education sciences

MDPI

Article

# Use of Technological Resources for the Development of Computational Thinking Following the Steps of Solving Problems in Engineering Students Recently Entering College

**Ronald Paucar-Curasma [1,*], Klinge Orlando Villalba-Condori [2], Jorge Mamani-Calcina [3], David Rondon [4], Mario Gustavo Berrios-Espezúa [5] and Claudia Acra-Despradel [6]**

1 Grupo de Investigación TIC Aplicadas a la Sociedad, Universidad Nacional Autónoma de Tayacaja Daniel Hernández Morillo, Pampas 09156, Peru
2 Facultad de Ciencias y Tecnologías Sociales y Humanidades, Universidad Católica de Santa María, Arequipa 04001, Peru
3 Departamento de Estudios Generales, Universidad Tecnológica del Perú, Arequipa 04001, Peru
4 Departamento Estudios Generales, Universidad Continental, Arequipa 04001, Peru
5 Facultad Ciencias Histórico Sociales, Universidad Nacional de San Agustín de Arequipa, Arequipa 04001, Peru
6 Departamento de Investigación, Universidad Nacional Pedro Henríquez Ureña, Santo Domingo 10100, Dominican Republic
* Correspondence: rpaucarc@unat.edu.pe

**Abstract:** In this work, the authors propose the use of technological resources to develop computational thinking following the steps or phases of problem-solving for first-year students. During the development of the activities using technological resources (Arduino board, sensors, electronic devices, and mBlock) the students carried out activities, such as algorithm development and programming of the Arduino board and sensors from a friendly and playful interface such as the mBlock, as well as the debugging of programs until obtaining the expected results. These activities had an impact on the cognitive processes, practices, and technological perspectives of the students. Causality has been shown to exist between computational thinking skills and problem-solving phases in an environment of engineering students entering college. For the analysis of the relationship between computational thinking skills and problem-solving, Pearson's statistical correlation test was used through SPSS software.

**Keywords:** computational thinking; problem-solving; first-year students; technological resources; technological projects; skills; STEM

check for updates

**Citation:** Paucar-Curasma, R.; Villalba-Condori, K.O.; Mamani-Calcina, J.; Rondon, D.; Berrios-Espezúa, M.G.; Acra-Despradel, C. Use of Technological Resources for the Development of Computational Thinking Following the Steps of Solving Problems in Engineering Students Recently Entering College. *Educ. Sci.* **2023**, *13*, 279. https://doi.org/10.3390/educsci13030279

Academic Editors: José Miguel Vílchez-González, Palma Tonda Rodríguez and José Luis Lupiáñez Gómez

Received: 20 January 2023
Revised: 1 March 2023
Accepted: 2 March 2023
Published: 5 March 2023



## 1. Introduction

In Latin America, countries are heterogeneous and each country is heterogeneous within itself. There are differences between rural and urban areas, students of high and low socioeconomic levels, different cultural levels, and pronounced differences between rural and urban schools concerning educational quality. In the last two PISA assessments in Latin American countries, the results showed that Peru was the most critical, where 90% of students did not achieve the required level of reading skills, and 95% of students did not achieve the required level of mathematical skills, considered key for citizens to develop in today's world and contribute to the development of the country [1]. Inequalities in the education sector manifest themselves in various ways, one of them being the effective use of ICTs, not only as simple users of technologies, but also to strengthen cognitive skills [2]. These inequalities in the education sector are very marked in the regions of Peru, resulting in poor academic training in students who are entering public universities in the Huancavelica region, where students have low skills in problem-solving, mathematical reasoning, logic, reading comprehension, abstraction skills, critical thinking, pattern recognition, and teamwork [3,4].

In recent studies, computational thinking (CT) is receiving greater interest at all levels of the education sector as a fundamental support in strengthening problem-solving skills in students, even more so in students with limited competencies in mathematical reasoning and logic [5,6]. CT is the type of thinking that helps in solving problems, where solutions are represented as steps and computational algorithms [7,8]. In the latest definitions of CT, apart from computational skills, the authors highlight the development of competencies in people; these competencies are skills to face complex problems, persistence, treatment, and confidence in solving complex problems. Therefore, CT is an important skill for problem-solving in academic and social fields [9–11], not only for STEM students but also for students from other disciplines [12,13].

Several authors state that computational thinking and problem-solving techniques complement each other in the generation of academic competencies in students [14,15]. One of the most used techniques in problem-solving is the proposal by Pólya [16], which is formed by four phases or processes for solving problems in a sequential and orderly manner. The phases are: understanding the problem, preparing the plan, executing the plan, and reviewing of the solution. The advantages of using the Pólya method allows engineering students to improve their analytical capacity and understanding of the problem, strengthen their skills to propose strategies in an orderly and sequential manner in problem-solving, employ strategies to successfully implement a business plan developed in the previous phase, and finally strengthen the critical capacity to evaluate the functionalities of the product and its respective validation [5].

Concerning the use of technological resources for solutions STEM, Sobreira et al. [4] presented the Snap4 Arduino visual programming platforms to be used by students starting their engineering careers, adding the AppInventor tool and IoT (Internet of Things) devices for activities related to their geographical area. Diaz et al. [17] promotes the Codeblocks tool integrated with Tinkercad, which generates motivation in students, the key factors being diversity in the composition of the student group, availability of 3D printers to materialize designs, and a test environment. Gao et al. [18] explored AppInventor for integration into undergraduate computer science and engineering courses, thereby introducing computational thinking in the context of creating mobile apps, and recommended it for beginning students to help reduce barriers to programming. They also stated that block programming-based application development reduces syntax errors and encapsulates mobile device functions in high-level abstractions that are easy to incorporate into applications. Recently, Trilles et al. [19,20] promoted computational thinking at pre-university students through the Sucre4Stem project, using block programming, assembly of sensors and actuators in microcontrollers, network connectivity, and remote data sharing. Through the components of Sucre4Stem, students designed, created, and programmed collaborative sensorization projects that recreate real situations of the IoT.

This article develops computational thinking skills following the phases of problem-solving in students recently entering the engineering career at a public university located in the Andes of Peru. The educational strategy is based on the proposal of technological projects in the classroom; once the technological projects were finished, the computational thinking skills and problem-solving phases were evaluated in two periods in 2021-II and 2022-I, identifying relationship between computational thinking skills and problem-solving phases using Pearson's correlation.

