see Fig. 4). The mat has a number from 1 to 5. It is the team that will create the algorithm that decides which card is the first instruction until the fifth.

On the carpet are two people, one representative from each team. And for example team 1 uses team 2's algorithm and vice versa. If team 1 starts and the participant will follow the instructions of card 1 of team 2's algorithm and then the participant of team 2 will follow the instruction of card 1 of the algorithm created by team 1 and so on.

The winning condition of the game is to follow the algorithm using only the hands and feet. You cannot use elbow or knee supports and you can only take one hand or foot out of the way for the next step presented in the algorithm. The teams were competing with each other until they had the winning team.

After the activity the students and the teacher who was just watching the activity were interviewed. Following are the questions from the student interview:

- In your opinion, is the proposed activity related to Computational Thinking?
- Did you have difficulty in the subject of Computational Thinking?
- In your opinion, activities like these can help in solving problems?

Fig. 4. Cards.

Questions directed at the teacher can be seen below:

- Would you like to know about your opinion about the activity using the PROGWISTER game?
- Do you think that activities like PROGWISTER, which is unplugged, can help students to apply knowledge in problem-solving?
- Do you think that creativity can be applied in the classroom to any discipline like computational thinking or does it have a limit?

4.4 Results

The Analysis was carried out based on the transcripts of the semi-structured interviews.

When asked if students thought the activity was related to computational thinking. Most answered yes and justified their claim through the following arguments: Student 1: "The activity introduced commands to achieve a goal". Student 2: " The activity worked on abstract thinking. It is necessary to imagine the person's position on the carpet".Student 3: "Presented a problem to be solved using an algorithm". Student 4: "Helped in the development of Algorithmic thinking". Student 5: "Dividing the problem into smaller parts".

When asked about the difficulty in the discipline of Computational Thinking, some students reported that they had difficulty in some activities proposed by the discipline's teacher. Well, you need to use decomposition to solve the problem of similar flags.

The last question: In your opinion, activities like these can help in solving problems? Some answers to follow: Student 6: "It can help for those who are starting their studies in programming". Student 7: "It helps to train problem-solving, as you will have a model to follow with similar problems." Student 8: "Maybe if I made an association with something digital it would be more interesting". Student 9: "Activities like these help to change the idea that computational thinking can only be applied using the computer".

Additionally, when asked "Would you like to know about your opinion about the activity using the PROGWISTER game?", the teacher said:

> *So, they understood the instructions very well and this question of motivation, of playfulness together with the activity the impression it gives is that they will internalize here without realizing that they were working on this concept. So much so that it happened during the game there, now I won't remember exactly how the sentence was, but I remember them saying "about the algorithm.*

A normal person playing twister will not talk about an algorithm. And I realized that this call was happening and that it is something that will help them. So, I think that this contextualization with the game allows a fixation with the content that is indirect, implicit and sometimes this ends up being the best type of learning that you have. Why that learning that comes without you even realizing and stays. Why is it something that you used to do something else and not say: "now you will learn this here".

In the interview, the question about the teacher's opinion is whether the activity can help students apply their knowledge in solving problems. She answered:

> *I think so, certainly, my dream of consumption, that all classes could be like this. That's it, it's ideal for me, that they can apply. In the flag algorithm, which I told you that I wanted them to do, in any way, using the procedures to design certain types of flags that were similar, I said explicitly, that I wanted them to use decomposition, that this it was the focus of the exercise, even though several didn't use it and I kept thinking like that, they didn't see, that it was necessary to use decomposition, that it was useful for something, I think they didn't understand, the purpose, what I will use this decomposition if I can solve it this way here. Well, I forced them to use a concept, only apparently, it was not the best way to do that. I should create a situation where that concept was really needed, so that they would seek to solve a problem using this concept here and not say, look I have this problem and I want you to use this concept here to solve it. So, thinking about it, I killed the student's creativity there. And when I was correcting it, I realized that I hadn't thought about how to solve it, this way is correct. I can't tell the student that it doesn't work, it works, just because he didn't use what I wanted him to use, will I give a low grade? So in correcting the activities, I was feeling this, the student did not do what I wanted, but he did it in another way. And I think this is a big problem in traditional education, which is a big difference from your proposal, this*

question of how to work, how it gives space for creativity within the evaluation process. So, if the student brings the tools and concepts alone. Now it is very difficult to think about activities, to create this whole world so that things can happen.

Finally, when giving her opinion if creativity can be applied to any discipline as computational thinking uses it, there is a limit, the teacher replied:

I think it has no limit. I think that many times people think there is and I sometimes end up kind of believing, because I usually give subjects that are easier to innovate, some teacher comes to me and says in calculations, I can't do any of that . I keep thinking, it is. But I'm already changing my mind because everything can be applied in some context. So, I think it can be done in all subjects, but then the teacher also has to be creative. For the teacher to work on the students' creativity, he has to be creative too, because if he doesn't develop his creativity, then it gets difficult. After all, then he doesn't see, how he will make the students develop it.

