Mathematical Modelling of Engineering Problems
Vol. 12, No. 8, August, 2025, pp. 2931-2944
Journal homepage: http://iieta.org/journals/mmep

Promoting Gender Equity Through a STEM Kit and Visual Programming to Develop Computational Thinking in the Early Years of University Education

Ronald Paucar-Curasma[1*], Roberto Florentino Unsihuay-Tovar[2], Omar Felix Illesca-Cangalaya[2], Nolan Jara-Jara[2], Rosa Quispe-Llamoca[3], Anieval Cirilo Peña-Rojas[4], Klinge Orlando Villalba-Condori[5]

[1] Grupo de Investigación TIC Aplicadas a la Sociedad, Universidad Nacional Autónoma de Tayacaja Daniel Hernández Morillo, Pampas 09156, Peru
[2] Facultad de Ingeniería Electrónica y Eléctrica, Universidad Nacional Mayor de San Marcos, Lima 15001, Peru
[3] Carrera de Economía, Universidad de Lima, Surco 15023, Peru
[4] Facultad de Ingeniería de Sistemas, Universidad Nacional del Centro del Perú, Huancayo 12001, Peru
[5] Vicerrectorado de Investigación, Universidad Católica de Santa María, Arequipa 04001, Peru

Corresponding Author Email: rpaucarc@unat.edu.pe

https://doi.org/10.18280/mmep.120834

Received: 1 June 2025
Revised: 19 July 2025
Accepted: 24 July 2025
Available online: 31 August 2025

Keywords:
computational thinking, visual programming, STEM educational kit, computational concepts, computational practices, engineering students, technology projects

ABSTRACT

This article analyzes the development of computational thinking dimensions by gender among students from Industrial Engineering and Systems Engineering programs at universities in the Andean region of Peru. Two key dimensions were assessed: computational concepts (including sequence, events, conditionals, loops, operators, data, and parallelism) and computational practices (experimenting and interacting, testing and debugging, reusing previous projects, and abstracting and modularizing). The study employed a post-test quasi-experimental design with intentional non-probability sampling. Technological projects with a contextual and community-based focus—related to agriculture, livestock, environment and safety—were developed using a STEM Learning Kit with sensors, actuators, and the mBlock visual programming environment. Results showed no statistically significant differences between male and female students in overall computational thinking performance. However, when analyzing achievement levels, the researchers found that notable differences emerged: most women in Systems Engineering reached expected or outstanding conceptual levels, while most students in both programs and genders remained at the beginning level in computational practices. Women in Industrial Engineering exhibited greater variability in practical achievement, suggesting higher potential for progress. These findings confirm that integrating contextualized technological projects with visual programming is an effective pedagogical strategy to enhance computational thinking across genders and promote gender equity in STEM from the early years of university education.

1. INTRODUCTION

It is widely recognized that there is a significant gender gap in Science, Technology, Engineering, and Mathematics (STEM) career choices, with women disproportionately underrepresented. This disparity suggests that women may feel less engaged in solving context-specific problems, a challenge particularly pronounced in rural areas where opportunities for women are even more limited. However, numerous studies highlight global efforts to enhance education and promote equality across various sectors [1]. This gap largely stems from low female participation in STEM fields worldwide; only 10% of women choose to pursue STEM careers globally, and in Peru, only 29% of those who opt for science and technology fields are women, due to gender-related barriers [2]. In response, UNESCO emphasizes the need to cultivate interest in science and technology from an early age, dismantle stereotypes, train educators to encourage girls to pursue STEM fields, develop gender-sensitive curricula, and provide guidance to challenge preconceived notions.

Despite growing global interest in promoting gender equity in STEM education, there remains a significant lack of inclusive interventions during the early years of university education [3, 4]. While several studies have addressed gender gaps in computational thinking at the secondary education level [5, 6], university-level efforts—especially in rural or underserved regions—are still underexplored. This lack of research at the university level, particularly in marginalized regions like the Andes, highlights a critical gap in understanding how inclusive, gender-sensitive approaches may foster computational thinking in higher education. Most interventions fail to consider the intersection of gender, geographic location, and educational equity, leaving students in marginalized contexts without access to strategies that could enhance their participation and success in STEM fields [7].

In Peru, data on PRONABEC [8] scholarship recipients as of the 2015-II semester reveals that 88% of graduates are pursuing studies in engineering and technology, with a gender distribution of 73% male and 27% female. The rising number of female graduates has led to increased participation in fields such as Art and Architecture (78%), Economics and related disciplines (57%), Basic Sciences (57%), and Agriculture and related fields (56%). In the Huancavelica region, Beca 18 has supported 2,655 young people between 2012 and 2015 in accessing higher education; of these, 55% of scholarships were awarded to men and 45% to women. Notably, 69% of scholarships funded engineering and technology studies, with most awarded to men, while 25% went to business-related studies, among others. Regarding researchers in science and technology by gender in Peru, men account for 68% and women for 32%; in the Huancavelica region specifically, 77% are men and 23% are women [9].

According to the United Nations Educational, Scientific and Cultural Organization [10], advances in STEM disciplines have driven progress in various aspects of life, including health, agriculture, infrastructure, and renewable energy. However, a persistent issue is the underrepresentation of women in STEM fields, often stemming from early exposure to prejudices and stereotypes that discourage interest in science and technology. To encourage greater female participation in STEM, computational thinking has been identified as a promising strategy through classroom activities involving technological resources [11]. STEM fields naturally align with computational thinking through activities such as algorithm development, programming, modeling, simulation, and experimental methods [12].

In academia, computational thinking has been adopted as a classroom strategy to inspire more women to pursue STEM disciplines. This approach involves activities related to STEM, such as utilizing technological resources (e.g., microcontrollers, sensors, actuators) alongside problem-solving methods [13]. STEM disciplines and computational thinking share common elements, including tasks that involve algorithm development, coding, using technological tools, and teamwork [12].

Today, electronic prototypes—such as robotics kits, sensors, and actuators equipped with visual programming environments—play a crucial role for students beginning their university studies. These tools facilitate conceptual learning by simplifying fundamental programming and logic concepts essential for research. They also encourage creativity, allowing students to experiment and innovate without the constraints of textual code. Moreover, they foster collaborative skills by including group components that promote teamwork and communication across genders [14].

While many studies focus on robotics or conventional STEM programs, the present study stands out by integrating community-centered problem-solving, a visual programming environment, and a gender perspective [13, 15, 16]—an innovative combination rarely explored in current STEM education research.

This study aims to address this gap by evaluating the effectiveness of community-based technological projects—integrating a STEM Kit and visual programming environment—in developing computational thinking among male and female students in Industrial and Systems Engineering programs in Peru's Andean region. Although the data is drawn from a specific national context, the educational challenges it highlights—such as underrepresentation of women in STEM and the limited availability of inclusive, practice-oriented learning experiences—are common across many global contexts [17]. Therefore, the results and methodology presented may be relevant and transferable to other regions facing similar equity challenges.

2. RELATED WORK

2.1 Computational thinking in the early years of college

Computational thinking is a crucial 21st-century skill that students need to develop to effectively solve problems across various domains. According to existing literature, computational thinking originally focused on skill development for primary and secondary education students. However, successful interventions in university settings have emerged, particularly during the initial years of study. Best practices suggest introducing computational thinking to first-year students in ICT or computing courses, as well as incorporating it into non-computing courses [18]. This approach establishes a reference framework for computational thinking that educators can apply across diverse courses. Many authors agree that "we must go beyond merely training students to solve problems using programming languages" and instead emphasize nurturing their skills and motivations, which are essential for enhancing student performance.

Terreni [19] describes computational thinking as encompassing a range of complex skills, commonly linked with computer programming. He emphasizes that computational thinking involves a sequence of processes, starting with problem comprehension and definition, followed by identifying alternative solutions, argumentation, using technological tools, executing activities, testing performance, and gathering feedback. These processes can be applied to various disciplines, depending on the curriculum design.

In terms of computational tools, a variety of resources are available. Common tools include programming languages and IDEs, with Python being a typical choice in introductory courses for computer science and engineering students. Through programming, students engage with computational concepts and develop tailored applications. Additionally, pre-programming gamified experiences are popular, as are educational tools for teaching algorithms, programming structures, and variables using platforms like Lightbot, mBlock, and educational robotics kits, which emphasize foundational programming and computational thinking skills [20]. As a cognitive problem-solving process, computational thinking comprises five key skills: algorithmic thinking, decomposition, pattern recognition, abstraction, and simplified presentation, and evaluation for decision-making [21].

