Mathematical Modelling of Engineering Problems
Vol. 12, No. 1, January, 2025, pp. 241-252
Journal homepage: http://iieta.org/journals/mmep

Arduino and mBlock Projects to Enhance Computational Thinking in First-Year Engineering Students

Ronald Paucar Curasma*, Freddy Toribio Huayta-Meza, Nolan Jara Jara, Omar Felix Illesca-Cangalaya, Luis Enrique Porras-Ccancce, Angel Junker Romero-Gabino, Zenón Manuel López-Robles, Rosa Quispe Llamoca

Department of General Studies, National Autonomous University of Tayacaja Daniel Hernandez Morillo, Pampas 09156, Peru

Corresponding Author Email: rpaucarc@unat.edu.pe

https://doi.org/10.18280/mmep.120125

Received: 29 October 2024
Revised: 20 December 2024
Accepted: 26 December 2024
Available online: 25 January 2025

Keywords:
computational thinking, mBlock platform, programming with Arduino, engineering students, problem-solving

ABSTRACT

Computational thinking skills are essential in education, especially for students at the beginning of their engineering studies. Therefore, it is crucial to integrate these skills with other techniques and methods that can further enhance computational thinking in engineering students. This research aimed to strengthen computational thinking skills by developing technological projects that address the specific needs of the local context for industrial and systems engineering students. To achieve this, students engaged in classroom activities utilizing technological resources, where they developed algorithms, programmed Arduino boards, configured sensors, debugged programs, and created applications using the mBlock platform. The research methodology employed a quantitative approach with a quasi-experimental post-test design and intentional non-probabilistic sampling proportional to the number of students. Results indicated that working on technological projects equally motivated and sensitized both female and male students to engage in technological activities successfully, thereby strengthening computational thinking skills in students across both fields. Furthermore, the results revealed that systems engineering students demonstrated greater improvement in computational thinking skills compared to industrial engineering students. This difference can be attributed to the systems engineering curriculum being more oriented toward the use and development of technology, while industrial engineering focuses more on management.

1. INTRODUCTION

In Peru and other Latin American countries, significant inequalities in education persist, particularly in the teaching of STEM (Science, Technology, Engineering, and Mathematics) disciplines, where students often achieve low competency levels [1]. According to the World Bank, approximately 40% of students in the region fail to reach satisfactory levels in mathematics and science, as evidenced by PISA results [2]. These challenges limit the development of cognitive skills necessary for addressing the demands of the 21st century, where STEM knowledge is essential for technological innovation and societal advancement [3, 4].

Computational thinking (CT) has emerged as a critical skill in STEM education, equipping students with problem-solving abilities, algorithmic reasoning, and persistence [5]. It enables them to deconstruct problems, identify patterns, and propose algorithmic solutions, fostering creativity and innovation in real-world contexts [6, 7]. While developed countries have successfully integrated CT into their curricula, its application in developing regions remains underexplored.

This research aims to address this gap by investigating how technological projects, designed to tackle local challenges, can strengthen computational thinking skills in first-year engineering students. Specifically, this study seeks to answer the following questions: How do technological projects enhance computational thinking skills in engineering students? And are there significant differences in skill development between students from different engineering disciplines?

By focusing on local contexts and utilizing tools such as Arduino boards and the mBlock platform, this study explores innovative methods to foster critical skills and contribute to educational equity in STEM disciplines.

2. RELATED WORK

The existing literature highlights the importance of computational thinking as an essential skill in 21st-century education, particularly for engineering students who face an increasingly technological environment. Various studies have explored the integration of computational thinking into education, emphasizing the effectiveness of methods such as project-based programming, the use of interactive platforms like mBlock, and the configuration of technological devices like Arduino. However, these investigations often address these practices in a general manner, without considering the specific needs of local contexts or the curricular differences between engineering disciplines. Furthermore, there is a lack of studies examining the implications of gender in the motivation and performance of students in technological activities. This research seeks to fill these gaps by focusing on strengthening computational thinking through technological projects designed for local contexts, while also analyzing how these practices impact students in industrial and systems engineering differently. In doing so, it contributes to understanding how technology-based pedagogical approaches can be adapted to maximize their effectiveness in various engineering fields.

2.1 Computational thinking and problem-solving in higher education

In the various studies carried out by the scientific community, they have agreed on the benefits that computational thinking provides in the educational sector; both in regular basic education and in higher education. In higher education, the ability to abstract and algorithmic thinking has been considered important; and they have been applied in the disciplines of communication and mathematics to solve complex problems [8, 9]; also, Wilson et al. [10] pointed out that the development of algorithms and coding contribute to the understanding and development of mathematical problems, logic, reading comprehension and other disciplines that have been characterized by their abstraction and complexity; to develop these activities involves using computers, microcontrollers and computer programs that help the student solve mathematical problems and other related disciplines. Shyamala et al. [3] points out that the use of tools based on block programming and hardware in different activities generates greater motivation in students; likewise, teamwork and gender equality have been strengthened in these activities; during the execution of the activity; also, the steps of the problem solving were practiced, and in each step the skills of decomposition, abstraction, algorithmic development, pattern recognition, among others, are applied [11]. Kules [12] in one of his investigations points out critical thinking as an important component of reasoning and argumentation before planning activities through computational thinking skills and problem-solving.

Various authors suggest a range of computational thinking skills, such as abstraction, recursion, interaction, patterns, synectics, creativity, and simulation. However, most agree on five core skills of computational thinking: decomposition, abstraction, algorithmic design, pattern recognition, and evaluation [13, 14]. Abstraction focuses on identifying key characteristics; decomposition breaks down complex problems into smaller, more manageable parts; generalization involves recognizing patterns across different contexts; algorithmic design involves proposing a solution in a structured, step-by-step manner and executing the activity; and evaluation involves reviewing the solution and assessing the efficiency of the resources used.

In the initial definition of computational thinking, Wing [15] described it as "a process that involves problem-solving, system design, and understanding human behavior using the core concepts of computer science". From this first definition, various authors also conceptualized or pointed out that the main reason for "computational thinking is problem-solving, and problem-solving is composed of a set of phases that are used to obtain the solution; also, researchers consider it important to add other thoughts to computational thinking; such as, critical thinking, which takes on greater value when it is applied in a phase before computational thinking. This set of thoughts has only one objective, to strengthen the different skills; for example, communication when disseminating research results and teamwork to achieve common objectives; thus, as the search, analysis, and synthesis of information, and continuous learning [16, 17]. Starting from the definition of computational thinking that is linked to problem-solving, there are two fundamental components, which are: a set of skills required for problem-solving and an approach to using these skills in problem-solving [18].

