VIDYA

VIDYA, v. 45, n. 2, p. 679-692, jul./dez. - Santa Maria, 2025.

Recebido em: 26.05.2025. Aprovado em: 03.02.2026.

ISSN 2176-4603 | DOI: doi.org/10.37781/vidya.v45i2.5393

A WINNING STRATEGY ANALYSIS IN THE GAME "RACE TO 20" THROUGH A COMPUTATIONAL THINKING PERSPECTIVE

UMA ANÁLISE DA ESTRATÉGIA VENCEDORA NO JOGO "CORRIDA AO 20" ATRAVÉS DE UMA PERSPECTIVA DO PENSAMENTO COMPUTACIONAL

UN ANÁLISIS DE LA ESTRATEGIA GANADORA EN EL JUEGO "CARRERA AL 20" DESDE UNA PERSPECTIVA DEL PENSAMIENTO COMPUTACIONAL

JOSÉ VITOR RAMOS DE LIMA[1]
PÉRICLES DE LIMA SOBREIRA[2]
FRANCK GILBERT RENÉ BELLEMAIN[3]
AIRTON TEMISTOCLES GONÇALVES DE CASTRO[4]

ABSTRACT

This article aims to analyze the winning strategy of Race to 20 in the light of the Computational Thinking (CT) pillars. This study was developed collaboratively between members of the Atelier Digitas research group at the Federal University of Pernambuco (UFPE, Brazil), and the Department of Engineering and Computer Science at the Université du Québec en Outaouais (UQO, Canada). The methodological approach included the mathematical modeling of the winning strategy based on the Theory of Combinatorial Games, followed by its analysis according to CT pillars: problem formulation, decomposition, abstraction, pattern recognition, algorithm design, and debugging. The results revealed the potential of problem solving through educational games, and the analysis under the CT framework enabled possibilities of investigation for applying Race to 20 in different educational levels (from primary to higher education), especially in the teaching of progressions, patterns, and mathematical induction.

Keywords: Computational Thinking; Problem Solving; Mathematical Games; Race to 20.

RESUMO

Este artigo tem como objetivo analisar a estratégia vencedora do jogo "Corrida ao 20" à luz dos pilares do Pensamento Computacional (PC). Este estudo foi desenvolvido de forma colaborativa entre membros do grupo de pesquisa Atelier Digitas da Universidade Federal de Pernambuco (UFPE, Brasil) e do Departamento 'd'informatique et d'ingénierie' da Université du Québec en Outaouais (UQO, Canadá). O percurso metodológico incluiu a modelagem matemática da estratégia vencedora com base na Teoria dos Jogos Combinatórios e, posteriormente, sua análise segundo pilares do PC: formulação de problema, decomposição, abstração, reconhecimento de padrões, produção de algoritmos e depuração. Os resultados revelaram o potencial da resolução de problemas a partir de jogos educacionais, assim como a análise sob à luz do PC viabilizou uma investigação de possibilidades da aplicação do Corrida ao 20 em diferentes níveis de educação (do primário ao ensino superior), especialmente no ensino de progressões, padrões e indução matemática.

Palavras-chave: *Pensamento Computacional; Resolução de Problemas; Jogos Matemáticos; Corrida ao 20.*

1 Doutorando em Educação Matemática e Tecnológica. Universidade Federal de Pernambuco. E-mail: jose.lima@ufpe.br. ORCID: https://orcid.org/0000-0001-6431-4832

2 Doutor em Informatique. Université du Québec. E-mail: pericles.delimasobreira@uqo.ca. ORCID: https://orcid.org/0000-0002-4869-3155

3 Doutor Didactique des Mathématiques. Universidade Federal de Pernambuco. E-mail: franck.bellemain@ufpe.br. ORCID: https://orcid.org/0000-0001-5358-2057

4 Doutor em Matemática. Universidade Federal de Pernambuco. E-mail: airton.castro@ufpe.br. ORCID: https://orcid.org/0000-0003-3096-2529

RESUMEN

Este artículo tiene como objetivo analizar la estrategia ganadora del juego "Carrera al 20" a la luz de los pilares del Pensamiento Computacional (PC). El estudio se desarrolló de manera colaborativa entre miembros del grupo de investigación Atelier Digitas de la Universidade Federal de Pernambuco (UFPE, Brasil) y del Departamento de Informática e Ingeniería de la Université du Québec en Outaouais (UQO, Canadá). El recorrido metodológico incluyó la modelización matemática de la estrategia ganadora basada en la Teoría de los Juegos Combinatorios y, posteriormente, su análisis según los pilares del PC: formulación del problema, descomposición, abstracción, reconocimiento de patrones, producción de algoritmos y depuración. Los resultados evidencian el potencial de la resolución de problemas mediante juegos educativos, así como que el análisis desde la perspectiva del Pensamiento Computacional posibilita la investigación de distintas formas de aplicación del juego Carrera al 20 en diferentes niveles educativos (desde la educación primaria hasta la educación superior), especialmente en la enseñanza de progresiones, patrones e inducción matemática.