Brennan and Resnick [22] argue that computational thinking thrives in design-based learning activities, such as creating interactive media, facilitated by visual software environments or block programming. This approach is generally defined by two primary dimensions: computational concepts and practices. Many authors highlight visual programming as a powerful tool for developing computational thinking, as it provides an intuitive and accessible means to understand and apply computational concepts. Furthermore, it promotes creativity, logical reasoning, and problem-solving skills, all of which are essential for engaging with STEM disciplines. Table 1 outlines the main dimensions of computational thinking.

Table 1. Dimensions of computational thinking

Computational Dimensions	Indicators	Definition
Computational Concepts	Sequences	Activity that is expressed through a series of instructions that the computer executes.
	Cycles	It executes a sequence of instructions repetitively.
	Events	It is related to "when an event occurs, then it causes another event to happen."
	Parallelism	A sequence of instructions is executed simultaneously.
	Conditionals	They are decisions or alternatives to choose to solve a problem.
	Operators	Mathematical, logical and string expressions that are used in programs.
	Data	They are related to variables that store data, such as numbers, characters, among others.
Computing Practices	Incremental and iterative	These are iterative steps that are taken when developing a program (for example: developing little by little, then testing, and continuing to develop a little more)
	Testing and debugging	It refers to constant "trial and error" testing; also, to requesting support from a third party or a community.
	Reusing and remixing	Developing a program based on other pre-existing programs.
	Abstracting and modularizing	"Characterizing the process of building something large, by adding sets of smaller elements."

2.2 Computational thinking and gender

Gender is a significant factor in education, as differences in attitudes and performance between girls and boys are evident, even in basic tasks like reading and writing. However, research comparing the development of computational thinking skills between genders, particularly in K-12 robotics activities, remains limited [23]. Social stereotypes surrounding computer science can negatively impact women's motivation to engage in computational activities [24].

Several studies have highlighted gender as a relevant variable in the development of computational thinking skills. For instance, one study found that computational thinking activities often favored boys in regular education settings [25]. Meta-analyses also suggest that computational thinking may exhibit a moderate gender bias, as many activities tend to be more male-oriented [26]. Furthermore, it appears that the types of projects preferred by boys often require more programming complexity, resulting in higher computational thinking scores compared to girls. This suggests that gender differences in computational thinking can be influenced by the nature of the projects undertaken [27].

Significant differences have also been observed in the strategies and approaches boys and girls use during computational thinking activities. When teaching techniques are designed to support a comprehensive understanding of computational thinking principles and to inspire exploration among both genders, the gender gap in computational thinking competency nearly disappears. It has been suggested that instructional design should accommodate these gender-specific strategies, as girls often approach learning with different methods than boys. Adapting coding activities to these differences, by offering differentiated support, can lead to more equitable learning outcomes [28].

By implementing appropriate teaching strategies and tools—such as educational games, virtual programming languages, and robotics—specifically designed with a gender-based approach, recent years have seen substantial progress in reducing the gender gap [29]. Additionally, research has shown that collaboration and teamwork can play a crucial role in bridging gender differences. For example, pairing female students with male peers showed that female students performed on par with male students, demonstrating that both genders benefit equally from teamwork in computational problem-solving [30].

These insights contribute to the development of age-appropriate, evidence-based pedagogies and learning progressions for computational thinking [31]. While scientific literature reflects growing interest in computational thinking and related skills, gender-specific research in this area remains relatively sparse. However, interest is increasing, particularly in K-12 education, where there is a push to incorporate educational strategies that encourage more women to pursue technology-related careers. In this context, computational thinking plays a vital role in addressing the gender gap.

2.3 STEM Kit with visual programming environment

To carry out the activities, an educational STEM Kit was used, consisting of sensors, actuators, and a microcontroller board compatible with visual programming environments [32]. This allowed the implementation of technological classroom projects in an accessible and interactive way. The board and its components were integrated into the mBlock environment, a block-based interface known for its intuitive design, which facilitates the acquisition of basic programming concepts and computational thinking skills among first-year university students [11].

The use of such tools aligns with previous studies that highlight their potential to democratize programming education, especially for students with no prior experience [14, 33]. However, it is important to acknowledge some of the limitations of the mBlock environment. One limitation is its restricted scalability to more advanced programming levels, which may limit the development of more sophisticated skills in higher-level courses [34, 35]. Additionally, some functionalities rely on internet connectivity, which can be a barrier in rural or low-resource settings [12].

Despite these limitations, the tool is well suited for the objectives of this study, which aim to foster computational thinking among first-year students. Its use facilitates experimentation, collaboration, and the creation of visual narratives—key elements for motivating female students in particular to develop technological skills [36, 37].

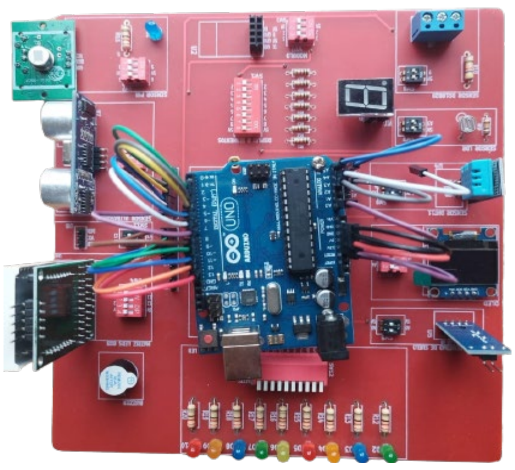


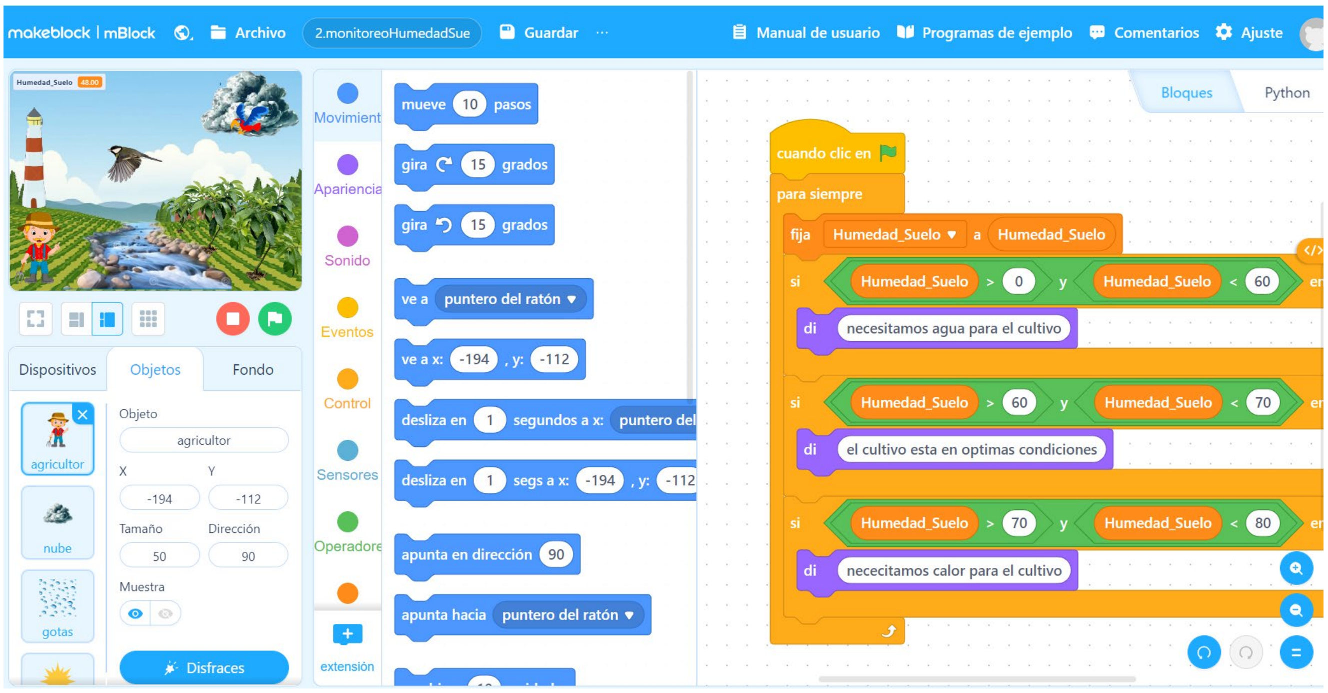
Figure 1. STEM educational kit

Overall, the STEM Kit offers an accessible and user-friendly platform that effectively supports the educational needs of first-year university students. It enables them to engage in hands-on projects that address real-world challenges relevant to their local communities. Figure 1 illustrates the components of the STEM Education Kit.

The software component of the STEM Kit features a visual programming environment with blocks specifically designed to interact with sensors and actuators. These programming blocks were created using Python libraries within the mBlock development platform [38]. At the core of the STEM Kit is this visual programming environment based on mBlock, allowing students to represent solutions for their projects in a straightforward and intuitive manner, minimizing cognitive complexity. This is particularly suitable for university freshmen. The visual nature of the programming environment engages students in scientific and technological activities, enabling them to see results instantly and continue refining their projects to meet their objectives.