Several researchers have explored the connection between computational thinking skills and the phases of problem-solving, yielding positive results in enhancing computational thinking among students in both school and higher education [19, 20]. They have defined that the five key computational thinking skills align with the four phases of problem-solving [15]. Specifically, the "understand the problem" phase corresponds to the abstraction skill; the "prepare the plan" phase is associated with decomposition and generalization skills; the "execute the plan" phase relates to algorithmic design; and the "review the solution" phase corresponds to the evaluation skill [21]. Table 1 illustrates the relationship between computational thinking skills and the problem-solving phases.

Table 1. Relationship between problem-solving phases and computational thinking skills [21]

Problem-Solving Phases	Computational Thinking Skills				
	Abstraction	Decomposition	Generalization	Algorithmic design	Evaluation
Understand the problem	X				
Prepare the plan		X	X		
Execute the plan				X	
Review the solution					X

2.2 Technological resources and computational thinking

Currently, various activities utilize technological resources, such as hardware and software, to enhance computational thinking through classroom educational activities. Therefore, it is essential to train students in computational thinking and problem-solving skills using both digital tools and non-digital (unplugged) methods [22]. Research has demonstrated that lessons in computational thinking can improve students' response inhibition, planning, and coding abilities. As these skills become increasingly important in the fast-evolving 21st century, education in computational thinking holds great potential for better preparing students for future careers and active citizenship [23].

Various technological products have been developed for teaching science and technology from elementary to higher education, aiming to introduce computational thinking and programming through the use of physical components and social interaction. These tools motivate students to learn coding and foster greater interest in STEM disciplines. Most of these platforms support activities such as block-based programming, assembling sensors and actuators on microcontrollers, network connectivity, and remote data sharing [24, 25]. Technological resources help students understand the foundation of science and how it has been used technically in the world; for example, robotics allow students to understand the fundamentals of basic programming and develop their computational thinking with STEM education in an attractive way [26, 27].

Researchers recommend adding technological resources in various courses to help better understand the topics and implicitly strengthen computational thinking skills: abstraction, decomposition, generalization, algorithm, design, data, and representation; Common technological resources are educational robotics, code-based (C++, Python, Arduino, etc.) and block-based (Scratch, Alice, mBlock, etc.) programming languages [28, 29]; also, sensors (distance, temperature, humidity, etc.), actuators (motor, display, LCD, etc.) and LEDs are added, which allow the development of activities that involve engineering processes [30, 31]. Therefore, schools and universities that incorporate STEM into their curricula are already fostering the development of computational thinking in various aspects throughout the teaching process. In other words, the integration of STEM disciplines naturally encourages the cultivation of computational thinking skills [32].

2.3 Technological projects in engineering classrooms

The experiences and learning of university students play a crucial role in fostering a shift towards a culture of sustainability. Higher education should, therefore, contribute to the development of sustainability competencies, such as critical and creative thinking, problem-solving, the ability to take action, collaboration, and systemic thinking. These skills help shape potential change agents who can build more sustainable societies, particularly within their own environments [33].

Several international studies, including those by prominent organizations like the OECD, highlight the need for a significant shift in teaching methods within university classrooms. Achieving competency-based learning (encompassing knowledge, abilities, skills, and attitudes across various contexts) and the holistic development of individuals, as sought by our current educational system, requires fostering motivation and creating meaningful, transferable, functional, and lasting learning experiences [34]. To this end, it is recommended to adopt active and contextualized methodologies that encourage student participation, involvement, and the application of knowledge in real-world situations [35].

Combining problem-solving methods and technology involves the creation of realistic projects that develop, simultaneously and integrate, the curricula of scientific-technological subjects [36]. Thus, the development of a final product or prototype generates a process of complex open tasks that involve research, solving authentic problems, and designing strategies and/or experiments; these strategies applied in the classroom increase the motivation and confidence of students, improving their attitudes towards learning and reducing absenteeism, above all generating greater identity and sensitivity to solve the real needs of society and the community or city where they live [37].

The development of technological projects in the classroom stems from the urgent need to transform university education and align it with modern advancements. This strategy incorporates elements that help boost motivation, enhance academic performance, foster entrepreneurship and creativity, and strengthen the training of ethical, responsible, and effective professionals. These efforts aim to prepare students for success in the era of knowledge and complexity, contributing to greater social equity [38]. This could imply that environments that motivate innovation from an industry perspective should be introduced into teaching and that consider solutions to problems of interest in society involving the economy, ecology, and equity of all communities [39]. Likewise, in engineering education, strategies have been incorporated to exploit creativity, which corresponds to the moment of the “big event” that leads to the occurrence of great creative leaps, fundamental in innovation and education of the new century, without forgetting the quality of teaching as a motivating and renewing entity of teaching methods and instructional materials [40].

3. METHODOLOGY

3.1 Research design

The methodology employed in this research follows a mixed qualitative and quantitative approach, utilizing a post-test quasi-experimental design. Participants were selected through intentional non-probabilistic sampling proportional to the number of students in each professional program involved. Two groups were formed: one composed of 37 Industrial Engineering students from the National Autonomous University of Tayacaja Daniel Hernández Morillo (UNAT), and another of 49 Systems Engineering students from the National University of Huancavelica (UNH). Both groups consisted of first-year students, aged between 16 and 17, enrolled in the Information Management course during the 2022-II academic period. The intervention consisted of activities specifically designed to strengthen computational thinking skills through the development of technological projects addressing contextual and local problems. These activities included the design and programming of algorithms, configuration of sensors and actuators, and the creation of applications using the mBlock platform. The assessment of computational thinking skills was conducted using the Computational Thinking Test (CTT) by Román-González [41], previously validated in terms of criterion and convergence. Results were analyzed to identify significant differences in skill development between students of both programs, considering curricular and contextual differences that could influence the outcomes.

