Zetetiké

DOI: 10.20396/zet.v31i00.8672180

Potentials of Python Programming for the Development of Creative Thinking in Mathematics

Potencialidades da Programação em Python para o Desenvolvimento do Pensamento Criativo em Matemática

Rosane Rossato Binotto[1]

Marcus Vinicius Maltempi[2]

Rogério Aparecido Batista da Silva[3]

Abstract

This article presents the results of a study aimed at identifying and analyzing indications of mathematical creativity in programming activities developed in Python, throughout a professional development course involving undergraduate students majoring in Mathematics, High school students, and teachers from K-12 Education. The participants responded to an open-ended question and implemented their solutions in Python, with the code analyzed in this article. The research followed a qualitative approach, utilizing Content Analysis to categorize the obtained data. Through the analysis, we found that Python programming for open-ended problem-solving holds potentials for fostering creative thinking in mathematics, facilitating simulation, debugging, reflection on the process, motivation, and engagement, leading to creative solutions.

Keywords: Mathematical Creativity; Teacher Education; Computational Thinking; K-12 Education.

Resumo

Este artigo apresenta os resultados de uma pesquisa cujo objetivo foi identificar e analisar indícios de criatividade matemática em atividades de programação desenvolvidas em Python, ao longo de um curso de formação continuada que contou com a participação de estudantes de Licenciatura em Matemática, do Ensino Médio e de professores que atuam na Educação Básica. Os cursistas responderam a uma questão aberta e implementaram suas soluções em Python, cujos códigos são analisados neste artigo. A pesquisa seguiu uma abordagem qualitativa e utilizou a Análise de Conteúdo para categorizar os dados obtidos. Por meio da análise, constatamos que a programação em Python para a solução de problemas abertos possui potencialidades para desenvolver o pensamento criativo em matemática facilitando a simulação, depuração, reflexão sobre o processo, motivação e engajamento, resultando em soluções criativas.

Submetido em: 21/12/2022 – **Aceito em**: 01/08/2023 – **Publicado em**: 19/12/2023

[1] PhD in Mathematics from the State University of Campinas (Unicamp). Professor at the Federal University of the Southern Border (UFFS), Brazil. Email: rosane.binotto@uffs.edu.br. ORCID: https://orcid.org/0000-0001-9420-9312.

[2] PhD in Electrical and Computer Engineering from the State University of Campinas (Unicamp). Professor at the São Paulo State University (Unesp), Brazil. Email: marcus.maltempi@unesp.br. ORCID: https://orcid.org/0000-0001-5201-0348.

[3] Master's student in Mathematics Education at the São Paulo State University (Unesp), Brazil. Email: rogerio.batista@unesp.br. ORCID: https://orcid.org/0009-0004-8311-0842.

DOI: 10.20396/zet.v31i00.8672180

Palavras-chave: Criatividade em Matemática; Formação de Professores; Pensamento Computacional; Educação Básica.

Introduction

The complexities and constant changes in the global landscape demand citizens who can adapt and participate in these changes, possessing the necessary training, technical knowledge, and scientific skills to critically, creatively, and collaboratively solve problems, using digital technologies when required.

In this regard, it is crucial to encourage young individuals to think critically and solve problems from K-12 Education onwards, as proposed by the National Curricular Parameters for High School Education, focusing on essential citizenship competencies and performance. This involves skills such as,

> [...] the ability to abstract, develop systemic thinking, as opposed to a partial and fragmented understanding of phenomena, creativity, curiosity, the ability to think of multiple alternatives to solve a problem, in other words, the development of divergent thinking, the ability to work in teams, the willingness to seek and accept criticism, the willingness to take risks, the development of critical thinking, the ability to communicate, and the ability to seek knowledge (Ministério da Educação, 2000, p. 11).

We emphasize the importance of fostering creativity and divergent thinking to prepare students to handle citizenship and tackle unpredictable problems. Therefore, it is crucial for teachers to propose tasks, activities, methods, and environments that stimulate creative thinking and creativity in mathematics.

According to Alencar (1990), Gontijo (2007), and Gontijo, Carvalho, Fonseca and Farias (2019), to foster creativity in mathematics, teachers should create a classroom environment that allows students to demonstrate fluency, flexibility, originality, and elaboration in their work. These are characteristics of creative thinking that promote the development of divergent thinking.