Using mBlock, students can break down a problem into smaller parts and understand how to sequence steps toward a solution. The programming blocks cover fundamental concepts such as loops, conditionals, and events, which are essential for algorithmic thinking. Working with these concepts helps students learn to decompose and structure problems—core skills in computational thinking. This approach not only encourages programming but also facilitates the application of knowledge to fields like electronics, physics, and creative design. Through these activities, students develop competencies that integrate various areas of knowledge, promoting a comprehensive, multidisciplinary learning experience.

The STEM Kit also supports project-based learning, allowing students to learn actively by creating tangible solutions. This fosters collaboration, as projects can be shared and improved upon in teams, enhancing teamwork and communication skills. Figure 2 shows the mBlock-based visual programming environment.



```
cuando clic en
para siempre
  fija Humedad_Suelo a Humedad_Suelo
  si Humedad_Suelo > 0 y Humedad_Suelo < 60 en
    di necesitamos agua para el cultivo
  si Humedad_Suelo > 60 y Humedad_Suelo < 70 en
    di el cultivo esta en optimas condiciones
  si Humedad_Suelo > 70 y Humedad_Suelo < 80 en
    di nececitamos calor para el cultivo
```

Figure 2. Visual programming environment based on mBlock

2.4 Importance of visual programming for beginner students

Block-based programming languages have become a popular, low-cost method for teaching programming and computational thinking to students and educators with limited computing experience, as well as for beginner students. Studies have shown that block-based programming is an effective approach to developing skills in computer science and computational thinking [39]. For instance, programs like Scratch and mBlock [40] enable novice students to create projects, such as city stories set in specific historical periods, where they apply skills and knowledge from mathematics, technology, communication, and social sciences [41]. Common programming tools for beginners include Scratch, mBlock, and AppInventor, while visual assessment tools, like Dr. Scratch, are also block-based [42]. These tools, with their virtual programming environments, enhance the teaching and learning of computational thinking. The visual programming environment creates a drag-and-drop interface, using functional and control blocks to describe programs. Scratch and mBlock are now benchmarks in visual programming, useful for everything from introductory programming to more complex application development [43]. The colored block categories assist students in selecting the correct block, reducing some of the barriers to programming by preventing errors and minimizing cognitive overload. Syntax errors common in text-based languages are largely avoided or eliminated [44].

Different visual programming environments serve as both tools and pedagogical resources that strengthen students' cognitive and problem-solving skills. Educators can select these tools based on the specific educational context. However, the diverse range of languages and tools presents a challenge, as trends, development environments, and curricular needs constantly evolve [45]. These software tools are supported by constructivist theory, promoting knowledge construction as a dialectical interaction between the knowledge of the teacher and the student. This interaction fosters intrinsic motivation, class integration and participation, student-centered focus, interaction and feedback, and seamless integration of educational content into problem-solving [46]. Additionally, there is broad consensus that programming, in its various forms, is essential for developing computational concepts and best practices that form the core of computational thinking [47].

Selecting appropriate visual programming environments is crucial, as it impacts student learning outcomes and the development of computational thinking and creativity. Most visual programming environments are free or partially free, compatible with various platforms, and often web-based, requiring a continuous internet connection. This can be a limitation in certain educational settings, such as rural areas with slow connectivity [48].

3. METHODOLOGY

3.1 Research approach and participants

This study employed a quasi-experimental design with pretest and posttest measurements, framed within a quantitative approach. The participants were first-year students enrolled in the "Information Management" course, part of the Industrial and Systems Engineering programs at a public university located in the Andean region of Peru. The course lasted 16 weeks, with weekly sessions of 4 hours, and was aligned with pedagogical objectives aimed at promoting computational thinking, strengthening problem-solving skills related to the profession, and developing technological competencies from the early stages of the academic cycle. Some of the co-authors of this study served as instructors for the course.

The sample consisted of 95 university students from the Industrial and Systems Engineering programs. Of these, 70 were male and 25 females, reflecting a representative gender distribution in these fields, particularly in rural contexts where female participation in STEM areas remains low [4]. Although the difference in group sizes could limit some comparative analyses, all available participants were included in the study. The students' ages ranged from 17 to 20 years. Table 2 presents the distribution of participants by gender.

Table 2. Students who participated in the study

Population/Sample	Men	Women	Total
Industrial Engineering 2022-II	32	16	48
Systems Engineering 2022-I	38	09	47
Total	70	25	95

3.2 Analysis instruments

To assess computational thinking in university students, two instruments were used, one for each dimension: computational concepts (7 items) and computational practices (4 items). Achievement levels in both dimensions were evaluated based on the Peruvian Ministry of Education's grading system (2024), which uses a literal scale ranging from 0 to 20: AD (Outstanding Achievement, 18–20), A (Expected Achievement, 14–17), B (In Progress, 11–13), and C (Beginning, 0–10).

A. Instrument for the Computational Concepts Dimension

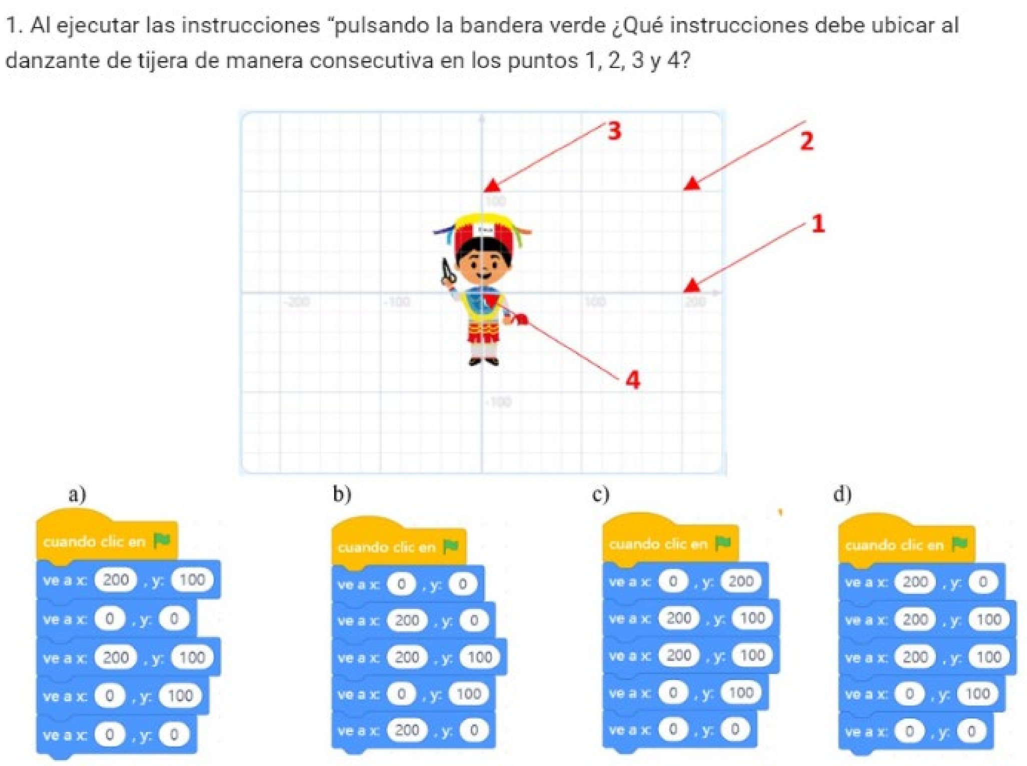
For this dimension, a multiple-choice objective test was used, including visual activities and closed-ended questions, with a total score of 20 points. The instrument was adapted from the computational thinking test developed by Román-Gonzalez [49], originally designed for secondary education, to fit the university context. This adaptation is justified by the compatibility between the indicators used to assess computational concepts and the digital competencies required during the early semesters of engineering programs. Furthermore, previous studies have validated this framework in higher education contexts [50], confirming its relevance for evaluating transversal skills related to computational concepts. Table 3 and Figure 3 present the instrument used for this dimension. Internal consistency was assessed using Cronbach's alpha coefficient, yielding a value of $\alpha = 0.81$, indicating acceptable to good reliability.