## 2. Review of Literature

### 2.1. Computational Thinking in Higher Education

Several authors have expressed the importance and benefits of computational thinking in higher education, highlighting the ability of abstraction and algorithmic thinking in strengthening reading comprehension and in solving complex problems following algorithmic methods [21–25]. In addition, Wilson et al. [26] in an applied study pointed out that programming helps to understand and develop activities of mathematics and other disciplines that can be abstract or complex; this involves developing programs through the computer to solve problems in mathematics or other areas. Shyamala et al. [3] pointed out that the use of tools based on block programming and hardware generates interest and motivation in students, as well as teamwork and problem-solving through abstraction, decomposition, and algorithmic thinking skills. It also directly relates to the development of common skills through creative programming and innovation [27–29]. Finally, Kules [30] added critical thinking as a form of reasoning and exchange of ideas before solving problems through computational thinking skills. Concerning the key skills of CT, most researchers consider abstraction, decomposition, algorithmic design, generalization, and evaluation [31,32]. Abstraction is the process of deciding or ignoring the details or characteristics of a thing; decomposition is the process of breaking down a complex problem into much smaller and more feasible parts; generalization includes discovering similarities or patterns in any complex problem or broken problem; algorithmic design is a set of rules or instructions, well posed, ordered sequentially, and finite, which allows a task to be performed or executed following steps established successively in a safe way to solve an identified problem; and evaluation is recognizing and determining the scope of performing processes, in terms of efficiency and use of resources.

### *2.2. Problem-Solving*

Problem-solving or solving a problem is a cognitive process to obtain a goal for the individual who solves it [33]. The processes of problem-solving are shaped by cognitive factors: planning, critical thinking, and debate to make decisions [34].

In the scientific literature, problems and problem-solving techniques or methods have been contextualized [11,35–39]; where each problem has its particularity, so there is no single procedure that guarantees its solution [40], but several procedures indicate the steps or phases to follow in order to solve a problem [41]. In general, four stages, phases, or processes can be identified in the solution of any problem: Understanding the problem, drawing up the plan, executing the plan, and reviewing the solution [5,16,42]. These four phases allow solution of the identified problem in a sequential way. The advantages of using Pólya's method in the resolution of problems are that it allows students to improve their analytical capacity and understanding of the problem, strengthen skills to propose strategies in an orderly and sequential way in the resolution of problems, use strategies to correctly execute a business plan drawn up in the previous phase, and strengthen their critical capacity to evaluate the functionalities of the product and its respective validation.

### *2.3. Computational Thinking and Problem-Solving*

Wing [12] defined computational thinking as problem-solving, systems design, and understanding human behavior using the fundamental concepts of computer science. Several authors have also contributed to the definition, pointing out that one of the fundamental reasons for computational thinking is problem-solving, and to solve problems, a set of processes or phases must be followed to reach the solution. Additionally, computational thinking is joined by other problem-based learning approaches, which strengthen critical thinking, considered as a phase before computational thinking. In addition, these approaches strengthen communication skills when presenting project results. They can also provide opportunities for teamwork, searching, analyzing, synthesizing research materials, and lifelong learning [12,13,43,44]. Rsecently, Román-González [45] defined computational thinking as "the (human) ability to solve problems and express ideas making use of concepts, practices, and perspectives of Computer Science". In addition, Rabiee [46] stated that the applicability of computational thinking should be seen in practice as a universal concept and a real-world problem-solving tool.

From that perspective, computational thinking has gained popularity and has been emphasized as an effective means of understanding and solving complex problems by using computer science concepts and techniques [47]. As a computational process, computational thinking begins by confronting problems, the solution of which involves the use of the skills of decomposition, generalization, abstraction, automation, algorithms, and evaluation [47–50]. Based on the similarity between computational thinking skills and problem-solving phases, which has also been recognized in the research of Voogt [51], it is stated that computational thinking is a specific form of problem-solving, where each computational thinking skill is located within a problem-solving process or phase, but also provides a more specific description of learning processes that reflect techniques and concepts of computer science. From such a perspective, a student involved in computational thinking can also be considered involved in the problem-solving phases [52]. To know the relationship between computational thinking skills and problem-solving phases, different studies have been analyzed, whose results are shown in Table 1. Abbreviations for computational thinking skills are ABS (Abstraction), DES (Decomposition), GEN (Generalization), ALG (Algorithmic Design), and EVA (Evaluation). Concerning the problem-solving phases, the authors agree with most of the phases.

**Table 1.** Problem-solving phase and computational thinking skills.

| Phases of Resolution of Problems | Computational Thinking Skills | | | | |
|---|---|---|---|---|---|
| By Ubaidullah [14] | ABS | DES | GEN | ALG | EVA |
| Understanding/definition | X | | | | |
| Planning | | X | X | | |
| Design | | | | X | |
| Codification | | X | | X | X |
| Evaluation | | | X | | X |
| By Jeng [53] | ABS | DES | GEN | ALG | EVA |
| Recognition of the problem. | X | | | | |
| Solution strategy development | | X | X | | |
| Organization of knowledge about the problem | | | | X | |
| Solution evaluation | | | | | X |
| Por Joshua [15] | ABS | DES | GEN | ALG | EVA |
| Simplifying the problem | X | | | | |
| Dividing the problem into smaller parts | | X | | | |
| List of steps to resolve | | | X | | |
| By Maharani [54] | ABS | DES | GEN | ALG | EVA |
| Decision on the subject matter | X | | | | |
| Solution formulation | | | X | | |
| Division of complex problems | | X | | | |
| Step-by-step design to solve the problem | | | | X | |
| Identification to correct errors | | | | | X |
| By Kale [52] | ABS | DES | GEN | ALG | EVA |
| Understanding the problem | X | | | | |
| Plan and monitoring | X | X | X | | |
| Execution | | | | X | |
| Check/reflect | | | | | X |
| By Rabiee [46] | ABS | DES | GEN | ALG | EVA |
| Identification/understanding of the problem | X | | | | |
| Breakdown of the main problem | | X | | | |
| Solution development | | | X | X | |
| Implementation | | | | X | X |
| Validation | | | | | X |
| By Pedaste [55] | ABS | DES | GEN | ALG | EVA |
| Problem identification | X | X | | | |
| Selection of strategies | | | | X | |
| Strategy execution | | | | X | |
| Review of results | | | X | | X |

