PRÁTICAS EDUCATIVAS, MEMÓRIAS E ORALIDADES

Rev.Pemo – Revista do PEMO

The Turtle's Path: Python and the Turtle Library in Education

ARTICLE

Francisco Davi Camilo Ribeiro[i]
Instituto Federal de Educação, Ciência e Tecnologia do Ceará, Fortaleza, CE, Brasil
Francisca Risolene Fernandes Rocha[ii]
Universidade Federal do Ceará, Fortaleza, CE, Brasil
Vitória Chérida Costa Freire[iii]
Secretaria Municipal de Educação de Fortaleza, Fortaleza, CE, Brasil

Abstract
New technologies have changed the way we live in the current century, generating impacts on the educational environment, which can be enhanced with new methodological possibilities in order to foster student learning. Thus, the objective of this article is to discuss how the application of the teaching methodology proposed by the Logo language facilitates the teaching of programming and some mathematical fundamentals, even when aimed at individuals with little prior knowledge of this language. The study, which adopts a qualitative approach, was developed through interactions in the Python environment and its Turtle library, which allowed the execution of several programming commands to illustrate how this language is relevant to the teaching of programming and some mathematical notions. It was evident that this feat enables more constructive, interactive and meaningful learning, when compared to traditional forms of teaching, since the learning subject is encouraged to construct their own knowledge based on their mistakes, through an interactive relationship within the environment, providing meaningful learning. It is concluded that the programming language is an important mechanism for diversifying formal schooling, especially when it comes to teaching programming and mathematics.
Keywords: New technologies. Programming language. Teaching. Logo.

O caminho da Tartaruga: Python e a biblioteca Turtle na educação

Resumo
As novas tecnologias modificaram as formas de viver no século atual, gerando impactos no ambiente educativo, o qual pode ser incrementado com novas possibilidades metodológicas, a fim de desenvolver a aprendizagem discente. Dessa maneira, o objetivo deste artigo é discutir como a aplicação da metodologia de ensino proposta pela linguagem Logo facilita o ensino da programação e de alguns fundamentos matemáticos, mesmo quando voltado para pessoas com pouco conhecimento prévio acerca dessa linguagem. O estudo, de abordagem qualitativa, foi desenvolvido mediante interações no ambiente Python e em sua biblioteca Turtle, que permitiram a execução de alguns comandos de programação para exemplificar como a referida linguagem é pertinente para

DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

o ensino de programação e de algumas noções matemáticas. Evidenciou-se que tal abordagem possibilita uma aprendizagem mais construtiva, interativa e significativa, se comparada às formas tradicionais de ensino, uma vez que o sujeito aprendiz é instigado a construir o seu próprio aprendizado com base nos seus erros, por meio de uma relação interativa dentro do ambiente, proporcionando um aprendizado com significado. Conclui-se que a linguagem de programação é um mecanismo importante para diversificar a escolarização formal, principalmente quando se trata do ensino de programação e de matemática.
Palavras-chave: Novas tecnologias. Linguagem de Programação. Ensino. Logo.

1 Introduction

Each historical period, considering its particularities, requires a specific type of educational training, so that, over time, we can observe a direct relationship between education and society (Durkheim, 1978). Thus, given that contemporary times are marked by technological advances in various fields, the educational process is not exempt from this climate of transformation and is directly affected by it – whether through the proposal and implementation of new methods of teaching and learning or through the (positive and negative) developments that new technologies, even unconsciously, have generated within classroom contexts (Castells, 2002).

From this perspective, according to Moran (2009), formal education in the 21st century – that is, the one developed in spaces specifically designed for that purpose – is a process markedly different from all previous models of education, given the particularities of the context inaugurated by the emergence of Information and Communication Technologies (ICTs). Since it is impossible to disregard this scenario, as it is present in everyone's life, both teachers and students alike, it is worth considering ways to make good use of new technologies so as to treat them as allies in the learning process (Castells, 1999). Based on this principle, a variety of virtual environments and interactive programming languages have been highlighted for their ability to support learning (Calvo, 2016), such as *Logo*, which can be used to teach programming and certain mathematical concepts to diverse audiences.

DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

Logo is a programming language developed by Seymour Papert, Cynthia Solomon, and Wally Feurzeig in the late 1960s at the Massachusetts Institute of Technology (MIT), designed for teaching children, adolescents, and even adults. Its main goal is to encourage users to learn by exploring, investigating, creating, and discovering on their own – not only computational but also mathematical concepts (Coluci, 2022).

