

# Development of computational reasoning through block programming in primary school students at the Universidad Peruana Los Andes

*Desarrollo del razonamiento computacional mediante programación en bloques en estudiantes de Educación Primaria de la Universidad Peruana Los Andes*

**Received:** 05/08/2025 - **Accepted:** 03/11/2025

**Zenón Manuel López Robles**

<https://orcid.org/0000-0003-0744-6075>

[d.zlopez@ms.upla.edu.pe](mailto:d.zlopez@ms.upla.edu.pe)

Universidad Peruana Los Andes. Huancayo, Peru

**Dustin Douglas Ticse Alfaro**

<https://orcid.org/0000-0001-7780-817X>

[d.dticse@ms.upla.edu.pe](mailto:d.dticse@ms.upla.edu.pe)

Universidad Peruana Los Andes. Huancayo, Peru

**Franklin Moises Gilvonio Yaranga**

<https://orcid.org/0000-0003-1804-0169>

[d.fgilvonio@ms.upla.edu.pe](mailto:d.fgilvonio@ms.upla.edu.pe)

Universidad Peruana Los Andes. Huancayo, Peru

**Elsa Marleni Egoavil Victoria**

<https://orcid.org/0000-0002-3118-3773>

[d.eegoavil@ms.upla.edu.pe](mailto:d.eegoavil@ms.upla.edu.pe)

Universidad Peruana Los Andes. Huancayo, Peru

**Florinda Martha Castro Ingaroca**

<https://orcid.org/0009-0006-4408-0980>

[d.fcastroin@ms.upla.edu.pe](mailto:d.fcastroin@ms.upla.edu.pe)

Universidad Peruana Los Andes. Huancayo, Peru

## Abstract

The purpose of this research is to determine the impact of block programming on the development of computational reasoning in third-year students of the Bachelor's Degree in Primary Education at the Universidad Peruana Los Andes during the year 2024. The study adopts a quantitative approach and a quasi-experimental design with experimental and control groups, in which pre- and post-intervention measurements were applied to a sample of 60 students in the third cycle of Primary Education, selected by simple random sampling. Inclusion and exclusion criteria were also established to ensure adequate participation. Data collection was carried out through surveys, and the analysis was performed using descriptive statistics and the Wilcoxon test, with the aim of evaluating changes in computational reasoning before and after the intervention. The main results showed that block programming has a significant impact on the development of students' computational reasoning. Similarly, a positive effect was observed on problem-solving skills ( $Z = -5.000$ ,  $p = 0.000$ ), pattern recognition ( $Z = -5.385$ ,  $p = 0.000$ ), abstraction ability ( $Z = -5.568$ ,  $p = 0.000$ ), and algorithm design ( $Z = -4.243$ ,  $p = 0.000$ ). In conclusion, block programming strengthened students' fundamental computational thinking skills.

**Keywords:** computational reasoning, block programming, primary education.

## Resumen

La presente investigación tiene como propósito determinar el impacto de la programación en bloques en el desarrollo del razonamiento computacional en los estudiantes del III ciclo de la Licenciatura en Educación Primaria de la Universidad Peruana Los Andes, durante el año 2024. El estudio adopta un enfoque cuantitativo y un diseño

cuasiexperimental con grupos experimental y de control, en los cuales se aplicaron mediciones pre y posintervención a una muestra de 60 estudiantes del III ciclo de Educación Primaria, seleccionados mediante muestreo aleatorio simple. Asimismo, se establecieron criterios de inclusión y exclusión que garantizaron una participación adecuada. La recolección de datos se realizó mediante encuestas, y el análisis se llevó a cabo utilizando estadística descriptiva y la prueba de Wilcoxon, con el objetivo de evaluar los cambios en el razonamiento computacional antes y después de la intervención. Los principales resultados evidenciaron que la programación en bloques impacta significativamente en el desarrollo del razonamiento computacional de los estudiantes. De igual manera, se observó un efecto positivo en la habilidad para la resolución de problemas ( $Z = -5.000$ ,  $p = 0.000$ ), el reconocimiento de patrones ( $Z = -5.385$ ,  $p = 0.000$ ), la capacidad de abstracción ( $Z = -5.568$ ,  $p = 0.000$ ) y el diseño de algoritmos ( $Z = -4.243$ ,  $p = 0.000$ ). En conclusión, la programación en bloques fortaleció las habilidades fundamentales del pensamiento computacional en los estudiantes.