From the results obtained from Table 1, approximately 30 phases have been found that indicate the steps or phases to follow to solve a problem. The other phases were mentioned with different vocabularies (synonyms) by the authors. Therefore, in any problem four stages or phases can be identified: understanding the problem, drawing up the plan, executing the plan and reviewing the solution; these four phases can be seen in other research [5,16,42,51], For each of the four phases of problem-solving there is similarity or equivalences identified in the study. The phase of understanding of the problem is similar or equivalent to the phase of "understanding/definition", "recognition of the problem", "simplification of the problem", "decision on the object", "understanding of the problem", "identification/understanding of the problem", and "identification of the problem". The plan development phase is similar to the "planning" phase, "development of solution strategies", "division of the problem into smaller parts/list of steps to solve", "formulation" the solution/vision of complex problems", "plan and monitoring", "breakdown of the main problem", and "selection of strategies". The implementation phase of the plan is similar or equivalent to the phase of "design and coding", "step-by-step design to solve the problem", "execution", "solution development/implementation", and "strategy execution". The revision phase of the solution is equivalent to the phase of "evaluation", "evaluation of the solution", "identification to correct errors", "verification/reflection", "validation", and "review of results". Table 2 summarizes the relationship between the four phases of problem-solving and the five key skills of computational thinking.

**Table 2.** Summary of the relationship between problem-solving phases and computational thinking skills.

| Problem-Solving Phases | Computational Thinking Skills | | | | |
|---|---|---|---|---|---|
| | ABS | DES | GEN | ALG | EVA |
| Understanding the problem | X | | | | |
| Preparation of the plan | | X | X | | |
| Implementation of the plan | | | | X | |
| Solution Review | | | | | X |

## 3. Materials and Methods

The participants in this research were 37 students of the industrial engineering career of the Universidad Nacional Autónoma de Tayacaja Daniel Hernández Morillo (UNAT) and 49 students of the systems engineering career of the Universidad Nacional de Huancavelica (UNH), both located in the province of Tayacaja in the Huancavelica region, in the Andes of Peru. The participants were students recently entering UNAT and UNH in the period 2021-II and 2022-I, respectively, and were students in the first year of studies at the university. The students' sampling was intentional, not probabilistic, according to the authors' criteria; that is, the similarity of careers, students of the teacher (author) in the semester, etc. Therefore, under these conditions, the investigation could be carried out efficiently.

To know the initial computational thinking skills of the newly admitted students, 5 reagents on computational thinking were used [56–58]. These reagents are related to the skills of decomposition, abstraction, generalization, algorithmic design, and evaluation of computational thinking.

For the development of technological projects, students used technological resources consisting of hardware and software. The hardware used were Arduino, environment temperature sensors, distance sensors, obstacle detection sensors, LEDs, displays, etc. For software, the mBlock was used, a programming interface based on blocks which allows interaction with the sensors through Arduino. The use of a block-based programming interface allowed students to focus on computational concepts rather than the syntax of programming languages, while the presence of electronic sensors and output devices allowed students to enthusiastically view the actual movement/consequence of the program occurring in the physical world, generating immediate visual feedback from the programming and motivating beginner students to more easily test their hypothesize and refine their ideas.

The instrument used for data collection corresponding to computational thinking skills is the Computational Thinking Test (TPC) of Román-Gonzalez [59]. The test has been validated in criterion and convergence [60,61]. Initially (version 1.0) the instrument was composed of 40 items in length; currently, the instrument has 28 items (version 2.0). Each of the 28 items was designed and characterized in five axes (computational concept addressed, item interface-environment, style of the response alternatives, existence or non-existence of nesting, and required task). The instrument is correctly adapted to the cognitive level of newly admitted students, mostly from rural schools with low educational quality, who range in age from 16 and 17 years old.

The instrument for data collection corresponding to problem-solving is based on the proposal of Molina [5,62], and structured according to the problem-solving method of Pólya [16]. This method is composed of four phases: understanding the problem, preparing the plan, executing the plan, and verifying the solution. It is formed by 22 items in total. The phase of understanding the problem consists of 7 items, the phase of preparation of the plan consists of 5 items, the phase of execution of the plan consists of 5 items, and the phase of revision of the solution consists of 7 items. Each item is answered according to the Likert scale, where a 1 is a "no" and a 5 is a "yes", while the intermediate values take the values according to the degree of agreement or disagreement.

For the processing of data obtained from the research, statistical tools were used, such as descriptive statistics (mean and standard deviation), as well as inferential statistics (Pearson's correlation) to correlate computational thinking skills and problem-solving. The computer tool for data interpretation that was used is the statistical software SPSS in its 25 version.

## 4. Results

### *4.1. Educational Strategy through the Proposal of Technological Projects*

Before assigning the technological projects to the group of students, at the beginning of the semesters or academic periods 2021-II and 2022-I an initial test was applied based on five items to determine the preliminary computational thinking skills of recently admitted students at the university. Table 3 shows the technological projects proposed to the students according to the results obtained in the initial test. This form of distribution of technological projects was carried out with the purpose that the members of each group start with the same skills (rhythms, styles, and learning processes) so that they had the same opportunities to execute the activities of the projects, and so that the teachers could use various educational strategies to meet the academic needs of each group. Technological projects are related to the problem of the context of students.

The execution of the technological projects was completed over 16 weeks in the classroom, following the four phases of problem-solving, for both semesters. The activities of the phase of understanding the problem were distributed over 5 weeks. In this phase, the classroom teacher provided exercises on abstraction. The students presented the problem statement of their projects through visual organizers. The activities of the preparation phase of the plan lasted 2 weeks. In this phase, the teacher provided exercises of decomposition and generalization. Students identified activities from other projects to use in their projects, where they presented a set of activities to solve the identified problem. The activities of the execution phase of the plan lasted 5 weeks. In this phase, the teacher provided practical exercises on algorithmic design applied to the common tasks of people [32], as well as exercises using the mBlock programming environment, Arduino board, and uses of sensors (temperature, humidity, ultrasound, light, obstacle, etc.). The students presented advances of their projects, showing programs in mBlock, circuit implementations, programming of microcontrollers and sensors. The activities of the review phase of the solution lasted 1 week. The students evaluated the operation of their prototypes, and the teacher provided feedback to improve the operation of their prototypes. Figure 1 shows the results of the executed technological projects identified by their ID, which consist of prototypes developed by students in the 2021-II and 2022-I semesters, including hardware and software components.

