A Scratch Programming Workshop for High School Girls: Engagement, Learning, and Inclusion in Computing

Fernanda Coutinho[1], Isadora de Souza[1], Lara Dias[1], Juliana Lindebeck[1], Pedro Henrique Dias Valle[2], Alessandreia Marta de Oliveira[1]

[1] Universidade Federal de Juiz de Fora – DCC/UFJF – Juiz de Fora, MG, Brasil
[2]Universidade de São Paulo – IME/USP – São Paulo, SP, Brasil

{fernanda.coutinho, isadora.desouza, lara.dias}@estudante.ufjf.br
juliana.lindebeck@estudante.ufjf.br
pedrohenriquevalle@usp.br, alessandreia.oliveira@ufjf.br

Abstract. *Computational Thinking (CT) has been recognized as a key component in students' development, strengthening abilities such as creativity, autonomy, and problem-solving. This work presents a workshop for high school girls from a public school using Scratch to introduce programming concepts and encourage girls' participation in Computing. The initiative provides an accessible and interactive learning experience that supports digital skills and promotes a more inclusive approach to programming education. Results show that most students achieved "Adequate" and "Excellent" levels in skills involving abstraction, algorithms, and data analysis, suggesting assimilation of CT concepts and the potential of the workshop to foster girls' engagement in Computing.*

1. Introduction

Computing in Brazil still presents a marked gender imbalance: fewer than 20% of entrants and fewer than 16% of graduates in Information Technology (IT) higher education programs are women (2023) . This underrepresentation is related to structural and historical factors, such as gender stereotypes, low visibility of female role models, and the absence of efforts to encourage girls' interest in Computing in Basic Education [Deus et al. 2020, Souza et al. 2025]. Thus, the inequality is not primarily the result of individual lack of interest, but rather of limited educational opportunities and a socially constructed sense of non-belonging [Santos and Marczak 2023].

In this context, adopting educational strategies to increase women's participation in Computing is an effective way to reduce disparities. Among these strategies, Computational Thinking (CT) stands out as a set of problem-solving skills involving decomposition, abstraction, pattern recognition, and algorithm development [Wing 2021]. When embedded in inclusive learning environments, mediated by women and centered on student protagonism, CT-oriented practices help reshape perceptions of the computing field, reduce gender stereotypes, and strengthen self-confidence. Thus, integrating CT into accessible activities can increase girls' engagement and broaden views on tech careers.

Teaching CT in Basic Education requires selecting appropriate tools. In this context, Scratch[1] stands out as one of the most widely used platforms to support practices aimed at developing CT in educational initiatives and academic research

[1]https://scratch.mit.edu/

[Resnick et al. 2009]. A study by Garcia and Bittencourt (2023) indicates that Scratch was employed in 56.9% of the analyzed works on CT development, surpassing approaches such as Code.org[2], App Inventor[3], Kodu[4], and Blockly[5]. This predominance is related to its block-based visual interface, which reduces the cognitive load associated with the syntax of textual languages and enables novice students to focus on understanding algorithmic reasoning and structuring solutions. Moreover, the environment encourages an exploratory approach, where error is seen as part of the learning process, contributing to the gradual development of students' confidence.

In addition to addressing educational demands, this initiative aligns with the United Nations Sustainable Development Goals (SDGs), particularly SDG 4 (Quality Education) and SDG 5 (Gender Equality) [United Nations 2015]. By promoting learning experiences in Computing for girls in public high schools through an accessible, playful, and authorship-centered approach, the workshop helps expand opportunities and reduce gender barriers in technology. In this sense, the proposal aligns with international efforts to foster inclusion, diversity, and women's participation in STEM, while remaining aligned with national guidelines and the public school context.

Despite these advances, a significant gap remains in accessible school-based interventions that can effectively engage high school girls in introductory programming while fostering CT in a meaningful and inclusive manner. In this work, we present an experience report on implementing an introductory Scratch programming workshop for girls in public schools. The workshop was developed within the scope of the outreach Meninas Programadoras JF [Coutinho et al. 2025] , in partnership with Escola de Games UFJF [Genesio et al. 2023] , to encourage women in Computing and promote the development of CT and programming skills. The remainder of this paper is organized into four sections: related work, materials and methods, results and discussion, and final considerations.

2. Related Work

Recent literature reports initiatives for teaching Computing in Basic Education through workshops, active methodologies, and introductory programming, including efforts to promote girls' participation. In this context, Pereira et al. (2025) adopted Problem-Based Learning (PBL) in high school, integrating Computing with Digital Culture in a 16-hour course, using quizzes and Dr. Scratch for evaluation. Results showed good learning outcomes, although abstraction remained limited. Pinto et al. (2020) proposed activities for 37 girls from public schools that combined texts, comics, unplugged tasks, and Scratch, which proved effective despite differences in prior knowledge.

