Unplugged Memory: A Report of an Unplugged Activity in High School Education with a Technical Specialization in Brazil

Pedro Clarindo da Silva Neto[a], Arthur Octavio Confessor[b], Suellen Lages[c], João Paulo Delgado Preti[d], Tiago de Alameida Lacerda[e] and Thiessa Esteves Leite[f]

Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso, Campus Cuiabá - Cel. Octayde Jorge da Silva, Rua Zulmira Canavarros, 95 - Centro, Cuiabá, Brazil

Keywords: Unplugged Computing, Computational Thinking, High School Education, Unplugged Memory.

Abstract: Courses such as Algorithms and Programming Language face significant challenges in High School technical education in Brazil due to the high levels of abstraction required, leading to low grades, high repetition, and dropout rates. To address these issues, unplugged activities were implemented in the Programming Fundamentals course to foster Computational Thinking concepts: decomposition, pattern recognition, abstraction, and algorithm design. These activities included the creation of board games designed to simplify complex topics such as dynamic memory allocation and pointers. By breaking down abstract concepts into tangible and playful activities, students demonstrated greater engagement and reduced learning barriers. The feedback of the students indicated high satisfaction with the approach, highlighting its potential to bridge the gap between foundational and advanced technical concepts. This paper introduces Unplugged Computing as a practical strategy for improving the teaching-learning process in Programming Fundamentals during the year 2023.

1 INTRODUCTION

Dropout rates in Brazilian technical education are notably high, particularly in subjects such as programming fundamentals, where students face significant abstraction challenges. According to (Brackmann, 2017), (Giraffa and Mora, 2013), and (Rodrigues et al., 2015), introductory programming topics such as Algorithms, Programming Logic, or Fundamentals of Programming contribute significantly to failures and retention, as they are prerequisites for subsequent courses. This, in turn, exacerbates dropout rates, as illustrated by data from the Basic Education School Census (INEP, 2023).

Although enrollment in high school technical programs has increased by 32. 2% in the past five years (Figure 1), dropout rates remain a significant problem (Figure 2). According to the most recent data from 2019, a rising trend of dropouts—defined as students ceasing to attend classes for an entire academic year—has been observed, particularly in federal institutions. Factors such as academic under performance and the challenges of transitioning to a full-time technical education environment with higher expectations contribute to this problem (da Silva, 2020).

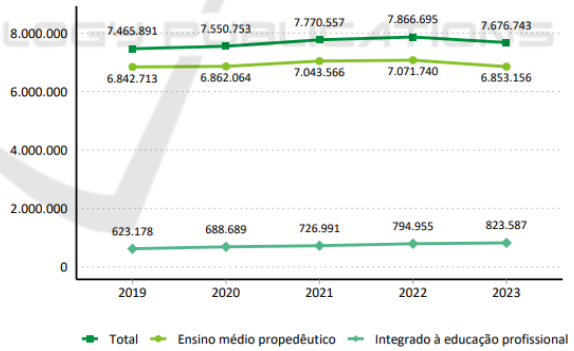

Figure 1: Number of Enrollments in High School Education (Total, Integrated, and Not Integrated with Professional Education).

To address these challenges, innovative teaching methodologies are essential. Previous studies have shown that active learning strategies and tools such

[a] https://orcid.org/0000-0002-4195-624X
[b] https://orcid.org/0009-0009-0784-3661
[c] https://orcid.org/0009-0002-5350-1629
[d] https://orcid.org/0009-0002-2808-2311
[e] https://orcid.org/0009-0002-0689-1783
[f] https://orcid.org/0009-0009-5872-0012

Neto, P. C. S., Confessor, A. O., Lages, S., Preti, J. P. D., Lacerda, T. A. and Leite, T. E.
Unplugged Memory: A Report of an Unplugged Activity in High School Education with a Technical Specialization in Brazil.
DOI: 10.5220/0013185200003932
In *Proceedings of the 17th International Conference on Computer Supported Education (CSEDU 2025) - Volume 2*, pages 649-653
ISBN: 978-989-758-746-7; ISSN: 2184-5026