**Table 3.** Proposal of technological projects.

| Initial Test Results | ID | Technological Projects 2021-II |
|---|---|---|
| **Correct answers** | P1–2021 | Monitoring of vegetable production in greenhouses in the city of Pampas, Tayacaja, Huancavelica region. |
| | P2–2021 | Implementation of a water level monitoring prototype in the Viñas reservoir of the city of Pampas, Tayacaja. |
| **1 incorrect answer** | P3–2021 | Implementation of a control and security system in a food market. |
| **2 incorrect answers** | P4–2021 | Prototype of automatic switching off and on of public lighting for the city of Pampas, Tayacaja, Huancavelica. |
| **3 incorrect answers** | P5–2021 | Smartboard for learning in single-teacher classrooms in the city of Pampas, Tayacaja, Huancavelica. |
| **4 incorrect answers** | P6–2021 | Monitoring of solid waste in the city of Pampas, Tayacaja, Huancavelica. |
| **Initial test results** | **ID** | **Technology projects 2022-I** |
| **Correct answers** | P1–2022 | LED games in the teaching of basic mathematical operations for primary school students of the city of Pampas, Tayacaja, Huancavelica. |
| | P2–2022 | Prototype of automatic distance detection alarm for vehicles in the Huancavelica region. |
| **1 incorrect answer** | P3–2022 | Smart cane with sensors for visually impaired people in the city of Pampas, Tayacaja, Huancavelica. |
| **2 incorrect answers** | P4–2022 | Monitoring of temperature and humidity with an automated irrigation system in vegetable production in the city of Pampas, Tayacaja, Huancavelica. |
| | P5–2022 | The animal safety system in the Huancavelica region. |
| **3 incorrect answers** | P6–2022 | Monitoring and control of humidity and temperature in the greenhouse in the Huancavelica region. |
| | P7–2022 | Home automation for the security and tranquility of homes in the city of Pampas, Tayacaja, Huancavelica. |
| **4 incorrect answers** | P8–2022 | Implementation of a biosafety prototype against COVID-19 in the professional school of systems engineering. |
| | P9–2022 | Monitoring of solid waste in homes in the city of Pampas, Tayacaja, Huancavelica. |

### *4.2. Evaluation of Computational Thinking and Problem-Solving*

In the 2021-II semester, the data collected from computational thinking skills and problem-solving phases correspond to 37 students of the industrial engineering career, while for the 2022-I semester, the data collected correspond to 49 students of the systems engineering career. Table 4 shows the reliability according to Cronbach's alpha of the data collected about computational thinking and problem-solving.

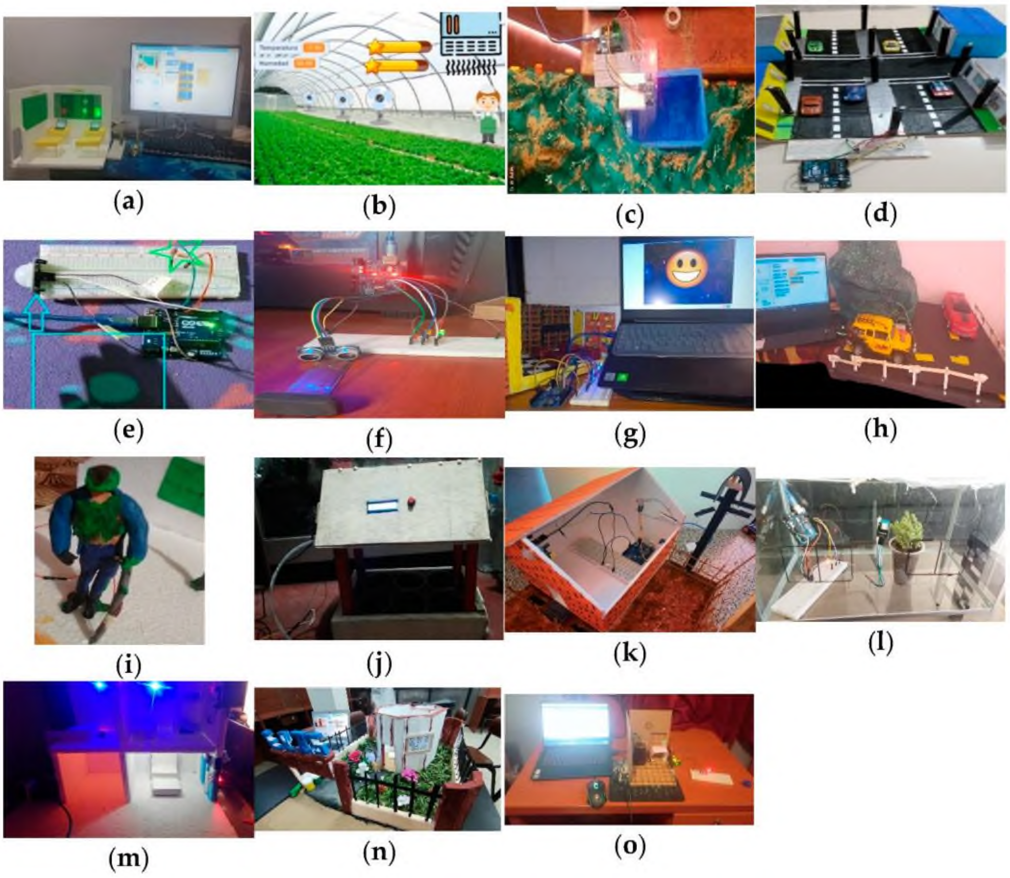

**Figure 1.** Results of the technological projects executed in the period 2021-II and 2022-I: (**a**) Project ID P1-2021; (**b**) Project ID P2-2021; (**c**) Project ID P3-2021; (**d**) Project ID P4-2021; (**e**) Project ID P5-2021; (**f**) Project ID P6-2021; (**g**) Project ID P1-2022; (**h**) Project ID P1-2022; (**i**) Project ID P1-2022; (**j**) Project ID P1-2022; (**k**) Project ID P1-2022; (**l**) Project ID P1-2022; (**m**) Project ID P1-2022; (**n**) Project ID P1-2022; (**o**) Project ID P1-2022.