One way to promote such an environment is using digital technologies, particularly computer programming for teaching and learning mathematics. Seymour Papert (1928-2016) proposed that the computer be used for learning, allowing students to construct their own knowledge. In this sense, education has the role of “creating the appropriate contexts for learning to develop naturally” (Papert, 1997, p. 8).

The use of computers in education gained new momentum in 2006 with Jeannette Wing’s work on *Computational Thinking*, which “[...] builds on the power and limits of computing processes, whether they are executed by a human or by a machine” (Wing, 2006, p. 33). This author also describes it as an effective problem-solving process, achievable by both humans and machines (Wing, 2011). Other definitions of Computational Thinking have emerged, with no consensus on a single definition. We highlight the understanding provided

DOI: 10.20396/zet.v31i00.8672180

by Brackmann (2017), who considers Computational Thinking a cognitive process for problem-solving, connected to four pillars: pattern recognition, decomposition, abstraction, and algorithms.

When solving problems using computers, the choice of programming languages is essential. In this work, we consider the Python language, an interpreted, dynamic, functional, open-source language with syntax resembling the English language, suitable for K-12 Education (Leonardo, 2020).

To contribute to research on activities and environments that promote creative thinking and/or creativity in mathematics within the school context, we conducted a study in which participants solved an open-ended problem and described their solutions in Python programs. In this context, we were guided by the overarching question: can Python programming activities foster creativity in mathematics? Therefore, our goal is to identify and analyze indications of mathematical creativity in the Python programming activities.

We adopted a qualitative approach, considering five activities developed by participants in an extension course. Based on an open-ended question, these learners were required to assign grades, calculate averages, and indicate the minimum passing or failing grade, as well as implement a Python program with the proposed solution.

To analyze the data, we utilized Bardin's Content Analysis (2016), establishing two post hoc analysis categories aligned with the guiding question, the proposed objective, and the theoretical framework of the research.

In this article, we discuss the theoretical framework of the study, focusing on creative thinking, creativity in mathematics, constructionism, Computational Thinking, and creative learning. We then present the methodological design of the study. In the results and discussions section, we present the five programs developed by the participants and analyze the data produced through the established categories. We conclude with final remarks.

Theoretical Framework

Constant changes have exposed young people to unfamiliar, uncertain, and unpredictable situations. Thus, it is essential for these young individuals to develop the ability to think creatively. To achieve this, it is crucial for them to engage in activities from an early age that stimulate creativity, not only in the arts but also in other areas of knowledge, including mathematics.

In this section, we initially present the fundamentals of creative thinking and creativity in mathematics and discuss possibilities for their integration into the school environment.

Creative Thinking and Creativity in Mathematics

DOI: 10.20396/zet.v31i00.8672180

One of the pioneers in studies on mathematical creation was the French mathematician Henri Poincaré (1854-1912), who studied the theory of Fuchsian groups and Fuchsian functions. For Poincaré, mathematical creation

> […] does not consist in making new combinations of known mathematical entities. Anyone could do that, but the combinations that could be obtained this way would be limited in number and, for the most part, entirely devoid of interest. Creation consists precisely in not constructing useless combinations, but in those that are useful and that are in very small minority. Creating is discerning, choosing (Gontijo et al., 2019, p. 41).

In other words, he considered the creative process in the sense of new discoveries, not merely modifications or combinations of existing mathematical findings. Based on this experience, Poincaré developed questions that were applied to other mathematicians to understand their creative processes and the involved factors. Although Poincaré did not approach creativity in mathematics in the contemporary sense, his work influenced later research, such as that conducted by another French mathematician, Jacques S. Hadamard (1865-1963).

Hadamard described four stages for creative problem-solving: preparation, incubation, insight, and verification. This involves preparation with prior knowledge, incubation to connect and organize information, illumination to generate ideas, and verification to maintain scientific rigor (Hadamard, 2009; Gontijo et al., 2019; Gontijo, Fonseca, Carvalho & Bezerra, 2021). Hadamard emphasized that creativity in mathematics requires knowledge, not just inspiration.