In general terms, *Logo* can be understood as an environment in which a robot responds to the user's commands, as it employs an interactive language that develops progressively through the directions received, in a process of construction (Pinheiro *et al.*, 2024). Such a language fosters the teaching of programming and mathematical concepts, since users build their knowledge through an interactive relationship with the robot – represented by the Turtle – which, in turn, receives the knowledge transmitted by the person interacting with it. Therefore, there is a direct relationship between the *Logo* environment and formal education, particularly concerning the teaching of programming and mathematical concepts – two areas often regarded as difficult by many. This points to the importance of employing mechanisms that make learning more accessible (Calvo, 2016).

Logo has its own language and programming environments designed specifically for it. However, due to the decline in its use and the incompatibility of some of these environments, we will employ the Python language and its native *Turtle* library, which was implemented as a geometric drawing tool based on *Logo*, with all its functionalities adapted for use with Python.

The aim of addressing this topic is not to invalidate or replace other teaching methodologies, but rather to demonstrate that knowledge can be constructed and developed in ways different from those traditionally employed in formal educational settings (Saviani, 2007). From this perspective, with the technological resource now emphasized, it becomes possible to develop teaching and learning in a more interactive way – engaging students so that they are not confined to traditional teaching methods and can instead learn through what is perhaps the most common method in programming studies: learning from

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

their own experiences. This approach introduces programming education in a lighter and more engaging manner (Parpet, 1980).

Accordingly, this study arises from the following question: how does the *Logo* language contribute to the learning of programming and certain mathematical concepts for young people, children, and adults, making this process more enjoyable? Therefore, the objective is to discuss how applying the teaching methodology proposed by the *Logo* language facilitates the teaching of programming and, consequently, some mathematical foundations, even when aimed at individuals with little prior knowledge of this language. To this end, the Python language and its *Turtle* library are used to produce sample programming codes that illustrate geometric figures, demonstrating the results of code execution and the Turtle's trajectory.

The significance of this work lies in presenting a distinctive possibility for facilitating the teaching of programming and mathematical concepts, whether for children, youth, or adults. Thus, it engages in dialogue between the fields of programming and education, the latter being the one most emphasized in this study, as the *Logo* environment was conceived with the aim of improving the teaching–learning relationship in educational contexts. Therefore, this study underscores the potential of new digital information and communication technologies to enhance learning within formal educational settings (Moran, 2009).

For better reader comprehension, the manuscript is organized as follows: first, this introduction presents the topic under discussion, outlining the nature of the *Logo* environment, as well as clarifying the study's objective, guiding question, and relevance. Next, the methodological approach used to achieve the proposed objective is discussed, followed by the presentation of *Logo* as a language that enhances programming and mathematics learning, addressing its relevance to teaching environments through the presentation of the Turtle's path. Finally, the concluding section revisits the objective and summarizes the main reflections that emerged from the discussions.

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

2 Methodology

This study adopts a qualitative approach (Minayo, 1994), as it is carried out through an interpretive lens, under the perspective of subjectivity linked to the universe of variable operationalizations, without focusing on data quantification or the pursuit of an absolute and unquestionable truth. As Minayo (1994) explains, the qualitative approach sheds light on the understanding of social phenomena through a detailed analysis of study objects that would not be comprehensible from a positivist standpoint, and such objects may be investigated from various perspectives. Therefore, it is important to emphasize the influence of this research approach on the present manuscript, whose core lies in the discussion of educational issues at the intersection with new technologies, specifically regarding the *Logo* language for the teaching of programming and mathematics.

To achieve this goal, since the *Logo* language is quite old and practically obsolete, the integrated development environment (IDE) of the Python programming language and the *Turtle* library were used. Through this library, examples were simulated to demonstrate how *Logo* contributes to the learning of programming and mathematics. Specifically, the *Turtle* library was used, which is a tool that enables the creation of simple drawings and animations by moving a virtual turtle.

Among many possibilities, illustrations of plane geometric figures (a line, triangle, rectangle, and circle) were produced using programming codes. The code examples and the figures related to them were documented as images and are presented within the body of this paper to better support the discussions and achieve the proposed objective.

Finally, regarding ethical aspects, it is important to clarify that the study complies with Resolution No. 510 of April 7, 2016, a mechanism established by the *Conselho Nacional de Saúde* (CNS – National Health Council) to define the standards to be followed in research within the fields of Human and Social Sciences. This is because the present study does not involve human participants, which exempts it from the requirement of submission to a research ethics committee.

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

3 The *Logo* Philosophy and Its Potential for Education

The *Logo* programming language features a graphical Turtle, which acts as a robot ready to interact with the user's written commands. Since the language is interpreted and interactive, the result is displayed immediately after each command is executed (Papert, 1980). In *Logo*, users learn from their own mistakes, experiencing and transferring this knowledge to the Turtle. If there is a flaw in their reasoning, the error becomes clearly visible on the screen through the *Logo* environment, prompting the learner to reflect on what might have gone wrong and to seek, based on the mistakes experienced, the correct solutions to the problems encountered.