**Table 4.** Cronbach's alpha of collected data.

| | **Computational Thinking** | |
|---|---|---|
| Semester | Alfa de Cronbach | N of elements |
| 2021-II | 0.793 | 28 |
| 2022-I | 0.799 | 28 |
| | **Problem-Solving** | |
| Semester | Alfa de Cronbach | N of elements |
| 2021-II | 0.965 | 24 |
| 2022-I | 0.924 | 24 |

The evaluation of computational thinking skills was carried out through the method of Román-Gonzalez [59], where, the ability is related to the 28 items of the test [63,64] generalization (4–6, 8–12, 14, 15, 17–18, 20, 22, 23 and 25–28), algorithmic design (1–28), and evaluation (3, 7, 10, 11, 15, 16, 19, 20 and 23–28). Figure 2 shows the average correct percentages of items related to computational thinking skills in the 2021-II and 2022-I semesters. In the most recent semester, the students got more than 60% of skills right. It is also observed, in the period 2021-II, that the highest percentage of successful ability is related to abstraction, followed by algorithmic design, generalization, decomposition, and evaluation, while for the period 2022-I, the highest percentage of successful ability is related to algorithmic design, followed by generalization, evaluation, abstraction, and decomposition.

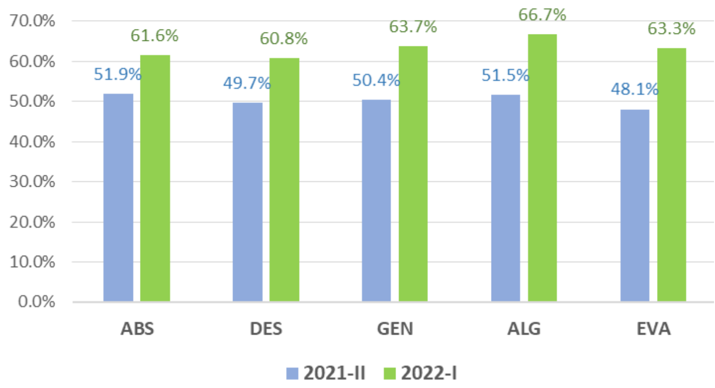


**Figure 2.** Means of the correct percentages of computational thinking.

For the evaluation of the phases of problem-solving, data were collected according to the Likert scale, where a 1 is a "no" and a 5 is a "yes", while the intermediate values range according to the degree of agreement or disagreement. Table 5 shows the averages rated for each phase of problem-solving by students in the 2021-II and 2022-I semesters. The problem-solving phases are abbreviated: PRO = Understanding the problem, ELA = Elaboration of the plan, EJE = Execution of the plan, and REV = Review of the solution. For the period 2021-II, the phase of understanding the problem had the best assessment, followed by the revision of the solution, execution of the plan, and preparation of the plan. Regarding the standard deviation, the phase of understanding the problem has the lowest value (0.73152), followed by preparation of the plan (0.87987), revision of the solution (0.91321), and execution of the plan (0.91936). For the period 2022-I, the execution phase of the plan presented the best evaluation by the students, followed by understanding the problem, reviewing the solution, and preparing the plan. Regarding the standard deviation, the phase of revision of the solution has the lowest value (0.68461), followed by understanding the problem (0.68602), execution of the plan (0.69586), and preparation of the plan (0.74024). For the semesters 2021-II and 2022-I, the average of the problem-solving phases is almost homogeneous in their assessment according to the Likert scale.

**Table 5.** Total average of the problem-solving phase.

| | 2021-II | | | 2022-I | | |
|---|---|---|---|---|---|---|
| | N | Mean | Deviation Standard | N | Mean | Deviation Standard |
| PRO | 37 | 3.5981 | 0.73152 | 49 | 3.6765 | 0.68602 |
| ELA | 37 | 3.3838 | 0.87987 | 49 | 3.2531 | 0.74024 |
| EJE | 37 | 3.5243 | 0.91936 | 49 | 3.7510 | 0.69586 |
| REV | 37 | 3.5246 | 0.91321 | 49 | 3.5133 | 0.68461 |

### *4.3. Evaluation of Pearson's Correlation between Problem-Solving and Computational Thinking*

Table 6 shows the results of Pearson's statistical correlation test between computational thinking skills and problem-solving phases. For the period 2021-II, there is a moderate positive correlation between the execution phase of the plan and the algorithmic design ability, a weak positive correlation between the revision phase of the solution and the ability to evaluate, and a weak positive correlation between the phase of understanding the problem and the ability to abstract. There is no correlation between the plan-making phase and the skills of decomposition and generalization. For the period 2021-II, there is a moderate positive correlation between the execution phase of the plan and the algorithmic design ability, a weak positive relationship between the solution review phase and the evaluation skill, a weak positive relationship between the phase of understanding the problem and the ability to abstract, and a weak positive correlation between the plan-making phase and the skills of decomposition and generalization.

**Table 6.** Pearson's correlation.

| | | 2021-II | | | | | 2022-I | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **ABS** | **DESC** | **GEN** | **ALG** | **EVA** | **ABS** | **DESC** | **GEN** | **ALG** | **EVA** |
| PRO | Pearson correlation | 0.352 * | | | | | 0.366 ** | | | | |
| | Sig. (bilateral) | 0.033 | | | | | 0.010 | | | | |
| | N | 37 | | | | | 49 | | | | |
| ELA | Pearson correlation | | 0.292 | 0.287 | | | | 0.340 * | 0.339 * | | |
| | Sig. (bilateral) | | 0.079 | 0.085 | | | | 0.017 | 0.017 | | |
| | N | | 37 | 37 | | | | 49 | 49 | | |
| EJE | Pearson correlation | | | | 0.491 ** | | | | | 0.492 ** | |
| | Sig. (bilateral) | | | | 0.002 | | | | | 0.000 | |
| | N | | | | 37 | | | | | 49 | |
| REV | Pearson correlation | | | | | 0.381 * | | | | | 0.415 ** |
| | Sig. (bilateral) | | | | | 0.020 | | | | | 0.003 |
| | N | | | | | 37 | | | | | 49 |

**. The correlation is significant at level 0.01 (bilateral). *. The correlation is significant at the 0.05 level (bilateral).

## 5. Discussion

According to the statistical results of the Pearson correlation, the abilities of abstraction, decomposition, generalization, algorithmic design, and evaluation are related to the phases of problem understanding, plan development, plan execution, and solution review, respectively. In the phase of comprehension of the problem, the ability to abstract the problematic situation of the technological project using mental maps, highlighting the main problem, causes, and effects, was observed in the students [65]; these activities have strengthened abstraction skills in students [14]. In the phase of elaboration of the plan, the students proposed various solutions, breaking them down into a set of manageable activities; for example, they proposed the activity of acquisition of electronic devices, hardware design, hardware implementation, program development, etc. These activities strengthened the decomposition ability in the students [15,53]. In the plan development phase, the students searched for background or solutions in other projects, identifying activities to be applied in their projects; in this way the students strengthened their generalization skills [52]. In the execution phase of the plan, the students executed or developed activities established in the previous phase. These activities were carried out in an orderly manner, step by step. First, they implemented the Arduino board with sensors; second, they developed algorithms; third, they debugged programs, etc. This way of working in an orderly manner, step by step, strengthened the ability of algorithmic design in students [46]. In the solution review phase, the students evaluated the operation of the components of their projects; for example, the circuitry and programming interface based on mBlock were assessed. In addition, they evaluated the final product or prototype integrated into a model. These activities strengthened the evaluation ability of the students [14,55].

