IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 12, No. 5, May, 2025, pp. 1594-1602
Journal homepage: http://iieta.org/journals/mmep

Playful Tools to Foster the Development of Computational Thinking in School-Age Children in the Andes of Peru

Ronald Paucar Curasma[1*], Claudia Acra-Despradel[2], Roberto Florentino Unsihuay Tovar[3], Omar Felix Illesca Cangalaya[3], Nolan Jara Jara[3], Rosa Quispe Llamoca[4], Juan Humberto Quijada Villegas[5], Klinge Orlando Villalba-Condori[6]

[1] ICT Research Group Applied to Society, Universidad Nacional Autónoma de Tayacaja Daniel Hernández Morillo, Pampas 09156, Peru
[2] Vicerrectorado de Investigación, Universidad Nacional Pedro Henríquez Ureña, Santo Domingo 10203, Dominican Republic
[3] Facultad de Ingeniería Electrónica y Eléctrica, Universidad Nacional Mayor de San Marcos, Lima 15001, Peru
[4] Carrera de Economía, Universidad de Lima, Surco 15023, Peru
[5] Facultad de Ingeniería de Sistemas, Universidad Nacional del Centro del Perú, Huancayo 12001, Peru
[6] Vicerrectorado de Investigación, Universidad Católica de Santa María, Arequipa 04001, Peru

Corresponding Author Email: rpaucarc@unat.edu.pe

https://doi.org/10.18280/mmep.120514

Received: 16 February 2025
Revised: 3 May 2025
Accepted: 8 May 2025
Available online: 31 May 2025

Keywords:
computational thinking, computational thinking skills, Lightbot, mBlock, school-age children

ABSTRACT

This study describes the design, implementation, and evaluation of a computational thinking workshop for children in the Andean regions of Peru, using playful tools such as Lightbot and mBlock. The objective was to strengthen skills such as algorithmic design, decomposition, debugging, loops, and pattern recognition. The participants were children aged 8 to 11 years, who engaged in activities aimed at developing these abilities. Descriptive analyses and statistical tests (Student's t-test) were applied to identify gender-based differences. The results showed that both girls and boys performed similarly, with no significant differences, suggesting that these tools promote gender equity in rural educational settings. Furthermore, according to the Peruvian educational evaluation scale, the skills of loops and debugging were identified as needing greater reinforcement, as they were concentrated in the lowest performance category (Category C). The study concludes that this type of workshop can help reduce the digital divide and strengthen 21st-century skills in vulnerable contexts, contributing to a more inclusive and contextualized education.

1. INTRODUCTION

In Latin America, computational thinking workshops are mostly carried out in urban areas and often require participation fees. Although there is growing expectation for teachers to incorporate technology into their classrooms, computational thinking and programming are not mandatory subjects in public institutions. This hinders the development of computational skills in Latin American children, and it is expected that the digital divide between urban and rural populations will continue to widen over time, especially in socioeconomically vulnerable communities [1, 2].

Currently, in the Peruvian educational system [1], most primary-level schools follow a traditional teaching model that excludes the use of technology and, even more so, the integration of computational thinking. This form of instruction fails to foster creative and innovative capacities in children, limiting knowledge generation and often leading to student frustration. This situation is especially pronounced in the region of Huancavelica, located in the Andes of Peru, where the geography is rugged and where, in terms of development in education, health, and related indicators, the region is often ranked among the lowest in the country.

We believe it is essential to develop computational thinking skills in rural areas as a means of promoting equity with children from urban zones. Moreover, various researchers argue that computational thinking helps children develop key competencies such as the ability to deal with complex problems, persevere in daily activities, work collaboratively with gender equity, and build confidence in managing complexity [2]. In this regard, computational thinking is a key skill for solving complex problems, interpreting data, and communicating results to others through computers or other agents [3].

To obtain reliable and meaningful results regarding the development of computational thinking among children and adolescents in Huancavelica, appropriate digital tools were selected to promote motivation and support playful learning experiences. These tools were combined with educational applications that allow for a gradual increase in complexity to enhance the development of computational thinking [4, 5].

This study adopts a quantitative approach with a quasi-experimental design, as it involved a non-randomized group and sought to observe the effects of an educational intervention on computational thinking. This design is well-suited for real-world contexts where complete variable control is not feasible, yet cause-effect relationships can still be explored. To assess computational thinking, an adapted version of the Computational Thinking Test (CTt), developed by Román-González [6], was used due to its scientific validity, age appropriateness (for ages 8 to 16), and its ability to directly measure key computational skills such as algorithmic design, loops, debugging, decomposition, and pattern recognition. The test was shortened to 20 items to facilitate application with students aged 8 to 11. The results were analyzed using descriptive statistics and Student's t-test for independent samples, enabling a comparison between boys and girls. Finally, the scores were interpreted using the Peruvian educational system's qualitative scale [7], which categorizes performance into levels AD (Outstanding Achievement), A (Expected Achievement), B (In Progress), and C (Beginning), thereby providing meaningful pedagogical interpretation within school environments.