Palabras clave: *Pensamiento Computacional; Resolución de Problemas; Juegos Matemáticos; Carrera al 20.*

INTRODUCTION

Computational Thinking (CT) has gained prominence in recent years, especially in educational contexts, due to its applicability across various fields of knowledge. Jeannette M. Wing, a computer scientist and professor at Columbia University, played a key role in popularizing this theoretical-methodological approach (Wing, 2006). As stated by Haimar (2021), one of Wing's goals was to highlight the relevance of Computer Science in different areas and to promote its learning through the concept of Computational Thinking.

Other researchers have also adopted Computational Thinking in his pedagogical practices, such as Brackmann (2017), who defines it as a set of skills that enable the integration of computational concepts into problem-solving situations. The author also emphasizes that its application is not limited to computing, but may extend to various disciplines and educational contexts.

CT strengthened the idea of integrating computing into educational programs by associating it with core concepts of Computer Science, such as abstraction and algorithm design, to solve problems and design systems. Wing (2006) argued that this approach could be adopted by everyone, fostering a mindset similar to that of a computer scientist, whether through digital tools or logical reasoning inspired by computational methods.

With the advancement of research on Computational Thinking, Wing (2021) proposed a methodology for solving problems and overcoming obstacles using Computer Science foundations. This has led to the emergence of various studies in Mathematics Education, particularly those examining problem-solving through CT, such as the works developed by Padilha *et al.* (2024), Perceval *et al.* (2024), Rocha and Basco (2024), and Padilha *et al.* (2023).

Padilha *et al.* (2024) analyzed the resolution of a mathematical problem based on CT processes, and obtained significant results by demonstrating how the CT foundations were applied during the solution of a mathematical task. This research highlights the importance of aligning Computational Thinking with the resolution of mathematical activities. This is also evident in Rocha and Basco (2024), which emphasize the importance of CT in global educational contexts, especially when scaffolded by technology. In this sense, educators and researchers aim to contribute to the integration of CT into school technological settings, enhancing student learning.

Based on this, we investigate how the pillars of Computational Thinking can be used to analyze the mathematics behind the winning strategy in Race to 20. This game, discussed by Guy Brousseau in *Théorie des Situations Didactiques* (Brousseau, 1998), can be explored across different educational levels (from primary to higher education) with various educational purposes (Lima, 2023). It involves players alternating turns by adding numbers until someone reaches 20, requiring the identification of patterns and development of strategies to win. Given the relevance of games in mathematics education, as highlighted by various studies (Lima, 2023; Boller, 2018; Brasil, 2018), this paper explores how CT can further enhance the use of educational games.

This article presents excerpts from the first author's master's research in Brazil (2023), as well as early results from a study conducted by him during his doctoral sandwich program in Canada (2025). It aims to analyze the mathematics behind the winning strategy of Race to 20 in light of CT foundations.

Accordingly, the reminder of this article is organized into five sections, as it follows. In the first one, we introduce the article's motivation and objectives. Next, we present the pillars of Computational Thinking that support this study. After, we describe the research's methodological approach followed by Race to 20 and the mathematics that scaffolds its winning strategy. Next, we analyze the strategy in light of the CT foundations, and finally, we conclude the work.

PILLARS OF COMPUTATIONAL THINKING AND THEIR CONNECTION TO MATHEMATICS EDUCATION

Although Wing (2006) conceptualized the idea of Computational Thinking (CT), Navarro and Sousa (2019) emphasize that CT emerged in Mathematics Education through the Logo programming language, in a research conducted by Seymour Papert, who aimed to make ideas more accessible (Papert, 1980). However, over time both (CT and Logo) lost prominence in Mathematics Education, mainly due to insufficient teacher training and the lack of effective implementation of them in school curricula (Savioli *et al.*, 2023).

With the advancement of technologies, new terms such as Digital Technologies, Information Technologies, and Information and Communication Technologies emerged in school environments, shifting the focus from programming and algorithms to the use of technologies in general. As a result, CT became a field restricted to computer scientists, but has recently regained attention within Mathematics Education (Padilha *et al.*, 2024).

Studies such as those developed by Canal *et al.* (2024) and Rocha and Basso (2024) highlight the relevance of CT in Mathematics Education. Although their research follows different directions, both bring important reflections on the integration of CT in initial and continuing teacher education, as well as its connection with technological resources like Scratch[5].

Considering the different conceptions of CT, this work adopts the definitions presented by Espadeiro (2021) and Padilha *et al.* (2023), which outline the following pillars of Computational Thinking: problem formulation, decomposition, pattern recognition, abstraction, algorithm design, and debugging, defined as follows:

5 Programming language created in 2007 by the MIT Media Lab. Since 2013, Scratch 2 has been available online and as an application for Windows, OS X, and Linux (https://scratch.mit.edu).