## 6. Conclusions

The execution of technological projects following the problem-solving phases has contributed to the development of computational thinking skills in engineering students recently admitted to the university. The use of technological resources, such as the Arduino board, electronic sensors, and mBlock software to solve problems in the context of their city or region, motivated the students to carry out activities such as the development of algorithms, programming the Arduino board, configuration of sensors, and development of a friendly and playful interface through the mBlock, as well as the debugging of programs until the expected results were obtained. This set of activities had an impact on the cognitive processes of the students (reasoning, decision making, understanding of the environment, logic, etc.), as well as in technological practices and perspectives. These activities could easily be developed in algorithms, introduction to programming, information management, and related courses, and would be a practical platform to help students acquire important skills in the current context of the 21st century, not only for engineering students but also in other disciplines such as social sciences, communications, arts, etc.

Problem-solving using technological resources fosters the development of computational thinking in engineering students recently admitted to college. To obtain optimal results in the development of computational thinking skills in students, the classroom teacher must constantly monitor the execution of projects at each stage or phase of problem-solving. The teacher must also provide feedback on the project activities, especially for students from remote or rural areas who are often interacting with electronic devices and programming software for the first time.

During the application of the educational strategy in the classroom through technological projects for the development of computational thinking, it is recommended to apply an initial test to determine the computational thinking skills of students recently admitted to university. According to the results of the test, groups or work teams should be formed to execute the project in the classroom. In this way, there will be groups with homogeneous members with similar rhythms, styles, and learning processes. In addition, the teacher will be able to use various educational strategies to meet the academic needs of each group.

The limitations of the research: The observation and evaluation of the students were online due to COVID-19. In addition, only 89 students participated. The sampling of students was intentional because the teacher had not taught other students in the scenario of a pandemic.

**Author Contributions:** Conceptualization, R.P.-C. and K.O.V.-C.; methodology and formal analysis, R.P.-C. and K.O.V.-C.; investigation, R.P.-C. and K.O.V.-C.; resources and data curation, R.P.-C., J.M.-C., D.R., M.G.B.-E. and C.A.-D.; writing—original draft preparation, R.P.-C.; project administration and funding acquisition, R.P.-C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been funded by R.P.-C.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Trucco, D. *Educación Y Desigualdad En América Latina*; Naciones Unidas Santiago, Chile, 2016. [En línea]. Available online: https://repositorio.minedu.gob.pe/bitstream/handle/20.500.12799/3053/EducacionydesigualdadenAL.pdf (accessed on 15 May 2020).
2. Sunkel y, G.; Trucco, D. *TIC Para La Educación En América Latina*; Riesgos y Oportunidades: Santiago, Chile, 2010.
3. Shyamala, C.K.; Velayutham, C.S.; Parameswaran, L. Teaching Computational Thinking to Entry-Level Undergraduate Engineering Students at Amrita University. In Proceedings of the 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, Greece, 25–28 April 2017; pp. 1731–1734. [CrossRef]