Therefore, *Logo* provides an environment that positions the student as an active participant in the learning process, not in the sense of dismissing or undervaluing the role of the teacher, but in the sense that it is the user who actively seeks knowledge through interaction with the Turtle, which, in turn, stimulates the learner's thinking. From this perspective, the student does not receive ready-made knowledge, as happens in traditional education, which treats them as repositories of information deemed important by teachers, society, or the curriculum (Saviani, 2007). Instead, they learn through experience while interacting with the Turtle, becoming active constructors of their own learning.

Thus, *Logo* proposes a teaching methodology that aims, through a simple, practical language that closely resembles natural language, to facilitate communication between the user and the computer, enabling the creation of graphic models through geometric forms and logical reasoning. Yet, beyond its technical features, the primary goal of *Logo* lies in its broader pedagogical purpose: the belief that students should be active builders of their own knowledge, thereby developing their reasoning skills as they are encouraged, through interaction with the *Logo* language, to reflect on new ways of solving presented problems and to continuously apply innovative approaches.

Moreover, because its foundation rests on meaningful learning (Ausubel, 1982), the *Logo* philosophy regards error as an essential component of the learning process, offering opportunities for students to understand the causes of their mistakes and to seek

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

new solutions through investigation, exploration, and self-discovery. In other words, the ultimate goal is learning through discovery. As Papert (1980, p. 93) explains:

> When an error occurs, it becomes an object of analysis so that it can be identified and reformulated, triggering learning and development. This process establishes a cycle of description–debugging–reflection–debugging, which was implemented in computer programming. (Papert, 1980, p. 93).

This passage makes it evident how much the *Logo* language contributes to meaningful learning, since, upon identifying an error, it presents it to the student as an object of analysis, initiating the processes of description–debugging and reflection–debugging, in other words, the search for possible ways to solve the problem at hand.

At this point, we can understand the importance of the Turtle in the approach proposed by the *Logo* philosophy. The Turtle represents the image displayed to the user during interaction with the language, resembling an actual turtle, hence its name within the programming context. The Turtle assists the user by clearly displaying successes and errors, allowing control over movement speed, and enabling observation of when and where the logic used may have become inconsistent.

Because of these characteristics, the *Logo* language embodies, in several aspects, the principles of constructionist philosophy as conceived by Papert himself, a view known as *Computational Constructionism*, which studies the development and use of technology, particularly computers, in creating educational environments (Papert, 1980). This demonstrates the significant value of *Logo* as a teaching–learning tool, emphasizing its potential to enhance educational development, especially in programming and mathematics, whether in formal education settings through teachers' initiatives or through individual efforts by learners independently exploring the language (Coluci, 2022).

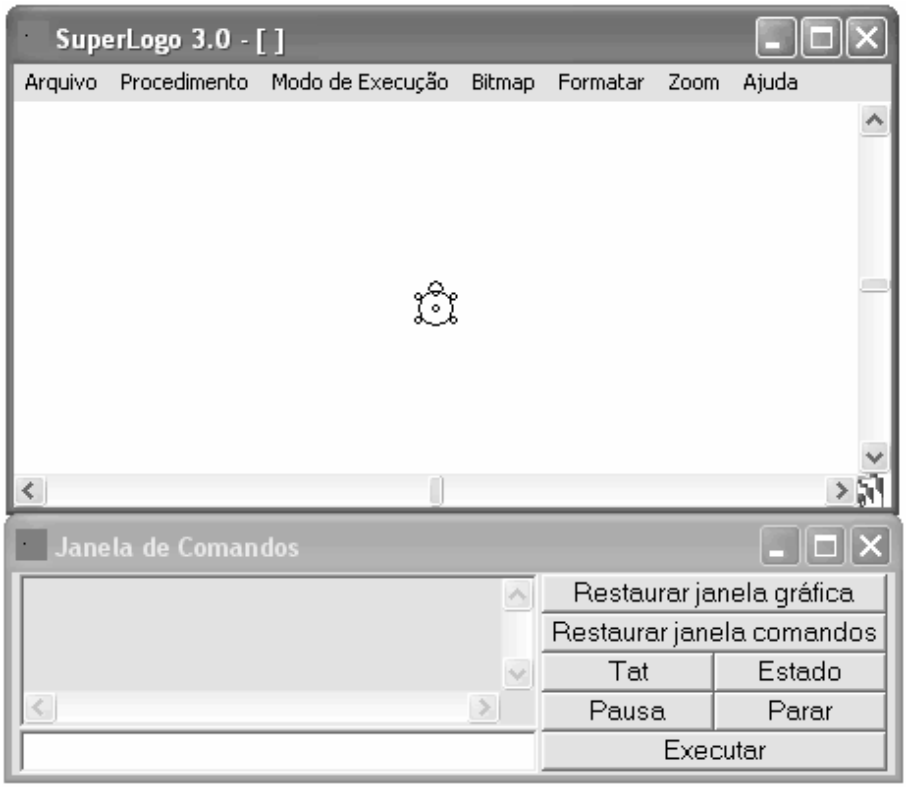
The *Logo* philosophy, therefore, is rooted in the idea that knowledge is constructed through an individual's experiences and their interaction with the surrounding environment, and that one can be encouraged to improve and learn even through mistakes. When these assumptions are aligned with educational theories, it becomes evident that the way knowledge is approached in *Logo* strongly relates to the developmental view of learning