More recently, Aiken (1973) proposed an understanding of creativity in mathematics, considering both the production process and the final product. In the cognitive process of mathematical creation, related to production, qualities such as "the ease and freedom to switch from one mental operation to another or, alternatively, the ability to analyze a problem from different perspectives, observing specific characteristics and identifying similarities and divergences among the involved elements" (Gontijo et al., 2019, p. 44) may be involved. The final product can be original and involve new methods for solving mathematical problems. Furthermore, "creativity in its product aspect also refers to the ability to formulate numerous, different, and appropriate questions when mathematical situations are presented graphically or in the form of a sequence of actions" (Gontijo et al., 2019, p. 45).

Haylock (1987) recognizes creativity in mathematics "both in problem formulation and problem-solving, as well as differentiated problem-solving methods" (Gontijo et al., 2021, p. 13). In this assertion, we perceive the relationship between creativity in mathematics and the formulation and solution of problems, adopting different strategies for their resolution and presenting more than one solution to the problem.

Considering the various perspectives on creativity in mathematics, Gontijo (2007, p. 37) defines it as

DOI: 10.20396/zet.v31i00.8672180

> [...] the ability to present numerous appropriate solution possibilities for a problem situation, focusing on distinct aspects of the problem and/or different ways of solving it, especially uncommon ways (originality), both in situations that require problem-solving and problem elaboration and in situations that demand classification or organization of mathematical objects and/or elements based on their properties and attributes, whether textually, numerically, graphically, or in the form of a sequence of actions.

This definition emphasizes strategies to stimulate creativity in mathematics through problem-solving and elaboration and through reorganization or redefinition of mathematical data and/or elements. Moreover, as stated by Alencar (1990), Gontijo (2007), and Gontijo et al. (2019), for creative production in mathematics to occur, teachers must create a conducive environment in which students demonstrate the skills of fluency, flexibility, and originality in their work. In this context,

> a) Fluency represents the quantity of different ideas generated that constitute suitable solutions for the proposed problems;
> b) Flexibility refers to the number of different categories in which the generated solutions for each problem can be classified;
> c) Originality corresponds to the infrequency or non-conventionality of the generated ideas, that is, solutions that are suitable and diverge from the large group of proposed solutions are considered original (Gontijo et al., 2019, p. 81).

The creative process in mathematics can involve cognitive, emotional, and motivational aspects, meaning that students need to be interested, motivated to solve a problem, and feel a need to solve it. This process can promote the development of mathematical skills (Gontijo et al., 2019).

Furthermore, regarding methodological strategies to promote creativity in mathematics, Gontijo et al. (2019) address the following: open-ended problem-solving, problem elaboration, and redefinition of mathematical elements. According to these authors, open-ended problems are those that have more than one solution. When seeking these solutions, students are not bound by predetermined results and have the opportunity to obtain

> [...] a range of solutions through divergent thinking, some correct, others incorrect, some well-elaborated, others in the process of structuring, some considered valid, others not accepted, and among all these, a smaller quantity of original answers, just as occurs in the process of real-life problem-solving (Gontijo et al., 2019, p. 62).

Additionally, students rarely leave this type of problem unanswered. When solving open problems, students are responsible for decision-making, developing their autonomy, no longer relying on the teacher or ready-made rules and models presented in educational materials.

In the literature, several works mention activities that have the potential to develop mathematical creativity in their resolutions, with a highlight on Alencar (1990), Gontijo (2007), Gontijo et al. (2019), Pereira (2008), and Oliveira (2016). The first three works present various activities characterized by an open problem statement, addressing

DOI: 10.20396/zet.v31i00.8672180

mathematical subjects such as geometry, combinatorial analysis, equations, numbers, mathematical operations, among others, with a focus on K-12 Education.

The work developed by Pereira (2008, p. 8) conducted a study with the purpose of "identifying and analyzing aspects related to creativity present in activities that used Mathematical Modeling as a teaching and research methodology [...] and producing indicators about the relationship between Mathematical Modeling and Creativity." This author used some dissertations developed in Postgraduate Programs of Brazilian universities and showed, among other aspects, "that Mathematical Modeling, when addressing situations from students' reality, can awaken greater interest in Mathematics and, consequently, promote the development of creativity-related skills in Mathematics" (Pereira, 2008, p. 8).

The work developed by Oliveira (2016, p. 8) aimed to "investigate the importance of Digital Information and Communication Technologies (TDIC), in enhancing mathematical creativity and knowledge," through a didactic experience with 9th-grade students of Elementary Education, using the GeoGebra software for problem-solving in Knowledge Projects. The results indicated that the use of "TDIC mediated by Knowledge Projects promotes creativity in mathematics with consequent progress of students in problem-solving competence" (Oliveira, 2016, p. 8).