- **Problem formulation**: This process[6] involves designing the problem by considering its needs, objectives, and possible solutions. It includes reflecting on what should be considered, what variations are involved, what actions can be taken, and which plans are feasible, based on appropriate techniques and repertoires (Espadeiro, 2021, p. 5).
- **Decomposition:** This consists of breaking down a complex problem into smaller and more manageable parts, facilitating its resolution through the resolution of its subproblems. It involves identifying familiar or unfamiliar parts, possible approaches, and how the situation's details can support understanding (Espadeiro, 2021, p. 6).
- **Pattern recognition:** This process involves identifying constant or variable elements in the data, based on prior experiences and mathematical/computational repertoires. Questions such as "what patterns are repeated?", "what relationships exist between the parts?", and "how do these regularities guide solutions?" help with analysis and strategy development (Espadeiro, 2021, p. 6).
- **Abstraction:** Refers to focusing only on what is essential to solve the problem, ignoring irrelevant information. It involves identifying relevant data, clearly representing it, and relating it to the starting point and the intended outcome (Espadeiro, 2021, p. 7).
- **Algorithm design:** Refers to creating steps or rules to solve problems using natural language, mathematics, programming, among others. It includes defining clear stages, required information, and the best way to structure the sequence of actions (Espadeiro, 2021, p. 7).
- **Debugging:** Refers to identifying and correcting errors, including testing, refining, and optimizing the proposed solution. It may involve reviewing or redefining strategies based on result analysis and ongoing adjustments (Espadeiro, 2021, p. 7).

METHODOLOGICAL STRATEGY

This research is methodologically grounded by the CT foundations, taking Race to 20 as its object of study. As previously informed, we are guided by the CT pillars outlined by Espadeiro (2021) and Padilha *et al.* (2023).

This study was started in the *Atelier Digitas* research group of the Center for Arts and Communication and the Graduate Program in Mathematics and Technology Education (EDUMATEC) at the Federal University of Pernambuco (UFPE). This group comprises researchers from various national and international institutions and is dedicated to the design, development, and validation of digital tools aiming to manage educational situations. The active involvement of its participants in the design and discussion of Race to 20 played a significant role in deepening the analysis presented in the present study.

This analysis was structured in three main stages: (i) specification of the mathematical object involved in Race to 20 (developed in a digital version on the Scratch platform); (ii) modeling of the winning strategy based on the assumptions of the Theory of Combinatorial Games; and (iii) analysis of the strategy in light of the pillars of Computational Thinking.

This digital version aimed to explore the pedagogical potential of the mathematical resource, although it was not applied with students at this stage of the research since it was not the central objective of this article, mainly due to space limitations for discussion. The analysis of the winning strategy in Race to 20 was carried out with the support of members of the *Atelier Digitas* group (UFPE) and the

6 In this work, processes and pillars should be understood as synonyms to refer to the foundations of Computational Thinking.

Department of Computer Science and Engineering at the Université du Québec en Outaouais (UQO), during the doctoral sandwich program performed by the first author of this article.

It is important to note that although this work focuses on the mathematical strategy behind the game, it was enriched through several exchanges between both groups of research, which reinforced the theoretical and methodological coherence of the investigation. These interactions also contributed to the formulation of questions for future discussions, including the possibility of practical applications and evaluations in educational contexts.

As part of the study, a detailed examination of Race to 20 was conducted, including an introductory presentation of its rules and mathematical structure. The digital version developed in Scratch is presented as a complementary resource intended to teachers or researchers in Mathematics Education (its technical development is not the focus of this article). Following the theoretical analysis of the game, the pillars of Computational Thinking mobilized in the formulation of the winning strategy were identified and discussed.

RACE TO 20: WHAT KIND OF GAME IS THIS?

In addition to providing entertainment, some games are also intended to stimulate autonomy, reasoning, cooperation, collaborative work, understanding of rules, self-assessment, among other skills. In such a way, games can be used as tools for overcoming mistakes by guiding students in a certain direction, prompting them to create meaning and look for solutions or adaptations for solving problem situations (Lima, 2023).

As previously mentioned, we present here Race to 20, which is a relatively simple game played by two opponents. One of them starts by choosing between two options: numbers 1 or 2. The opponent then adds 1 or 2 to the number previously stated, saying only the result. Alternately, the game continues with such rules until a player says the number 20 first (which will be considered the winner of the game) (Brousseau, 1998). An example of a match in Race to 20 can be seen in Table 1 (in this table, time runs from left to right, and from top to bottom).

Table 1 - Example of a match in Race to 20.

Player (P1)	Player (P2)
3 + 2 = 5	5 + 2 = 7
7 + 1 = 8	8 + 1 = 9
9 + 2 = 11	11 + 2 = 13
13 + 1 = 14	14 + 2 = 16
16 + 1 = 17	17 + 2 = 19
19 + 1 = 20 (the winner)	

Source: adapted from Lima (2023).

Notice that the player referred to as P1 started the game by saying two. At this point, player P2 had two possible moves: saying three or saying four. By choosing three, P2 added one to the previously stated number. On P1’s second move, they added two and said five, and the match continued until P1 won by saying twenty. This example represents just one of the many possible configurations of Race to 20.