Calderon et al. (2025) developed a progressive learning pathway with awareness activities, Scratch, and Robotics (LEGO EV3), highlighting student agency, especially in mentoring roles. Similarly, Santos et al. (2024) explored storytelling in workshops, increasing engagement and strengthening university–school collaboration. In the same direction, Gomes et al. (2014) also used Scratch to foster STEM learning, with students applying programming concepts despite initial difficulties.

[2]https://code.org/pt-BR
[3]https://appinventor.mit.edu/
[4]https://www.kodugamelab.com/
[5]https://developers.google.com/blockly?hl=pt-br

Finally, Ferreira et al. (2025) addressed the transition from visual to textual languages. In a 20-hour course grounded in PBL and gamification, the authors found that methodological adaptation and inclusive environments help mitigate performance drops when moving from Scratch to Python, reinforcing the role of visual tools as an entry point.

Figure 1 synthesizes the assessment instruments and results, providing a comparative overview. Our study combines Scratch, girls' agency, women-led workshops, and activities that foster authorship and autonomy.

Reference	Assessment Instruments	Main Results	Relation to This Study
Pereira et al. (2025)	Automatic analysis (Dr. Scratch) and gamified quizzes	Projects classified as "In Development": good logic but low abstraction	Presents strategies for contextualizing programming learning in educational settings
Pinto et al. (2025)	Activity observation and project analysis	Initial programming proficiency and algorithmic understanding	Shows early experiences aimed at increasing interest in programming
Calderon et al. (2025)	Motivation and performance in questionnaires and competitions	The students' protagonism consolidated technical self-confidence	Highlights actions designed to encourage girls' participation in Computing
Santos et al. (2024)	Qualitative discussions and personal narratives (Storytelling)	The subjective connection generated immediate engagement and strengthened support	Emphasizes inclusive and engaging approaches for teaching programming
Gomes et al. (2025)	Final project analysis and activity observation	Mastery of control structures, interative project development and collaborative skills	Reinforces the potential of creative programming activities in education
Ferreira et al. (2025)	Performance rankings and learning curve analysis	Identifying of performance degradation during the transition to text-based programming (Python)	Highlights Scratch as an initial step in learning programming concepts

Figure 1. Overview of related works: assessment instruments and main results.

3. Material and Methods

The methodology was based on active learning and project-based learning [Bender 2012, Prince and Felder 2006, Santiago et al. 2023], which place students at the center of the learning process, fostering autonomous and reflective learning through practice and experimentation. The **Workshop Planning** aimed to develop CT and programming skills, based on the BNCC Computing competencies [Brasil 2022] (Table 1). Strategies focused on discovery and problem-solving. Activities were designed to promote engagement and original solutions, encouraging programming practice in a playful, accessible way while considering students' prior knowledge. As part of this planning, we conducted an **Infrastructure Mapping** to identify available spots with functional computers with Internet access, ensuring equitable participation and adequate resources. The **Definition of Resources** included equipment and materials for instruction and practice. Computers with Internet access and a projector were used to present Scratch and instructional materials. These slides[6] , developed with Meninas Programadoras JF and Escola de Games UFJF , supported the explanation of concepts and guided the activities.

3.1. Workshop Structure and Implementation

The **Workshop Implementation** followed a structure of progressive meetings that combined expository moments, guided exploration, and practical tasks. The main resource produced was the instructional material conceived as a mediation script that alternates concise explanations, interactive examples, and proposals for experimentation. This organization contributed to content comprehension and continuous participant engagement

[6] https://www.canva.com/design/DAG0gOSkJ90/IgHYJg_sttpHYs-s8OKqzA/view

Table 1. BNCC for Computing skills developed

Skill	Pedagogical relationship with Scratch activities
EM13CO01	Related to the identification and formulation of problems addressed through computational solutions. Developed in the workshop based on the definition of objectives and the expected behavior of the projects created in Scratch.
EM13CO02	Associated with abstraction and problem decomposition processes, explored through the organization of ideas into actors, events, and distinct scripts within the Scratch environment.
EM13CO05	Linked to the analysis and comparison of different computational solutions for the same problem, encouraged through collective discussion of projects and reflection on alternative implementation strategies.
EM13CO06	Related to software evaluation, considering characteristics such as functionality, clarity, usability, and adequacy to the proposed objectives, based on the critical analysis of the programs developed in Scratch.
EM13CO15	Explored transversally through the analysis of the interaction between students and the artifacts created in Scratch, considering user experience, interface clarity, and the suitability of the projects to educational and playful contexts.

throughout the activities. We organized the workshop into four practical classes and one final meeting, each lasting one hour, totaling five hours of instruction, with a logical progression of content, as shown in Figure 2.