Table 3. Instrument for the evaluation of computational concepts

Computational Dimensions	Indicators	Items	Score
Concepts	Sequences	Item1	2
	Cycles	Item2	3
	Events	Item3	3
	Parallelism	Item4	3
	Conditionals	Item5	3
	Operators	Item6	3
	Data	Item7	3

B. Instrument for the Computational Practices Dimension

For this dimension, an analytical performance rubric was used, focusing on the direct observation of computational thinking and programming skills during project execution using mBlock. The rubric was developed based on the cognitive domain levels of Bloom's taxonomy, as adapted by Selby [51]. Level "1" represents a basic performance, where the student is able to identify a specific pattern; level "3" corresponds to an intermediate performance, where the student recognizes the need to apply a practice and develops a simple solution; and level "5" indicates a competent performance, where the student is able to choose among various implementation strategies for a given practice. Table 4 and Figure 4 present the instrument used to assess this dimension. Internal consistency was measured using Cronbach's alpha coefficient, yielding a value of $\alpha = 0.77$, which indicates acceptable to good reliability.


1. Al ejecutar las instrucciones "pulsando la bandera verde ¿Qué instrucciones debe ubicar al danzante de tijera de manera consecutiva en los puntos 1, 2, 3 y 4?

a)
```
cuando clic en 🏁
ve a x: 200 , y: 100
ve a x: 0 , y: 0
ve a x: 200 , y: 100
ve a x: 0 , y: 100
ve a x: 0 , y: 0
```
b)
```
cuando clic en 🏁
ve a x: 0 , y: 0
ve a x: 200 , y: 0
ve a x: 200 , y: 100
ve a x: 0 , y: 100
ve a x: 200 , y: 0
```
c)
```
cuando clic en 🏁
ve a x: 0 , y: 200
ve a x: 200 , y: 100
ve a x: 200 , y: 100
ve a x: 0 , y: 100
ve a x: 0 , y: 0
```
d)
```
cuando clic en 🏁
ve a x: 200 , y: 0
ve a x: 200 , y: 100
ve a x: 200 , y: 100
ve a x: 0 , y: 100
ve a x: 0 , y: 0
```

Figure 3. Item 1 corresponding to the instrument on computational concepts

Table 4. Instrument for the evaluation of computational practices

Score	Computational Practices: Indicators			
	Incremental and Iterative	**Testing and Debugging**	**Reusing and Remixing**	**Abstracting and Modularizing**
1	Experiment and iterate minimally during the execution of the activity	Test and debug minimally during the execution of the activity	Reuses previous projects (set of blocks) minimally in the proposed solution	Minimally abstract and modularize during the execution of the activity
3	Experiment and iterate during the execution of the activity	Test and debug during the execution of the activity	Reuses previous projects (set of blocks) moderately in the proposed solution	Medium abstract and modularize in the proposed solution
5	Experiment and iterate until the proposed solution is completed	Test and debug until the proposed solution is complete	Reuses previous projects (set of blocks) of large size in the proposed solution	High abstraction and modularization in the proposed solution

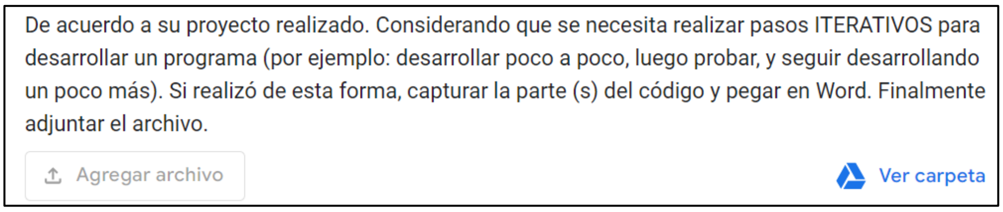

De acuerdo a su proyecto realizado. Considerando que se necesita realizar pasos ITERATIVOS para desarrollar un programa (por ejemplo: desarrollar poco a poco, luego probar, y seguir desarrollando un poco más). Si realizó de esta forma, capturar la parte (s) del código y pegar en Word. Finalmente adjuntar el archivo.

Figure 4. Item 1 corresponding to the instrument on computational practices

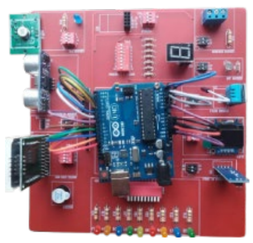
3.3 Proposal and development of technological projects in the classroom

To strengthen the dimensions of computational thinking, technological projects were implemented (see Table 5) focused on solving contextualized problems related to areas such as livestock, environment, agriculture, safety, and education, based on the students' local context. These projects were structured around the four phases of Pólya's method [13]: understanding the problem, designing activities, executing activities, and reviewing the solution. In each phase, students carried out tasks aimed at enhancing both computational concepts and practices. They used the STEM educational board along with various sensors, including an ultrasonic distance sensor (HC-SR04), a temperature and humidity sensor (DHT11), an infrared sensor (HC-SR501), a light sensor (LDR), and a multicolor LED. Additionally, they developed applications using the visual programming environment mBlock to contextualize and simulate solutions to the identified problems.

The planning of activities was structured around the four phases of Pólya's method: understanding the problem, planning activities, executing activities, and reviewing the solution. These sessions were carried out over 16 academic weeks, progressively strengthening the dimensions of students' computational thinking. The activities were part of the "Information Management" course, corresponding to the first year of both academic programs, with a load of 4 hours per week over 16 sessions. All activities were conducted in the classroom, under the constant supervision and continuous feedback of the instructor (co-author of the study).

- Understanding the problem (5 sessions): Students investigated the assigned issue using tools such as ChatGPT, Scopus, and SciELO. They created descriptive summaries with scientific citations using the Mendeley reference manager and visually represented cause-effect relationships through graphic organizers.
- Planning activities (3 sessions): They identified scientific background, analyzed prior experiences, and designed feasible activities contextualized to the local environment, considering technical feasibility and the use of the STEM educational kit.
- Executing activities (2 sessions): Students were trained in using the STEM Kit, assembled circuits, and programmed applications in mBlock to monitor parameters such as air quality, water turbidity, body temperature, among others. They also built models and wrote articles describing their experiences.
- Reviewing the solution: Students evaluated the functioning of the developed models, optimized results based on teacher feedback, and completed their research articles. The solutions addressed contextualized health issues, such as anemia, stomach infections, and risks in fish farms, validated through the use of sensors and visual programming.

Table 5. Proposed technological projects and STEM educational kit

Technological Project	Sensor	STEM Educational Kit
Monitoring soil moisture in vegetable crops to prevent anemia in school-aged children in the district of Acraquia, Tayacaja province	Capacitive soil moisture sensor	
Monitoring river flow to prevent flooding in the city of Huancavelica	Ultrasonic distance sensor	
Monitoring environmental parameters to protect crops from frost	Temperature and humidity sensor (DHT11)	
Monitoring animals to prevent predator attacks in the highlands of Huancavelica	PIR infrared sensor	

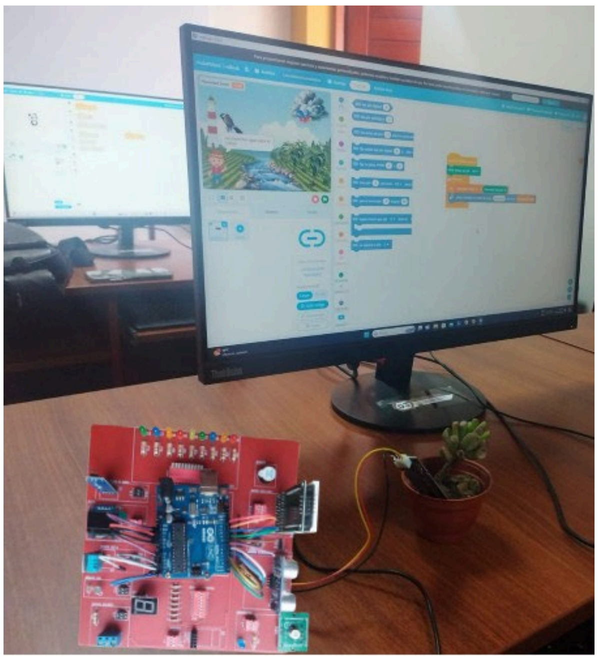


Figure 5. Classroom activities using the STEM Kit and mBlock

Figure 5 illustrates the activities carried out by the students during the implementation of the project "Monitoring soil moisture in vegetable crops to prevent anemia in school-age children in the district of Acraquia, Tayacaja Province."