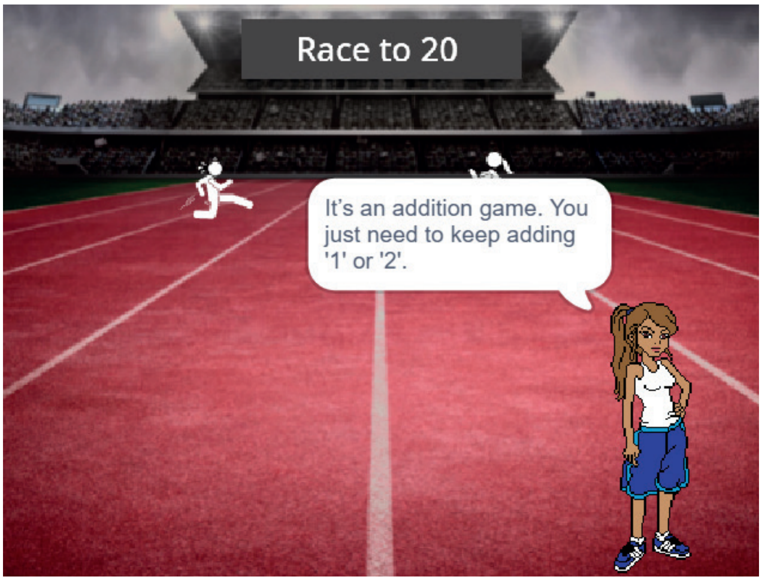
VIDYA, v. 45, n. 2, p. 679-692,
jul./dez. - Santa Maria, 2025.

From this brief analysis of a match, we can raise some questions, as outlined by Bittar (2017): (a) Is there a strategy to always win the game? (b) Player P1won the game, but was it intentional? (c) At any point, did the player P2 have a chance to win as well? These questions might be answered intuitively, but there are assumptions from the Theory of Combinatorial Games that can help us discover the winning strategy, as it will be presented later in this article. In the meantime, reader, you can start thinking about it.

To better visualize this game, we decided to develop its digital version in Scratch, in such a way to engage its users (students and teachers) to manipulate/program it without requiring of them an extensive knowledge in Computer Science. Moreover, students and teachers can examine the underlying structure of the game to explore improvements by modifying its code blocks. Thus, Scratch enables both teachers and students to test, run, and, through a trial-and-error approach, generate new learning experiences (which aligns with the Computational Thinking theory).

Reader, you are invited to play a few rounds, formulate ideas, and even try to identify patterns in the Scratch version of Race to 20 developed in this research. For this, you should visit the link https://scratch.mit.edu/projects/1180534119/. Figure 1 shows its initial interface (in there, just click on it to start a match).

Figure 1 - Interface of the digital version of Race to 20.



Source: Digital game developed in Scratch by authors.

As you can see, Race to 20 is not a complex game, however, it involves knowledge and mathematical concepts that can be explored through it. We will analyze it based on the assumptions of the game “Nim”, which was one of the first games to be studied mathematically as part of Combinatorial Game Theory and has a complete mathematical theory. (Bouton, 1901). Which has the following properties: finite mathematical game that cannot end in tie; two players (P1 and P2) taking turns; complete information; and no element of chance. Later, we will also use the pillars of Computational Thinking to support systematization.

VIDYA, v. 45, n. 2, p. 679-692,
jul./dez. - Santa Maria, 2025.

THE RELATIONSHIP BETWEEN NIM AND RACE TO 20

Since we are interested in investigating the winning strategy behind Race to 20, which corresponds to our problem formulation, it is important to understand the theoretical assumptions of games categorized as Nim-type, which will assist in the process of identifying the strategy.

The first scientific publication on Nim was by mathematician Charles Leonard Bouton in 1901, in his article titled “Nim, a Game with a Complete Mathematical Theory.” However, its exact origin is unknown. Some believe it originated in China, while others trace it to Old English. Interestingly, some have noted that ‘NIM’ means ‘to take’, and when reversed to ‘WIN’, it can mean ‘to win’ (Bouton, 1901).

The game has several versions, such as Dice Nim, Starred Nim, One-Pile Nim, Three-Pile Nim, Chain Game, Circular Nim, Queen Game, Horse Game, LIM, among others. These are classified as Nim games because they meet the following conditions: exactly two players; players take turns; complete information; finite; no possibility of a tie; deterministic; and impartial. More about Nim can be found in the works of Pessoa (2013), Lima (2023), Torres (2017), and Moreira (2014).

Note that Race to 20 satisfies all these conditions, and we can therefore affirm that it is a Nim-type game, though we will provide further discussion throughout this article. Among the versions mentioned, the one that most closely resembles Race to 20 is One-Pile Nim. This version consists of a pile of pieces in a finite quantity and includes the following rules: two players; each player, on their turn, may remove a certain number of pieces (in our case, $S = \{1, 2, 3\}$); the winner is the one who removes the last piece(s).