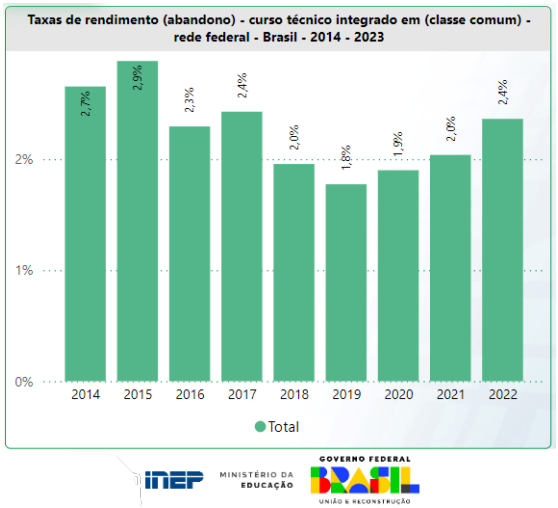

Figure 2: Students dropout rate in high school education with technical specialization in the federal education network in Brazil.

as Scratch (Rocha et al., 2013) can improve engagement and reduce failure rates. Scratch, originally developed by MIT, has been widely used to teach programming concepts in a playful and accessible manner (Rocha et al., 2013). Similarly, Computational Thinking (CT) has emerged as a critical competency in technology education, fostering skills such as decomposition, pattern recognition, abstraction, and algorithm design (BNCC, 2017). These skills, fundamental to problem-solving, are integral to students' ability to engage with abstract programming concepts (Costa. et al., 2021).

Unplugged Computing, as disseminated by (Tim Bell and Fellows, 2015), provides a means of teaching Computational Thinking through playful, hands-on activities that do not require computers. This approach is particularly advantageous in contexts with limited infrastructure, offering a transparent and engaging way to introduce fundamental computing concepts (Araújo et al., 2021). By focusing on real-world problems and collaborative tasks, unplugged activities foster deeper engagement and understanding.

In this context, this article aims to present an unplugged activity focused on "dynamic memory allocation", developed during the third term of the Informatics course (a technical specialization course in high school), within the subject of Programming Fundamentals. This activity leverages Computational Thinking concepts to aid the teaching-learning process, providing students with an innovative and accessible approach to mastering complex programming topics.

2 METHODS

The activities were carried out with high school first-year students (informatics technical course) at the Federal Institute of Mato Grosso and applied in the school subject of Programming Fundamentals. The unplugged activities took place in the second half of the year. Each lesson lasted 50 minutes, the institution standard lesson time. The activities were structured over three sessions: the first dedicated to a theoretical lecture covering the key concepts, while the second and third sessions focused on the development and application of hands-on activities.

Programming Fundamentals is a first-year course discipline; topics such as recursion, memory pointers, and dynamic memory allocation are covered in the third quarter. These topics often require students to transition from the concrete knowledge gained in elementary education to abstract thinking, which is a significant cognitive leap. This transition is particularly challenging in technical courses, where students are required to grasp concepts such as memory structures and algorithmic thinking early in their academic journey.

To mitigate these challenges, simpler activities, such as unplugged computing exercises, were introduced. These activities were designed to foster engagement and facilitate understanding by breaking down abstract concepts into tangible and interactive components. The main goal was to create a supportive learning environment that encourages exploration and participation while reducing cognitive overload.

The unplugged activities were carefully designed to incorporate key concepts of Computational Thinking (CT), which is increasingly recognized as a critical competency in technology education. CT skills, such as decomposition, pattern recognition, abstraction, and algorithm design, were explicitly embedded in the activities to ensure that students not only understood theoretical concepts but also applied them in practice:

- **Decomposition.** Students were tasked with breaking down complex programming concepts into smaller, manageable components. For instance, in board game design, they separated the concept of dynamic memory allocation into its core elements: data types, memory addresses and pointers pointers operations. This incremental approach allowed the students to address individual aspects of the problem in a systematic way.
- **Pattern Recognition.** Students analyzed how different data types address different memory spaces and patterns in memory allocation. This activity

helped them internalize these patterns, which are essential for understanding the principles of efficient memory management.

- **Abstraction.** Students learned to focus on the essential elements of memory allocation while ignoring less relevant details. By simplifying complex problems into general principles, students improved their ability to approach abstract programming challenges.
- **Algorithm Design.** The creation of board games required students to develop rules and step-by-step procedures that simulated real-world programming tasks. This hands-on experience fostered a deeper understanding of algorithms, as students had to ensure their game mechanics accurately reflected the logical operations of memory management.