**Palabras clave:** razonamiento computacional, programación en bloques, educación primaria.

## Introduction

Computational reasoning is fundamental to contemporary education, as it fosters problem-solving skills, logical thinking, and analysis, competencies that are essential in the 21st century (Wing, 2017). Block-based programming serves as an educational technology tool that enables students to cultivate analytical thinking by abstracting information and manipulating it in a visual and active manner (Grover & Pea, 2013). Its implementation in primary education not only facilitates comprehension of programming logic but also promotes learner independence, analytical skills, and critical thinking. In this context, Tejera-Martínez et al. (2020) demonstrated that computational thinking and programming significantly contribute to the development of digital and cognitive skills from an early age, underscoring the importance of incorporating these tools into educational curricula.

Computational reasoning is defined as a set of cognitive skills and mental processes that allow individuals to formulate problems and design solutions that can be executed by computers. This definition has been the subject of extensive research due to its close relationship with logical thought, problem-solving, and technology-mediated learning (Wing, 2017). According to Roig-Vila and Moreno-Isac (2020), computational reasoning encompasses more than programming; it includes problem decomposition, pattern recognition, abstraction, and algorithm design. Block-based programming is approached as a teaching method that employs the visual organization of blocks to construct programs. Maloney et al. (2010) argue that this approach simplifies programming by minimizing syntactic difficulties, thus enabling a focus on algorithmic design.

Several studies, such as that of Quiroz Vallejo et al. (2021), highlight the establishment of computational thinking as a high-impact global research area, emphasizing the need to enhance its integration into basic education curricula. In the Peruvian context, the teaching of block-based programming remains limited, creating a gap in digital and computational skills compared to other countries (Ministerio de Educación del Perú [MINEDU], 2023). Previous research has shown that the use of block-based programming software, such as Scratch, improves logical-mathematical and cognitive skills among primary education students. Collado-Sánchez et al. (2021) found that its use increases logical-mathematical abilities by up to 65% and strengthens computational thinking. Similarly, Vargas Montenegro (2021) concluded that Scratch significantly contributes to the development of new cognitive abilities in primary school students, while Pérez López (2020) demonstrated that activities based on Scratch enhance algorithmic thinking and problem-solving abilities.

Reinhold et al. (2025) support the view that Scratch-based block programming, combined with real-world devices like robots, represents a promising strategy for fostering computational thinking in primary education. Additionally, Pérez-Marín et al. (2020) showed, in a quasi-experimental study with 132 students, that computational thinking can be improved through the use of metaphors and programming with Scratch. Likewise, Çakiroğlu and Özyilmaz Şahin (2026) observed that real-world problem contexts, along with Scratch's characteristics, encourage abstraction behaviors and the optimization of solutions. Durango-Warnes et al. (2020) also evidenced the advantages of Scratch for enhancing meaningful mathematical learning in third-grade students.

In the III Cycle of Primary Education at the Universidad Peruana Los Andes, students face a significant challenge: the lack of training in block-based programming. Although they have access to digital tools, the current curriculum does not provide structured instruction in computational thinking. This deficiency limits their development in the technological domain and leaves them inadequately prepared for academic and professional challenges in the future.

The issues identified in the Faculty of Education are related to the insufficient training of teachers in innovative methodologies for teaching computational reasoning through block programming, as well as the absence of this tool in the curricular framework (Román-González et al., 2017). Furthermore, some students in

the III Cycle mistakenly perceive programming as being solely relevant to engineering careers, which restricts their learning and engagement.

Failure to foster computational reasoning through block programming means that students do not enhance essential skills such as analytical, logical, and critical thinking, thereby creating disadvantages in the job market, where digital competencies are increasingly valued (Brennan & Resnick, 2012). A lack of a solid foundation in computational thinking exacerbates the digital divide and limits young people's opportunities in an increasingly digital society. Holstein and Cohen (2025) warn that for a successful adoption of constructivist pedagogies, it is crucial to understand the tensions teachers perceive in integrating computational thinking into curricula.