Based on the studies by Castro *et al.* (2019), we bring some explanations and definitions. A winning strategy is one where Player 1 has such a strategy that, for every move made by Player 2, there is at least one response by Player 1 that leads them to victory. Thus, regardless of what Player 2 does, Player 1 will win the game. However, to establish this, it is necessary to perform what is called move mapping: determining each position and verifying who has the winning strategy - Player 1 or Player 2. We define these positions as Losing Position (P) and Winning Position (G): all final positions are G; any position that leads only to G is a P; any position that leads to at least one P is a G.

From this, we can map the One-Pile version of the Nim game using, for example, a match with 16 pieces. It is important to note that the number of pieces can vary, as long as it is finite. Position ‘0’ means no pieces are left to be taken, and thus it is a losing position (P).

Note that positions ‘1’, ‘2’, and ‘3’ are winning positions, because when the player reaches one of these positions, they can remove that number of pieces and win the game. These initial positions are shown in Table 2.

Table 2 - Mapping the One-Pile version.

0	1	2	3	4	5	6	7	8
P	G	G	G					

Source: prepared by authors.

Returning to the definition of losing position (P) and winning position (G), we have the possible moves from position ‘4’: removing one piece, leaving the opponent in position ‘3’; removing two pieces, leaving the opponent in position ‘2’; or removing three pieces, leaving the opponent in position ‘1’.

Therefore, position ‘4’ is a losing position (P), since any move will leave the opponent in a winning position (G).

Continuing the mapping, we find that position ‘5’ is a winning position (G), because there is at least one move that leaves the opponent in a losing position - removing one piece and leaving them in position ‘4’. See the full mapping in Table 3.

Table 3 - Mapping of the One-Pile Version; numerical trio (S = {1, 2, 3}).

0	1	2	3	4	5	6	7	8
P	G	G	G	**P**	G	G	G	**P**
9	10	11	12	13	14	15	16	
G	G	G	**P**	G	G	G	**P**	

Source: prepared by the authors.

From this analysis, we can recognize a specific pattern: the losing positions (P) form the set of multiples of 4. Thus, we have P = {0, 4, 8, 12, 16, ..., 4n, ...}. Other studies, such as Costa *et al.* (2021), discuss different analyses and variations for other numerical trios. Based on these assumptions from Nim, we will now investigate Race to 20.

MATHEMATICAL ELEMENTS IN RACE TO 20.

As previously mentioned, Race to 20 is a variation of Nim. Therefore, we can analyze it through move mapping, attempt to identify patterns, formulate conjectures, and even prove them using Mathematical Induction, which is a method of mathematical proof used to demonstrate the truth of an infinite number of propositions. It is also worth noting that in the next section, we will present the pillars of Computational Thinking to synthesize what has been developed in these initial sections.

In Race to 20, a common move-mapping strategy consists of analyzing the game situation in advance. For instance, if it is your turn and your opponent has just said 19, the winning move is to add 1, reaching 20 and ending the game. Similarly, if you receive the number 18, victory is still possible, but only if you add 2 and reach 20. Adding only 1 in this situation would result in 19, which would allow the opponent to win on the next turn.

This example helps us revisit previously discussed concepts, such as the winning strategy, where one of the players knows the path to victory. And what about you - would you make the last move, leaving 19 for your opponent? Let’s analyze a few more positions.

Would you consider position 17 a winning or losing one? Note that we have two possibilities: $17 \Longrightarrow \{17+2=19, 17+1=18\}$. Therefore, if you find yourself at position 17, both possible moves allow your opponent to win the match - unless they make a mistake. However, we are assuming the players know the winning strategy.

Keep in mind that you must include position zero in the mapping, as it is the starting point of the game. Position 20 is a losing position, since if it’s my turn and I receive 20, the game has already ended on the previous move and I have lost.

As mentioned, it is easier to carry out the mapping from the end of the game to the beginning. You can use the chart presented in the previous section to complete your mapping and return here to review it.

- [20] is a losing position (P); no valid move exists.
- [19] is a winning position (G); there is only one move, and it places the opponent in a losing position.
- [18] is a winning position (G); there are two possible moves - one bad, 18 + 1 → 19 (G), which favors the opponent; and one good, 18 + 2 → 20 (P), which places the opponent in a losing position. If there is at least one good move, the position is winning.
- [17] is a losing position (P); both moves are bad: 17 + 1 → 18 (G) and 17 + 2 → 19 (G). There is no good move.
- [16] is a winning position (G); two moves are available - one bad, 16 + 2 → 18 (G), and one good, 16 + 1 → 17 (P), which places the opponent in a losing position. If there is at least one good move, the position is winning.
- [15] is a winning position (G); two moves are available - one bad, 15 + 1 → 16 (G), and one good, 15 + 2 → 17 (P). If there is at least one good move, the position is winning.

Look at how the mapping is so far here in table 4:

Table 4 - Mapping Race to 20.

20	19	18	17	16	15	14	13	12
P	G	G	**P**	G	G			

Source: prepared by the authors.