Supporting these authors, we agree that to develop creative potential, activities should prioritize allowing students to seek multiple solutions to the same problem, elaborating and reflecting on different strategies to solve it, and redefining mathematical elements if necessary. School can become a conducive environment for developing creativity in mathematics, with flexible and willing teachers to implement activities and innovations, understanding that "creative productions tend to occur in structured social contexts and not in isolation" (Martindale, 1999, p. 165).

Furthermore, Pereira (2008, p. 44), based on the studies of Martindale (1999) and Fleith (2007), among others, argues that "the use of techniques to stimulate creative potential is more productive if done in groups." When students work in groups, a greater range of ideas emerges, new possibilities for problem-solving are discussed, and communication and collaborative work are valued.

While identifying students' creative potential is a challenge, there are tests, instruments, and procedures that can be used to evaluate creativity. They are based on the "valorization of divergent thinking [...] in the analysis of production, both in the broad context of creativity and in the context of specific creativity" (Gontijo et al., 2019, p. 81). These tests seek to assess creative thinking skills, such as fluency, flexibility, and originality of thought.

In the next section, we will discuss computer programming in education, referring to the theory of Constructionism and its relationship with programming and Computational Thinking.

DOI: 10.20396/zet.v31i00.8672180

Constructionism, Creative Learning, and Computational Thinking

Digital technologies are revolutionizing people's daily lives, study, and work environment, influencing "the way of being, living, doing, and learning of most people, so that having technology in the service of knowledge transmission is no longer sufficient" (Maltempi, 2005, p. 2). This revolution is also causing changes in the school environment, with the incorporation of digital technologies into pedagogical practice, including the creation of activities, methods, and environments for teaching and learning involving these technologies.

Papert advocates for the construction of learning environments from a Piagetian perspective, using computers. He is responsible for creating the Logo programming language[4] and is considered the founder of Constructionism, a theory that suggests the use of computers to assist students in constructing their own knowledge and development of thought. According to Papert, when a student uses the computer, they visualize their constructions, relating the concrete and the abstract through an interactive process that can generate a sequence of constructions and mental abstractions, influencing learning. Moreover, according to Papert (1994, p. 158), computers can and should be used "as instruments for working and thinking, as means of carrying out projects, as a source of concepts for thinking new ideas," and not just as a form of support for automated instruction. The activity of programming stimulates "thinking with" the machines and "thinking about" one's own thinking.

We also consider what was proposed by Resnick, a student of Papert and the creator of the Scratch programming environment[5], a block-based programming environment. This author believes that technologies, on one hand, accelerate the pace of changes in society, emphasizing the need for creative thinking in all aspects of these people's lives. On the other hand, these technologies "have the potential, if properly designed and used, to help people develop as creative thinkers, so they are better prepared for life [...]" (Resnick, 2007, p. 19). This author argues that people should not only acquire knowledge but also find creative solutions to unexpected problems, as "success is based not only on what you know or how much you know, but on your ability to think and act creatively" (Resnick, 2007, p. 18), which he calls a Creative Society.

According to Resnick (2007, p. 18), when developing a project in Scratch, students "imagine what they want to do, create a project based on their ideas, play with their creations, share their ideas and creations with others, and reflect on their experiences - all of which leads them to imagine new ideas and news projects", going through a process called "the creative thinking spiral". This process comprises the following steps: imagine, create, play,

[4] From: https://www.nied.unicamp.br/biblioteca/super-logo-30/. Visited: 07 Aug. 2023.

[5] From: https://scratch.mit.edu/. Visited: 12 Jan. 2023.

DOI: 10.20396/zet.v31i00.8672180

share, reflect, and imagine again. Thus, for this author, Scratch is part of a group of new technologies designed to help prepare students for the Creative Society, enabling the incorporation of computer programming to address creative thinking in schools.

Resnick also created a movement called Creative Learning, with the aim of schools creating opportunities for children from the early years to develop the potential of creative thinkers. Creative learning is a process that encompasses four pillars: Projects, Passion, Peers and Play, as "the best way to cultivate creativity [is] by helping people work on *projects* based on their *passions*, in collaboration with *peers*, and maintaining the spirit of *thinking playing*" (Resnick, 2020, p. 15).