The CTT has been validated for criteria and convergence [42, 43] by 20 experts who reviewed and evaluated the instrument. It includes 28 items, each designed and categorized into five dimensions: computational concept addressed, item environment interface, style of response alternatives, presence or absence of nesting, and the required task. The instrument is tailored to the cognitive level of the students, who are predominantly 16 to 17 years old and in their first year of university studies. Table 2 displays the items from the Román-González test related to the skills of abstraction, decomposition, generalization, algorithmic design, and evaluation [44, 45]. A correct answer is scored as one point, while an incorrect answer receives zero points. Therefore, the maximum possible scores are as follows: 16 points for abstraction, 16 points for decomposition, 19 points for generalization, 28 points for algorithmic design, and 14 points for evaluation.

Table 2. Items of the test to assess computational thinking skills

Computational Thinking Skills	Number of Items	Marcos Román-González Test Items
Abstraction	16	1 al 3, 7, 11 al 15, 21 al 23 y 25 al 28
Decomposition	16	4 al 7, 10 al 13, 15, 21 al 23 y 25 al 28
Generalization	19	4 al 6, 8 al 12, 14, 15, 17, 18, 20, 22, 23 y 25 al 28
Algorithmic design	28	1 al 28
Evaluation	14	3, 7, 10, 11, 15, 16, 19, 20, 23 al 28

3.2 Proposal for technological projects

Figure 1 illustrates the framework for developing technological projects in the classroom aimed at enhancing computational thinking in engineering students, particularly in relation to ICT. This proposal draws on the technological principles of constructionism [46] and follows Pólya's problem-solving method [47], which consists of four phases: understanding the problem, developing the plan, executing the plan, and reviewing the solution.

The scheme represents the direction and sequence of execution of the technological project; where, in the "understanding the problem" phase, activities are carried out to strengthen the abstraction skill; In the "make the plan" phase, activities are carried out to strengthen decomposition and generalization skills; in the "execute the plan" phase, activities are carried out to strengthen the algorithmic design skill; and in the "review the solution" phase, activities are carried out to strengthen the evaluation skill.

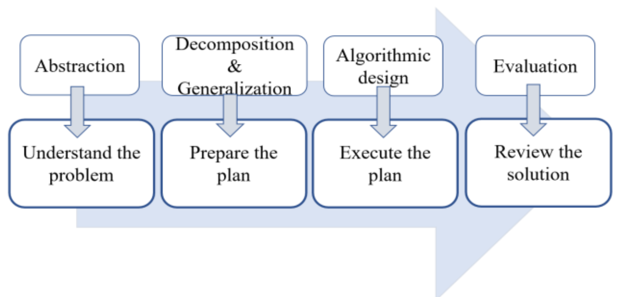


Figure 1. Execution scheme for technological projects

Given the nature and vocation of engineering students related to ICT, various technological projects have been proposed that focus on solving problems within the community or context where the students reside. These types of projects have been effective in motivating students of all genders to engage in project execution within the classroom, as they tackle real-world issues that are directly relevant to people's lives, addressing the needs of their city or region and contributing to community well-being and protection [48, 49].

Table 3 presents the technological projects aimed at offering alternative solutions from the classroom, addressing the specific needs of the province of Tayacaja in the Huancavelica region of Peru. These projects aimed to foster awareness and social sensitivity among first-year university students, from an educational perspective, resulting in the creation of prototypes or models that represent potential solutions.

The projects covered a range of topics, including greenhouse cultivation monitoring, water quality monitoring, physical security, solid waste management, student health monitoring, animal protection from predators, educational support tools, and automatic crop irrigation. Throughout the execution of these projects, students utilized microcontrollers, sensors, actuators, and programming with the mBlock platform.

The execution of technological projects was carried out in the Information Management course during a 16-week academic semester in the 2022-II period, with 4 hours of weekly sessions. The framework of Pólya's problem-solving method structured the activities into four distinct phases: "understand the problem" (5 weeks), "make the plan" (3 weeks), "execute the plan" (6 weeks), and "review the solution" (2 weeks). In the "understand the problem" phase, students identified key elements of the challenges presented, such as the scope and context of the problem, and represented these elements through mind maps and problem statements. In the "make the plan" phase, they developed detailed strategies to address the identified problems, including the design of circuits, selection of hardware components (e.g., Arduino boards and sensors), and preliminary algorithm outlines. During the "execute the plan" phase, students implemented their designs, programmed the hardware using the mBlock platform, and debugged their systems to ensure functionality. Finally, the "review the solution" phase focused on iterative testing and evaluation, where students critically assessed the performance of their prototypes, optimized both hardware and software components, and presented their findings to peers and instructors for feedback.

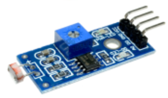
Projects were carried out in groups of 3 to 5 students to encourage collaboration and teamwork, while also ensuring that every participant had hands-on experience with the tools and methods used. Students were tasked with documenting their progress and sharing code snippets for review and debugging during class sessions. Each group also showcased their final prototype through a presentation that included visual demonstrations and a detailed explanation of the problem-solving process. This collaborative and iterative approach ensured that the technological projects not only aligned with the course objectives but also provided a comprehensive, practical learning experience.

4. RESULTS

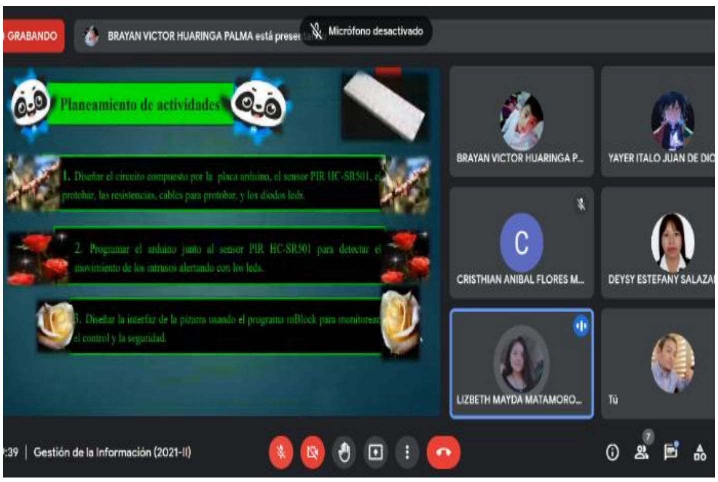
4.1 Execution of technological projects to strengthen computational thinking

To illustrate the execution details of the technological project and the enhancement of computational thinking skills, the project titled "Implementation of a Water Level Monitoring Prototype in the Viñas Reservoir in the City of Pampas, Tayacaja Province" was selected. Figure 2 displays the results of the project's execution, aligned with the problem-solving phases. These phases include understanding the problem, developing the plan, executing the plan, and reviewing the solution, showcasing how each phase contributed to the successful development of the prototype.