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

proposed by Piaget, through his theory of Constructivism. For this educational psychologist, knowledge is built by the learner through experiences and interactions with their environment, rather than being deposited by others, as occurs in traditional teaching models (García, 2002). This distinction defines the educational meaning of the *Logo* approach.

The *Logo* programming language varies depending on the implementation designed by the developers of a particular IDE. Over time, *Logo* has received numerous adaptations by programming enthusiasts and scholars worldwide, including in Brazil. In this study, the *Logo* environment used is *SuperLogo 3.0*, which consists of a screen displaying a Turtle, the graphical robot that follows the user's commands as they are entered sequentially in a command line, a command window that receives native functions of the *Logo* language, and several buttons used to send primary commands, as illustrated below:

Image 1 – *SuperLogo 3.0* Environment



Source: Author's own work (2025).

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

As shown in the image, the home screen of the *SuperLogo 3.0* environment is a blank field with the image of a Turtle at the center, and in the upper and lower right corners there are several commands available to the user, such as “Execution Mode,” “Zoom,” and “Restore Graphic Window.” Therefore, it is an environment that is easy to interact with and accessible to a wide range of users, from children to adults.

Although *SuperLogo 3.0* initially works with only one command at a time, it is possible to write customized, more complex functions and load them using a specific command, just as we do in other programming languages. Everything depends on the user’s choice. Thus, the *Logo* language can execute more complex commands, allowing users to create and instruct the Turtle to draw more elaborate shapes, as well as to use repetition and recursive functions, which are comparatively less complex. Consequently, the complexity level of commands depends on the user’s decisions while interacting with the environment and attempting to solve the problems that may arise.

Among the functions available in the “Command Window,” which, as shown in the previous image, is located at the bottom of the *SuperLogo 3.0* environment, there is a function that executes commands typed in the command prompt, a button that clears the entire history of previously entered commands, and a reset function that erases all of the Turtle’s progress, returning it to the starting point and clearing the screen of any previously drawn figures.

Regarding the *Logo* language itself, it is important to note that it is an interpreted programming language, meaning that its actions are executed by an interpreter that checks each line of code and analyzes it, carrying out instructions in real time – a crucial factor for user interaction (Papert, 1980). Moreover, the language features dynamic, strong, and implicit typing, and it is classified under the following programming paradigms: functional, procedural, and reflective.

Since the *Logo* language is quite old (created in the 1960s), it was necessary to execute the commands used as examples here through Python and the *Turtle* library. Python is a programming language developed by Guido van Rossum in the late 1980s,

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

with its first version released in 1991. From the beginning, Python's main goal was to prioritize code readability over computational effort, resulting in a language that is more user-friendly and closer to human language. After all, van Rossum's intention was to create a language as easy to understand as English itself (Pinheiro *et al.*, 2025).

Over the years, Python has gained increasing popularity due to its fast learning curve, even for beginner programmers, as well as its versatility across a wide range of projects. Today, Python is used for data analysis, *machine learning*, web programming, among others. Therefore, it is a highly accessible language with diverse applications, suitable for users of different ages.

Due to the great curiosity and interest that the *Logo* language has sparked, especially among those who use it as an introduction to programming, several tools have been developed inspired by the *Logo* environment. Among these, one of the most complete and faithful adaptations was implemented as a standard component in Python: the *Turtle* library. This library almost entirely reproduces the behavior and features of a *Logo* environment, including the graphical Turtle, along with a graphical window where users can observe the Turtle's movements and the drawings it creates, with the possibility of adjusting the speed to follow its step-by-step actions.

Furthermore, the Python environment becomes even more similar to a traditional *Logo* setup when used through Python's standard development environment, known as the *Integrated Development and Learning Environment* (IDLE). Therefore, the choice to use this programming language, along with its *Turtle* library, is justified as a means of exemplifying how the *Logo* language contributes to learning in mathematics and programming.