2. RELATED WORK

2.1 Computational thinking in the school

In the late 1950s, Alan Perlis, a pioneer in computing, emphasized the value of coding as a mental tool to understand problems [8]. During the 1960s, various authors argued that computing provides cognitive tools that are useful in everyday life [9]. In 1967, Papert developed Logo, a programming environment that allowed children to program a turtle [10]; this tool was considered a powerful pedagogical resource that complemented Polya's proposal on the phases of problem solving: understanding the problem, planning, execution, and reviewing the solution [11, 12].

Building on these developments, the term computational thinking has gained relevance, involving the use of fundamental computer science concepts [13], and is now seen as an essential 21st-century skill, on par with reading, writing, or arithmetic [14].

According to a widely cited definition [15], computational thinking encompasses problem-solving, system design, and understanding human behavior using core principles of computing. Within this framework, abstraction is highlighted as a key component for effectively addressing problems [16-18]. Furthermore, computational thinking supports the development of creativity, critical thinking, teamwork, and the ability to deal with complex situations [2, 19].

In the past ten years, definitions of computational thinking have expanded, with many researchers highlighting frameworks based on multiple categories [20, 21], including problem-solving applied to academic disciplines and real-world issues, block-based programming, and the three dimensions proposed by Brennan and Resnick [22]: computational concepts, practices, and perspectives. Operational definitions, in turn, refer to it as a set of sequential processes in problem-solving: formulation, organization, analysis, automation, representation, implementation, and transfer [23, 24].

2.2 Tools to develop programming: Lightbot and mBlock

The Lightbot APP [25] is an educational tool for mobile devices designed to teach children the basics of programming and foundational computer science principles. Its gameplay mechanics, centered around guiding a robot to solve puzzles, help children become familiar with essential programming concepts such as loops, conditionals (e.g., "if-then"), and procedures, all without the need to write actual code. Lightbot is organized into progressively challenging levels: Basic, Procedures, and Loops. Each level includes a set of exercises with visual commands, making it a playful and intuitive way to introduce programming. The APP is free and widely recognized as an effective resource for training computational thinking in early and primary education.

mBlock [26], on the other hand, is a visual programming platform based on Scratch 2.0 and developed by Makeblock. It is aimed at teaching STEAM competencies and has become one of the most prominent tools for learning coding in school contexts. mBlock enables children to design games, interactive stories, and simulations by dragging and dropping colored blocks that represent programming instructions, making it more accessible than traditional text-based languages. The platform promotes creativity, logical thinking, and collaborative learning, and also encourages students to engage with a global community by sharing their projects online. It is recommended for use from grades 3 to 6 of primary education as part of computational thinking and coding programs [27-34].

Table 1. Computational thinking skills

Computational Thinking Skills	Definition
Algorithmic design	The ability to design step by step an operation/action on how problems are solved.
Pattern recognition	Discover similarities or patterns in any complex problem or decomposed problem.
Decomposition	The ability to break down complex problems into simpler ones that are easier to solve.
Cycle or loops	Sequence of code instructions that is executed repeatedly.
Depuration	The ability to identify, eliminate, and correct errors.

2.3 Computational thinking skills

Currently, there are various approaches to computational thinking skills in school education. Puhlmann [35] identified four core skills: abstraction, decomposition, pattern recognition, and algorithmic design. He also proposes that these skills are structured into three computational dimensions: concepts (sequence, loops, events, parallelism, conditionals, operators, and data), practices (experimentation and interaction, testing and debugging, reuse of previous projects, and abstraction/modularization), and perspectives (forms of self-expression, connection with others, and questioning). Several authors agree that abstraction is one of the most essential skills in computational thinking, especially in the formulation of problems [14, 36], while problem-solving involves a set of steps and computational algorithms that encompass processes such as formulation, organization, analysis, automation, representation, implementation, and transfer [23, 24, 37]. In this study, the Lightbot APP helped develop skills such as algorithmic design, decomposition, and pattern recognition, as well as the computational concept of loops and the practice of testing and debugging [25, 38].

Meanwhile, mBlock enabled the comprehensive application of the three dimensions of computational thinking proposed by Puhlmann [35], through activities that integrate concepts, practices, and perspectives. Table 1 summarizes the specific computational thinking skills that can be developed using Lightbot and mBlock.

3. METHODOLOGY

3.1 Participants and research approach

The children who participated in the study attended various primary schools located in rural areas of the province of Huancavelica, a region characterized by high poverty rates and limited access to technological resources. In total, 11 children (6 boys and 5 girls), aged between 8 and 11 years, took part. Most of them come from families engaged in subsistence farming, with monthly incomes below the minimum wage and limited access to computers or a stable internet connection at home. Regarding prior experience, none of the students had formal exposure to programming environments or educational robotics before the workshop. Their contact with technology was limited to occasional use of mobile devices at home or in public internet cafes. This characterization helps to understand the scope and limitations of the intervention, as well as the generalizability of the findings to populations in similar conditions in the Peruvian Andes or other rural regions in Latin America.

This research was conducted under a quantitative approach with a quasi-experimental design, working with a single group without random assignment, with the purpose of observing the effects of an educational intervention based on playful programming tools such as Lightbot and mBlock. This type of design allows for the analysis of intervention impacts in real-world contexts, especially in vulnerable populations where establishing control groups is not feasible.