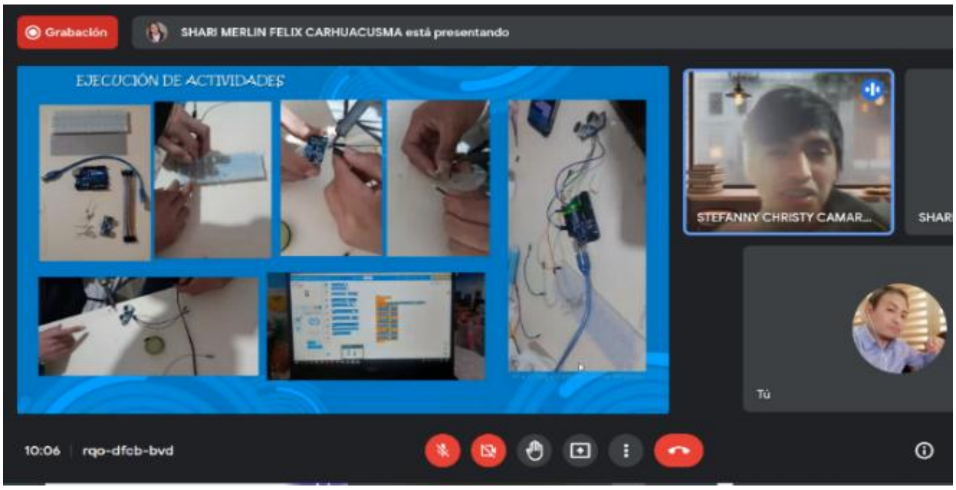
Table 3. Proposal of technological projects and electronic devices

Technological Projects	Sensors		Career
Monitoring of vegetable production in greenhouses, Pampas, Tayacaja province, Huancavelica region	Sensor DHT11		
Implementation of a water level monitoring prototype in the Viñas reservoir in the city of Pampas, Tayacaja province	Ultrasonic Sensor HC-SR04		
Implementation of a control and security system in a market	Infrared Sensor PIR HC-SR501		Industrial engineer
Prototype of automatic on/off for public lighting, Huancavelica region	LDR Sensor (Light Dependent Resistor)		
Solid waste monitoring in the Huancavelica region.	Ultrasonic Sensor HC-SR04		
Automatic Distance Detection Alarm Prototype for Vehicles in Pampas.	Ultrasonic Sensor HC-SR04		
Sensor-Equipped Smart Cane for Visually Impaired Individuals in Pampas.	Ultrasonic Sensor HC-SR04		
Monitoring of temperature and humidity with an automated irrigation system in vegetable production in the city of Pampas, Tayacaja, Huancavelica.	DHT11 Sensor		
The monitoring system for animal safety in the Huancavelica region.	LDR Sensor (Light Dependent Resistor)		Systems engineer
Monitoring and control of humidity and temperature in the greenhouse in the Huancavelica region.	Sensor DHT11		
Home automation for the security and tranquility of homes in the city of Pampas, Tayacaja, Huancavelica.	Infrared Sensor PIR HC-SR501		
Implementation of a biosafety prototype against COVID-19 in the professional school of systems engineering.	Infrared Sensor PIR HC-SR501		
Monitoring of solid waste in homes in the city of Pampas, Tayacaja, Huancavelica.	Ultrasonic Sensor HC-SR04		

(a)



(b)



(c)

(d)

Figure 2. Execution of the technological project: (a) activities of the understanding the problem phase; (b) activities of the plan development phase; (c) activities of the plan execution phase; (d) activities of the solution review phase

Table 4 outlines the activities undertaken to strengthen the five computational thinking skills in alignment with the problem-solving phases. During the "understand the problem" phase, abstraction skills were enhanced as students focused on identifying the key elements of the problem. In the "make the plan" phase, both decomposition and generalization skills were developed, enabling students to break down the problem into manageable parts and recognize patterns. The "execute the plan" phase focused on algorithmic design, where students created step-by-step solutions. Finally, in the "review the solution" phase, evaluation skills were strengthened as students assessed the effectiveness and efficiency of their solutions.

In the development of technological projects, Arduino boards and the mBlock platform were used as the main tools to strengthen computational thinking skills. Arduino boards were employed for implementing electronic circuits that integrated sensors and actuators, allowing students to interact directly with real hardware and understand basic concepts of electronics and programming. For instance, in the water level monitoring project, students configured the HC-SR04 distance sensor to measure water levels, programmed the Arduino board to process the collected data, and activated an alert system based on predefined parameters. The mBlock platform, based on block programming, facilitated the creation of algorithms and visual applications, enabling students to simulate and control the designed systems. This tool was also key in introducing advanced programming logic concepts, such as the use of conditional structures, loops, and functions. These activities not only helped develop skills such as abstraction, decomposition, and algorithmic design but also fostered creativity and problem-solving by providing a practical and visually intuitive environment.

4.2 Computational thinking assessment

The following sections present the results of the statistical analysis related to the evaluation of the enhancement of computational thinking skills—specifically, abstraction, decomposition, generalization, algorithmic design, and evaluation—through the development of technological projects in the classroom for industrial and systems engineering students. These results provide insights into the effectiveness of using technological projects as a method for strengthening these critical skills within the educational context.