4.5 Discussion

In this section, we discuss the evaluation of the Conceptual Model. The model was evaluated in a study carried out in a Computational Thinking course within a Computer Science Course. The study started with a research question that sought to verify if creative learning is a factor that influences the development of Computational Thinking.

About the activity carried out using the Conceptual Model, there was a great deal of student involvement in solving the proposed problem, whose resolution was completed by all groups. In addition, the solution developed by the student groups presented characteristics of creative learning, because it caused a transformation in students from obtaining new solving problems skills by applying the knowledge of Computational Thinking. Such skills can be seen through engagement in carrying out the activity making them creative. In the process, there was personalization of information, comparison with knowledge based on interpretation and own points of view, and the generation of new ideas. Thus, the proposed activity verified the hypothesis that creative learning helps in the development of computational thinking and solving problems.

Regarding the students' responses in the semi-structured interview, the majority managed to identify elements of Computational Thinking in the activity as abstraction, which is a fundamental element in problem solving. Since, it is through abstraction that we evaluate relevant aspects of a problem and extract a representation of it taking into account a factor to propose the solution.

About the teacher's speech, she observed that the game had implicit elements of Computational Thinking and that the activity fulfilled its object, which is to help solve problems and also work on creativity and algorithmic. Besides, the teacher talks about the importance of developing creative and playful learning activities, stating that these types of activities, in addition to being fun, engage the student and fulfill their goal of helping students' learning. In order to develop creative activities, the teacher must also be creative in their pedagogical practice.

5 Conclusion

The skills of the future require professionals to be creative in solving increasingly complex problems. This issue raises the question of how to promote creative teaching and learning [1]. In this sense, the development of Computational Thinking can help in solving problems, as it has four pillars that allow the individual to seek creative solutions [5].

This research is in line with the presented context, proposing a model in which creative learning can be used to solve problems through the pillars of Computational Thinking.

To assess the applicability of the conceptual model, a quasi-experiment was carried out in an undergraduate course in Computing in the discipline of Computational Thinking, to understand the students' perception of the relationship between the activity with the PROGWISTER game and problem-solving. The research also aimed to investigate the teacher's perceptions about the activity and how it can help students learn. The study showed that creative learning contributed to problem solving and the development of Computational Thinking through the Conceptual Model that served as the foundation of the proposed activity. Since the students reported through the semi-structured interview that the activity helped in the development of algorithmic thinking, in the decomposition of the problem, and in abstraction.

In this perspective, the educational environment is relevant to the development of creativity, especially in the teacher's actions, through creative teaching and proposals for playful activities to engage students and teach them how to solve problems by seeking creative solutions. This approach places students as protagonists of their learning process, encouraging their active participation, and helping them to enhance their ability to propose creative solutions while promoting the development of Computational Thinking, thus confirming the hypothesis of this study.

As discussed in this article, creativity is a skill and can be taught and learned, in addition to being an expected competence for professionals of the future. The model presented here supports the systematization of this process.

This study showed, through the teacher's perception, that activities like these are relevant to the learning that the teacher called "implicit", because in the teacher's speech the activity promoted engagement in the students in the activity, even if at first "noise and agitation" were observed, students were not dispersed. Still, the teacher of the discipline, highlighted the importance of the teacher using creativity in her classes, so that the students are involved in their learning.

Furthermore, this article presents the second study using the conceptual model, showing its versatility, as it can even be used for different purposes. It contributes positively to the development of Computational Thinking and to solving problems algorithmically.

5.1 Paper Contribution

Considering the perspective of using creative learning to solve problems and promote Computational Thinking, the research contributed to i) create, evaluate and refine the Conceptual Model for the development of Computational Thinking; ii) help improve students' ability to solve problems; iii) insert creative learning in the classroom; iv) students' self-assessment of their learning and the ability to solve problems.

5.2 Limitations

The limitations of this research are as follows: i) the choice of students who participated in the research was not random; ii) the activity was not proposed for the whole discipline, being limited to just one activity; iii) the interview with students and teacher was conducted only once; iv) observation of the teacher's methodology was not carried out to analyze whether she used creativity in the classroom. Despite these limitations, the research has shown promising results.

5.3 Future Works

As future work, we intend to use the Conceptual Model in the Introduction to Programming discipline and integrate it with the agile learning methodology model. Since its values are geared towards meaningful learning, a collaboration between participants (students and teachers), and adaptability are promoted [25]. They can help both with the skills of 21st-century professionals and in solving problems and developing Computational Thinking. Furthermore, we intend to add other forms of analysis of the research, such as observation of the methodology of the teacher's classes, self-assessment questionnaires for students and teacher.