Data collection was carried out using an adapted version of the Computational Thinking Test (CTt), developed by Román-González [6], a validated instrument that directly assesses skills such as algorithmic design, debugging, loops, decomposition, and pattern recognition. The test was reduced to 20 items to suit the cognitive level of the children (aged 8 to 11), allowing for feasible application within school sessions. Table 2 shows the test items related to the skills of algorithmic design, decomposition, pattern recognition, loops, and debugging [35, 39].

Table 2. Test items to evaluate computational thinking skills

Computational Thinking Skills	Evaluated by Marcos Román-González Test?	Marcos Román-González Test Items
Algorithmic design	Yes	1 to 6; 8 to 20
Pattern recognition	Yes	4 to 6; 8 to 12; 14, 15, 17 and 18
Decomposition	Yes	4 to 7; 10 to 13 and 15
Cycle or loops	Yes	5 al 20
Debugging	Yes	3, 7, 11, 16 and 19

The data were processed using descriptive statistics and inferential analysis, applying Student's t-test for independent samples to identify significant differences between boys and girls. Additionally, the scores were interpreted using the qualitative scale of the Peruvian educational system [40], which classifies student performance into four levels: AD (Outstanding Achievement), A (Expected Achievement), B (In Progress), and C (Beginning).

This methodological approach not only allowed for the evaluation of computational thinking skills in a rural context, but also enabled the analysis of gender equity and the identification of specific pedagogical reinforcement needs, providing contextualized evidence on the use of educational technologies in highly vulnerable areas.

3.2 Workshop proposal for strengthening computational thinking skills

The workshop was implemented as part of the university social responsibility (RSU) project of the Universidad Nacional Autónoma de Tayacaja Daniel Hernández Morillo, entitled "Workshop on 21st-Century Skills in the Huancavelica Region." The program lasted 16 hours, held twice a week with 2-hour sessions. It was delivered in both asynchronous and synchronous formats, using Google Meet for videoconferencing, as illustrated in Figure 1.

Figure 1. Children participating in the workshop

Table 3. Structure of the workshop and skills developed by students

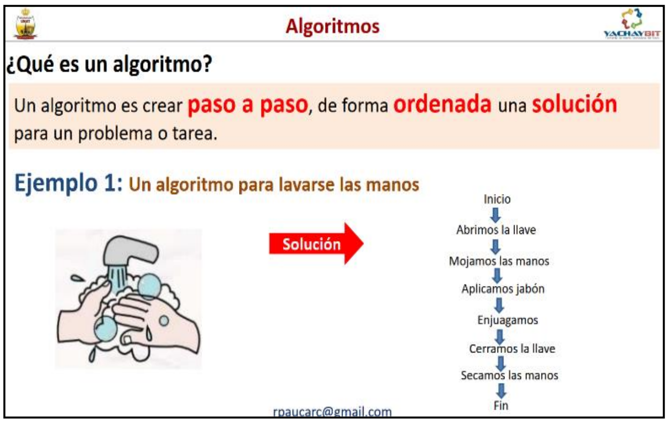
Workshop Activity	Tool	Skills Developed
Real-life algorithms (paper and pencil)	None	Algorithmic design
Programming in Lightbot – Basic Level	Lightbot	Algorithmic design, debugging
Programming in Lightbot – Procedures	Lightbot	Decomposition, pattern recognition
Programming in Lightbot – Loops	Lightbot	Loops, debugging
Game creation (Ping-Pong, Scissors Dancer)	mBlock	Algorithmic design, decomposition, pattern recognition, loops, debugging

Table 3 presents the structure of the workshop and the computational thinking skills addressed in the students. The workshop began with the definition of an algorithm and its application in real-life situations, followed by robot programming using Lightbot in Basic Mode (levels 1 to 8) and Procedure Mode (levels 1 to 6). The next stage involved working with mBlock, starting with an introduction to the development environment and its various programming blocks. This was followed by the creation of interactive applications such as the "Ping-Pong" game and the "Scissors Dancer" animation.

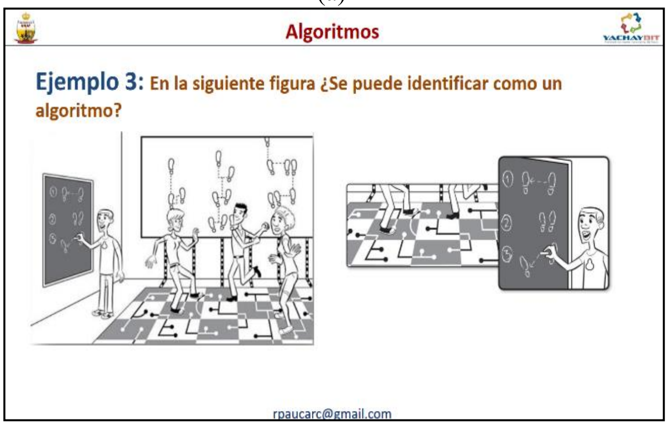
The following section outlines the activities carried out for each computational thinking skill.

3.2.1 Algorithmic design

To help children grasp the concept of algorithms, they were encouraged to generate real-life examples illustrating how algorithms are used in everyday situations. As shown in Figure 2, the students then developed their own algorithmic exercises using pencil and paper, systematically following a series of steps until reaching a solution.