- [14] is a losing position (P); both available moves are bad: 14 + 1 → 15 (G) and 14 + 2 → 16 (G). There is no good move.
- [13] is a winning position (G); there are two moves - one bad, 13 + 2 → 15 (G), and one good, 13 + 1 → 14 (P), placing the opponent in a losing position. If there is at least one good move, the position is winning.
- [12] is a winning position (G); two possible moves - one bad, 12 + 1 → 13 (G), and one good, 12 + 2 → 14 (P), placing the opponent in a losing position. If there is at least one good move, the position is winning.
- [11] is a losing position (P); both available moves are bad: 11 + 1 → 12 (G) and 11 + 2 → 13 (G). There is no good move.

In Table 5, we present the complete mapping of all moves. Can you identify a pattern?

Table 5 - Mapping Race to 20.

20	19	18	17	16	15	14	13	12	11	10
P	G	G	**P**	G	G	**P**	G	G	**P**	G
9	8	7	6	5	4	3	2	1	0	
G	**P**	G	G	**P**	G	G	**P**	G	G	

Source: prepared by the authors.

Note that the pattern of losing positions is P = {2, 5, 8, 11, 14, 17, 20}, which follows an arithmetic progression. In other words, the winning strategy forms a numerical sequence in which each term, starting from the second, is equal to the previous term plus a constant r (in this case, 3), where r is the common difference of the arithmetic progression.

Table 6 - Mapping Race to 20.

Thus, we have an = 3n - 1 with n ∈ {1, 2, 3, 4, 5, 6, 7}; a1 = 2 and a6 = 17. This fact can be demonstrated through Mathematical Induction, which is a method used to prove an infinite number of propositions. To do this, suppose we are given a statement a(n) that depends on n ∈ N, such that: 1. 1. a(1) is true.. 2. For k ∈ N, a(k + 1) is true whenever a(k) is true. Therefore, a(n) is true for all n ∈ N. In our case, we have an = 3n - 1 with n ∈ {1, 2, 3, 4, 5, 6, 7}, which corresponds to the losing positions P = {2, 5, 8, 11, 14, 17, 20}. To demonstrate this proposition: We have that a(1) is true, since an = 3n - 1 → 3(1) - 1 = 2. We assume as our induction hypothesis the proposition: a(k) = 3k - 1 is valid, where '3k - 1' is our assumption. Now for n = k + 1, we have: 3(k + 1) - 1 = 3k + 3 - 1 = 3k + 2 = 3k - 1 + 3. Note that '3k - 1' is our hypothesis and '3' is the common difference.

Source: prepared by the authors.

Thus, the proposition (k + 1) is also true. As a conclusion, we can affirm that, by the Principle of Mathematical Induction, the proposition a(n) holds for all n.

ANOTHER MATHEMATICAL APPROACH TO RACE TO 20

Here we present another mathematical approach to Race to 20. This approach can be applied using any target number where the aim is to introduce the concept of Euclidean division.

Now consider another version of Race to 20, now as Race to 25, with move options ranging from 1 to 2. We could divide 25 by 3 (defined as 1+2) because the goal would be to group the total accordingly. Thus, we would obtain a quotient of 8 and a remainder of 1. By projecting numbers in steps of 3, we would get the winning sequence: 1, 4, 7, 10, 13, 16, 19, 22, and 25. To win, the starting player should choose 1.

We will use "Race to x" for the following generalization:

- Move (M1): Choose a number from 1 to y to start the game, where y ∈ N represents the maximum value allowed in a single move.
- Move (M2): Choose a number from 1 to y to add to the opponent's number.
- The game continues this way, alternately, and the player who says the number x first wins.

Players, by adding numbers from 1 to y until reaching x, are effectively creating a method of division through successive subtractions, which is related to the idea of equal sharing. Thus, a possible strategy to win the game would be to divide x by 1 + y, identifying the remainder of this division, which could be used as the starting move to always win.

In Race to 20, with move options from 1 to 2, we can apply this strategy as shown below:

- 20 divided by 1 + 2, in other words, 20 divided by 3.
- Result = Quotient 6 and remainder 2.
- To win, the starting player should choose 2.
- By projecting numbers in steps of 3, we would get the winning sequence: 2, 5, 8, 11, 14, 17, 20.

Therefore, this is another approach to analyze Race to 20, now using a different mathematical concept: Euclidean division.

The mathematical analyses presented here supported the development of the winning strategy, since it is necessary to go through the Computational Thinking process, including problem formulation, decomposition, pattern recognition, abstraction, algorithm design, and debugging. In the next section, we will analyze Race to 20 through the lens of the pillars of Computational Thinking.

ANALYSIS OF RACE TO 20 THROUGH THE LENS OF COMPUTATIONAL THINKING PILLARS

After the mathematical exploration aimed at developing the winning strategy in Race to 20, we can now conduct an analysis based on the Computational Thinking processes outlined earlier in this article. This analysis was carried out considering the interaction between different processes that make up the pillars of Computational Thinking. Although we do not intend to examine the development of the game's digital version, we do focus on problem formulation, decomposition, pattern recognition, abstraction, algorithm design, and debugging. The analysis below presents how each of these stages of Computational Thinking is related to the strategy discussed.