The *Turtle* library's purpose is to draw on a blank canvas using a graphical robot called the Turtle, which functions like a pencil tip, drawing according to the commands it receives. Based on logically structured commands, it is easy to perceive the range of possibilities this tool offers. Focusing specifically on geometry, various formulas can be executed as Python code so that the Turtle generates well-known plane geometric figures, serving as visual proof of the correctness of the applied concepts.

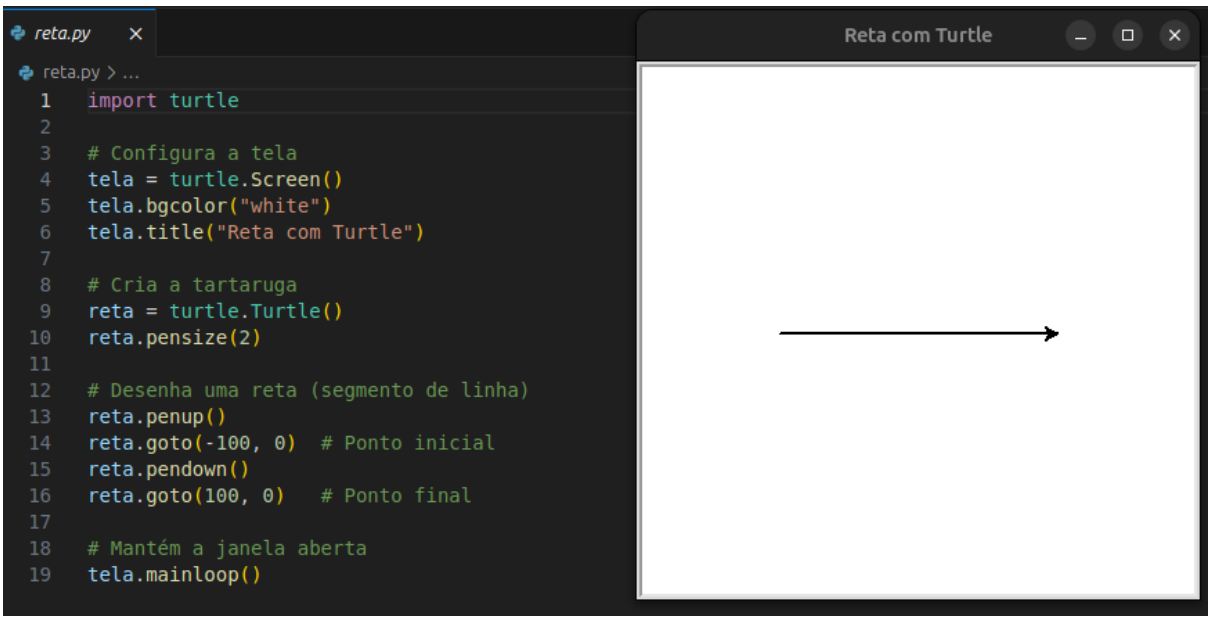
Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

Below are the programming codes used in the environment described earlier, designed to instruct the Turtle to draw certain plane geometric figures, namely: a line, a triangle, a rectangle, and a circle. Before that, however, it is necessary to understand what a point represents within this language, as it serves as the foundation for drawing geometric shapes.

A point is a geometric entity that cannot be defined. What we know about it are its characteristics, as it represents the most primitive geometric notion that exists. Here, we will address the point as a pixel on the screen, since the *Turtle* library considers it the basic unit of measurement and the smallest unit of space it can comprehend. Every time we move the Turtle across the screen, we are using a certain number of pixels to define the distance it should travel.

Thus, a line consists of a sequence of consecutive points drawn by the Turtle, running from point A to point B with no gaps in between. Below is the code that instructs the Turtle to draw a line, followed by the resulting drawing:

Image 2 – Code that instructs the Turtle to draw a line



Source: Author's own work (2025).

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

The codes illustrated on the left side of the page instruct the setup of the *Turtle* library's screen, the creation of the Turtle, and the drawing of the line, establishing the starting and ending points.