4. Sobreira, P.D.L.; Abijaude, J.W.; Viana, H.D.G.; Santiago, L.M.S.; El Guemhioui, K.; Wahab, O.A.; Greve, F. Usability evaluation of block programming tools in IoT contexts for initial engineering courses. In Proceedings of the 2020 IEEE World Conference on Engineering Education (EDUNINE), Bogota, Colombia, 15–18 March 2020; pp. 1–5. [CrossRef]
5. Molina, N.; Adamuz, Y.; Bracho, R. La resolución de problemas basada en el método de Polya usando el pensamiento computacional y Scratch con estudiantes de Educación Secundaria, Handb. *Handb. Educ. Psychol.* **2020**, *49*, 287–303. [CrossRef]
6. Paucar-Curasma, R.; Villalba-condori, K.O.; Jara, N.; Quispe, R.; Cabrera, J.; Ponce-Aguilar, Y.M. Computational Thinking and Block-Based Programming for Beginning Engineering Students: Systematic Review of the Literature. In Proceedings of the 2021 XVI Latin American Conference on Learning Technologies (LACLO), Arequipa, Peru, 19–21 October 2021; pp. 530–533.
7. Grover, S.; Pea, R. Computational Thinking in K-12: A Review of the State of the Field. *Educ. Res.* **2013**, *42*, 38–43. [CrossRef]
8. Aho, A.V. Computation and Computational Thinking. *Comput. J.* **2012**, *55*, 832–835. [CrossRef]
9. Lee, I.; Martin, F.; Apone, K. Integrating computational thinking across the K–8 curriculum. *ACM Inroads* **2014**, *5*, 64–71. [CrossRef]
10. Weintrop, D.; Beheshti, E.; Horn, M.; Orton, K.; Jona, K.; Trouille, L.; Wilensky, U. Defining Computational Thinking for Mathematics and Science Classrooms. *J. Sci. Educ. Technol.* **2015**, *25*, 127–147. [CrossRef]
11. Neo, C.H.; Wong, J.K.; Chai, V.C.; Chua, Y.L.; Hoh, Y.H. Computational Thinking in Solving Engineering Problems-A Conceptual Model Definition of Computational Thinking. *Asian J. Assess. Teach. Learn.* **2021**, *11*, 24–31. [CrossRef]
12. Wing, J.M. Computational Thinking. *Commun. ACM* **2006**, *49*, 267–271. [CrossRef]
13. Allen, Y.D.E.; Duch, B.J.; Groh, S.E. The power of problem-based learning in teaching introductory science courses. *New Dir. Teach. Learn.* **1996**, *1996*, 43–52. [CrossRef]
14. Ubaidullah, N.H.; Mohamed, Z.; Hamid, J.; Sulaiman, S.; Yussof, R.L. Improving Novice Students' Computational Thinking Skills by Problem-Solving and Metacognitive Techniques. *Int. J. Learn. Teach. Educ. Res.* **2021**, *20*, 88–108. [CrossRef]
15. Weese, J.; Feldhausen, R. STEM Outreach: Assessing Computational Thinking and Problem Solving. In Proceedings of the 2017 ASEE Annual Conference & Exposition, Columbus, Ohio, 24–28 June 2017. [CrossRef]
16. Polya, G. *How to Solve It*, 2nd ed.; Princeton University Press, Doubleday Anchor Books: New York, NY, USA, 1945.
17. Diaz, L.M.; Hernandez, C.M.; Ortiz, A.V.; Gaytan-Lugo, L.S. Tinkercad and Codeblocks in a Summer Course: An Attempt to Explain Observed Engagement and Enthusiasm. In Proceedings of the 2019 IEEE Blocks and Beyond Workshop, Memphis, TN, USA, 18 October 2019; pp. 43–47. [CrossRef]
18. Gao, M.; Johnson, J.; Reed, D.; Sheller, C.; Turbak, F. Using app inventor in introductory CS courses. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education, Kansas City, MO, USA, 4–7 March 2015; pp. 346–347. [CrossRef]
19. Trilles, S.; Monfort-Muriach, A.; Cueto-Rubio, E.; Granell, C.; Juan, P. Sucre4Stem: Internet of things in classrooms», en 15th International Conference of Technology, Learning and Teaching of Electronics. In Proceedings of the XV Congreso Tecnología Aprendizaje y Enseñanza de la Electrónica, Teruel, España, 29 June–1 July 2022; pp. 6–9. [CrossRef]
20. Trilles, S.; Monfort-Muriach, A.; Gomez-Cambronero, A.; Granell, C. Sucre4Stem: Collaborative Projects Based on IoT Devices for Students in Secondary and Pre-University Education. *Rev. Iberoam. Tecnol. Prendiz.* **2022**, *17*, 150–159. [CrossRef]
21. Alegre, F.; Moreno, J.; Dawson, T.; Tanjong, E.E.; Kirshner, D.H. Computational Thinking for STEM Teacher Leadership Training at Louisiana State University. In Proceedings of the 2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology, Portland, OR, USA, 10–11 March 2020; Volume 1, pp. 1–2. [CrossRef]
22. Harangus, K.; Kátai, Z. Computational Thinking in Secondary and Higher Education. *Procedia Manuf.* **2020**, *46*, 615–622. [CrossRef]
23. Da Silva, T.J.C.; De Melo, B.; Tedesco, Y.P. The creative process in the development of computational thinking in higher education. *CSEDU* **2020**, *1*, 215–226.
24. Wu, J.; Wang, Y.; Kong, H.; Zhu, L. How to Cultivate Computational Thinking-Enabled Engineers: A Case Study on the Robotics Class of Zhejiang University. In Proceedings of the 2019 ASEE Annual Conference & Exposition, Tampa, FL, USA, 15 June–19 October 2019. [CrossRef]
25. Delgado, R.; Yacchirema, S.; Coral, F.; Celi, J. Platform for university teaching applying Computational Thinking at the Armed Forces University ESPE. In Proceedings of the 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 19–22 June 2019. [CrossRef]
26. Wilson, J.W.; Fernandez, M.L.; Hadaway, Y.N. *Mathematical Problem Solving*; The University of Georgia: Athens, GA, USA, 1993. Available online: http://jwilson.coe.uga.edu/emt725/PSsyn/Pssyn.html (accessed on 19 January 2023).
27. Fang, A.-D.; Chen, G.-L.; Cai, Z.-R.; Cui, L.; Harn, L. Research on Blending Learning Flipped Class Model in Colleges and Universities Based on Computational Thinking—"Database Principles" for Example. *Eurasia J. Math. Sci. Technol. Educ.* **2017**, *13*, 5747–5755. [CrossRef]
28. Weese, J.L. Mixed Methods for the Assessment and Incorporation of Computational Thinking in K-12 and Higher Education. In Proceedings of the 2016 ACM Conference on International Computing Education Research, Melbourne, VIC, Australia, 8–19 September 2016; pp. 279–280. [CrossRef]
29. Villalba-Condori, K.O.; Cuba-Sayco, S.E.C.; Chávez, E.P.G.; Deco, C.; Bender, C. Approaches of Learning and Computational Thinking in Students that get into the Computer Sciences Career. In Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 24 October 2018; pp. 36–40. [CrossRef]
30. Kules, B. Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes. *Proc. Assoc. Inf. Sci. Technol.* **2016**, *53*, 1–6. [CrossRef]