Figure 2. Visual representation of the workshop structure.

The first session began with interaction among participants and instructors to establish a welcoming, safe environment appropriate to students' levels of technological familiarity. Next, an introduction to the basic functioning of computers addressed concepts related to input, output, and data-processing devices. This stage enabled students to understand how computers receive, process, and return commands and information, providing a conceptual foundation for the logical principles explored in later sessions.

Next, we presented the concept of a programming language, then introduced Scratch. The interface organization was explored in detail, highlighting essential functionalities for project development and reducing difficulties during practical activities. We also guided students in creating platform accounts, which enabled them to save and resume their work later. Additionally, examples of games produced by the instructor and other creators were presented, illustrating the platform's variety of development possibilities, particularly in themes close to participants' daily lives.

In the subsequent sessions, the activities focused on exploring the Scratch block categories, starting with fundamental elements and progressing to more complex resources. In practice, the proposed exercises directly revisited the concepts presented in the expository moments, ensuring coherence between theory and experimentation. Students developed the activities mostly individually, with instructor mediation provided on demand, a strategy that fostered student autonomy, protagonism in the learning process, and independent problem-solving.

We conducted the **Final Project Application**, consisting of creating an original project in Scratch. To encourage creative expression, no specific theme was imposed, nor was a minimum technical difficulty set. In this activity, each participant applied the knowledge acquired throughout the workshop. This stage served as a moment of synthesis and reflection, allowing students to share their work, exchange experiences, and collaboratively analyze their learning processes. During development, we observed student autonomy in problem-solving and in combining blocks, events, and control structures. The most recurrent difficulties concerned the appropriate use of conditionals and coordination across different parts of the code, requiring targeted instructor interventions to advance the participants' conceptual understanding.

3.2. Participants and Ethical Considerations

The characterization of participants' profiles was conducted based on the analysis of a questionnaire administered before the workshop (Figure 3). To ensure transparency and reproducibility, the original data collection instruments (questionnaires and blank consent forms) and the anonymized raw dataset utilized in this study are publicly available at an online repository[7]. The collected data indicate that the group was composed of a total of eight participants. The majority consisted of six high school students aged 16-19 years, enrolled in public schools in the region. Additionally, two undergraduate students from the initial semesters of the Exact Sciences and Information Systems programs participated, resulting in a heterogeneous group in terms of educational background. Although the workshop focused on Computing, responses revealed diverse professional interests across the life sciences, humanities, and exact sciences, and many students remained undecided about their career choice.

Regarding prior programming knowledge, students reported basic logic skills and prior exposure to textual languages. This initial background facilitated exploratory engagement with Scratch, enabling participants to broaden their understanding of representation forms and algorithmic construction. Concerning motivations for participating in the workshop, the analysis of responses about factors that sparked students' interest revealed active engagement beyond initial curiosity. An explicit pursuit of technical skills, especially related to "programming" and "Computing". Moreover, interest in "creating a game" and "learning tools" indicates that participants envisioned a practical application of knowledge, demonstrating a desire to apply learning through original projects.

The workshop was offered as an extracurricular activity open to any girl interested in participating, without a formal selection process based on prior knowledge of Computing. The initiative was announced in classrooms at the partner school, and participation was voluntary, with registration via an online form. The number of participants can be partially explained by contextual factors, particularly the workshop schedule, which took place at the end of regular classes, during lunchtime. This timing may have limited participation due to the students' daily routines. On the other hand, small group enabled closer mediation, individualized support, and more detailed observation of learning processes.