Table 4. Activities developed to strengthen computational thinking skills

Problem-Solving Phases	Computational Thinking Skills
Understand the problem	Abstraction The teacher developed exercises on the skill of abstraction [14] and provided instructions to elaborate on the problematic situation of the project assigned to the group. The students consulted the scientific databases: Google Academic, Dialnet, Scielo, and ALICIA; After reviewing the selected information regarding the problematic situation of the Viñas reservoir overflow, they represented the causes and effects on a mental map; thus, managing to abstract the complex problem; They also prepared a paragraph with the most important points of the problem to be solved; each paragraph was referenced.
Prepare the plan	**Decomposition** The teacher developed exercises on the decomposition skill [14], and the students created or decomposed a list of activities: Understand the important points of the causes of the problem of the overflow of the reservoir. Design the circuit composed of the Arduino board and the distance sensor. Program the Arduino to read the distance of the water level. Develop the graphical interface for monitoring water level monitoring through the mBlock platform. Integrate the sensors and graphic interface into a model that represents the overflow of the vineyard reservoir. **Generalization** The teacher developed exercises on generalization skills, such as pattern recognition [14]; he also gave instructions on ways to search for background information related to the project "Implementation of a water level monitoring prototype in the Viñas reservoir in the city of Pampas in the province of Tayacaja." The students consulted the scientific databases: Google Academic, Dialnet, Scielo, and ALICIA; where they identified the components or activities or patterns that they would use in their projects; for example, the use of sensors, monitoring parameters, circuit design, etc.
Execute the plan	**Algorithmic design** The teacher developed exercises on developing algorithms and programs in mBlock [14]. The students executed the activities planned in the previous phase, and implemented the circuit composed of the Arduino board and distance sensor; They also developed a monitoring application through mBlock; where, they represented the reservoir of vineyards, with objects that represent rain, sun, cloud, water and the measured values of the distance from the water level; For the development of the application they used control blocks (yes, else, repeat, etc.) to make decisions when the water level rises to the surface; The water level reading was carried out with the HC-SR04 distance sensor; During the development of these activities, the programming logic and programming sequence were observed; In this way they developed the skill of algorithmic design.
Review the solution	**Evaluation** The teacher developed exercises on the method and techniques for evaluating the results [14]. At the end of the project, the students obtained a prototype or product; the integration of the hardware components (circuits, microcontrollers, and sensors) and software (graphical user interface in mBlock or application) was carried out through a model, where they represented a physical scenario simulating the monitoring of the Viñas reservoir in the city of Pampas of the province of Tayacaja. They verified the operation by component and its integrity; When difficulties arose in operation during the tests, they proceeded to evaluate the prototype starting from the hardware component, to find the problem and then continue with the program codes until the error was found. Also, the evaluation was carried out during the progress of the execution of the technological project in each problem-solving phase, with the help of the classroom teacher; In this way, the students perfected the evaluation skill.

Table 5. Sample means and standard deviation

Computational Thinking Skills	Career	N	Average	Standard Deviation
Abstraction	Industrial engineering	37	51.8581	19.36855
	Systems engineering	49	61.6071	18.22172
Decomposition	Industrial engineering	37	49.6622	19.81610
	Systems engineering	49	60.8418	17.10565
Generalization	Industrial engineering	37	50.3549	20.13821
	Systems engineering	49	63.6945	16.51258
Algorithm design	Industrial engineering	37	51.5449	18.77726
	Systems engineering	49	66.6910	16.78884
Evaluation	Industrial engineering	37	48.0689	19.96680
	Systems engineering	49	63.2645	18.20972

Table 5 shows the results of sample means and standard deviation of computational thinking skills for students studying industrial engineering and systems engineering.

To decide the type of parametric or non-parametric statistical test, normality analysis with Shapiro Wilk is used due to having less than 50 students in the sample in each career. Table 6 shows the normality tests with Shapiro-Wilk.

According to Table 6 the results of normality tests, it is concluded that all those with a P value greater than 0.05 (significance) have a normal distribution (a parametric test is applied). In those who do not have this condition, a non-parametric test is used Mann-Witney U test for independent groups.

Table 7 shows the t Student test for independent groups and determines whether there are differences between systems engineering and industrial engineering students for abstraction and decomposition.

From the Table 7 results of the previous table in abstraction and decomposition skills there are significant differences ($p<0.05$), so it can be stated that systems engineering students developed these skills better than industrial engineering students.

Table 6. Normality tests with Shapiro-wilk

Computational Thinking Skills	Career	Shapiro-Wilk		
		Statistical	**gl**	**p-value**
Abstraction	Industrial engineering	0.969	37	0.385
	Systems engineering	0.960	49	0.093
Decomposition	Industrial engineering	0.957	37	0.163
	Systems engineering	0.969	49	0.217
Generalization	Industrial engineering	0.963	37	0.256
	Systems engineering	0.937	49	0.011
Algorithm design	Industrial engineering	0.954	37	0.131
	Systems engineering	0.932	49	0.007
Evaluation	Industrial engineering	0.939	37	0.044
	Systems engineering	0.928	49	0.005

Table 7. t student test for independent samples

Independent Samples Test						
		Levene's test for equality of variances		t test for equality of means		
		F	p-value	t	gl	p-value (bilateral)
Abstraction	Equal variances	0.13	0.72	-2.4	84	0.019
Decomposition	Equal variances	1.37	0.24	-2.8	84	0.006

Table 8. Mann-Whitney U test for non-parametric

Test Statistics[a]			
	Generalization	Algorithm design	Evaluation
Mann-Whitney U test	552.500	486.500	516.000
Wilcoxon W	1255.500	1189.500	1219.000
Z	-3.099	-3.673	-3.426
Pvalue. asymptotic (bilateral)	0.002	0.000	0.001

a. Grouping variable: Career

Table 8 shows the Mann-Whitney U test to determine differences in generalization, algorithmic design, and evaluation skills.

From Table 8, it is evident that there are significant differences (P-value < 0.05) in the skills of generalization, algorithmic design, and evaluation. The data indicate that systems engineering students demonstrated superior development in these skills compared to industrial engineering students.

5. DISCUSSION

The implementation of technological projects utilizing resources such as Arduino boards, sensors, actuators, and the mBlock programming environment proved to be an effective approach for students beginning their engineering studies. This aligns with previous studies that have applied similar educational strategies to strengthen computational thinking (CT) skills and problem-solving abilities [50]. Designing technological projects to address local issues not only motivated students and raised awareness about their community but also ensured equitable participation among both male and female students from Industrial and Systems Engineering disciplines. These activities included creating programs in mBlock, configuring sensors, programming Arduino boards, and developing functional prototypes in a collaborative classroom setting [51].

Each phase of Pólya's problem-solving framework contributed to the development of specific CT skills. In the "understanding the problem" phase, students developed abstraction skills by representing complex issues using mind maps and simplifying them into basic representations [52, 53]. This process helped students focus on essential elements of the problem and ignore irrelevant details.