Problem formulation - The central challenge of Race to 20 is to develop a winning strategy in a turn-based addition game between two players. The objective is to be the first to reach the number 20, with each player allowed to add either 1 or 2 to the current number. The initial task is to decide which number to add in order to prevent the opponent from reaching 20 first. Therefore, the problem lies in formulating the correct strategy and determining the actions that guarantee victory.

Decomposition - The solution to the problem can be structured into several stages, each representing a subproblem within the overall context of the game. First, it is necessary to understand the basic structure of the game:

- The starting number is 0, and the player who begins must choose to add either 1 or 2;
- The game proceeds in turns until a player reaches 20. This scenario also involves analyzing different game positions, considering the upcoming move and the strategy to be implemented, allowing players to focus on specific stages such as the correct choice of adding 1 or 2 to ensure victory.

Abstraction - In Race to 20, abstraction refers to simplifying the problem by ignoring irrelevant details and focusing on the essence of the game - that is, the moves that lead to victory. By abstracting the game into a mathematical model, we can reduce the interactions to "positions" and "moves," eliminating repetitive or trivial aspects. Abstraction allows the focus to be placed on positions that are either "losing" or "winning," removing information not relevant to the formulation of a winning strategy.

Pattern Recognition - Pattern recognition is a key process for understanding and anticipating moves in the game. When analyzing the game, note that the losing positions (P) follow a mathematical pattern, as discussed in the previous section. Recognizing this pattern allows the player to adopt a strategy based on a logical and predictable sequence of moves, leading to victory by forcing the opponent into one of these losing positions. Furthermore, pattern recognition can be useful in constructing algorithms and strategies for the game.

Algorithm Design - In the context of the game, algorithm design involves creating a set of logical and structured steps that guide the player in making winning moves. In the case of Race to 20, the algorithm consists of identifying the "winning" and "losing" positions at each stage of the game and, based on this information, determining whether to add 1 or 2 in order to move to a winning position -

always forcing the opponent into a losing one. Another point worth noting is that the algorithm can be implemented digitally, as in the Scratch version of the game, where the code checks the current game position and makes decisions to reach victory.

Debugging - Debugging is a continuous process of verification, in which the moves and decisions made are checked against the game's objective. If a strategy does not lead to victory, it is necessary to review the decisions and adjust the approach. In Race to 20, debugging involves testing different sequences of moves and verifying whether the choices follow the logic of winning and losing positions, reviewing the algorithm's performance, and ensuring the game is configured to achieve a consistent and logical outcome.

Thus, by analyzing the pillars of Computational Thinking in the development of a winning strategy for Race to 20, it is possible not only to understand the strategies involved but also to gain a deeper understanding of the mathematical dynamics that underlie the game, promoting a more efficient approach to problem-solving and the development of solutions in educational contexts. Other Computational Thinking-based approaches identified as winning strategies can be shared through a forum available on the Atelier Digitas research group website, named < ATELIER DIGIT^2@S>.

We also invite the reader to explore the winning strategy in the game "Les secrets du six" and to share their solution based on the pillars of Computational Thinking in the forum. This game was presented in the book *Les maths c'est Magique!* and follows these rules: (a) ask a friend to say a number from 1 to 5; (b) you choose a number from 1 to 5 and add it to your friend's number; (c) the winner is the first to reach 50 (Ball, 2006).

CONCLUSIONS

This article aimed to analyze the mathematics behind the winning strategy of the game "Race to 20", through the lens of the pillars of Computational Thinking. Throughout this study, it was possible to observe how each of the pillars - problem formulation, decomposition, abstraction, pattern recognition, algorithm design, and debugging - contributes to a solid and critical mathematical analysis.

We also emphasize that through the solution presented for the mathematical problem - finding the winning strategy - we demonstrated that Race to 20 can be used both in Basic and Higher Education, since it mobilizes various concepts and knowledge through gameplay. For instance, in Basic Education, concepts such as arithmetic progression and multiples of a number can be explored, while in Higher Education, the game can be used to introduce topics such as Mathematical Induction, in addition to pattern recognition, conjecture formulation, mapping of possibilities, among others.

For future studies, we intend to analyze the construction of the games presented here on the Scratch platform from the perspective of Computational Thinking, and use these findings to implement a didactic sequence in the classroom based on the theoretical assumptions of the Theory of Didactical Situations (Brousseau, 1998). Moreover, you, reader/teacher, if you apply the approach presented in this article with your students, are encouraged to share your experience in the forum available on the Atelier Digitas research group website.

Finally, this study proposes a reflection on the application of the pillars of Computational Thinking in problem-solving within mathematical games and the possibility of expanding this analysis to other educational contexts. It also becomes clear that there is great importance in continuing to explore and develop games as pedagogical tools, especially those that integrate concepts of Computational Thinking and Mathematics. The use of such approaches, combined with the creation of new technological resources, has the potential to enrich teaching and learning practices. Thus, we also propose that this article serve as an interactive text, allowing readers to engage via the forum, play the game alongside the reading, and find inspiration to further this line of research.