[7]`https://drive.google.com/drive/folders/15E892qt54VVNat_wmItn4l76GRTTyzz-`

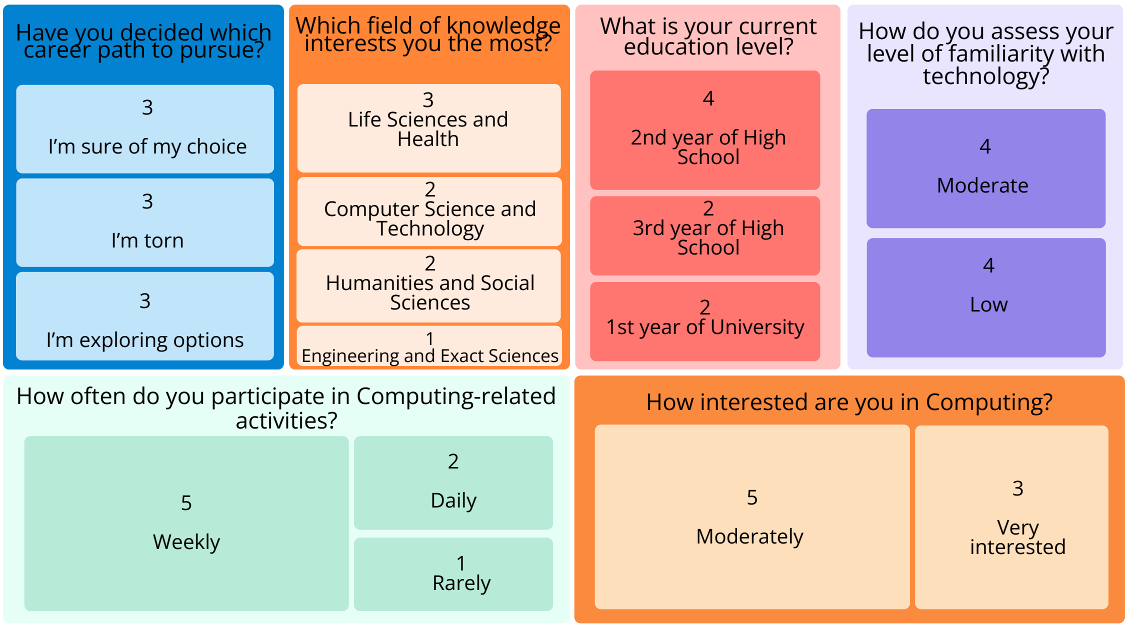

Figure 3. Demographic profile and characteristics of the participating students.

This work was developed within the scope of a university outreach activity, with educational and formative purposes, aimed at the practical application of knowledge and the dissemination of technological expertise. The workshop implementation was preceded by formal authorization from the partner educational institution, which reviewed and approved the proposal and validated the adequacy of the physical and computational infrastructure needed to conduct the activities.

Regarding ethical aspects of student participation, all procedures adopted followed ethical principles governing research and educational activities involving human subjects. For underage students, a Free and Informed Consent Form (ICF) signed by parents or legal guardians authorizes participation in the activities and the use of data, visual records, and academic productions for scientific and dissemination purposes, ensuring anonymity and confidentiality. Additionally, to respect the participants' autonomy and decision-making rights, a Free and Informed Assent Form (IAF) was used, through which the students voluntarily and knowingly agreed to participate in the workshop. The participants were duly informed about the objectives of the activity, the procedures involved, and the possibility of withdrawing at any time, without any academic or institutional prejudice. The information collected during the workshop, including questionnaire responses, photographic records, and Scratch projects, was treated as strictly confidential. For purposes of analysis and dissemination of results, the participants' names were suppressed or replaced with alphanumeric codes, ensuring that no student could be identified directly or indirectly. Thus, the work preserves the privacy, dignity, and rights of the participants and follows ethical principles for studies and experience reports in education.

3.3. Project Evaluation Criteria and Data Analysis

Finally, the **Results Analysis** was based on a formative, continuous evaluation, prioritizing observations of interactions, participation, and students' progression throughout the meetings. A predominantly qualitative approach was adopted, focused not only on final products but also on the knowledge construction process, considering the understanding of the concepts addressed and engagement in the proposed activities.

Based on the projects developed at the end of the workshop, an integrated qualitative and quantitative analysis of participants' learning was conducted. In this analysis, both the use of the Scratch tool's features and the logical construction of the implemented solutions were considered. The evaluation followed the criteria established in the activity description to assess the complexity, the diversity of the elements used, and student autonomy during solution development. To align the evaluation with the pillars of CT, the criteria are presented below with their respective CT mappings:

1. **Abstraction:** The development or adaptation of original scenarios incorporating multiple entities, such as characters or objects.
2. **Algorithmic Thinking:** Use of at least three distinct code blocks, not necessarily associated with the same character;
3. **Pattern Recognition and Logical Reasoning:** Use of at least one control structure block (such as repetition or condition);
4. **Data Collection and Analysis:** Inclusion of at least one sensor block;
5. **Mathematical Reasoning:** Use of at least one operator;
6. **Decomposition:** Creation and manipulation of a variable;
7. **Algorithms:** Implementation of some form of interaction with the player, either through inputs, character movement, or object collection.

The degree of mastery of each CT-related skill was identified using criteria to minimize subjectivity. As shown in Table 2, the 'Insufficient' (0) and 'Not evidenced' categories indicate conceptual gaps or absences, whereas the 'Adequate' (1) and 'Excellent' (2) levels distinguish between basic execution and advanced logical reasoning.