This research aims to implement computational reasoning through block programming as an effective teaching methodology for students in the III Cycle of Primary Education. According to Denning and Tedre (2022), computational reasoning consists of a set of skills based on computational models that facilitate problem-solving and idea abstraction. Its social and educational impact is significant, as it fosters the development of critical and competent citizens in a digital environment, capable of structuring their logical thinking and understanding the fundamental principles of computing.

Moreover, the implementation of block programming contributes to reducing the digital divide and promotes equality of opportunity from an early age, benefiting educators and families while integrating computational thinking into the holistic education of students. Theoretically, this study is grounded in constructivist learning theories and cognitive development, drawing on research by Wing (2017) and Grover and Pea (2013), who demonstrated that block programming effectively strengthens students' cognitive abilities, with results applicable to the Peruvian context.

Consequently, this research highlights the benefits that III Cycle Primary Education students experience through block programming, emphasizing its positive impact on the development of computational thinking.

### **General objective**

To determine the impact of block programming on the development of computational reasoning among students in the III Cycle of Primary Education at the Universidad Peruana Los Andes (2024).

### **General hypothesis**

Block programming significantly impacts the development of computational reasoning among students in the III Cycle of Primary Education at the Universidad Peruana Los Andes (2024).

### **Specific hypotheses**

- Block programming significantly impacts problem decomposition.
- Block programming significantly impacts pattern recognition.
- Block programming significantly impacts abstraction capacity.
- Block programming significantly impacts algorithm design.

### **Methodology**

The research was conducted using a quantitative approach and a hypothetical-deductive method, beginning with the observation of the problem to formulate and verify the hypotheses through deductive processes. The study type was applied, aimed at transforming knowledge through practical solutions with real impact. The level was explanatory, focusing on the cause-and-effect relationship between block programming and computational reasoning. The design was quasi-experimental, consisting of an experimental group and a control group, where pre- and post-intervention measurements of computational reasoning were taken.

The population comprised 60 students from the III Cycle of Primary Education at the Universidad Peruana Los Andes, selected through simple random sampling to ensure equal probability of selection. Inclusion criteria considered students enrolled and regularly attending during the second academic period of 2024, in addition to informed consent from parents or guardians. Students who did not attend the intervention sessions and those with conditions limiting their effective participation were excluded.

To collect data, a survey was administered to describe and explain the phenomena under study. The analysis incorporated descriptive statistics and the Wilcoxon test, aimed at evaluating changes in computational reasoning before and after the intervention with block programming, allowing for the assessment of the educational strategy's impact.

## Results and discussion

The following table presents the detailed program of the six learning sessions developed to strengthen computational thinking through block programming. Each session was designed with a clear purpose, specific competencies, and practical activities aimed at integrating theoretical knowledge and skills in an active pedagogical environment, utilizing the Scratch platform and its variants as facilitators of learning.

Learning Session	Subject	Section	Competency / Learning Outcome	Purpose	Knowledge	Activities	Time
<b>1. Introduction to computational thinking</b>	ICT in Education	B1	Recognizes the components of computational thinking and its application in early education.	Understand the concept and relevance of computational thinking.	Definition, components (decomposition, patterns, abstraction, algorithms).	Puzzle sequence dynamics, participatory presentation, introductory video, team activity, and group reflection.	90 minutes
<b>2. Getting to know scratch: our first block application</b>	ICT in Education	B1	Explores the block programming environment (Scratch) and its basic elements.	Familiarize with the Scratch interface and basic functions.	Interface, movement blocks, events.	Visual presentation, guided exploration, practical activity creating scenes, experience sharing, and project socialization.	90 minutes
<b>3. Storytelling with blocks</b>	ICT in Education	B1	Designs an interactive story using block programming.	Apply blocks to represent a narrative.	Narrative structure, control blocks, events, characters.	Short reading, story planning, implementation in ScratchJr, digital presentation, and self-assessment.	90 minutes
<b>4. Creating sequences to solve problems</b>	ICT in Education	B1	Develops simple algorithms to solve tasks or educational games.	Use logical sequences and educational conditionals.	Algorithms, sequences, conditionals (if-then).	"Follow the Instructions" game, hands-on programming, introduction to conditional blocks, feedback, and reflection.	90 minutes
<b>5. Educational games programmed by me</b>	ICT in Education	B1	Creates interactive resources based on block programming for early education.	Design a simple educational game.	Game design, interactivity, control and repetition blocks.	Review of examples, collaborative planning, coding in Scratch, peer testing, and feedback.	90 minutes
<b>6. Programming to learn: final presentation</b>	ICT in Education	B1	Presents a final project integrating communicative, logical, and creative skills.	Integrate learnings through an educational application in blocks.	Project evaluation, rubric, oral and written presentation.	Project preparation and presentation, practical demonstration, group evaluation, and final reflection.	90 minutes