Students were encouraged to design and implement their own board games using simple materials such as paper, glue, pens, scissors and poster board. This approach promoted creativity, collaboration, and critical thinking. Unlike traditional teaching methods, that rely heavily on passive learning, these activities placed students at the center of the learning process, empowering them to actively construct their understanding.

To assess the effectiveness of these activities, an online questionnaire was sent to the students. The questionnaire consisted of five questions, including an open-ended question and four closed-ended questions. This approach enabled the collection of quantitative and qualitative data, providing insights into students' perceptions of the unplugged computing approach and highlighting areas for potential improvement.

3 RESULTS AND DISCUSSION

This study aimed to evaluate the effectiveness of unplugged activities in promoting engagement and understanding among first-year technical students. The results demonstrated that the activities not only facilitated the understanding of abstract programming concepts, but also promoted a collaborative and interactive learning environment.

The first unplugged activity involved a game designed to visually represent how information is stored in a computer's memory (Figure 3). The game employed different colored pieces of paper to symbolize different data types, each occupying a specific amount of memory. By interacting with these physical figures, students gained a clearer understanding of concepts such as memory storage and the use of operators such as "sizeof" to calculate memory requirements.

Figure 3: Board game on memory allocation.

The second activity, "Deu Alloc na Memória", introduced the concept of pointers and dynamic memory allocation (Figure 4). The board game included a dice in which the faces represented a data type or a pointer-related action. The game execution consists of players moving around the board by rolling the dice, drawing cards and executing actions that simulate a memory operation. Through this playful interaction, students engaged with critical concepts such as memory addresses, pointer usage and memory allocation strategies.

Figure 4: Board Game "Deu alloc na memória.".

Classroom discussions and questionnaire responses provided second insights. Students frequently reported that developing their own unplugged games reinforced their understanding of programming concepts by requiring them to actively apply and explain these ideas. This observation aligns with prior research on active learning, which highlights the

value of student-led, hands-on activities in promoting deeper understanding and retention of abstract topics.

Since it was not possible to quantitatively measure knowledge retention at this stage, qualitative feedback indicated that students were more engaged and demonstrated greater interest in the subject matter compared to previous cohorts. Teachers noted that students asked more insightful questions during lessons and were better prepared to tackle advanced topics. The collaborative nature of the activities also helped build a supportive learning environment, promoting peer interaction and critical thinking.

3.1 Assessment of the Activities Developed

Near the end of the term, nine students voluntarily completed an online questionnaire. The results revealed unanimous satisfaction with the unplugged activities, with 100% of respondents stating that the exercises enhanced their understanding of the material. When asked about their preferred teaching method, all students indicated a preference for a combination of traditional lectures and unplugged activities.

Additionally, 88.9% of participants expressed interest in using unplugged computing to explore other programming topics. Suggestions for future activities included data structures such as stacks, queues, and linked lists, as well as advanced topics like artificial intelligence and applied mathematics (Figure 5).

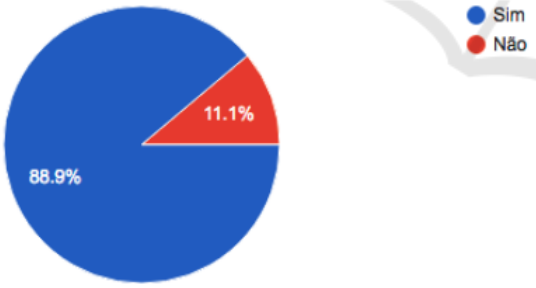

Figure 5: Responses on the use of Unplugged Computing in teaching other concepts.

The feedback provided by students underscores the versatility and potential of unplugged computing as a pedagogical strategy. Moving forward, future studies will aim to incorporate standardized evaluation methods, such as pre- and post-activity assessments, to quantitatively measure the impact of unplugged activities on students' academic performance and knowledge retention.

4 CONCLUSION

Teaching programming in technical courses presents a major challenge for educational institutions: reducing the number of failures and retentions in these subjects without compromising the quality of education, ensuring that improvements are not merely numerical. To address this, new methodologies have been sought to assist in the teaching-learning process. Unplugged Computing, one of the techniques of Computational Thinking, has proven to be a strong ally in this process.