Table 2. Assessment Rubric for CT Skills based on Proficiency Levels.

Skill	Insufficient (0)	Adequate (1)	Excellent (2)	Not evidenced
Abstraction	Only one character is present, or the default scenario is used without modification.	Includes two or more characters OR an original/adapted scenario.	Includes two or more characters AND a purposeful original/adapted scenario.	The project is empty or lacks visual elements.
Algorithmic Thinking	Use of only 1 or 2 isolated code blocks with no clear execution logic.	Use of at least 3 distinct code blocks, even if across different characters.	Use of multiple (3+) code blocks organized in logical, functional sequences.	No code blocks are present.
Pattern Recognition and Logical Reasoning	Attempted use of control blocks (loops/conditionals) that are incorrect or non-functional.	Use of at least one functional control structure (loop or conditional).	Strategic use of control structures to optimize code or create complex events.	No control structures are used.
Data Collection and Analysis	Sensor blocks are present but not connected to any meaningful action or event.	Includes at least one sensor block that triggers a basic functional reaction.	Uses sensors to create dynamic interactivity or complex decision-making.	No sensor blocks are used.
Mathematical Reasoning	Use of operators in a way that does not alter the program's behavior.	Functional use of at least one operator (arithmetic, relational, or logical).	Integrates operators into conditionals or variables for precise calculations.	No operators are used.
Decomposition	A variable is created but never manipulated (value never changes or is never read).	Successfully creates and manipulates (changes the value of) at least one variable.	Uses variables to manage complex states (e.g., scoring, health, or timing).	No variables are used.
Algorithms	Attempted interaction, but inputs/commands do not respond or are broken.	Implementation of one functional form of interaction (input, movement, or collection).	Implementation of multiple fluid interactions that engage the user/player.	The project is a static, passive animation.

To provide a visual synthesis of the methodological path adopted, Figure 4 presents the flow of the six main stages described in this section, serving as a summarized guide to the methodological proposal. The structure, materials, and evaluation instruments described in this study were designed to support replication in similar contexts.

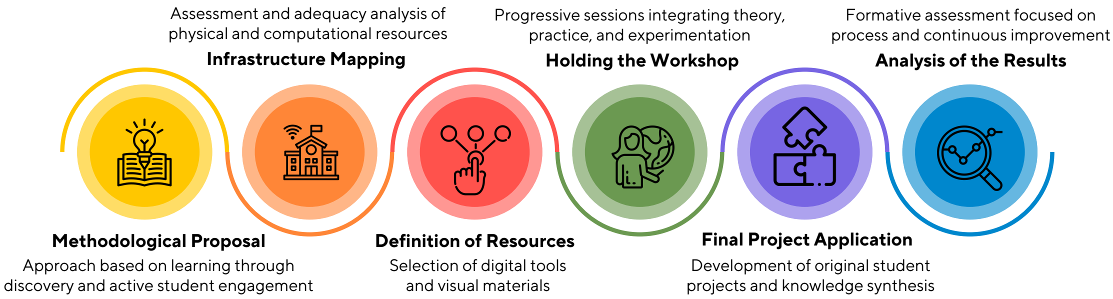

Figure 4. Workshop implementation overview: methodological flow.

4. Results and Discussion

To prioritize direct evaluation over student perceptions, the results analysis focused on the objective assessment of the source code produced during the workshop. A relevant aspect was the authorship of the final Scratch projects. All participants completed their projects, partially or fully meeting the defined criteria. These criteria aimed not only to assess the technical execution of the games but also to examine cognitive skills related to CT pillars.

As presented in Table 3, the criteria for evaluating the final projects were established to operationalize, within the context of the workshop, the CT skills described in the BNCC for Computing. The inclusion of components such as actors, variables, sensors (for detecting actions or conditions), operators, conditional structures, repetition loops, and user interactions was intended to demonstrate competencies in decomposition, abstraction, pattern recognition, and algorithmic thinking.

Table 3. Relationship between final project criteria and developed skills

Criterion	CT Concept	Developed Skill
Actors and Scenarios	**Decomposition:** breaking a complex problem into smaller and manageable parts.	Division of the project into characters, scenes, and responsibilities.
Sensors	**Algorithmic Thinking:** organizing instructions that allow the program to respond to events and inputs.	Programs react to user input through event-driven logic.
Conditional Structures	**Algorithmic Thinking:** defining decision rules that control the program's behavior.	Using conditions to determine different actions during program execution.
Variables	**Abstraction:** selecting and representing the most relevant elements of the problem while ignoring unnecessary details.	Representing relevant information such as score or time.
Operators	**Abstraction:** representing and manipulating data through expressions and comparisons.	Composing expressions that control rules, comparisons, and calculations within the project.
User Interaction	**Algorithmic Thinking:** organizing sequences of instructions that respond to user actions.	Creating control mechanisms that make the project responsive and coherent.
Repetition Structures	**Pattern Recognition:** identifying repeated actions and generalizing them.	Optimizing code by eliminating unnecessary repetitions through behavior automation.