(a)



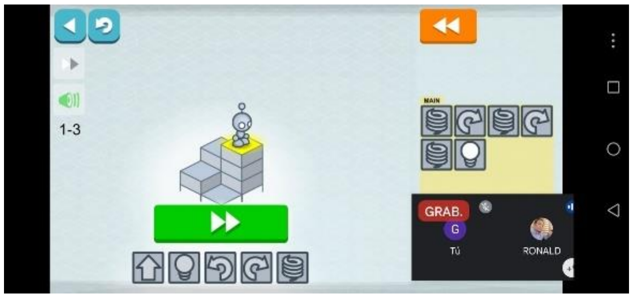
(b)

Figure 2. Algorithms session: a) Algorithm definition, b) Algorithm examples session

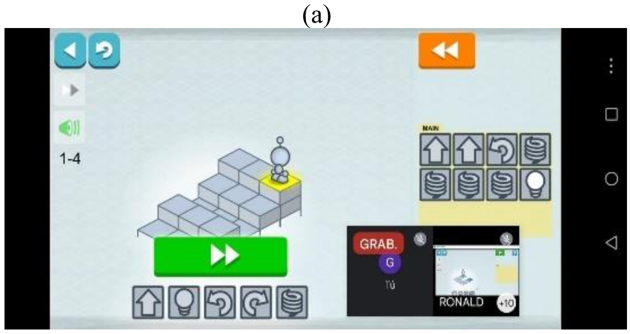
During the algorithm practice sessions with Lightbot, the children were introduced to the fundamentals of algorithmic design. They worked through the BASIC level of Lightbot, which consisted of eight exercises (1-1 to 1-8) focused specifically on developing algorithmic thinking. The children followed four key programming steps: planning, coding, execution, and debugging, to successfully program the robot. As shown in Figure 3, they programmed the robot to illuminate the blue tile using commands such as "Walk forward," "Light up," "Turn left," "Turn right," and "Jump." The students constructed step-by-step sequences of instructions and inserted them in order within the MAIN area until they successfully completed the task of lighting the blue box.

3.2.2 Pattern recognition

This section addresses a computational thinking skill developed during the PROCEDURES level of Lightbot, which includes four sub-levels (2-1 to 2-3). At this stage, children continue to apply the programming cycle—planning, coding, execution, and debugging—for each exercise. As shown in Figure 4, the virtual robot is programmed through a sequence of commands or instructions. A key feature introduced at this level is the use of the P1 command, which allows students to define reusable sets of instructions. To program the robot effectively, children first identify recurring patterns in the tasks. These repeated patterns are then coded within the PROC1 area and subsequently inserted into the MAIN area to complete the exercise efficiently.

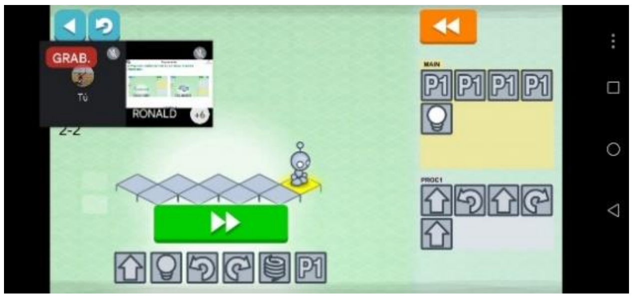


(a)

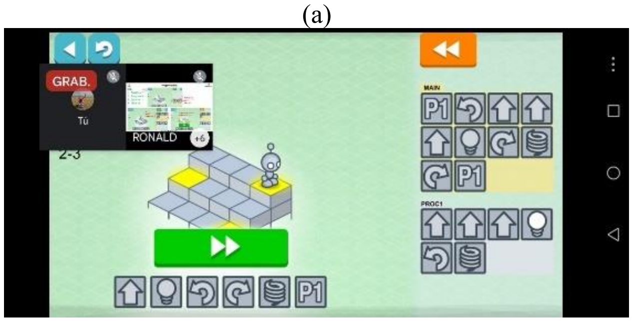


(b)

Figure 3. Algorithmic exercises with Lightbot: a) Basic level exercise 3, b) Basic level exercise 4



(a)

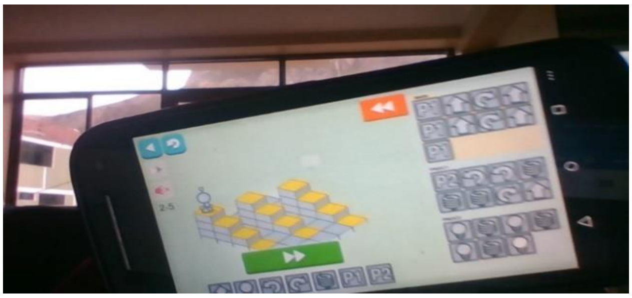


(b)