31. Selby, C.C. Relationships: Computational thinking, Pedagogy of programming, And bloom's taxonomy. *ACM Int. Conf. Proceeding Ser.* **2015**, *64*, 80–87. [CrossRef]
32. Bordignon, F.; Iglesias, Y.A. *Introducción Al Pensamiento Computacional*; EDUCAR S.E: Buenos Aires, Argentina, 2018.
33. Mayer, R.E.; Wittrock, Y.M.C. Problem solving. In *Handbook of Educational Psychology*; Alexander, P.A., Winne, P.H., Eds.; Macmillian: New York, NY, USA, 2006.
34. Mayoral, S.; Roca, M.; Timoneda, C.; Serra, Y.M. Mejora de la capacidad de planificación cognitiva del alumnado de primer curso de Educación Secundaria Obligatoria. *Aula Abiert* **2015**, *43*, 9–17. [CrossRef]
35. Kwon, K.; Ottenbreit-Leftwich, A.T.; Brush, T.A.; Jeon, M.; Yan, G. Integration of problem-based learning in elementary computer science education: Effects on computational thinking and attitudes. *Educ. Technol. Res. Dev.* **2021**, *69*, 2761–2787. [CrossRef]
36. Gouws, L.; Bradshaw, K.; Wentworth, P. First year student performance in a test for computational thinking. In Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference, New York, NY, USA, 7–9 October 2013; pp. 271–277. [CrossRef]
37. Iwata, M.; Laru, J.; Mäkitalo, Y.K. Designing problem-based learning to develop computational thinking in the context of K-12 maker education. *CEUR Workshop Proc.* **2020**, *2755*, 103–106.
38. Svarre, T.; Burri, Y.S. Problem based learning: A facilitator of computational thinking. *Proc. Eur. Conf. E Learn. ECEL* **2019**, *2019*, 260–267. [CrossRef]
39. Smith, O.S.; Dindler, R.C. From Computational Thinking to Computational Empowerment: A 21 st Century PD Agenda. In Proceedings of the 15th Participatory Design Conference, New York, NY, USA, 20–24 August 2018. [CrossRef]
40. Ziman, D. Solución de problemas en la investigación tecnológica. *J. Creat. Behav.* **1974**, *2*, 93–102.
41. Mcpherson, J.H. The People, The Problems and The Problem Solving Methods*. *J. Creative Behav.* **1968**, *2*, 103–110. [CrossRef]
42. Duderstadt, J.; Knoll, G.; Springer, Y.G. *Principles of Engineering*; Ilustrada: New York, NY, USA, 1982.
43. Marais, C.; Bradshaw, Y.K. Problem-solving ability of first year CS students: A case study and intervention. In Proceedings of the 44th Conference of the Southern African Computers Lecturers' Association, Johannesburg, South Africa, 1–3 July 2015; pp. 54–160.
44. Sáez, J.M.; Buceta, R.; De Lara, S. Introducing robotics and block programming in elementary education. *RIED Rev. Iberoam. Educ. A Distancia* **2021**, *24*, 95–113. [CrossRef]
45. Román-González, M. Pensamiento Computacional: Un Constructo Que Llega a La Madurez, Aula Magna 2.0, Revistas Científicas de Educación en Red, 2022. Available online: https://cuedespyd.hypotheses.org/11109 (accessed on 19 January 2023).
46. Rabiee, M.; Tjoa, Y.M. From Abstraction to Implementation: Can Computational Thinking Improve Complex Real-World Problem Solving? A Computational Thinking-Based Approach to the SDGs Maryam. In Proceedings of the Information and Communication Technologies for Development: 14th IFIP WG 9.4 International Conference on Social Implications of Computers in Developing Countries, ICT4D 2017, Yogyakarta, Indonesia, 22–24 May 2017; pp. 104–116.
47. Wing, J.M. Computational thinking and thinking about computing. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2008**, *366*, 3717–3725. [CrossRef]
48. Barr, V.; Stephenson, C. Bringing computational thinking to K-12. *ACM Inroads* **2011**, *2*, 48–54. [CrossRef]
49. Grover, S. Robotics and engineering for middle and high school students to develop computational thinking. In Proceedings of the Annual Meeting of the American Educational Research Association, New Orleans, LA, USA, 7–11 April 2011; Volume 650, pp. 1–15, [En línea].
50. Del Rey, Y.A.R.; Cambinda, I.N.C.; Deco, C.; Bender, C.; Avello-Martínez, R.; Villalba-Condori, K.O. Developing computational thinking with a module of solved problems. *Comput. Appl. Eng. Educ.* **2020**, *29*, 506–516. [CrossRef]
51. Voogt, J.; Fisser, P.; Good, J.; Mishra, P.; Yadav, A. Computational thinking in compulsory education: Towards an agenda for research and practice. *Educ. Inf. Technol.* **2015**, *20*, 715–728. [CrossRef]
52. Kale, U.; Yuan, J. Still a New Kid on the Block? Computational Thinking as Problem Solving in Code.org. *J. Educ. Comput. Res.* **2020**, *59*, 620–644. [CrossRef]
53. Jeng, H.-L.; Liu, L.-W.; Chen, C.-N. Developing a Procedural Problem-solving-based Framework of Computational Thinking Components. In Proceedings of the 2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI), Toyama, Japan, 7–11 July 2019; pp. 272–277. [CrossRef]
54. Maharani, S.; Kholid, M.N.; NicoPradana, L.; Nusantara, Y.T. Problem Solving in the Context of. *Infinty J. Math. Educ.* **2019**, *8*, 109–116.
55. Pedaste, M.; Palts, T.; Kori, K.; Sormus, M.; Leijen, A. Complex Problem Solving as a Construct of Inquiry, Computational Thinking and Mathematical Problem Solving. In Proceedings of the 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT), Maceio, Brazil, 15–18 July 2019; Volume 2161–2377X, pp. 227–231. [CrossRef]
56. Rojas-López, A.; García-Peñalvo, F.J. Increase of confidence for the solution of problems in preuniversity students through Computational Thinking. In Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 24–26 October 2018. [CrossRef]
57. Talent Search, Computer Olympiad—Computer Olympiad South Africa, 2015. Available online: http://olympiad.org.za/ (accessed on 5 September 2020).
58. UK Bebras Computational Thinking Challenge, What Is Bebras. 2015. Available online: https://www.bebras.org/ (accessed on 5 September 2020).

59. Román-Gonzalez, M.; Pérez-González, J.C.; Jiménez-Fernández, C. Test de Pensamiento Computacional: Diseño y psicometría general [Computational Thinking Test: Design & general psychometry]. In Proceedings of the III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015), 14 October 2015. Available online: https://innovacioneducativa.wordpress.com/2015/01/27/iii-congreso-internacional-sobre-aprendizaje-innovacion-y-competitividad-cinaic-2015/ (accessed on 19 January 2023). [CrossRef]
60. Román-González, M.; Pérez-González, J.-C.; Jiménez-Fernández, C. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Comput. Hum. Behav.* **2016**, *72*, 678–691. [CrossRef]
61. Román-González, M.; Pérez-González, J.-C.; Moreno-León, J.; Robles, G. Can computational talent be detected? Predictive validity of the Computational Thinking Test. *Int. J. Child Comput. Interact.* **2018**, *18*, 47–58. [CrossRef]
62. Ruipérez, B.O.; Brouard, M.A. Evaluar el Pensamiento Computacional mediante Resolución de Problemas: Validación de un Instrumento de Evaluación. *Rev. Iberoam. Evaluación Educ.* **2021**, *14*, 153–171. [CrossRef]
63. Viale, P.; Deco, Y.C. Introduciendo conocimientos sobre el Pensamiento Computacional en los primeros años de las carreras de ciencia, tecnología, ingeniería y matemáticas. *Energeia* **2019**, *16*, 73–78.
64. Puhlmann, C. Desenvolvimento do Pensamento Computacional Através De Atividades Desplugadas Na Educação Básica. Doctoral Thesis, Universidade Federal do Rio Grande do Sul, Farroupilha, Brazil, 2017.
65. Liu, J.; Wang, L. Computational Thinking in Discrete Mathematics. In Proceedings of the 2010 Second International Workshop on Education Technology and Computer Science, Wuhan, China, 6–7 March 2010; Volume 1, pp. 413–416. [CrossRef]