The distribution of these objective evaluation results is shown in Figure 5, which supported the technical analysis. In the graph, the vertical axis shows the number of projects, and the lines indicate the frequency of performance levels across the eight criteria. Two distinct scenarios can be observed: in most criteria, especially Characters, Interactions, Number of Blocks, and Scenarios, "Excellent" performance predominated, indicating mastery of Scratch's basic features and the development of CT practices, particularly decomposition and algorithmic thinking, supported by guided practice. Conversely, in criteria involving greater logical complexity, such as Sensors and Operators, there was a reduction in "Excellent" evaluations and an increase in "Insufficient" classifications. This indicates that features requiring higher levels of abstraction and algorithmic reasoning remain challenging, highlighting the need for more specific activities in future workshops. The "Not evidenced" evaluations refer to projects in which certain criteria could not be reliably analyzed, due to the absence of the feature in the code or implementation issues.

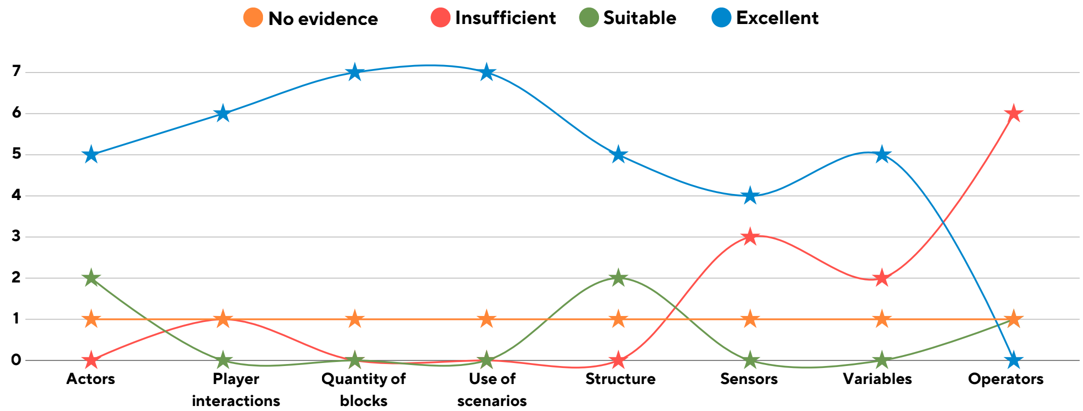

Figure 5. Assessment of CT concepts application in the students' final projects.

This result is particularly relevant, as it indicates students assimilated the presented content and highlights important lessons learned from experience: authorial project development functions as a powerful mediator for consolidating CT concepts among novice learners. Although initial difficulties arose, especially with conditional structures and loops, these challenges were progressively overcome through mediated support and experimentation. The cycle of trial, error, and refinement favored concept consolidation, enabling participants to internalize programming mechanisms in a meaningful way.

The engagement observed in the workshop supports the interpretation that designing and building a game was central to learning, in line with constructionist approaches that prioritize active learning and project-based learning as strategies for understanding abstract concepts. Thus, the fact that the students completed authorial solutions demonstrates conceptual mastery and shows autonomy, creativity, and protagonism in the process, elements essential to the development of competencies associated with CT.

In quantitative terms, the self-assessment presented in Figure 6 indicates an overall positive performance of the workshop. Most participants rated the experience as satisfactory to very satisfactory (levels 4 and 5), indicating a favorable perception of the clarity of explanations, the adequacy of the class pace, and the usefulness of the activities. The data suggest that the methodology was accessible, motivating, and aligned with the students' needs. On the other hand, although representing a minority, the concentration of responses at levels 1 and 2 in the Difficulty Level item indicates a specific challenge that should be considered in the planning of future workshop editions.

However, despite being a minority, responses at levels 1 and 2 in the Difficulty item indicate a cognitive challenge. This self-reported difficulty corroborates findings from the code analysis (particularly the low performance in Sensors and Operators), confirming which conceptual areas should be reinforced in future workshops.