Another movement that promotes the incorporation of computer programming in schools as a didactic tool for learning is associated with the development of Computational Thinking, which had its initial ideas with Papert but gained prominence from 2006, being the subject of study by various researchers, such as Wing (2006, 2011), Barr and Stephenson (2011), Brennan and Resnick (2012), Valente (2016), Denning (2017), Brackmann (2017), Barbosa and Maltempi (2020), Barichello (2021), among others.

For Brackmann (2017, p. 29), Computational Thinking consists of a

> [...] human creative, critical, and strategic capacity to know how to use the foundations of Computing, in various areas of knowledge, in order to identify and solve problems, individually or collaboratively, through clear steps, so that a person or a machine can execute them effectively.

Activities developed from this perspective aim to contribute to the construction of logical thinking, the ability to recognize patterns, and the development of reasoning, through decomposition, pattern recognition, abstraction of a problem, and algorithm design. Thus, in the context of problem-solving, computational thinking can be organized into four pillars: decomposition, pattern recognition, abstraction, and algorithms (Brackmann, 2017).

From the above, we can observe an alignment between Computational Thinking and creative thinking. Computational Thinking is a type of critical thinking associated with problem-solving, using computational foundations, with or without the use of a machine, in individual or collaborative work. It uses algorithms – one of the pillars for problem-solving – as a possibility for developing computational thinking. On the other hand, creativity in mathematics is related to the "ability to propose uncommon algorithms, as well as the ability to find multiple different answers to the same problem" (Gontijo et al., 2019, p. 46).

Furthermore, problem-solving through computer programming can stimulate the development of Computational Thinking, and we hope it can also contribute to the development of creativity in mathematics.

DOI: 10.20396/zet.v31i00.8672180

Methodological Approach

This work was guided by a qualitative research approach, as it sought to "[...] address aspects of the human condition without passing through the filters of measurement, without starting from previously defined methods, and therefore without being confined to quantifiers and resulting calculations" (Bicudo, 2019, p. 113). According to Bogdan and Biklen (1997, p. 209), qualitative studies "should reveal greater concern for process and meaning, and not for its causes and effects." It is worth noting that, being a qualitative research, the goal of data production was not to obtain a statistical picture of teachers' perceptions for the purpose of generalizations, but rather to consider their contributions, without concern for the size of the sample of respondents.

Nine individuals participated in this research[6], all participants of an extension course[7], including one student from K-12 Education, two Mathematics teachers working in K-12 Education, and six undergraduate students in Mathematics at the State University of São Paulo (Unesp), Rio Claro campus. After the course participants interacted with Python and completed some programming activities in the face-to-face course meetings, they solved an open-ended question proposed and implemented their solution using Python. Five solutions were produced for the proposed question, four of them developed in pairs and one individually.

The participants developed the programs using Google Colaboratory (Colab), which reads and executes Python language codes and works online, as shown in Figure 1, facilitating program execution as it doesn't consume much computer memory. In this figure, you can also observe the Google Colab interface, presenting a simple program that, when executed, prompts the user for the salary value and returns whether it is necessary to pay taxes or is exempt from them.

---

[6] Approved by the Research Ethics Committee of Unesp, Botucatu, SP, with the following information: CAAE: 38656322.8.0000.5411, Opinion Number: 5.462.880, dated June 10, 2022.

[7] This course was part of the postdoctoral internship activities of the first author of this article, carried out at Unesp, Rio Claro/SP, from March 2022 to February 2023, under the supervision of the second author.

DOI: 10.20396/zet.v31i00.8672180

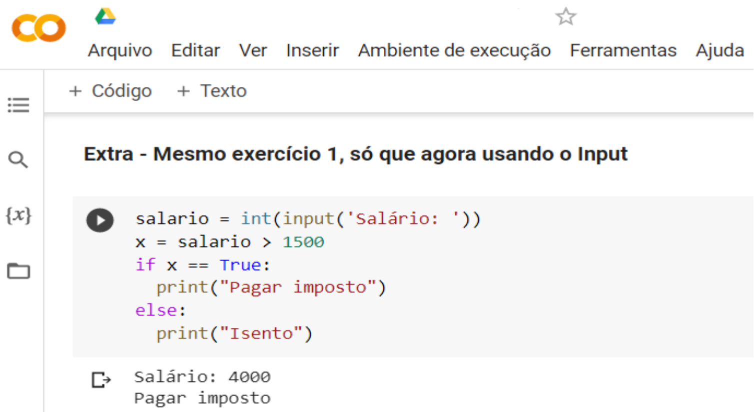

Figure 1 - Google Colab Interface and Programming Code
Source: Created by the authors of the article (2022)