During the "preparation of the plan" phase, students honed their decomposition skills by breaking down the problem into smaller, manageable subproblems. These included activities such as acquiring electronic components, designing and implementing hardware, and planning the development of software solutions. Additionally, this phase fostered generalization skills, as students were encouraged to identify patterns or similarities with previous problems [54, 55].

The "execution of the plan" phase was pivotal for the development of algorithmic design skills. Students systematically followed a sequential process to complete tasks, which included programming the Arduino with sensors, developing algorithms, debugging programs, and integrating all components into a cohesive prototype. This step-by-step approach exemplifies core aspects of algorithmic thinking and problem-solving [56].

Finally, in the "reviewing the solution" phase, students enhanced their evaluation skills by rigorously testing and debugging their prototypes. They used criteria such as readability, optimization, and performance to refine their solutions until achieving the desired outcomes [52, 57]. This iterative process not only strengthened their ability to assess technical outputs but also fostered critical thinking.

A comparison of outcomes between industrial and systems engineering students reveals important differences in CT skill development. Systems Engineering students showed superior improvements in abstraction and algorithmic design skills, likely due to their curriculum's emphasis on hardware and software technology from the first year of study. In contrast, Industrial Engineering students exhibited more moderate gains, which may be attributed to their program's focus on management-oriented problem-solving approaches [26, 27]. These findings highlight the importance of tailoring CT-based interventions to the specific needs and strengths of each academic discipline.

In summary, the results demonstrate that technological projects aligned with Pólya's framework effectively strengthen CT skills in engineering students. They also underscore the importance of integrating contextually relevant, hands-on activities to enhance both technical and cognitive skills. Future research should further explore these differences, examining how curricular design and student backgrounds influence CT skill development.

6. CONCLUSIONS

The findings of this study highlight the importance of using technological projects to strengthen computational thinking skills in engineering students. Beyond the specific activities performed, such as algorithm development and programming, this pedagogical approach proved effective in fostering fundamental skills like abstraction, decomposition, algorithmic design, and evaluation. These skills are not only essential in STEM disciplines but also critical for training future professionals capable of solving complex problems and contributing to technological development within their communities.

Additionally, a significant difference was observed in the development of these skills between Systems Engineering and Industrial Engineering students, reflecting the impact of technology-oriented curricula on the effectiveness of such interventions. This finding underscores the need to tailor educational strategies to the characteristics and requirements of each academic program.

The implications of this study extend beyond the classroom, as the use of context-specific projects not only motivates students but also promotes sensitivity to local problems, reinforcing their commitment to developing practical solutions for their environment. This presents opportunities to expand this approach to other disciplines and institutions, exploring its scalability and applicability in diverse contexts.

Finally, this study emphasizes the need to integrate technological tools and innovative methodological approaches in engineering education, contributing to the development of key competencies for the 21st century. However, future research should address the identified limitations, including exploring longer interventions and analyzing other technological tools, to further enrich our understanding of how to enhance computational thinking in educational contexts.

This study has several limitations that should be acknowledged to provide a balanced perspective. First, the intervention was conducted over a relatively short duration (16 weeks), which may not fully capture the long-term development of computational thinking skills. Second, the sample was limited to students from two specific universities in the Huancavelica region of Peru, which may introduce potential biases related to the local context and reduce the generalizability of the findings to other regions or educational settings. Third, the intentional non-probabilistic sampling method used in this study, while appropriate for the scope of the research, may limit the representation of the broader population of engineering students. Additionally, the technological projects developed were tailored to address local community problems, which could pose challenges to the scalability or replication of these projects in different geographical or institutional contexts.

Future studies should aim to expand on the findings of this research by exploring the integration of technological projects in diverse cultural and educational contexts. For instance, conducting similar studies in regions with varying levels of technological infrastructure or educational policies could provide valuable insights into the adaptability and effectiveness of these interventions. Additionally, future research could examine the impact of incorporating alternative technological tools, such as Raspberry Pi, micro:bit, or cloud-based platforms, to further enhance computational thinking skills. Comparative studies between different engineering disciplines or even non-STEM fields could also yield a broader understanding of how computational thinking methodologies can be tailored to various academic and professional domains. Finally, longitudinal studies assessing the long-term retention and application of computational thinking skills acquired through these projects would contribute significantly to the field, offering evidence of their enduring impact on students' academic and professional development.

REFERENCES

[1] Sunkel, G., Trucco, D. (2010). TIC para la educación en américa latina. Riesgos y oportunidades. Serie Políticas Sociales, 167: 1-7.

[2] Ferreyra, M.M., Avitabile, C., Álvarez, J.B., Paz, F.H., Urzúa, S. (2024). Momento decisivo La educación superior en América Latina y el Caribe. 2017. https://openknowledge.worldbank.org/server/api/core/bitstreams/921c0f2e-6aa3-512e-be0c-20efb0a3493b/content.

[3] Shyamala, C.K., Velayutham, C.S., Parameswaran, L. (2017). Teaching computational thinking to entry-level undergraduate engineering students at Amrita University. In 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, Greece, pp. 1731-1734. https://doi.org/10.1109/EDUCON.2017.7943084

[4] de Lima Sobreira, P., Abijaude, J.W., Viana, H.D.G., Santiago, L.M.S., El Guemhioui, K., Wahab, O.A., Greve, F. (2020). Usability evaluation of block programming tools in IoT contexts for initial engineering courses. In 2020 IEEE world Conference on Engineering Education (EDUNINE), Bogota, Colombia, pp. 1-5. https://doi.org/10.1109/EDUNINE48860.2020.9149481

[5] Grover, S., Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. Educational Researcher, 42(1): 38-43. https://doi.org/10.3102/0013189X12463051

[6] Ayuso, Á.M., Povedano, N.A., López, R.B. (2020). La resolución de problemas basada en el método de Polya usando el pensamiento computacional y Scratch con estudiantes de Educación Secundaria. Aula Abierta, 49(1): 83-90.