A triangle is a plane geometric figure composed of three sides, three vertices, and three angles. For the Turtle to draw this figure, it must be instructed with the following code:

Image 3 – Code that instructs the Turtle to draw a triangle

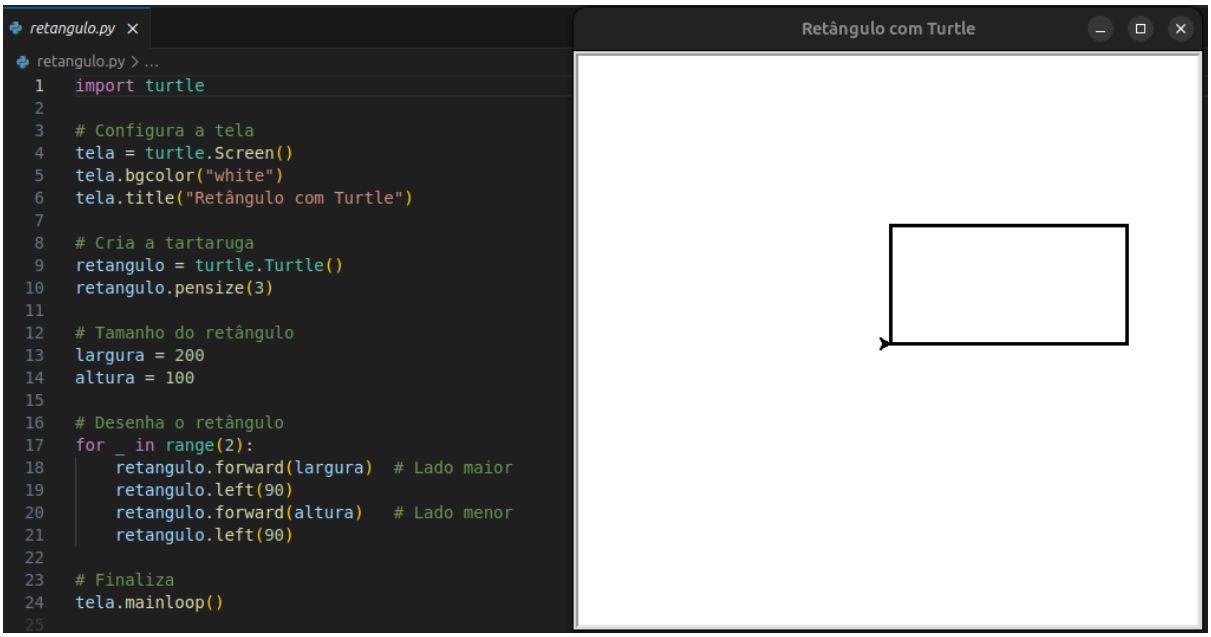
Source: Author's own work (2025).

For the triangle to be illustrated, the Turtle must receive commands to move forward 100 units and then turn 120 degrees to the left. In addition to these instructions, the *Turtle* library includes settings to configure the screen, create the Turtle, and end the execution.

Besides the line and the triangle, following the same logic of drawings made with straight lines, the Turtle can also illustrate a rectangle. This is a plane geometric figure with four sides (a type of quadrilateral), in which all internal angles measure 90° (right angles) and opposite sides are parallel and equal. The code that instructs the Turtle to draw a rectangle is as follows:

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

Image 4 – Code that instructs the Turtle to draw a rectangle



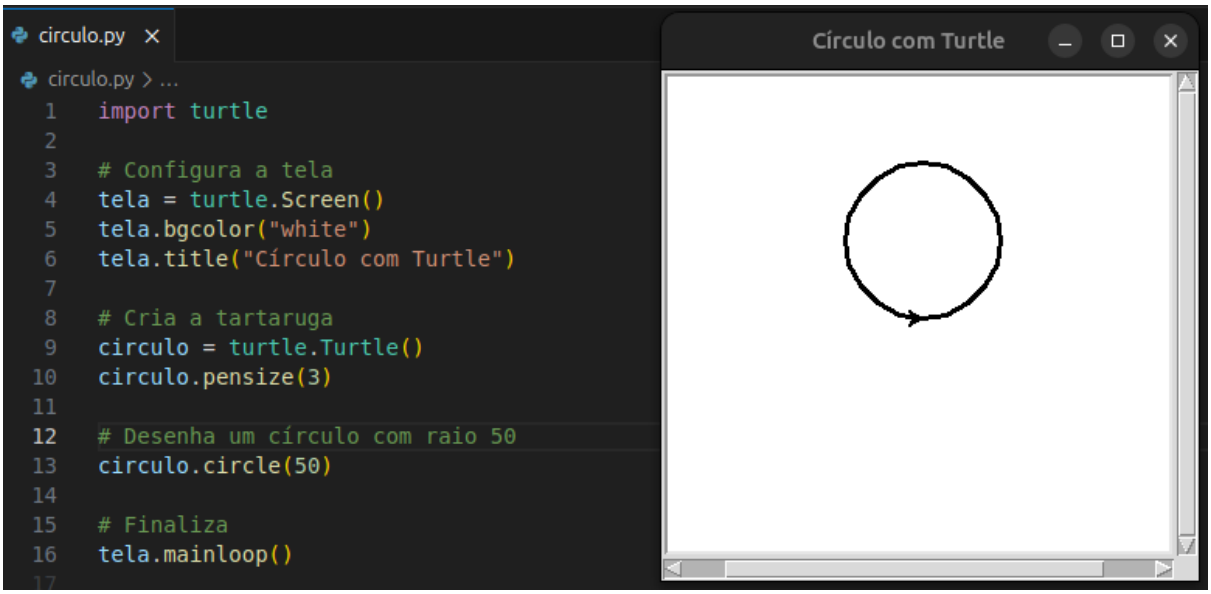
Source: Author's own work (2025).