**Table 1** displays the organization of each session, specifying the subject, section, competencies to be developed, conceptual contents, activities performed, and the duration of each session. This structure facilitated a systematic monitoring of the teaching-learning process, supporting the progressive acquisition of competencies and skills in programming and computational reasoning.

**Table 2**

*Distribution of performance levels in dimensions of computational thinking: Comparing Pre-Test and Post-Test*

Dimension	Level	Pre-test (n=60)	% Pre-test	Post-test (n=60)	% Post-Test
Computational reasoning	Low	27	45,0%	15	25,0%
	Medium	17	28,3%	18	30,0%
	High	16	26,7%	27	45,0%
Decomposition	Low	28	46,7%	14	23,3%
	Medium	18	30,0%	21	35,0%
	High	14	23,3%	25	41,7%
Pattern recognition	Low	30	50,0%	14	23,3%
	Medium	15	25,0%	18	30,0%
	High	15	25,0%	28	46,7%
Abstraction	Low	29	48,3%	13	21,7%
	Medium	19	31,7%	20	33,3%
	High	12	20,0%	27	45,0%
Algorithms	Low	22	36,7%	15	25,0%
	Medium	22	36,7%	18	30,0%
	High	16	26,7%	27	45,0%

Source: Authors' own elaboration

**Table 2** highlights a significant improvement across all evaluated dimensions following the block programming intervention. There is a notable reduction in the percentage of students at lower performance levels and an increase in those at higher levels of computational reasoning, problem decomposition, pattern recognition, abstraction, and algorithm design. These results demonstrate the positive impact of block programming on the development of computational skills among III Cycle primary education students.

**Table 3**

*Hypothesis testing*

Hypothesis	N Ranks+	Average Rank	Sum of Ranks	Z (Wilcoxon Test)	Asymptotic Sig. (2-tailed)	Conclusion
General hypothesis	23	12.00	276.00	-4,796	0.000	Reject Ho; block programming

						significantly impacts computational reasoning
<b>Specific hypothesis 1: problem decomposition</b>	25	13.00	325.00	-5.000	0.000	Reject Ho; block programming significantly impacts problem decomposition
<b>Specific hypothesis 2: pattern recognition</b>	29	15.00	435.00	-5.385	0.000	Reject Ho; block programming significantly impacts pattern recognition
<b>Specific hypothesis 3: abstraction</b>	31	16.00	496.00	-5.568	0.000	Reject Ho; block programming significantly impacts abstraction capacity
<b>Specific hypothesis 4: algorithm design</b>	18	9.50	171.00	-4.243	0.000	Reject Ho; block programming significantly impacts algorithm design

**Source:** Authors' own elaboration

The results of the hypothesis testing indicate that, both for the general hypothesis and the specific hypotheses related to problem analysis, pattern recognition, abstraction, and algorithm design, positive ranks predominated, accompanied by negative Z values and p-values below 0.05. This consistently allowed for the rejection of the null hypotheses, concluding that block programming has a statistically significant effect on the development of the evaluated competencies among students.

In relation to the general objective— “To determine the impact of block programming on the development of computational reasoning among students in the III Cycle of Primary Education at the Universidad Peruana Los Andes, 2024”—the statistical analyses revealed a significant improvement ( $\alpha = 0.000$ ), confirming that block programming substantially influenced the advancement of computational reasoning. These findings align with those reported by Huaripata Huaripata (2023), who noted significant improvements in reading comprehension among secondary students through the use of Scratch, with a 45% increase between initial and final evaluations.