3.4 Strengthening the dimensions of computational thinking

A. Computational Concepts

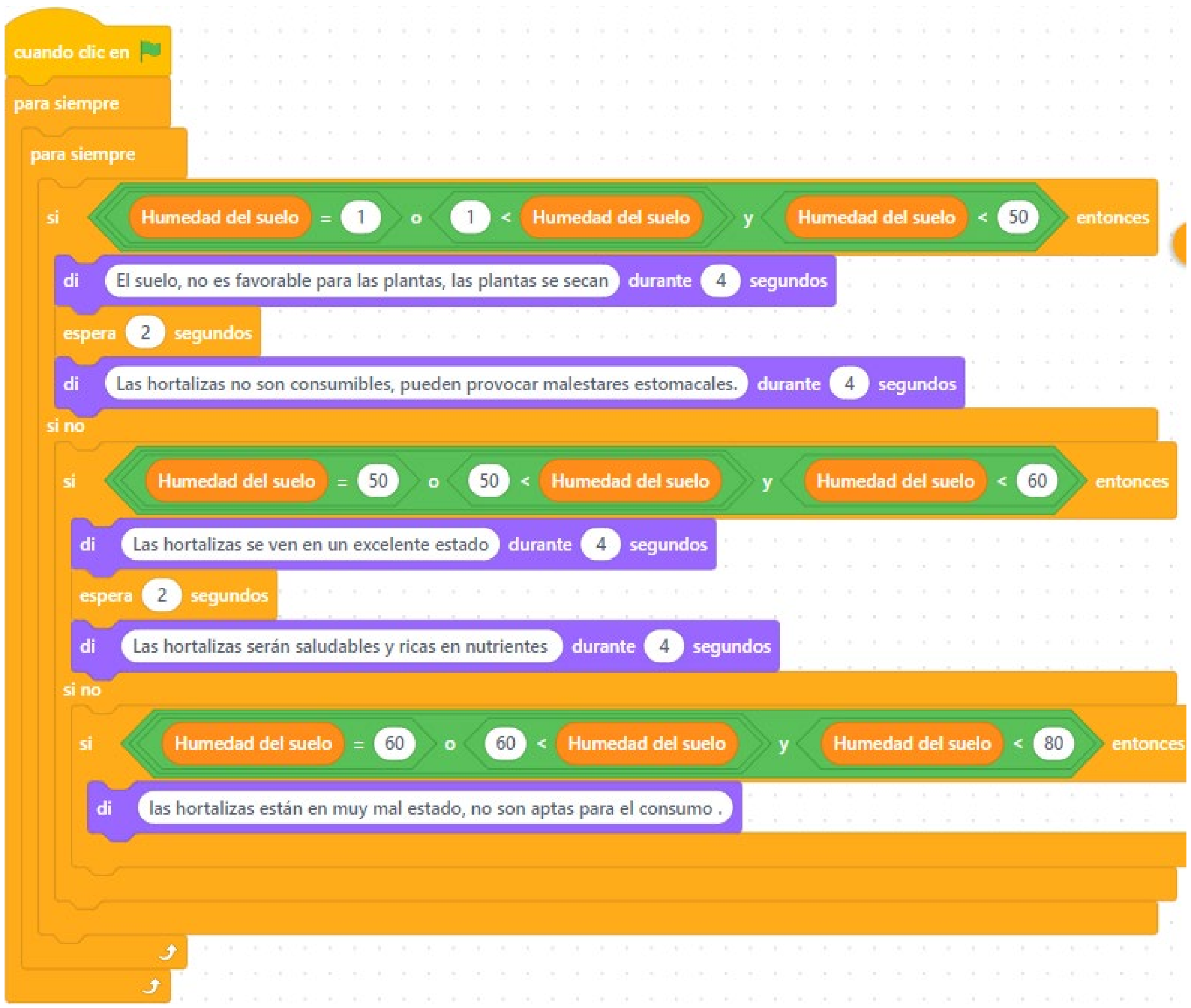
To strengthen computational concepts—sequence, loops, events, parallelism, conditionals, operators, and data—students carried out various activities within their assigned technological projects. As an illustrative example, Figure 6 shows a programming routine in mBlock corresponding to the project "Monitoring soil moisture in vegetable crops to prevent anemia in school-age children in the district of Acraquia, Tayacaja province", which uses a soil moisture sensor from the STEM educational kit.

Throughout the development of the project, students applied computational concepts in a contextualized manner. The sequence concept was addressed by executing instructions in a logical order, such as displaying messages or creating timed pauses. The event concept was demonstrated by starting the routine with the block "when green flag is clicked."

Conditionals were used through "if–then–else" structures to make decisions based on the value of the "Soil Moisture" variable. Operators were reinforced by applying logical comparisons (<, >, =, and, or) to define specific threshold values. The data concept was applied by manipulating the variable's value to control the program's flow. These activities promoted not only a theoretical understanding but also the practical application of computational concepts, linking them to a real-world agricultural problem relevant to the students' local context.

B. Computational Practices

The programming routine shown in Figure 6 demonstrates the strengthening of various computational practices among students. First, experimentation and iteration are evident, as students adjusted logical conditions to simulate different soil moisture levels and their effects on crops, conducting successive tests and refining their code until achieving a functional solution. Testing and debugging were also developed by verifying the program's behavior in response to different sensor values and correcting logical errors to ensure coherent responses. Additionally, reuse and remixing of prior structures can be seen, such as conditional blocks and familiar messages adapted to a new contextualized problem. Finally, abstraction and modularization were promoted by organizing the code into clearly differentiated conditional blocks based on moisture ranges, facilitating the understanding of the program's logical flow and preparing students for more structured programming. This integrative experience enabled the meaningful application of computational practices to real-world problems in their local context.



```
cuando clic en 🏳
para siempre
  para siempre
    si < <Humedad del suelo = 1> o <1 < Humedad del suelo> > y <Humedad del suelo < 50> entonces
      di El suelo, no es favorable para las plantas, las plantas se secan durante 4 segundos
      espera 2 segundos
      di Las hortalizas no son consumibles, pueden provocar malestares estomacales. durante 4 segundos
    si no
      si < <Humedad del suelo = 50> o <50 < Humedad del suelo> > y <Humedad del suelo < 60> entonces
        di Las hortalizas se ven en un excelente estado durante 4 segundos
        espera 2 segundos
        di Las hortalizas serán saludables y ricas en nutrientes durante 4 segundos
      si no
        si < <Humedad del suelo = 60> o <60 < Humedad del suelo> > y <Humedad del suelo < 80> entonces
          di las hortalizas están en muy mal estado, no son aptas para el consumo .
```

Figure 6. Programming routine in mBlock that strengthens computational concepts

4. RESULTS

The evaluation of computational thinking dimensions was conducted by gender for both Industrial Engineering and Systems Engineering programs. The dimensions assessed included computational concepts and practices, each involving specific indicators. Additionally, the level of achievement was evaluated according to the established grading scale for computational thinking dimensions: AD (Outstanding Achievement, 18-20), A (Expected Achievement, 14-17), B (In Progress, 11-13), and C (Beginning, 0-10).

4.1 Student distribution

Table 6 shows the percentage of enrolled students by gender in the Industrial Engineering and Systems Engineering programs. In both the 2020 and 2021 academic periods, the Industrial Engineering program had a higher percentage of female participation compared to 2022. The students from the 2022 period were enrolled in the Systems Engineering program, where it is evident that more women tend to prefer Industrial Engineering, possibly due to its less technical nature compared to Systems Engineering.

Table 6. Percentage of enrolled students by gender

Professional Careers	Men	Women
Industrial Engineering 2022-II	67%	33%
Systems Engineering 2022-I	81%	19%

4.2 Evaluation of computational thinking dimensions by gender and academic program

An inferential analysis was conducted using the Shapiro-Wilk test, which determined that the data for all subgroups did not follow a normal distribution (p-value < 0.05) for students in both Industrial Engineering and Systems Engineering programs. Consequently, the non-parametric Mann-Whitney U test for independent samples was used for each program. Tables 7-10 present the inferential analysis of computational thinking skills, specifically the computational concepts and practices, for students in the Industrial Engineering and Systems Engineering programs.

Table 7. Computational concepts by gender: Industrial Engineering

Computational Concepts	Average		U Statistician	p value
	Women	Men		
Sequences	1.41	1.35	256	0.851
Cycles	2.29	2.90	210	0.310
Events	1.76	1.74	262	0.970
Parallelism	2.12	1.65	222	0.297
Conditionals	1.06	1.74	204	0.139
Operators	1.76	2.52	198	0.060
Data	2.29	1.94	232	0.405

Table 8. Computational practices by gender: Industrial Engineering

Computational Practices	Average		U Statistician	p value
	Women	Men		
Incremental and iterative	1.353	0.581	188	0.057
Testing and debugging	1.235	0.677	2224	0.314
Reusing and remixing	0.824	0.613	258	0.872
Abstracting and modularizing	0.706	0.548	258	0.883

Table 9. Computational concepts by gender: Systems Engineering

Computational Concepts	Average		U Statistician	p value
	Women	Men		
Sequences	1.33	1.16	156	0.645
Cycles	2.67	2.68	170	0.980
Events	2.33	1.66	133	0.227
Parallelism	3.00	2.29	131	0.113
Conditionals	1.67	1.74	167	0.912
Operators	2.33	2.45	165	0.812
Data	2.00	1.42	138	0.310

Table 10. Computational practices by gender: Systems Engineering

Computational Practices	Average		U Statistician	p value
	Women	Men		
Incremental and iterative	0.778	0.895	166	0.894
Testing and debugging	1.556	1.079	128	0.205
Reusing and remixing	1.444	1.026	124	0.134
Abstracting and modularizing	0.556	0.737	161	0.761

The population means for female and male students in the Industrial Engineering program were compared. As shown in Table 7, there are no statistically significant differences between the groups for any of the computational concepts, as indicated by p-values greater than 0.05. This suggests that, based on these data, gender does not significantly influence scores for these computational concepts: sequences, loops, events, parallelism, conditionals, operators, and data.