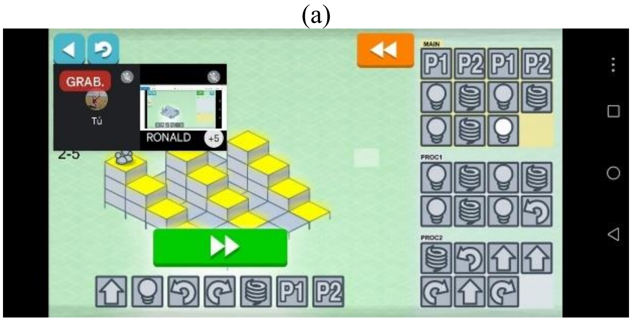
Figure 4. Pattern recognition exercises with Lightbot: a) Procedure level exercise 2, b) Procedure level exercise 3

3.2.3 Decomposition

This section describes a computational thinking skill developed through the PROCEDURES level of Lightbot, specifically within exercises 2-4 to 2-6. In these sublevels, children learn to program a virtual robot by following the standard programming cycle: planning, coding, execution, and debugging. As illustrated in Figure 5, the robot is programmed using a sequence of commands and instructions. At this stage, two procedure commands, P1 and P2, are introduced. To complete the exercises, students first identify tasks that can be decomposed into smaller parts with potential for reuse. These decomposed tasks are then programmed in the PROC1 and PROC2 areas and subsequently inserted into the MAIN area. As shown, children apply the reusable procedures P1 and P2 multiple times within the MAIN program. Through this process, they learn how to effectively break down problems and design efficient, modular programs.

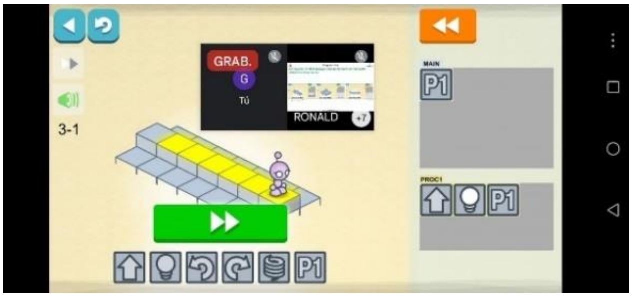


(a)

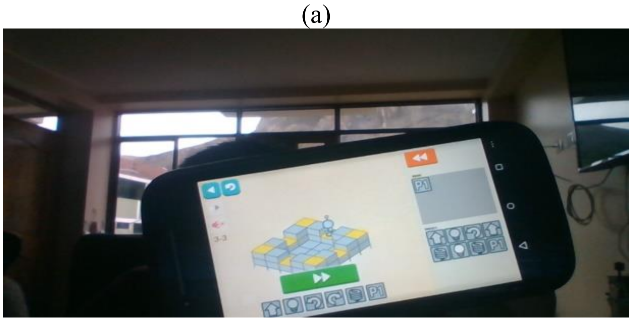


(b)

Figure 5. Decomposition exercises with Lightbot: a) Procedure level exercise 5, b) Procedure level exercise 4



(a)



(b)

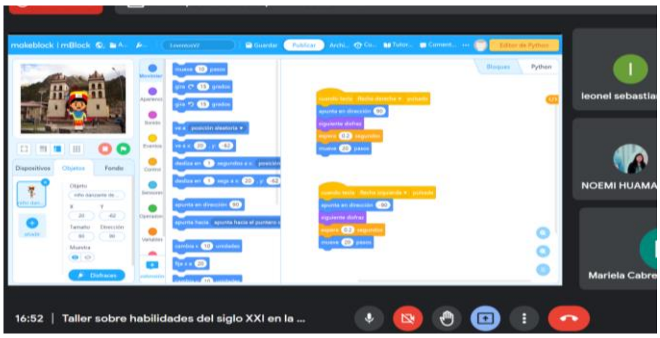
Figure 6. Loops exercises with Lightbot: a) Loop level exercise 1, b) Loop level exercise 3

3.2.4 Cycles or loops

This section highlights a computational thinking skill developed through the LOOPS level of Lightbot, which includes five exercises (3-1 to 3-5). In each activity, children followed the four-step programming cycle: planning, coding, execution, and debugging. As illustrated in Figure 6, the virtual robot was programmed using a sequence of structured commands. The children began by analyzing the tasks to determine whether loops could be applied to repetitive actions. Upon identifying a pattern suitable for looping, they used the P1 procedure within the PROC1 area to construct the loop. As shown in Figure 6, the P1 command was inserted both in the MAIN area and within the P1 block itself, allowing the program to iterate over the repeated tasks. This process helped children understand how to simplify programs through iteration and reuse.

3.2.5 Depuration

The children strengthened their programming skills by working with a virtual robot, following a four-step process: planning, programming, execution, and debugging. Particular emphasis was placed on debugging during exercises focused on decomposition, algorithmic design, and pattern recognition. After practicing with Lightbot, the children applied their skills to develop two video games using mBlock. This transition allowed them to deepen their understanding of programming blocks such as loops and conditionals. To further enhance their computational thinking, the children were tasked with programming two games using mBlock, employing block structures similar to those encountered in the Lightbot activities. One game—the “Scissors Dancer”—involved the use of sequential programming blocks, while the second—the “Ping-Pong” game—required the application of loops and conditional blocks. These student-created games are illustrated in Figure 7.