As shown above, to draw a rectangle, it is necessary to define its width and height, which in this example are 200 and 100, and then execute the drawing of the figure using distinct codes for the longer and shorter sides. In addition, the *Turtle* library provides options to configure the screen, create the Turtle, and finalize the drawing, as in the previous examples.

Unlike the previous illustrations, which consist of straight lines, the Turtle can also outline a circle, which is a plane geometric figure formed by all points equidistant (at a constant radius) from a fixed central point. The code required to sketch a circle is as follows:

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

Image 5 – Code that instructs the Turtle to draw a circle



Source: Author's own work (2025).

In this last example, we can see that, in addition to the fixed fields (configure screen, create the Turtle, and finalize), specific codes are used to instruct the Turtle to draw a circle with a radius of 50 units.

Based on the examples presented earlier, it becomes clear that the *Logo* language – used here through the Python programming language and its *Turtle* library – has great potential to enhance the teaching of programming and, at the same time, mathematical concepts. In the cases discussed, the user had to employ different codes to draw plane geometric figures, engaging in an interactive and constructive process within the Python environment. This makes learning more dynamic and meaningful for the learner, as they take on the role of constructing their own knowledge. If the user entered an incorrect code, they would need to find ways to solve the problem, being encouraged to think, rather than receiving ready-made answers from the learning environment.

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

4 Final considerations

The objective of this study was to discuss how the teaching methodology proposed by the *Logo* language facilitates the teaching of programming and certain mathematical fundamentals – in this case, plane geometric figures – even for individuals with little prior knowledge of this language. To achieve this goal, a qualitative study was conducted, developed through the Python programming language and its *Turtle* library, considering that *Logo* is an older language that has largely fallen into disuse. Within the mentioned environment, examples of codes were illustrated that generated images of geometric shapes, combining programming with mathematics.

The guiding question that inspired this work was: “How does the *Logo* language contribute to the teaching of programming and mathematics for young people, children, and adults, making this process more enjoyable?” At the end of the examples resulting from interactions within the Python environment using the *Turtle* library, it became evident that new technologies – in this case, a programming language – are valuable tools for promoting meaningful and interactive learning.

By executing programming codes to draw a line, a triangle, a rectangle, and a circle, it was possible to perceive that the user is immersed in an interdisciplinary environment, one that combines knowledge from more than one field – in this particular case, programming and mathematics. It thus became evident that the path traced by the Turtle (drawing an analogy with the *Logo* language) is a trajectory that contributes to education and can even be used in the classroom, whether in programming or mathematics lessons, to strengthen formal schooling.

Furthermore, since the language enables constructive learning based on individual errors, its potential for fostering learner autonomy is clear. The learner becomes responsible for seeking solutions to possible problems arising from errors in their code. Therefore, the use of the *Logo* language is significant for reinterpreting the educational environment, moving away from traditional techniques centered on the student as a passive recipient of information and transforming them into an active thinker who reasons independently.

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

It can be concluded that the examples and discussions presented here on programming and education reaffirm the importance of new information and communication technologies for the contemporary educational context. Thus, it is essential that programming languages, such as *Logo*, be used as instruments that strengthen learning in a constructive and meaningful way for learners of all ages. Given the potential of programming languages for education, it is important that this discussion be further explored in future studies, using different codes as the basis for drawing other figures.

References

AUSUBEL, David Paul. **A aprendizagem significativa: a teoria de David Ausubel.** São Paulo: Moraes, 1982.

BRASIL. **Ministério da Saúde. Conselho Nacional de Saúde.** Resolução nº 510, de 7 de abril de 2016. Dispõe sobre as normas aplicáveis a pesquisas em Ciências Humanas e Sociais. *Diário Oficial da União: seção 1*, Brasília, DF, n. 98, p. 44–46, 24 maio 2016.

CALVO, Alfredo Hernando. **Viagem à escola do século XXI: assim trabalham os colégios mais inovadores do mundo.** São Paulo: Fundação Telefônica Vivo, 2016.

CASTELLS, Manuel. **A era da informação: economia, sociedade e cultura.** Lisboa: Fundação Calouste Gulbenkian, 2002.