The implementation of unplugged activities in the Programming Fundamentals course demonstrated the potential of key Computational Thinking concepts—decomposition, pattern recognition, abstraction, and algorithm design—in simplifying complex topics. Decomposition allowed students to break down abstract problems, such as dynamic memory allocation, into smaller, manageable components, while pattern recognition enabled them to identify recurring structures in memory management. Abstraction helped students focus on the most relevant aspects of the problems, and algorithm design encouraged the creation of structured solutions through playful activities such as board games. These concepts not only facilitated understanding but also enhanced engagement and collaboration among students.

The observation that students better absorbed concepts through the creation of unplugged activities is supported by classroom discussions and questionnaire responses. Students actively applied and explained their understanding while designing their games, reinforcing their grasp of abstract programming topics. This aligns with prior research on active learning, which suggests that hands-on, student-led approaches promote deeper understanding and retention.

The fact that it does not require the use of any electronic devices makes Unplugged Computing a suitable activity for various contexts, including field trips, technical visits, and environments with limited infrastructure. This adaptability is particularly relevant in the Brazilian educational landscape, where resource constraints can hinder the adoption of technology-based teaching strategies. Moreover, the versatility of unplugged computing allows its application in other areas of knowledge, reinforcing its value as a flexible pedagogical tool.

However, it is important to acknowledge the limitations of this study. The evaluation relied primarily on self-reported feedback from students, which, while insightful, does not provide objective measures of knowledge retention or academic performance. Future research will incorporate standardized evaluation

methods, such as pre- and post-activity assessments, to objectively measure the impact of unplugged activities on students' learning outcomes. These assessments will enable a more comprehensive understanding of the effectiveness of this approach and its potential for broader application.

As future work, the goal is to expand the range of unplugged activities to address more complex topics that students find challenging, such as data structures and algorithms. Additionally, a broader research effort on Computational Thinking and the use of Unplugged Computing is planned, with the aim of developing training programs for educators. By equipping teachers with the skills to implement these techniques in their classrooms, we can further enhance the quality and accessibility of informatics education, bridging the gap between foundational knowledge and advanced technical competencies.

REFERENCES

Araújo, Q., Lima, A., Junior, A. C., and Oliveira, W. (2021). Uma proposta para uso de computação desplugada no ensino híbrido. In *Anais Estendidos do XXXII Simpósio Brasileiro de Informática na Educação*, pages 08–14, Porto Alegre, RS, Brasil. SBC.

BNCC (2017). Base nacional comum curricular.

Brackmann, C. P. (2017). *Desenvolvimento do Pensamento Computacional através de atividade desplugadas na Educação Básica.* Tese, UFRGS, Centro de Estudos Interdisciplinares em Nova Tecnologias na Educação, Porto Alegre, RS.

Costa., E. J. F., Vitorino., M. G. S. Q., Medeiros., J. M. L., Campelo., C. E. C., and Campos., L. M. R. S. (2021). A digital application to assist basic education teachers in the interdisciplinary development of computational thinking skills on the math discipline in brazilian learning context. In *Proceedings of the 13th International Conference on Computer Supported Education - Volume 1: CSEDU*, pages 475–482. INSTICC, SciTePress.

da Silva, F. (2020). Curso técnico em informática integrado ao ensino médio: Percepções sobre a evasão. Dissertação (mestrado em educação), Universidade do Vale do Sapucaí (UNIVÁS), Pouso Alegre, MG. Orientador: Rosimeire Aparecida Soares Borges.

Giraffa, L. and Mora, M. (2013). Evasão na disciplina de algoritmo e programação: Um estudo a partir dos fatores intervenientes na perspectiva do aluno. In *III Conferencia Latinoamericana sobre el Abandono en la Educación Superior (III CLABES)*.

INEP (2023). Censo escolar da educação básica 2023: Resumo técnico, versão preliminar. República Federativa do Brasil, Ministério da Educação (MEC).

Rocha, A., Silva, J., and Carneiro, D. (2013). Utilização do scratch como ferramenta de auxílio à aprendizagem de programação. In *XLI Congresso Brasileiro de Educação em Engenharia*.

Rodrigues, F. S., Brackmann, C. P., and Barone, D. A. C. (2015). Estudos da evasão no curso de ciência da computação da ufrgs. *Revista Brasileira de Informática na Educação*, 23(1).

Tim Bell, I. H. W. and Fellows, M. (2015). *Computer Science Unplugged*. CS Unplugged.