Based on the experience and results obtained, some lessons learned can be highlighted. First, authorial project development proved to be a powerful strategy for consolidating CT concepts among novice learners. Second, although visual programming

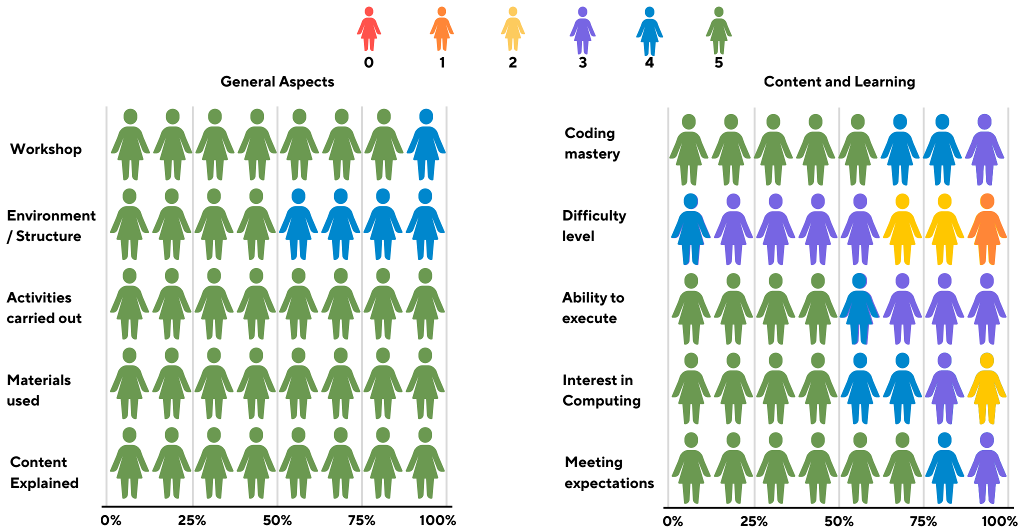

Figure 6. Student feedback on the workshop structure and learning outcomes.

environments such as Scratch facilitate initial engagement, specific support is still needed for higher-level concepts, such as conditionals and operators. Third, the presence of a supportive and inclusive learning environment, especially one composed exclusively of girls, contributed positively to participants' confidence and engagement. Finally, guided mediation played a crucial role in helping students overcome initial difficulties and progress in their understanding of programming concepts.

5. Final Considerations

We presented the conception, implementation, and evaluation of an introductory programming workshop using Scratch for public high school students. This work contributes by presenting a structured, replicable, and context-aware workshop model for introducing programming to high school girls, aligned with CT development. In addition, it provides empirical evidence on students' engagement and learning, as well as insights that support similar initiatives to promote gender inclusion in Computing education. The results indicate that the proposal fostered participants' autonomous engagement in project creation through a practical format that promoted experimentation and skills aligned with the BNCC Computing competencies. Beyond technical aspects, conducting the workshop in a girls-only learning space strengthened engagement in a context of gender inequalities. The study reinforces that integrating Computing in schools requires practices that promote authorship, experimentation, and recognition of students' capacity to create technology. As a limitation, the reduced sample size is due to infrastructure factors such as connectivity, scheduling, and equipment, which limited the generalizability of the results.

As future work, we intend to expand the workshop to other schools, increase the workload, and validate the methodology with a larger and more diverse sample. We also plan to improve CT assessment through formal rubrics or automated tools, such as Dr. Scratch, and conduct longitudinal analyses to measure long-term impacts, including engagement, self-confidence, and knowledge sharing in academic and school communities, as well as study the transition from Scratch to textual languages such as Python. In addition, the workshop model can be adapted and replicated in other public school contexts, contributing to initiatives aimed at promoting girls' engagement in Computing.

6. Acknowledgements

This research was funded by the Brazilian National Council for Scientific and Technological Development (CNPq), the Ministry of Science, Technology and Innovation (MCTI), and the Ministry of Women (MM) through the RENACEE project (Grant No. 440502/2024-1). The authors also acknowledge the support of PROEX/UFJF and the partnership of the Escola Estadual Duque de Caxias in the implementation of the workshops.

Use of Artificial Intelligence

Generative Artificial Intelligence tools were used only for language review and minor editorial support. ChatGPT was employed for text refinement, Gemini Colab was used to support the creation of illustrative figures, and Grammarly was used for minor proofreading. All methodological decisions, analyses, and conclusions remain the sole responsibility of the authors.

References

Bender, W. N. (2012). *Project-based learning: Differentiating instruction for the 21st century*. Corwin Press.

Brasil (2022). Computação na educação básica: Complemento à base nacional comum curricular (bncc). Aprovado pelo Parecer CNE/CEB nº 2/2022.