Although the topic under study differs, both works emphasize the educational value of Scratch for strengthening complex cognitive skills. Additionally, research by Vargas Montenegro (2021) corroborates these results by demonstrating that the use of Scratch contributed 71% to the development of logical thinking skills and new cognitive capacities in primary students, further reinforcing the positive impact of block programming on computational reasoning.

Moreover, the study by Durango-Warnes et al. (2020) showed that block programming stimulates logical-mathematical thinking, promoting meaningful learning and the development of key competencies such as creative thinking, collaborative work, and continuous learning. Conversely, Pérez López (2020) highlighted that activities with Scratch aimed at algorithmic thinking in secondary students yielded varied results: while many succeeded in passing, specific difficulties in problem-solving persisted, indicating the need for deeper pedagogical intervention in certain cases.

## Conclusions

The study concludes that block programming significantly impacts the development of computational reasoning among students in the III Cycle of Primary Education at the Universidad Peruana Los Andes, 2024. This finding was supported by the Wilcoxon signed-rank test, which yielded a negative rank of -4.796 and a significance

value of  $p = 0.000$  ( $< 0.05$ ), indicating a statistically significant improvement in computational reasoning skills following the intervention.

Additionally, it was determined that block programming plays a substantial role in enhancing students' problem decomposition capabilities. The Wilcoxon test revealed negative ranks of  $-5.000$  and a  $p$ -value of  $0.000$  ( $< 0.05$ ), suggesting a significant advancement in the ability to break down complex problems into simpler components, a fundamental aspect of computational thinking.

Furthermore, block programming demonstrated a relevant impact on pattern recognition. The results indicated negative ranks of  $-5.385$  and a  $p$ -value of  $0.000$  ( $< 0.05$ ), confirming a significant difference post-intervention. This suggests that students developed a greater capacity to identify sequences, regularities, and structures, which are essential for creating algorithms and understanding computational processes.

Moreover, it was determined that block programming has a significant effect on abstraction skills. The Wilcoxon test reported negative ranks of  $-5.568$  and a  $p$ -value of  $0.000$  ( $< 0.05$ ), demonstrating considerable improvement in students' ability to represent problems using conceptual and symbolic models.

Finally, the study concludes that block programming significantly influences algorithm design. The Wilcoxon test results—negative ranks of  $-4.243$  and a  $p$ -value of  $0.000$  ( $< 0.05$ )—indicate substantial enhancement in students' ability to organize logical sequences of instructions, develop ordered procedures, and solve problems through structured steps, which are essential competencies for algorithmic thinking.

## References

- Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking* (p. 25). <https://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>
- Çakiroğlu, Ü., & Özyılmaz Şahin, İ. (2025). Building blocks of abstraction: Exploring children's real-life problem solving in block-based programming. *Thinking Skills and Creativity*, 59, 101995. <https://doi.org/10.1016/j.tsc.2025.101995>
- Collado-Sánchez, M., García-Peñalvo, F. J., & Pinto-Llorente, A. M. (2021). Computational thinking competences training for primary education teachers. *Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21)*, 758–762. <https://doi.org/10.1145/3486011.3486544>
- Denning, P. J., & Tedre, M. (2022). Computational thinking: A disciplinary perspective. *Informatics in Education*, 20(3), 361–390. <https://doi.org/10.15388/infedu.2021.21>
- Durango-Warnes, C., & Ravelo-Méndez, R. E. (2020). Beneficios del programa Scratch para potenciar el aprendizaje significativo de las Matemáticas en tercero de primaria. *Trilogía Ciencia Tecnología Sociedad*, 12(23), 161–184. <https://doi.org/10.22430/21457778.1524>
- 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>
- Holstein, S., & Cohen, A. (2025). Scratch teachers' perceptions of teaching computational thinking with school subjects in a constructionist approach. *Thinking Skills and Creativity*, 56, 101772. <https://doi.org/10.1016/j.tsc.2025.101772>
- Huaripata Huaripata, A. (2023). *Aplicación del Programa Scratch para la comprensión lectora en estudiantes de secundaria de una institución educativa, Tongod—2022* [Tesis de licenciatura, Universidad César Vallejo]. <https://dialnet.unirioja.es/servlet/tesis?codigo=339284>
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4), 16:1–16:15. <https://doi.org/10.1145/1868358.1868363>
- Ministerio de Educación del Perú (MINEDU). (2023). *Reporte de monitoreo de tecnologías educativas en instituciones públicas* (Dirección de Innovación Tecnológica Educativa). MINEDU. <http://www.minedu.gob.pe/normatividad/pesem/informe-evaluacion-2023-pesem.pdf>
- Pérez López, A. F. (2020). *Actividades con Scratch para desarrollar el pensamiento algorítmico en estudiantes en una IEP de Chiclayo* [Tesis de Maestría, Universidad Católica Santo Toribio de Mogrovejo]. <http://tesis.usat.edu.pe/handle/20.500.12423/2736>
- Pérez-Marín, D., Hijón-Neira, R., Babelo, A., & Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and Scratch to teach computer programming to children? *Computers in Human Behavior*, 105, 105849. <https://doi.org/10.1016/j.chb.2018.12.027>
- Quiroz Vallejo, D. A., Carmona-Mesa, J. A., Castrillón Yepes, A., & Villa-Ochoa, J. A. (2021). Integración del pensamiento computacional en la educación primaria y secundaria en Latinoamérica: Una revisión sistemática de literatura. *RED: Revista de Educación a Distancia*, 21(68), 1–28. <https://dialnet.unirioja.es/servlet/articulo?codigo=8205366>