Proposals for using this programming language in High School exist, such as the work of Leonardo (2020), which proposes to work on basic notions of programming logic, introduce Python, model and solve problems using a programming language. Additionally, it recommends working on the pillars of Computational Thinking and fluency in the use of digital technology with the purpose of enabling student engagement as protagonists in the use and implementation of solutions.

The data produced in this research were submitted to a Content Analysis as a methodological alternative in the study of information, which, according to Bardin (2016), includes the stages of: pre-analysis, material exploration, and treatment of results and interpretation. In this context, pre-analysis involves the study and preparation of the Python material used in the face-to-face stage of the extension course, as well as the development of the open-ended question proposed to the participants of this course.

The exploration of the material is characterized by the research conducted, in which the participants responded to the proposed question and created the programs. Finally, the stage of treatment of results and interpretation considers the discussion of the proposal through the account of the experience and its analysis, described in the next section.

The analysis of the produced data was conducted based on two *a posteriori* established textual categories, namely: (i) Perceptions about creative thinking skills; (ii) Contributions of computer programming to creativity in mathematics.

In the first category, we listed subcategories to analyze the produced data, focusing on skills of fluency, flexibility, and originality of thought, in order to assess signs of the development of creative thinking by the participants. In the second category, we anchored ourselves in the pillars of Computational Thinking and computational tools for problem-solving, as well as elements such as motivation, decision-making, critical thinking, and creative problem-solving, among others.

DOI: 10.20396/zet.v31i00.8672180

Results and Analysis

We begin by describing the five Python programs developed for the proposed open-ended question. We then proceed to analyze the data.

Obtained Results

The **open-ended question** was as follows: Write a program that reads grades (input data), calculates the average, and decides whether the student is approved or failed. In this programming, it is necessary to define the minimum grade for a student to be approved, the number of grades (2, 3, or more grades), and how the average will be calculated. You can also add other information if you deem it necessary (Created by the authors of the article, 2022).

Program 1

Figure 2 illustrates a solution for this question (Python code) in which two grades are proposed, and the average calculation is performed using weighted average. The minimum grade to pass is 5.0; otherwise, the student is in recovery. The authors did not detail the conditions for approval or failure after recovery. We infer that the grade to fail, after recovery, is below 5.0.

```
[ ]
    n1 = float(input("Primeira nota: "))
    n2 = float(input("Segunda nota: "))
    mp = (6*n1 + 4*n2)/10

    if mp >= 5:
      print("Você foi aprovado com média: ",mp)
    if mp < 5:
      print("Você ficou de recuperação, pois sua média foi: ",mp)
```

```
Primeira nota: 4.5
Segunda nota: 5
Você ficou de recuperação, pois sua média foi:  4.7
```

Figure 2 - Program 1
Source: Research data (2022)

In addition to the code, this figure also portrays a situation in which the student would be in recovery. The program includes commands for reading and assigning data to variables, allowing decimal numeric values, and an expression to calculate the average. The if decision command is used, as well as the print command to display messages requesting the grades and the result of approval or recovery, including the obtained average.

Program 2

Figure 3 presents a solution to the question, where grades for two exams are proposed, and the average is calculated using arithmetic mean. The student is approved if the average is

DOI: 10.20396/zet.v31i00.8672180

greater than 5.0 and failed if it is lower. The authors did not describe what happens if the student obtains an average equal to 5.0. Additionally, they present a situation in which the student is approved.

```
[ ]  a = int(input(" Prova 1 "))
     b = int(input(" Prova 2 "))
     c = (a+b)/2
     if c > 5:
       print("O aluno foi aprovado")

     if c < 5:
       print("O aluno foi reprovado")

 Prova 1 5
 Prova 2 6
O aluno foi aprovado
```

Figure 3 - Program 2
Source: Research data (2022)

Like the previous program, commands for reading and assigning data to variables were used, allowing only integer numeric values. Average calculations were performed, utilizing the if conditional statement and the print command to display grades and the approval or disapproval message.