References

1. BBC, C.T.: Introduction to computational thinking (2017). https://www.bbc.com/education/guides/zp92mp3/revision/1. Accessed 3 Aug 2020
2. Bennedsen, J., Caspersen, M.E.: Teaching object-oriented programming-towards teaching a systematic programming process. In: Eighth Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts. Affiliated with 18th European Conference on Object-Oriented Programming (ECOOP). J.G (2004)
3. Gomes, A., Ke, W., Lm, S.K., Siu, A., Mendes, A.J., Marcelino, M.J.: A teacher's view about introductory programming teaching and learning-Portuguese and Macanese perspectives. In: 2017 IEEE Frontiers in Education Conference (FIE), pp. 1–8. IEEE (2017)
4. MacKinnon, D.W.: The nature and nurture of creative talent. Am. Psychol. **17**(7), 484 (1962)
5. National Research Council. Education for life and work: developing transferable knowledge and skills in the 21st century. In: National Academies Press [S.l.]
6. Osborn, A.: Your Creative Power [S.l.]. Read Books, London and New York (2008)
7. Sternberg, R.J.: Wisdom, Intelligence, and Creativity Synthesized. Cambridge University Press (2003)
8. Tschimmel, K.C: Sapiens e Demens no pensamento criativo do design, p. 574. Tese (Design) - Universidad e de Aveiro (2010)

9. Young what is creativity? J. Cretives, **19**(2), 77–87 (1985)
10. Wing, J.M.: Computational thinking. Commun. ACM **49**(3), 33–35 (2006)
11. Zavadil, P.: Creativity and analogical thinking. In: Katja Tschimmel. (Org.). The Creativity Virus - a Book about and for Creative Thinking. 1ed.Porto: Mindshake (2019)
12. Miller, L.D., et al.: Integrating computational and creative thinking to improve learning and performance in CS1. In: Proceedings of the 45th ACM Technical Symposium on Computer Science Education (2014)
13. Peteranetz, M.S., et al.: Examining the impact of computational creativity exercises on college computer science students' learning, achievement, self-efficacy, and creativity. In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education (2018)
14. Barr, V., Stephenson, C.: Bringing computational thinking to k-12: What is involved and what is the role of the computer science education community? ACM Inroads **2**, 48–54 (2011)
15. Proctor, C., Blikstein, P.: How broad is computational thinking? a longitudinal study of practices shaping learning in computer science. Int. Soc. Learn. Sci. Inc.[ISLS] (2018)
16. Piteira, M., Haddad, S.: Innovate in your program computer class: an approach based on a serious game. In: Workshop on Open Source and Design of Communication (2011)
17. Santiago, A.D., Kronbauer, A.: Um Modelo Lúdico para o Ensino de conceitos de programação de computadores. Braz. Symp. Comput. Educ. (Simp. Bras. Inform. Educ.-SBIE), **27**(1), 420–428 (2016)
18. Silva, T.S.C., Melo, J.C.B., Tedesco, P.C.A.R.: The creative process in the development of computational thinking in higher education. CSEDU (1) (2020)
19. Rais, A.E., Sulaiman, S., Syed-Mohamad, S.M.: Game-based approach and its feasibility to support the learning of object-oriented concepts and programming. In: Software Engineering (MySEC). 5th Malaysian Conference. Malaysian: [s.n.], pp. 307–312 (2011)
20. Kuo, W.-C., Hsu, T.-C.: Learning computational thinking without a computer: how computational participation happens in a computational thinking board game. Asia Pac. Educ. Res. **29**(1), 67–83 (2020)
21. Shell, D.F., Soh, L.K., Flanigan, A.E., Peteranetz, M.S., Ingraham, E.: Improving students' learning and achievement in CS classrooms through computational creativity exercises that integrate computational and creative thinking. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, pp. 543–548 (2017)
22. Davies, S.: The effects of emphasizing computational thinking in an introductory programming course. In: 2008 38th Annual Frontiers in Education Conference, pp. T2C–3 (2008)
23. Kazimoglu, C., Kiernan, M., Bacon, L., Mackinnon, L.: A serious game for developing computational thinking and learning introductory computer programming. Proc.-Soc. Behav. Sci. **47**, 1991–1999 (2012)
24. Conde, M.Á., Fernández-Llamas, C., Rodríguez-Sedano, F.J., Guerrero-Higueras, Á.M., Matellán-Olivera, V., García-Penãlvo, F.J.: Promoting computational thinking in K-12 students by applying unplugged methods and robotics. In: Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality. ACM, p. 7 (2017)
25. Delhij, A., van Solingen, R., Wijnands, W.: The eduScrum guide. EduScrum Team (2015)