https://doi.org/10.17811/rifie.49.1.2020.83-90
[7] Curasma, R.P., Villalba-Condori, K.O., Jara, N.J., Llamoca, R.Q., Chávez, J.C.C., Ponce-Aranibar, M.D.P. (2021). Computational thinking and block-based programming for beginning engineering students: Systematic review of the literature. In 2021 XVI Latin American Conference on Learning Technologies (LACLO), Arequipa, Peru, pp. 530-533. https://doi.org/10.1109/LACLO54177.2021.00096
[8] Harangus, K., Katai, Z. (2020). Computational thinking in secondary and higher education. Procedia Manufacturing, 46: 615-622. https://doi.org/10.1016/j.promfg.2020.03.088
[9] da Silva, T.S., Melo, J.C., Tedesco, P.C. (2020). The creative process in the development of computational thinking in higher education. In Proceedings of the 12th International Conference on Computer Supported Education (CSEDU 2020), United Kingdom, pp. 215-226. https://doi.org/10.5220/0009346502150226
[10] Wilson, J.W., Fernandez, M.L., Hadaway, N. (2022). Mathematical problem solving. The University of Georgia. http://jwilson.coe.uga.edu/emt725/PSsyn/Pssyn.html.
[11] Fang, A.D., Chen, G.L., Cai, Z.R., Cui, L., Harn, L. (2017). Research on blending learning flipped class model in colleges and universities based on computational thinking—"Database Principles" for example. Eurasia Journal of Mathematics, Science and Technology Education, 13(8): 5747-5755. https://doi.org/10.12973/eurasia.2017.01024a
[12] Kules, B. (2016). Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes. Proceedings of the Association for Information Science and Technology, 53(1): 1-6. https://doi.org/10.1002/pra2.2016.14505301092
[13] Selby, C.C. (2015). Relationships: Computational thinking, pedagogy of programming, and Bloom's Taxonomy. In Proceedings of the Workshop in Primary and Secondary Computing Education, United Kingdom, pp. 80-87. https://doi.org/10.1145/2818314.2818315
[14] Bordignon, F., Iglesias, A. (2018). Introducción al pensamiento computacional. Argentina: Educar S.E.
[15] Wing, J.M. (2006). Computational thinking. Communications of the ACM, 49(3): 33-35. https://doi.org/10.1145/1118178.1118215
[16] Duch, B.J., Groh, S.E., Allen, D.E. (2001). The power of problem-based learning: A practical" how to" for teaching undergraduate courses in any discipline.
[17] Saez-Lopez, J.M. (2021). Introducing robotics and block programming in elementary education. Revista Iberoamericana de Educación a Distancia, 24(1): 95-113. https://doi.org/10.5944/ried.24.1.27649
[18] Marais, C., Bradshaw, K. (2015). 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, pp. 154-160.
[19] Ubaidullah, N.H., Mohamed, Z., Hamid, J., Sulaiman, S. (2021). Discovering the role of problem-solving and discussion techniques in the teaching programming environment to improve students' computational thinking skills. International Journal of Information and Education Technology, 11(12): 615-623. https://doi.org/10.18178/ijiet.2021.11.12.1572
[20] Kale, U., Yuan, J. (2021). Still a new kid on the block? Computational thinking as problem solving in Code. org. Journal of Educational Computing Research, 59(4): 620-644. https://doi.org/10.1177/0735633120972050
[21] Mamani Calcina, J.G., Paucar-Curasma, R., Villalba-Condori, K.O., Rondon, D., Berrios-Espezúa, M.G., Acra-Despradel, C. (2023). Use of technological resources for the development of computational thinking following the steps of solving problems in engineering students recently entering college. Education Sciences, 13(3): 279. https://doi.org/10.3390/educsci13030279
[22] Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). Developing computational thinking in compulsory education-Implications for policy and practice (No. JRC104188). Joint Research Centre (Seville site). https://doi.org/10.2791/792158
[23] Vegas, E., Fowler, B. (2021). What do we know about the expansion of K-12 computer science education? Brookings. https://www.brookings.edu/research/what-do-we-know-about-the-expansion-of-k-12-computer-science-education/?fbclid=IwAR2uLNpVqVNbMXySZSbhTKa3JxRW1XTNOp64H3oXNiRhDdA0X855FJ0son4.
[24] Trilles, S., Monfort-Muriach, A., Gómez-Cambronero, Á., Granell, C. (2022). Sucre4Stem: Collaborative projects based on IoT devices for students in secondary and pre-university education. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, 17(2): 150-159. https://doi.org/10.1109/RITA.2022.3166854
[25] Trilles, S., Granell, C. (2020). Advancing preuniversity students' computational thinking skills through an educational project based on tangible elements and virtual block-based programming. Computer Applications in Engineering Education, 28(6): 1490-1502. https://doi.org/10.1002/cae.22319
[26] Tseng, T.H., Tai, Y., Tsai, S.P., Ting, Y.L. (2018). Students' self-authoring mobile App for integrative learning of STEM. International Journal of Electrical Engineering Education, 2018: 1-12. https://doi.org/10.1177/0020720918800438
[27] Chookaew, S., Howimanporn, S., Hutamarn, S. (2021). Investigating students' computational thinking through STEM robot-based learning activities. Advances in Science, Technology and Engineering Systems, 5(6): 1366-1371. https://doi.org/10.25046/aj0506164
[28] Sinkovits, R.S., Soto, O.D., Introducing computing and technology through problem-solving in discrete mathematics. In Practice and Experience in Advanced Research Computing, USA, pp. 429-435. https://doi.org/10.1145/3311790.3396620
[29] Tuan, B.A., Thao, D.B., Ho, T.D.C. (2022). Internet of things (IoT) in education for sustainability: Using stemtech model to design smart home system for experiential learning at secondary schools. In 5th International European Conference on Interdisciplinary Scientific Research, I, Valencia, pp. 7-22.
[30] Juškevičienė, A., Stupurienė, G., Jevsikova, T. (2021). Computational thinking development through physical computing activities in STEAM education. Computer Applications in Engineering Education, 29(1): 175-190. https://doi.org/10.1002/cae.22365
[31] Yin, Y., Khaleghi, S., Hadad, R., Zhai, X. (2022). Developing effective and accessible activities to improve and assess computational thinking and engineering