(a)

(b)

Figure 7. Debugging exercises with mBlock: a) Scissor dancer game, b) Pin pong game

4. RESULTS

To evaluate computational thinking, several analyses were conducted, including a quantitative comparison by gender, a statistical analysis of gender differences, and an evaluation based on the Peruvian educational system.

4.1 Quantitative comparison by gender

Table 4 presents the average scores by gender for each computational thinking skill.

These results support the idea that both girls and boys demonstrate similar levels of development in the evaluated skills. This is a positive indicator for promoting gender equity from an early age through the use of playful tools such as Lightbot and mBlock.

4.2 Statistical analysis of gender differences

Table 5 presents the results of the student's t-test comparing girls and boys for each computational thinking skill.

As shown, the p-values for all skills are greater than 0.05, indicating that there are no statistically significant differences between girls and boys in any of the evaluated skills. This reinforces the hypothesis of gender equity in the development of computational thinking skills through the use of these educational tools.

Table 4. Results of computational thinking skills

Gender	Algorithmic Design	Pattern Recognition	Decomposition	Loops	Debugging
Female	10.0	9.2	10.0	8.6	9.6
Male	11.8	9.0	12.8	9.8	9.3

Table 5. Student's t-test results for gender differences in computational thinking skills

Skill	t-Statistic	p-value
Algorithmic design	-1.026	0.3326
Pattern recognition	0.077	0.9406
Decomposition	-1.389	0.2005
Loop	-0.578	0.5786
Debugging	0.083	0.9353

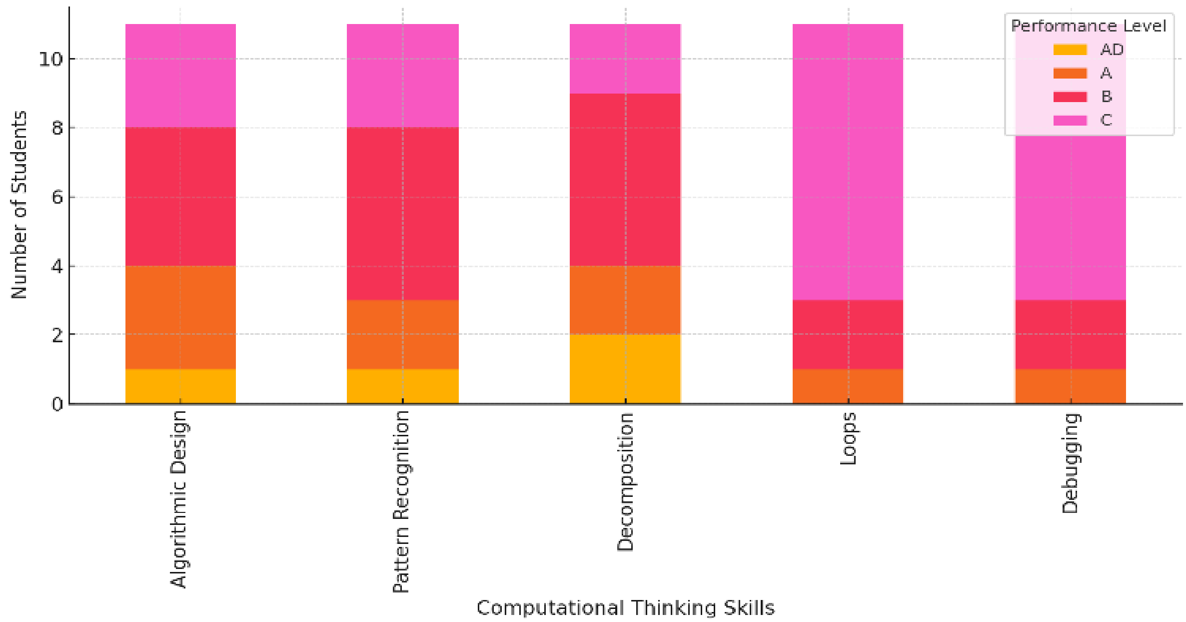


Figure 8. Score distribution by skill based on the Peruvian evaluation scale

4.3 Analysis based on the Peruvian educational evaluation system

According to the evaluation scale of the Peruvian educational system, computational thinking skills were categorized as follows: AD (Outstanding Achievement, 18–20), A (Expected Achievement, 14–17), B (In Progress, 11–13), and C (Beginning, 0–10). Figure 8 shows the distribution of student scores according to this scale for each computational thinking skill.

This visualization makes it possible to identify which skills require greater reinforcement. For instance, *Algorithmic Design* and *Decomposition* show a higher number of students in levels B and A, suggesting that these skills are in the process of consolidation. However, in skills such as *Debugging* and *Loops*, there is a greater concentration of students in level C, highlighting the need for additional educational interventions to strengthen these areas.

5. DISCUSSION

The results obtained show that both girls and boys developed computational thinking skills at similar levels, with no statistically significant differences. This supports the claims of Laura-Ochoa [38] and Urquizo [25], who argue that tools such as Lightbot and mBlock promote gender equity in educational contexts. This equity is especially relevant in rural areas such as Huancavelica, where access to technology is often limited [41].