The population means for female and male students in the Industrial Engineering program were compared. As shown in Table 8, there are no statistically significant differences between the groups for any of the computational practices, as indicated by p-values greater than 0.05. This suggests that, based on these data, gender does not significantly influence scores for these computational practices: incremental and iterative development, testing and debugging, reusing and remixing, and abstracting and modularizing.

The population means for female and male students in the Systems Engineering program were compared. As shown in Table 9, there are no statistically significant differences between the groups for any of the computational concepts, as indicated by p-values greater than 0.05. This suggests that, based on these data, gender does not significantly influence scores for these computational concepts: sequences, loops, events, parallelism, conditionals, operators, and data.

The population means for female and male students in the Systems Engineering program were compared. As shown in Table 10, there are no statistically significant differences between the groups for any of the computational practices, as indicated by p-values greater than 0.05. This suggests that, based on these data, gender does not significantly influence scores for these computational practices: incremental and iterative development, testing and debugging, reusing and remixing, and abstracting and modularizing.

4.3 Achievements in the development of computational thinking dimensions by gender and academic program

Figures 7 and 8 show the graphs corresponding to the achievement levels in computational concepts and practices of students from the Industrial and Systems Engineering programs, disaggregated by gender.

Figure 7, which presents the achievement levels in computational concepts by gender and academic program, reveals a more balanced distribution across the different performance levels. In Industrial Engineering, both women and men show significant percentages in all levels. Women stand out with 40% at the beginning level, 7% in progress, 40% at the expected achievement level, and 13% at the outstanding level. In contrast, men have a lower percentage at the beginning level (16%), but higher percentages in progress

(27%) and outstanding (18%).

In Systems Engineering, no women are found at the beginning level, while 67% reach the expected achievement level and 22% reach the outstanding level. Meanwhile, men show 21% at the beginning level, 24% in progress, 39% at the expected level, and 16% at the outstanding level. Overall, these results reflect a more evenly distributed performance across all levels compared to computational practices, and a greater presence of students—especially women in Systems Engineering—at the higher levels of conceptual achievement.

Figure 8 illustrates the achievement levels in computational practices by gender and academic program, and reveals that the majority of students are at the beginning level. In Industrial Engineering, 86% of women and 94% of men fall into this category. However, among women in this program, a greater diversity in performance levels is observed, with 7% in progress and another 7% achieving the outstanding level, while 6% of men reach the expected level. In the case of Systems Engineering, 100% of women and 95% of men are at the beginning level, with the latter group being the only one to show 5% in progress. Overall, the results reflect a predominant concentration at the beginning level in computational practices, with minimal representation in higher achievement levels, especially in Systems Engineering.

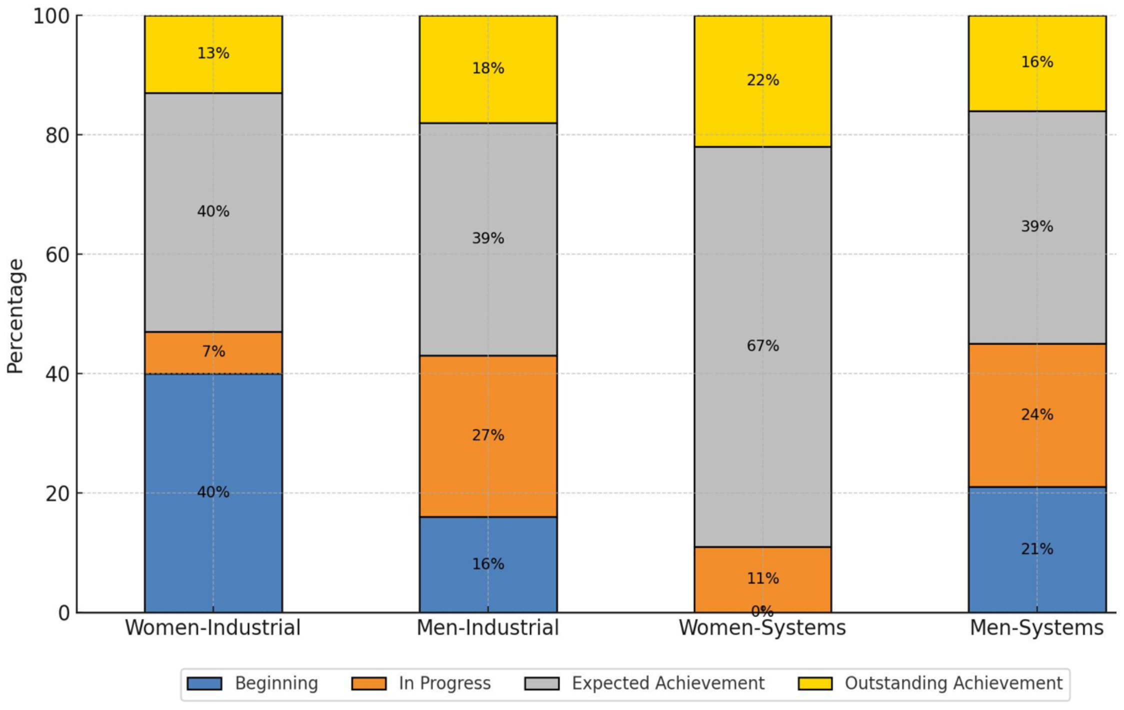


Figure 7. Achievement levels in computational concepts by gender and academic program

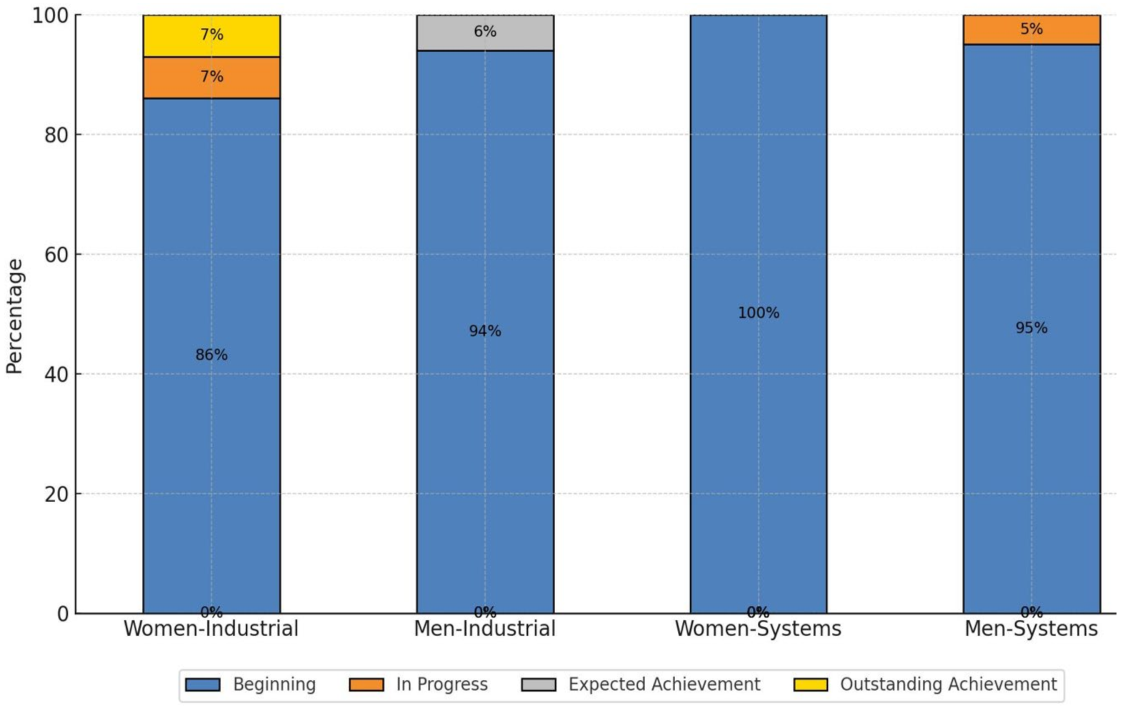


Figure 8. Achievement levels in computational practices by gender and academic program

5. DISCUSSION

The results show that while there are no statistically significant differences between male and female students in the dimensions of computational thinking—aligned with previous studies [52, 53]—there are notable distinctions when disaggregating achievement levels by gender and academic program. Specifically, women in the Systems Engineering program exhibit higher conceptual achievement levels, with 67% reaching the expected level and 22% the outstanding level. None remained at the beginning level. This contrasts with the 45% of male students in the same program who were at the beginning or in-progress stages.