REFERENCES

BALL, J. **Les maths c'est magique!** Paris: Gallimard Jeunesse, 2006.

BIANCHINI, B. L.; LIMA, G. L. (org.). **O pensamento matemático e os diferentes modos de pensar que o constituem**. São Paulo: Editora Livraria da Física, 2023.

BITTAR, M. Contribuições da teoria das situações didáticas e da engenharia didática para discutir o ensino de matemática. In: TELES, R.; BORBA, R.; MONTEIRO, C. (org.). **Investigações em didática da matemática**. Recife: Editora da UFPE, 2017. p. 101-132.

BOLLER, S.; KAPP, K. **Jogar para aprender**: tudo o que você precisa saber sobre o design de jogos de aprendizagens eficazes. São Paulo: DVS, 2018.

BOUTON, C. L. Nim: a game with a complete mathematical theory. **Annals of Mathematics**, [s. l.], v. 3, n. 6, p. 35-39, 1901.

BRACKMANN, C. **Desenvolvimento do pensamento computacional através de atividades desplugadas na Educação Básica**. 2017. Tese (Doutorado em Informática na Educação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**: educação é a base. Brasília: MEC, 2018.

BROUSSEAU, G. **Théorie des Situations Didactiques**. Grenoble: La Pensée Sauvage, 1998.

CANAL, A. P.; BISOGNIN, V.; ISAIA, S. M. A. Explorando o Pensamento Computacional com a recursão na formação inicial de professores de matemática: reflexões a partir de Robbie Case. **Vidya**, Santa Maria, v. 44, n. 2, p. 153-173, 2024.

CASTRO, A.; PESSOA, G.; ARAÚJO, T. Jogo do Nim não é brincadeira, haja matemática! In: SEMANA DA MATEMÁTICA DA UFRPE, 14., 2019, Recife. **Anais [...]**. Recife: UFRPE, 2019.

COSTA, F. A.; ANDRADE, M. D.; LIMA, J. V. R. Jogo do Nim como facilitador no desenvolvimento do pensamento científico. **Revista do Professor de Matemática**, São Paulo, v. 103, p. 50-54, jun. 2021.

ESPADEIRO, R. G. O Pensamento Computacional no currículo de Matemática. **Educação e Matemática**, Lisboa, n. esp., p. 5-10, 2021.

HAIMAR, W. **Étude de l'intégration de la pensée informatique dans les domaines ASHS au sein des universités**. 2021. Dissertação (Mestrado em Educação) - Université du Québec en Outaouais, Gatineau, 2021.

LIMA, J. **Prototipação de uma versão digital do Jogo do Nim com base no modelo de processo de software da Engenharia Didático-Informática**. 2023. Dissertação (Mestrado em Educação Matemática e Tecnológica) - Universidade Federal de Pernambuco, Recife, 2023.

MOREIRA, L. O. **O Jogo da Subtração**: uma ferramenta para aulas de matemática da Educação Básica. 2014. 37 f. Dissertação (Mestrado Profissional em Matemática) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2014.

NAVARRO, E. R. **O desenvolvimento do conceito de pensamento computacional na educação matemática segundo contribuições da teoria histórico-cultural**. 2021. Tese (Doutorado em Educação) - Universidade Federal de São Carlos, São Carlos, 2021.

PADILHA, L.; PRADO, S.; DANTAS, S. Pensamento computacional na resolução de um problema matemático. **Em Teia**: Revista de Educação Matemática e Tecnológica Iberoamericana, Recife, v. 15, n. 2, p. 81-97, 2024.

PADILHA, L. *et al.* Pensamento computacional: a construção do jogo Pac-Man no Scratch. **Revista Eletrônica DECT**: Debates em Educação Científica e Tecnológica, Vitória, v. 13, n. 1, p. 146-164, 2023.

PAPERT, S. **Mindstorms**: children, computers, and powerful ideas. New York: Basic Books, 1980.

PERCEVAL, J. *et al.* O Pensamento Computacional na Matemática do Ensino Médio: uma análise de livros didáticos. **Perspectivas da Educação Matemática**, Campo Grande, v. 17, n. 45, p. 1-21, 2024.

PESSOA, G. *et al.* Jogo do Nim: um primeiro olhar. In: GITIRANA, V. (org.). **Jogos com sucata na Educação Matemática**. Recife: Nemat, 2013. p. 78-92.

ROCHA, K. C.; BASSO, M. V. A. Polígonos no Scratch: um olhar para o pensamento computacional através de abstrações reflexionantes. **Vidya**, Santa Maria, v. 44, n. 1, p. 305-324, 2024.

TORRES, T. H. **Contribuição dos Jogos na Compreensão de Conceitos Matemáticos**. 2017. 92 f. Dissertação (Mestrado Profissional em Matemática - PROFMAT) - Universidade de Brasília, Brasília, 2017.

WING, J. M. Computational thinking. **Communications of the ACM**, New York, v. 49, n. 3, p. 33-35, 2006.

WING, J. M. Pensamento computacional. **Educação e Matemática**, Lisboa, n. esp., 2021. (Tradução).