Calderon, I., Araújo, V., Rios, M., and Tamada, M. (2025). Projeto meninas robóticas para incentivar a participação feminina na robótica: Um relato de experiência. In *Anais do XIX Women in Information Technology*, pages 819–830, Porto Alegre, RS, Brasil. SBC.

Coutinho, F., e Silva, A. J., Velten, H., Oliveira, R., Dias, L., Valle, P., and Oliveira, A. (2025). Transformando a presença feminina em computação a partir de oficinas de programação em python. In *Anais Estendidos do XXXI Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 215–221, Porto Alegre, RS, Brasil. SBC.

Deus, S., Freire, S., and de Farias, C. (2020). Um estudo sobre as dificuldades de inserção de meninas na computação. In *Anais do XIV Women in Information Technology*, pages 274–278, Porto Alegre, RS, Brasil. SBC.

Ferreira, S., Cavalcante, V., Oliveira, A., and Costa, C. (2025). Do scratch ao python: utilizando metodologias ativas no ensino de programação no projeto meninas digitais do acre. In *Anais do XXXIII Workshop sobre Educação em Computação*, pages 849–860, Porto Alegre, RS, Brasil. SBC.

Garcia, J. and Bittencourt, R. (2023). Um mapeamento sistemático da literatura sobre pensamento computacional na perspectiva dos fundamentos teóricos de aprendizagem. In *Anais do III Simpósio Brasileiro de Educação em Computação*, pages 01–12, Porto Alegre, RS, Brasil. SBC.

Genesio, N., Menezes, M., Almeida, J., Boaventura, A., and Valle, P. (2023). Aprendendo lógica de programação desenvolvendo jogos digitais: Um relato de experiência. In *Anais do XXIX Workshop de Informática na Escola*, pages 375–386, Porto Alegre, RS, Brasil. SBC.

Gomes, W., Louzada, C., Nunes, M., Salgueiro, E., and Andrade, B. (2014). Incentivando meninas do ensino médio à área de ciência da computação usando o scratch como ferramenta. In *Anais do XX Workshop de Informática na Escola*, pages 223–232, Porto Alegre, RS, Brasil. SBC.

Lopes, R., Maciel, B., Soares, D., Figueiredo, L., and Carvalho, M. (2023). Análise e reflexões sobre a diferença de gênero na computação: podemos fazer mais? In *Anais do XVII Women in Information Technology*, pages 68–79, Porto Alegre, RS, Brasil. SBC.

Pereira, W., Silva, M., Alves, M., Santana, M., and Rodrigues, R. (2025). Educação midiática e digital com o scratch. In *Anais do XXXIII Workshop sobre Educação em Computação*, pages 302–314, Porto Alegre, RS, Brasil. SBC.

Pinto, N., Araujo, A., Mendes, E., Pavani, G., Silva, K., Neto, R., Salles, K., and Rivero, L. (2020). Quando a aluna se torna a mestre: Um relato da experiência de alunas de graduação aplicando dinâmicas de ensino de computação para alunas de ensino médio. In *Anais do XXVIII Workshop sobre Educação em Computação*, pages 36–40, Porto Alegre, RS, Brasil. SBC.

Prince, M. J. and Felder, R. M. (2006). Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of Engineering Education*, 95(2):123–138.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. B. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11):60–67.

Santiago, C., Mendonça Menezes, J. W., and Alves de Aquino, F. J. (2023). Proposta e avaliação de uma metodologia de aprendizagem baseada em projetos em disciplinas de engenharia de software através de uma sequência didática. *Revista Brasileira de Informática na Educação*, 31:31–59.

Santos, A., de Oliveira, A., Dias, A., and França, J. (2024). Storygirl na prática: Um relato de experiência sobre as práticas colaborativas no projeto de extensão. In *Anais Estendidos do XIX Simpósio Brasileiro de Sistemas Colaborativos*, pages 170–173, Porto Alegre, RS, Brasil. SBC.

Santos, N. and Marczak, S. (2023). Fatores de atração, evasão e permanência de mulheres nas Áreas da computação. In *Anais do XVII Women in Information Technology*, pages 136–147, Porto Alegre, RS, Brasil. SBC.

Souza, A. P., Uchôa, A., Canedo, E. D., Pereira, J. A., Pereira, C. P., and Rocha, L. (2025). Overcoming obstacles: Challenges of gender inequality in undergraduate ict programs.

United Nations (2015). Transforming our world: the 2030 agenda for sustainable development. United Nations. Accessed: 2025.

Wing, J. M. (2021). Pensamento computacional. *Educação e Matemática*, (162):2–4.