In Industrial Engineering, both genders present more evenly distributed performance in conceptual understanding. Female students stand out with 40% in the beginning level and another 40% achieving the expected level, while 13% reached the outstanding level. Among males, 27% were in progress and 18% in the outstanding level. These results confirm that women can perform equally or even better in conceptual computational thinking when provided with supportive and inclusive environments [54, 55].

However, achievement in computational practices reveals a persistent concentration in the beginning level across all groups. In Systems Engineering, 100% of women and 95% of men remained in the beginning stage, with only 5% of men progressing. Similarly, in Industrial Engineering, 86% of women and 94% of men were in the beginning level. Only a small percentage of women (7%) reached the outstanding level, and 6% of men achieved the expected level. These data suggest a gap between conceptual understanding and practical application, as also reported in references [56, 57].

This trend can be explained by the high cognitive demand involved in computational practices, such as incremental and iterative design, testing and debugging, reusing and remixing solutions, as well as abstraction and modularization. These practices require not only technical skills but also advanced reasoning processes, complex problem-solving, and strategic thinking, which develop progressively through greater exposure to real-world practical scenarios [58]. Unlike computational concepts, which can be addressed through a more structured and sequential logic, computational practices demand greater autonomy, experimentation, and critical reflection, which may have hindered their mastery during the short intervention period. In this regard, the results do not indicate an ineffective intervention, but rather the need for a pedagogical approach that reinforces teacher guidance, iterative practical activities, and the use of cognitive scaffolding to support the progressive development of these competencies [59-61].

The use of project-based and community-driven technological activities in this study was a key strategy for engaging students equally. These projects, which focused on environmental protection and solving local problems, helped neutralize gendered preferences that may arise in activities like robotics [62, 63]. Women showed strong motivation when using microcontrollers and sensors with visual feedback, allowing them to experiment and improve functionality [33, 37]. Additionally, the block-based visual programming environment supported their creativity through interactive scenes and storytelling, which, according to Sáinz and Meneses [36], aligns with how many women prefer to express their creativity.

Finally, gender-based strengths were observed during classroom execution of technological projects. Men were more practical in executing tasks but showed less organizational structure, whereas women excelled in planning and teamwork—skills that enhanced collaborative learning and strengthened computational thinking across both genders [64]. These results reinforce the importance of designing learning environments that combine conceptual development with inclusive, practice-oriented experiences to close gender gaps in computational education [17].

6. CONCLUSIONS

The implementation of technology-based projects focused on real community challenges promoted the development of computational thinking among students in Industrial and Systems Engineering. Key concepts such as sequences, conditionals, loops, data, and parallelism—as well as computational practices like testing, debugging, and modularization—were strengthened through the use of the STEM Kit and the mBlock visual programming environment.

The results show that 67% of women in Systems Engineering reached the expected level in computational concepts, and 22% reached the outstanding level; none remained at the beginning level. In contrast, 45% of male students in the same program were still at the beginning or in-progress levels. In Industrial Engineering, female students achieved 40% at the expected level and 13% at the outstanding level, also outperforming their male counterparts. These findings highlight the potential of female students to reach high levels of conceptual understanding in inclusive educational settings.

However, over 90% of students, regardless of gender or academic program, remained at the beginning level in computational practices. This gap may be attributed to the high cognitive demand of these practices, which require autonomy, advanced reasoning, and real-world problem-solving experience. Therefore, there is a need to reinforce pedagogical strategies through structured guidance, scaffolding, and extended hands-on practice to support the progressive development of these skills.

The STEM Kit proved to be an effective educational tool, enabling learning through sensors, actuators, and an accessible visual programming interface. In addition to strengthening technical skills, it sparked greater motivation and interest among female students, fostering their inclusion in STEM fields.

It is recommended to integrate contextualized technological projects and visual programming environments from the early stages of university education. This approach can enhance students' scientific and technical competencies and contribute to closing gender gaps in STEM. Furthermore, the methodology is adaptable to basic education in areas such as Science, Technology and Environment, computing, and vocational orientation.

REFERENCES

[1] Torres-Torres, Y.D., Román-González, M., Pérez-González, J.C. (2020). Unplugged teaching activities to promote computational thinking skills in primary and adults from a gender perspective. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, 15(3):

225-232. https://doi.org/10.1109/RITA.2020.3008338

[2] UNICEF Perú, 200 adolescentes mujeres de Lima Norte reciben laptops para convertirse en programadoras web. https://www.unicef.org/peru/comunicados-prensa/200-adolescentes-mujeres-lima-norte-laptops-programadora-web-stem-ciencia-tecnologia-tic, accessed on May 16 2022.

[3] Master, A., Cheryan, S., Moscatelli, A., Meltzoff, A.N. (2017). Programming experience promotes higher STEM motivation among first-grade girls. Journal of Experimental Child Psychology, 160: 92-106. https://doi.org/10.1016/j.jecp.2017.03.013

[4] UNESCO, Girls' and women's education in science, technology, engineering and mathematics (STEM). https://www.unesco.org/en/gender-equality/education/stem?utm_source=chatgpt.com, accessed on Aug. 4, 2024.

[5] Sáez-López, J.M., Román-González, M., Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. Computers & Education, 97: 129-141. https://doi.org/10.1016/j.compedu.2016.03.003

[6] Wagner, M. Girls Who Game Empowers the Next Generation in STEM. Innovation. https://www.dell.com/en-us/blog/girls-who-game-empowers-the-next-generation-in-stem/, accessed on Apr. 26, 2025.

[7] OECD Indicators. (2021). Education at a Glance 2021: OECD Indicators. in Education at a Glance. Report, OECD. https://doi.org/10.1787/b35a14e5-en

[8] Programa Nacional de Becas y Crédito Educativo. (2016). MEMORIA INSTITUCIONAL 2012 - 2015 PROGRAMA NACIONAL DE BECAS Y CRÉDITO EDUCATIVO MINISTERIO DE EDUCACIÓN. Lima.

[9] CONCYTEC, Perú: Principales Indicadores en CTI. https://portal.concytec.gob.pe/indicadores/principales/, accessed on Oct. 3, 2024.

[10] UNESCO, Descifrar el código: La educación de las niñas y las mujeres en ciencias, tecnología, ingeniería y matemáticas (STEM). 2019. https://unesdoc.unesco.org/ark:/48223/pf0000366649?posInSet=1&queryId=d5f381da-86f6-442b-8f3b-a86a83220043.

[11] 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

[12] Dong, Y., Catete, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R., Andrews, A. (2019). PRADA: A practical model for integrating computational thinking in K-12 education. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education, pp. 906-912. https://doi.org/10.1145/3287324.3287431

[13] Paucar-Curasma, R., Villalba-Condori, K.O., Mamani-Calcina, J., 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

[14] 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

[15] Paucar-Curasma, R., Villalba-Condori, K.O., Jara, N.J., Llamoca, R.Q., Tovar, R.F.U., Malpartida, K.F.C. (2022). Technological project in the development of computational thinking and problem-solving. In 2022 XVII Latin American Conference on Learning Technologies (LACLO), Armenia, Colombia, pp. 1-6. https://doi.org/10.1109/LACLO56648.2022.10013445

[16] Paucar-Curasma, R., Cerna-Ruiz, L.P., Acra-Despradel, C., Villalba-Condori, K.O., Massa-Palacios, L.A., Olivera-Chura, A., Esteban-Robladillo, I. (2023). Development of computational thinking through STEM activities for the promotion of gender equality. Sustainability, 15(16): 12335. https://doi.org/10.3390/su151612335

[17] Cheryan, S., Ziegler, S.A., Montoya, A.K., Jiang, L. (2017). Why are some STEM fields more gender balanced than others? Psychological Bulletin, 143(1): 1. https://doi.org/10.1037/bul0000052

[18] Rojas López, A. (2019). Escenarios de aprendizaje personalizados a partir de la evaluación del pensamiento computacional para el aprendizaje de competencias de programación mediante un entorno b-Learning y gamificación. https://doi.org/10.14201/gredos.140444

[19] Terreni, L.G. (2021). Enseñanza del pensamiento computacional en la educación superior mediada por tecnología. Docentes Conectados, 4(8): 18-26.

[20] Urquizo, G., Vidal, E., Castro, E. (2021). Incorporación de Pensamiento Computacional en Ingenierías como soporte a la competencia de Desarrollo de Problemas: Jugando con Lightbot. Revista Ibérica de Sistemas e Tecnologias de Informação, (E42): 199-207.