CASTELLS, Manuel. **A sociedade em rede.** São Paulo: Paz & Terra, 1999.

COLUCI, Vitor. **Animações de conceitos da teoria de erros usando Manim/Python.** *Revista Brasileira de Ensino de Física*, v. 44, p. 1–8, 2022. Disponível em: https://www.scielo.br/j/rbef/a/YlfHKMG9B4HWKZfPtDNgPsn/?lang=pt. Acesso em: 3 jun. 2025.

DURKHEIM, Émile. **Educação e sociologia.** Petrópolis: Vozes, 1978.

GARCÍA, Rolando. **O conhecimento em construção: das formulações de Jean Piaget à teoria de sistemas complexos.** Porto Alegre: Artmed, 2002.

MINAYO, Maria Cecília de Souza. **Pesquisa social: teoria, método e criatividade.** Petrópolis: Vozes, 1994.

MORAN, José Manuel. **Novas tecnologias e mediação pedagógica.** Campinas: Papirus, 2009.

PAPERT, Seymour. **Logo: computadores e educação.** São Paulo: Brasiliense, 1980.

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

PINHEIRO, Éverton Leal; BARATA, Karla Miranda; NASCIMENTO NETO, Odemar Julião do; MARTINS, Ramon dos Santos; SUQUEIRA, Marcelo. **Aplicações computacionais em física: representação espectral da função de Green com Python.** *Revista Brasileira de Ensino de Física*, v. 46, p. 1–12, 2024. Disponível em: https://www.scielo.br/j/rbef/a/hstfJYCgS6hGrybTWc5GyKy/?lang=pt. Acesso em: 3 jun. 2025.

SAVIANI, Dermeval. **História das ideias pedagógicas no Brasil.** Campinas: Autores Associados, 2007.

[i] **Francisco Davi Camilo Ribeiro,** ORCID: https://orcid.org/0009-0000-4177-3614
Instituto Federal de Ciência e Tecnologia do Ceará, Curso de Engenharia da Computação
Graduando em Engenharia da Computação pelo Instituto Federal de Ciência e Tecnologia do Ceará (IFCE), campus Fortaleza.
Authorship contribution: execution of programming tests and drafting of the initial version of the text.
Lattes: http://lattes.cnpq.br/9150156436027825
E-mail: davicamiloribeiro@gmail.com

[ii] **Francisca Risolene Fernandes Rocha,** ORCID: https://orcid.org/0000-0002-9017-2142
Universidade Federal do Ceará, Programa de Pós-graduação em Artes, Prefeitura de Horizonte.
Mestre em Artes pelo PPG profissional em Artes da Universidade Federal do Ceará (UFC), Especialista em Alfabetização e Multiletramentos; em Gestão Pedagógica da Escola Básica; e em Língua Portuguesa e Literatura Brasileira, pela Universidade Estadual do Ceará (UECE); Graduada em Pedagogia pela Universidade Estadual Vale do Acaraú (UVA),
Authorship contribution: writing of the text and first revision.
Lattes: http://lattes.cnpq.br/1700981050573327
E-mail: profarisolenefernandes@gmail.com

[iii] **Vitória Chérida Costa Freire,** ORCID: https://orcid.org/0000-0002-8029-5907
Prefeitura Municipal de Fortaleza, Secretaria Municipal de Educação (SME)
Licenciada em Pedagogia (2015) pela Universidade Estadual do Ceará (UECE). Possui mestrado (2017) e doutorado (2022) em Educação. Estuda e desempenha pesquisas científicas na área educacional, principalmente sobre História da Educação no Brasil e no Ceará, Educação de Mulheres, Biografia, Escola Pública e Formação de Professores.
Authorship contribution: writing of the text and final revision.
Lattes: http://lattes.cnpq.br/3973477219174231
E-mail: vitoriacherida91@gmail.com

Rev. Pemo, Fortaleza, v. 7, 15701, 2025
DOI: https://doi.org/10.47149/pemo.v7.e15701
https://revistas.uece.br/index.php/revpemo
ISSN: 2675-519X

Responsible publisher: Genifer Andrade

***Ad hoc* experts:** Victor Hugo de Oliveira Henrique e Antônio Roberto Xavier.

How to cite this article (ABNT):
RIBEIRO, Francisco Davi Camilo.; ROCHA, Francisca Risolene Fernandes.; FREIRE, Vitória Chérida Costa. O caminho da Tartaruga: Python e a biblioteca Turtle na educação. **Rev. Pemo**, Fortaleza, v. 7, e15701, 2025. Disponível em: https://revistas.uece.br/index.php/revpemo/article/view/15701

Received on June 11, 2025.
Accepted on July 3, 2025.
Published in November 10, 2025.