Regarding pedagogical effectiveness, the results align with the research by Brennan and Resnick [22], who suggest that visual programming tools help children develop not only computational concepts (such as sequences, loops, and conditionals) but also practices (such as debugging and reuse) and perspectives (such as expression and connection). The activities implemented in this study demonstrated that participants were able to apply loops, sequences, and conditionals—core components of computational thinking [35].

Furthermore, the performance levels identified using the Peruvian evaluation scale are consistent with the findings of Sáez-López [33], who observed that visual programming tools like Scratch (and its derivative mBlock) enable progressive learning of complex skills. However, skills such as debugging and loops, which showed lower scores (Category C), require special attention in the design of future educational activities, as noted by Mukasherva and Omirzakova [32], who found that elementary students often struggle with these specific skills.

One notable finding of the study was the difficulty students experienced with debugging and loops, which had average scores in the lower range (Category C according to the Peruvian scale). This may be related to students limited prior exposure to meaningful technological experiences in rural settings. As noted by Paucar-Curasma [34], students with low exposure to computing environments tend to struggle more with skills that require structured logical thinking and error resolution, such as debugging. These skills are not typically emphasized in rural curricula, which may limit students' initial performance in tasks requiring analysis and correction [42].

Additionally, a comparison of the tools used revealed that Lightbot was particularly effective for introducing basic concepts such as sequences, loops, and pattern recognition through concrete visual challenges. In contrast, mBlock offered more opportunities for students to develop complex skills such as problem decomposition and algorithmic design. This differentiation is also supported by Paucar-Curasma [43], who stated that visual programming tools offer varying entry levels and depth, allowing for gradual progression in the development of computational thinking. In this study, Lightbot functioned as an introductory tool, while mBlock enabled more autonomous and creative application of learned concepts.

Finally, the virtual workshop strategy proved effective, supporting the observations of Simmonds [1], who noted that workshops—even in rural and vulnerable contexts—can be a powerful approach for introducing computational thinking, provided they are supported by trained teachers and contextualized resources [44].

6. CONCLUSIONS

The implementation of a computational thinking workshop using playful tools such as Lightbot and mBlock proved effective in developing key skills—algorithmic design, decomposition, pattern recognition, loops, and debugging—among children aged 8 to 11. No significant differences were found between boys and girls in any of the evaluated skills, suggesting that these tools can promote gender equity in learning computational thinking from an early age. The analysis using the Peruvian educational evaluation scale revealed that debugging and loops are areas requiring further reinforcement, highlighting the need for more targeted pedagogical strategies. The application of a quantitative quasi-experimental approach, combined with validated instruments, provided objective evidence of the intervention's impact, supporting the development of replicable proposals in similar contexts. This experience in rural areas of Peru demonstrates that it is possible to reduce the digital divide through the use of accessible and contextualized technologies, contributing to the development of 21st-century skills in students facing educational vulnerability. For future research, expanding the study to include more rural and urban schools will strengthen the external validity of the results and allow for exploration of contextual differences.

REFERENCES

[1] 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

[2] Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25: 127-147.https://doi.org/10.1007/s10956-015-9581-5

[3] Lee, I., Martin, F., Apone, K. (2014). Integrating computational thinking across the K-8 curriculum. ACM Inroads, 5(4): 64-71. https://doi.org/10.1145/2684721.2684736

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

[5] Kafai, Y.B., Burke, Q. (2014). Connected code: Why children need to learn programming. MacArthur Foundation Series on Digital Media and Learning. MIT Press. https://doi.org/10.7551/mitpress/9992.001.0001

[6] Román-González, M. (2015). Test de Pensamiento Computacional: Principios de diseño, validación de contenido y análisis de ítems Computational Thinking Test: Design guidelines, content validation and item analysis.

[7] MINEDU. (2024). Resolución Ministerial-MINEDU. Lima.

[8] Tedre, M., Denning, P.J. (2016). The long quest for computational thinking. In Proceedings of the 16th Koli Calling International Conference on Computing Education Research, Koli, Finland, pp. 120-129. https://doi.org/10.1145/2999541.2999542

[9] Forsythe, G.E. (1968). What to do till the computer scientist comes. The American Mathematical Monthly, 75(5): 454-462. https://doi.org/10.2307/2314698

[10] Papert, S. (1980). MINDSTORMS Children, Computers, and Powerful Ideas. Estados Unidos.

[11] Pólya, G. (1957). How to Solve, 2nd ed.

[12] Feurzeig, W., Papert, S., Bloom, M., Grant, R., Solomon, C. (1970). Programming-languages as a conceptual framework for teaching mathematics. ACM SIGCUE Outlook, 4(2): 13-17.

[13] Bento, W. (2017). O pensamento computacional no ensino profissional e tecnológico. Universidade Católica de Brasília.

[14] El pensamiento computacional 10 años después – Microsoft En Español, Microsoft, TechNet. hhttps://programamos.es/pensamiento-computacional-10-anos-despues/, accessed on Jun. 29, 2019.