[21] Bordignon, F.R.A., Iglesias, A.A. (2018). Introducción al Pensamiento Computacional. Argentina: EDUCAR S.E.

[22] Brennan, K., Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada, 1: 25.

[23] Demir-Kaymak, Z., Duman, I., Randler, C., Horzum, M.B. (2022). The effect of gender, grade, time and chronotype on computational thinking: Longitudinal study. Informatics in Education, 21(3): 465-478. https://doi.org/10.15388/infedu.2022.22

[24] Master, A., Meltzoff, A.N., Cheryan, S. (2021). Gender stereotypes about interests start early and cause gender disparities in computer science and engineering. Proceedings of the National Academy of Sciences, 118(48): e2100030118. https://doi.org/10.1073/pnas.2100030118

[25] Angeli, C., Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. Computers in Human Behavior, 105: 105954. https://doi.org/10.1016/j.chb.2019.03.018

[26] Román-González, M., Pérez-González, J.C., Moreno-León, J., Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. Computers in Human Behavior, 80: 441-459. https://doi.org/10.1016/j.chb.2017.09.030

[27] Niousha, R., Saito, D., Washizaki, H., Fukazawa, Y. (2022). Gender characteristics and computational thinking in Scratch. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education, pp. 1344-1344. https://doi.org/10.1145/3545947.3576290

[28] Papavlasopoulou, S., Sharma, K., Giannakos, M.N. (2020). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. Computers in Human Behavior, 105: 105939. https://doi.org/10.1016/j.chb.2019.03.003

[29] Hsu, T.C., Chang, C., Wong, L.H., Aw, G.P. (2022). Learning performance of different genders' computational thinking. Sustainability, 14(24): 16514. https://doi.org/10.3390/su142416514

[30] Buffum, P.S., Frankosky, M., Boyer, K.E., Wiebe, E.N., Mott, B.W., Lester, J.C. (2016). Collaboration and gender equity in game-based learning for middle school computer science. Computing in Science & Engineering, 18(2): 18-28. https://doi.org/10.1109/MCSE.2016.37

[31] Jiang, S., Wong, G.K. (2022). Exploring age and gender differences of computational thinkers in primary school: A developmental perspective. Journal of Computer Assisted Learning, 38(1): 60-75. https://doi.org/10.1111/jcal.12591

[32] PROCIENCIA-CONCYTEC, Investigadores de Huancavelica desarrollan dispositivo educativo para facilitar aprendizaje sencillo para universitarios - Noticias - Programa Nacional de Investigación Científica y Estudios Avanzados - Plataforma del Estado Peruano. https://www.gob.pe/institucion/prociencia/noticias/919234, accessed on Jun. 19, 2025.

[33] Kafai, Y.B., Burke, Q. (2014). Connected Code: Why Children Need to Learn Programming. MIT Press, Cambridge. https://doi.org/10.7551/mitpress/9992.001.0001

[34] Fischer, J., Romahn, N., Weinert, M. (2020). Fostering reflection in CS teacher education. A video-based approach to unveiling, analyzing and teaching novices' programming processes. In ISSEP (CEURWS Volume), pp. 128-139.

[35] Paucar-Curasma, R., Tovar, R.F.U., Jiménez, S.C.A.V., Jara, N.J., Agama, S.H.G., Quijada-Villegas, J.H. (2024). Developing IoT activities using the problem-solving method: Proposal for novice engineering students. Mathematical Modelling of Engineering Problems, 11(11): 3161-3172. https://doi.org/10.18280/mmep.111126

[36] Sáinz, M., Meneses, J. (2018). Brecha y sesgos de género en la elección de estudios y profesiones en la educación secundaria. Panorama Social, 27: 23-31.

[37] Ching, Y.H., Hsu, Y.C., Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. TechTrends, 62(6): 563-573. https://doi.org/10.1007/s11528-018-0292-7

[38] Makeblock Education, What is mBlock-Python Editor? Makeblock. https://education.makeblock.com/help/mblock-python-editor-what-is-mblock-python-editor/, accessed on Apr. 2, 2022.

[39] Weese, J.L. (2016). 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, pp. 279-280. https://doi.org/10.1145/2960310.2960347

[40] Makeblock, My Blocks Category. https://www.mblock.cc/doc/en/block-reference/DIY.html, accessed on Apr. 3, 2022.

[41] 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

[42] Romero, M., Lepage, A., Lille, B. (2017). Computational thinking development through creative programming in higher education. International Journal of Educational Technology in Higher Education, 14(1): 42. https://doi.org/10.1186/s41239-017-0080-z

[43] Melo, J., Fidelis, M., Alves, S., Freitas, U., Dantas, R. (2020). A comprheensive review of visual programming tools for arduino. In 2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE), Natal, Brazil, pp. 1-6. https://doi.org/10.1109/LARS/SBR/WRE51543.2020.9307023

[44] Fronza, I., Corral, L., Pahl, C. (2019). Combining block-based programming and hardware prototyping to foster computational thinking. In Proceedings of the 20th Annual SIG Conference on Information Technology Education, pp. 55-60. https://doi.org/10.1145/3349266.3351410

[45] Kuz, A., Ariste, M.C. (2022). Analysis and review of educational software to learn to program in playful environments. Tecné, Episteme y Didaxis: TED, (52): 117-136. https://doi.org/10.17227/ted.num52-13159

[46] 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

[47] Basogain, X., Olabe, M.Á., Olabe, J.C., Rico, M.J. (2018). Computational thinking in pre-university blended learning classrooms. Computers in Human Behavior, 80: 412-419. https://doi.org/10.1016/j.chb.2017.04.058

[48] João, P., Nuno, D., Fábio, S.F., Ana, P. (2019). A cross-analysis of block-based and visual programming apps with computer science student-teachers. Education Sciences, 9(3): 181. https://doi.org/10.3390/educsci9030181

[49] Román-Gonzalez, M. (2015). Test de pensamiento computacional: Principios de diseño, validación de contenido y análisis de ítems. pp. 1-19.

[50] Villalustre-Martínez, L. (2024). Analysis of the level of computational thinking of future teachers: A diagnostic proposal for the design of training actions. Pixel-Bit. Revista de Medios y Educación, 69: 169-194. https://doi.org/10.12795/pixelbit.101205

[51] 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, pp. 80-87. https://doi.org/10.1145/2818314.2818315

[52] Rodríguez-Abitia, G., Ramírez-Montoya, M.S., López-Caudana, E.O., Romero-Rodríguez, J.M. (2021). Factores para el desarrollo del pensamiento

computacional en estudiantes de pregrado. Campus Virtuales, 10(2): 153-164.

[53] Alvarez, R.H., León, C., Miranda, G., Segredo, E., Socas, O., Cuellar-Moreno, M., Caballero-Julia, D., García, L., Dıaz, Y. (2021). Promocion del Pensamiento Computacional en estudiantes pre-universitarios: ¿como se emocionan? Actas de las Jenui, 6: 219-226.

[54] Denner, J., Werner, L., Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? Computers & Education, 58(1): 240-249. https://doi.org/10.1016/j.compedu.2011.08.006

[55] Tan, W.L., Samsudin, M.A., Ismail, M.E., Ahmad, N.J. (2020). Gender differences in students' achievements in learning concepts of electricity via STEAM integrated approach utilizing scratch. Problems of Education in the 21st Century, 78(3): 423-448. https://doi.org/10.33225/pec/20.78.423

[56] 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

[57] Barr, V., Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM Inroads, 2(1): 48-54. https://doi.org/10.1145/1929887.1929905

[58] Fallas, J.M., Chavarría, L.S., Rodríguez, C.G., Torres, A.G., Zelaya, I.T. (2025). Fundamentos y perspectivas del pensamiento computacional: Un análisis integral para la investigación futura. Tecnología en Marcha, 38(1): 145-156. https://doi.org/10.18845/tm.v38i1.7055

[59] Wing, J.M. (2006). Computational thinking. Communications of the ACM, 49(3): 33-35. https://doi.org/10.1145/1118178.1118215

[60] Wing, J.M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366(1881): 3717-3725. https://doi.org/10.1098/rsta.2008.0118

[61] Wing, J.M. El pensamiento computacional 10 años después – Microsoft En Español. Microsoft, TechNet. https://blogs.technet.microsoft.com/microsoftlatam/2016/04/18/el-pensamiento-computacional-10-aos-despus/, accessed on Jun. 28, 2019.

[62] 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

[63] 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

[64] Espino, E., González, C. (2016). Estudio sobre pensamiento computacional y género. Revista Iberoamericana de Tecnologias del Aprendizaje, 4(3): 119-128.