learning. Educational Technology Research and Development, 70(3): 951-988. https://doi.org/10.1007/s11423-022-10097-w
[32] Bedar, R.A.H., Al-Shboul, M. (2020). The effect of using STEAM approach on developing computational thinking skills among high school students in Jordan. International Journal of Interactive Mobile Technologies, 14(14): 80-94. https://doi.org/10.3991/IJIM.V14I14.14719
[33] Segalàs Coral, J., Sánchez Carracedo, F. (2019). El proyecto EDINSOST. Formación en las Universidades españolas de profesionales como agentes de cambio para afrontar los retos de la sociedad. Revista de Educación Ambiental y Sostenibilidad, 1(1): 1204-1211.
[34] Benjumeda, F., Romero, I. (2017). Ciudad Sostenible: un proyecto para integrar las materias científico-tecnológicas en Secundaria. Revista Eureka, 14(3): 621-637. https://doi.org/10.25267/Rev_Eureka_ensen_divulg_ciencc.2017.v14.i3.08
[35] Martínez-Chico, M., Jiménez, M., López-Gay, R. (2016). Efecto de un programa formativo para enseñar ciencias por indagación basada en modelos, en las concepciones didácticas de los futuros maestros. Revista Eureka Sobre Enseñanza y Divulgación de Las Ciencias, 12(1): 149-166. https://doi.org/10.25267/rev_eureka_ensen_divulg_cienc.2015.v12.i1.10
[36] Sandia Saldivia, B., Montilva Calderon, J. (2020). Digital technologies in service learning for the formal education of the new millennium citizenship. Ried-Revista Iberoamericana de Educacion a Distancia, 23(1): 129-148. https://doi.org/10.5944/ried.23.1.24138
[37] Montilva, J., Montilva, W. (2018). An ontological and systemic method for the conceptual learning of digital technologies. Ciencia E Ingenieria, 39(3): 269-278.
[38] Vergara, M., Sandia, B., Arnal, G., Moreno, Y., Mujica, A. (2019). Reproducción del ambiente de innovación en el salón de clase como base para una educación universitaria sostenible. Revista Ibérica de Sistemas e Tecnologias de Informação, E20: 710-719.
[39] Al-Naqbi, A.K., Alshannag, Q. (2018). The status of education for sustainable development and sustainability knowledge, attitudes, and behaviors of UAE University students. International Journal of Sustainability in Higher Education, 19(3): 566-588. https://doi.org/10.1108/IJSHE-06-2017-0091
[40] Heidmets, M., Udam, M., Vanari, K., Vilgats, B. (2018). Good university and excellent professor: Competing quality perspectives in higher education. Sustainable Futures for Higher Education: The Making of Knowledge Makers, Springer, Cham, pp. 123-139. https://doi.org/10.1007/978-3-319-96035-7_13
[41] Román-Gonzalez, M. (2015). Test de pensamiento computacional: Principios de diseño, validación de contenido y análisis de ítems. In Jornada de Doctorandos, Madrid, pp. 1-19.
[42] Román-González, M., Pérez-González, J.C., Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. Computers in Human Behavior, 72: 678-691. https://doi.org/10.1016/j.chb.2016.08.047
[43] Román-González, M., Pérez-González, J.C., Moreno-León, J., Robles, G. (2018). Can computational talent be detected? Predictive validity of the computational thinking test. International Journal of Child-Computer Interaction, 18: 47-58. https://doi.org/10.1016/j.ijcci.2018.06.004
[44] Viale, P., Deco, C. (2019). Introduciendo conocimientos sobre el Pensamiento Computacional en los primeros años de las carreras de ciencia, tecnología, ingeniería y matemáticas. Energeia, 16(16): 73-78.
[45] Brackmann, C.P. (2017). Desenvolvimento do Pensamento Computacional através de atividades desplugadas na Educação Básica. Universidade Federal do Rio Grande do Sul.
[46] Papert, S., Harel, I. (2002). Situar el construccionismo. Alajuela: INCAE.
[47] Polya, G. (1945). How to Solve It (2nd ed.). New York: Princeton University Press, Doubleday Anchor Books.
[48] Casey, E., Jocz, J., Peterson, K.A., Pfeif, D., Soden, C. (2023). Motivating youth to learn STEM through a gender inclusive digital forensic science program. Smart Learning Environments, 10(1): 2. https://doi.org/10.1186/s40561-022-00213-x
[49] Ortiz-Martínez, G., Vázquez-Villegas, P., Ruiz-Cantisani, M.I., Delgado-Fabián, M., Conejo-Márquez, D.A., Membrillo-Hernández, J. (2023). Analysis of the retention of women in higher education STEM programs. Humanities and Social Sciences Communications, 10(1): 1-14. https://doi.org/10.1057/s41599-023-01588-z
[50] Ng, W., Fergusson, J. (2020). Engaging high school girls in interdisciplinary STEAM. Science Education International, 31(3): 283-294. https://doi.org/10.33828/sei.v31.i3.7
[51] Maass, K., Geiger, V., Ariza, M.R., Goos, M. (2019). The role of mathematics in interdisciplinary STEM education. ZDM, 51: 869-884. https://doi.org/10.1007/s11858-019-01100-5
[52] Ubaidullah, N.H., Mohamed, Z., Hamid, J., Sulaiman, S., Yussof, R.L. (2021). Improving novice students' computational thinking skills by problem-solving and metacognitive techniques. International Journal of Learning, Teaching and Educational Research, 20(6): 88-108. https://doi.org/10.26803/IJLTER.20.6.5
[53] Liu, J., Wang, L. (2010). Notice of Retraction: Computational thinking in discrete mathematics. In 2010 Second International Workshop on Education Technology and Computer Science, Wuhan, China, pp. 413-416. https://doi.org/10.1109/ETCS.2010.200
[54] Jeng, H.L., Liu, L.W., Chen, C.N. (2019). Developing a procedural problem-solving-based framework of computational thinking components. In 2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI), Toyama, Japan, pp. 272-277. https://doi.org/10.1109/IIAI-AAI.2019.00061
[55] Weese, J.L., Feldhausen, R. (2017). STEM outreach: Assessing computational thinking and problem solving. In 2017 ASEE Annual Conference & Exposition, Columbus, OH.
[56] Rabiee, M., Tjoa, A.M. (2017). From abstraction to implementation: Can computational thinking improve complex real-world problem solving? A computational thinking-based approach to the SDGs. In 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, pp. 104-116.

https://doi.org/10.1007/978-3-319-59111-7_10
[57] Pedaste, M., Palts, T., Kori, K., Sõrmus, M., Leijen, Ä. (2019). Complex problem solving as a construct of inquiry, computational thinking and mathematical problem solving. In 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT), Maceio, Brazil, pp. 227-231. https://doi.org/10.1109/ICALT.2019.00071