[15] Wing, J.M. (2006). Computational thinking. Communications of the ACM, 49(3): 33-35.

https://doi.org/10.1145/1118178.1118215
[16] 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
[17] Wing, J.M. (2001). The LINK, Computational Thinking--What and Why? https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why.
[18] Voogt, J., Fisser, P., Good, J., Mishra, P., Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. Education and Information Technologies, 20: 715-728. https://doi.org/10.1007/s10639-015-9412-6
[19] Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., Punie, Y. (2016). Developing computational thinking: Approaches and orientations in K-12 education. In EdMedia+ Innovate Learning, pp. 13-18. Association for the Advancement of Computing in Education (AACE). https://doi.org/10.2791/792158
[20] Zhong, B., Wang, Q., Chen, J., Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. Journal of Educational Computing Research, 53(4): 562-590. https://doi.org/10.1177/0735633115608444
[21] Zhang, L., Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. Computers & Education, 141: 103607. https://doi.org/10.1016/j.compedu.2019.103607
[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, p. 25. https://doi.org/10.1.1.296.6602
[23] Kalelioglu, F., Gülbahar, Y., Kukul, V. (2016). A framework for computational thinking based on a systematic research review. Baltic Journal of Modern Computing, 4(3): 583-596.
[24] Lockwood, J., Mooney, A. (2018). Computational Thinking in Secondary Education: Where does it fit? A systematic literary review. International Journal of Computer Science Education in Schools, 2(1): 41-60. https://doi.org/10.21585/ijcses.v2i1.26
[25] 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.
[26] Peng, L.H., Bai, M.H., Siswanto, I. (2020). A study of learning motivation of senior high schools by applying unity and mBlock on programming languages courses. Journal of Physics: Conference Series, 1456(1): 012037. https://doi.org/10.1088/1742-6596/1456/1/012037
[27] 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
[28] Chang, C.K. (2014). Effects of using Alice and Scratch in an introductory programming course for corrective instruction. Journal of Educational Computing Research, 51(2): 185-204. https://doi.org/10.2190/EC.51.2
[29] Giordano, D., Maiorana, F. (2014). Use of cutting edge educational tools for an initial programming course. In 2014 IEEE global engineering education conference (EDUCON), Istanbul, Turkey, pp. 556-563. https://doi.org/10.1109/EDUCON.2014.6826147
[30] Gutierrez, F.J., Simmonds, J., Hitschfeld, N., Casanova, C., Sotomayor, C., Peña-Araya, V. (2018). Assessing software development skills among K-6 learners in a project-based workshop with scratch. In Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training, pp. 98-107. https://doi.org/10.1145/3183377.3183396
[31] Moreno, J., Robles, G. (2014). Automatic detection of bad programming habits in scratch: A preliminary study. In 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, Madrid, Spain, pp. 1-4. https://doi.org/10.1109/FIE.2014.7044055
[32] Mukasheva, M., Omirzakova, A. (2021). Computational thinking assessment at primary school in the context of learning programming. World Journal on Educational Technology: Current Issues, 13(3): 336-353. https://doi.org/10.18844/wjet.v13i3.5918
[33] 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
[34] Paucar-Curasma, R., Frango, I., Rondon, D., López, Z., Porras-Ccancce, L. (2022). Analysis of the teaching of programming and evaluation of computational thinking in recently admitted students at a public university in the Andean Region of Peru Ronald. Congreso Internacional de Tecnología e Innovación Educativa, Arequipa, p. 10.
[35] Puhlmann, C. (2017). Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica. Universidade Federal do Rio Grande do Sul.
[36] 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
[37] Aho, A.V. (2012). Computation and computational thinking. The Computer Journal, 55(7): 832-835. https://doi.org/10.1093/comjnl/bxs074
[38] Laura-Ochoa, L. (2018). Teaching the introduction to programming using playful tools [Enseñanza de la Introducción a la Programación utilizando Herramientas Lúdicas]. In Proceedings of the LACCEI International Multi-Conference for Engineering, Education and Technology, Lima, Peru, pp. 18-20.
[39] 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.
[40] Ministerio de Educación, Minedu establece disposiciones para evaluar a estudiantes de escuelas pedagógicas - Noticias - Ministerio de Educación - Plataforma del Estado Peruano, 2020. https://www.gob.pe/institucion/minedu/noticias/304837-minedu-establece-disposiciones-para-evaluar-a-estudiantes-de-escuelas-pedagogicas, accessed on Oct. 05, 2024.

[41] Guadalupe, C., León, J., Rodríguez, J.S., Vargas, S. (2017). ESTADO DE LA EDUCACIÓNEN EL PERÚ: Análisis y perspectivas dela educación básica. Biblioteca Nacional del Perú. Lima.

[42] 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

[43] Paucar-Curasma, R., Villalba-Condori, K.O., Florentino Unsihuaytovar, R., Arias-Chavez, D., Gonzales Agama, S.H., Ureta-Orihuela, F.L. (2024). Visual programming and problem-solving to foster a positive attitude towards formative research. Visual Review, 1(17): 111-126. https://doi.org/10.62161/revvisual.v17.5373

[44] Ibrahim, A., Al Sayed, I.A., Jabbar, M.S., Almutairi, H., Sekhar, R., Shah, P., Al_Barazanchi, I.I. (2024). Evaluating the impact of emotions and awareness on user experience in virtual learning environments for sustainable development education. Ingenierie des Systemes d'Information, 29(1): 65-73. https://doi.org/10.18280/isi.290108