- Reinhold, F., Sprenger, P., & Staniczek, G. (2025). Introducing computational thinking to second and third graders: Programming whole paths outperforms step-by-step navigation in maze tasks. *Computers and Education Open*, 8, 100260. <https://doi.org/10.1016/j.caeo.2025.100260>
- Roig-Vila, R., & Moreno-Isac, V. (2020). El pensamiento computacional en educación: Análisis bibliométrico y temático. *Revista de Educación a Distancia (RED)*, 20(63). <https://doi.org/10.6018/red.402621>
- Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678–691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Tejera-Martínez, F., Aguilera, D., & Vílchez-González, J. M. (2020). Lenguajes de programación y desarrollo de competencias clave: Revisión sistemática. *Revista Electrónica de Investigación Educativa*, 22, 1–16. <https://doi.org/10.24320/redie.2020.22.e27.2869>
- Vargas Montenegro, A. K. (2021). *Uso de Scratch para el aprendizaje lógico de programación en estudiantes de nivel primaria, Huaraz-2019* [Tesis de pregrado, Universidad Católica Los Ángeles de Chimbote]. [https://repositorio.uladech.edu.pe/bitstream/handle/20.500.13032/25489/APRENDIZAJE\\_DESARROLLO\\_DE\\_CAPACIDADES\\_VARGAS\\_MONTENEGRO\\_AZUCENA\\_KATHERIN.pdf?sequence=1&isAllowed=y](https://repositorio.uladech.edu.pe/bitstream/handle/20.500.13032/25489/APRENDIZAJE_DESARROLLO_DE_CAPACIDADES_VARGAS_MONTENEGRO_AZUCENA_KATHERIN.pdf?sequence=1&isAllowed=y)
- Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7–14. <https://doi.org/10.17471/2499-4324/922>

## Authorship contribution

1. **Conceptualization:** López Robles, Zenón Manuel
2. **Data curation:** Gilvonio Yaranga, Franklin Moisés
3. **Formal analysis:** Castro Ingaroca, Florinda Martha
4. **Funding acquisition:** López Robles, Zenón Manuel
5. **Research:** Gilvonio Yaranga, Franklin Moisés
6. **Methodology:** Gilvonio Yaranga, Franklin Moisés
7. **Project direction:**
8. **Resources:** López Robles, Zenón Manuel
9. **Software:** Ticse Alfaro, Dustin Douglas
10. **Supervision:** Castro Ingaroca, Florinda Martha
11. **Validation:** Ticse Alfaro, Dustin Douglas
12. **Visualization:** Egoavil Victoria, Elsa Marleni
13. **Writing - original draft:** Egoavil Victoria, Elsa Marleni
14. **Writing - proofreading and editing:** Castro Ingaroca, Florinda Martha