Program 3

Figure 4 illustrates another solution to the proposed question.

DOI: 10.20396/zet.v31i00.8672180

```
[1] # Média
    print("Insira as notas para calculo da media")

    prova1 = int(input("Prova 1: "))
    prova2 = int(input("Prova 2: "))
    trab   = int(input("Trabalho(0 a 5): "))
    m      = int((prova1+prova2+trab)/2)

    print("Sua media foi: ")
    print(m)


    if m > 5:
      print("Congratulations")
    if m == 5:
      print("Congratulations")
    if m < 5:
      print("Exame")

    Insira as notas para calculo da media
    Prova 1: 3
    Prova 2: 0
    Trabalho(0 a 5): 5
    Sua media foi:
    4
    Exame
```

Figure 4 - Program 3
Source: Research data (2022)

In this program, in addition to assigning two test grades, each assumed to have a weight of 10.0, it indicates a grade for a project with a weight of 5.0. The average is calculated by summing all the grades and dividing the result by 2. We observe a potential error in the average calculation, as the student receives a bonus on the project grade, which could result in an average greater than 10.0. After the average calculation, there are conditions for passing and taking an exam. The conditions for the student’s passing after the exam are not presented.

The creators used commands to assign values to variables, allowing only integer numeric values. They performed the average calculation, used the if conditional statement, and the print command to display the grades, average, and the message *Congratulations* (implying approval) or Exam.

Program 4

Figure 5 shows a Python program where the user should include test grades in the code, separated by commas, thus defining the number of assessments to be used to calculate the average (arithmetic mean). The student passes if they achieve a minimum grade of 5.0 and fails with a lower grade. The authors present a situation where the student passes.

DOI: 10.20396/zet.v31i00.8672180

```
lista1 = [5,9,5,6,5] #coloquei aqui uma lista de notas, o usuário fica livre para digitar quantas notas quiser
z=len(lista1) #vai contar o número de elementos na lista anterior
y=sum(lista1) #soma os elementos da lista anterior
media = y/z #calcula a média
if media > 4.9: #Nota mínia para aprovação é 5
  print("Aprovado! Com média de " + str(media) + ".")
else:
  print("Reprovado! Com média de " + str(media) + ".")

Aprovado! Com média de 6.0.
```

Figure 5 - Program 4
Source: Research data (2022)

The creators of this program created a list with fixed values for five grades, used a command to count the number of elements in this list, and another to sum them, assigning the results to variables that were then used to calculate the arithmetic mean. They used the conditional commands if and else, in addition to print, to display the average and the approval or disapproval message.

Program 5

The program illustrated in Figure 6 was developed based on the idea that the average is initially calculated using the grades of two tests. If the student does not achieve the minimum grade of 5.0 in the average m1 for approval, an exam will be applied. If the student obtains a grade on the exam such that the arithmetic mean between *m1* and the exam *ex* is equal to or greater than 5.0, they pass; otherwise, they fail.

```
[ ] p1 = float(input("Qual foi a nota da primeira prova?  "))
    print("P1 = ", p1)
    p2 = float(input("Qual foi a nota da segunda prova?  "))
    print("P2 = ", p2)

    m1 = (p1 + p2)/2
    print("Média inicial = ", m1)

    if m1 >= 5:
      print("O aluno está aprovado com média final igual a ", m1)

    else:
      ex = float(input("Qual foi a nota do exame final?  "))
      m2 = (m1 + ex)/2
      if m2 >= 5:
        print("O aluno está aprovado com média final igual a ", m2)
      else:
        print("O aluno está reprovado com média final igual a", m2)

    Qual foi a nota da primeira prova?  9
    P1 =  9.0
    Qual foi a nota da segunda prova?  7
    P2 =  7.0
    Média inicial =  8.0
    O aluno está aprovado com média final igual a  8.0
    9
```

Figure 6: Program 5
Source: Research data (2022)

DOI: 10.20396/zet.v31i00.8672180

The program involves commands for reading and assigning data to variables, allowing fractional numerical values, expressions for calculating averages. Conditional statements if and else were used, as well as the print command to display grades, averages, and approval or disapproval messages.

Next, we analyze the produced data based on the proposed categories.

Data Analysis

In this section, we present the analysis of the data according to the two categories outlined.

(i) Perceptions about creative thinking skills

To categorize the data produced in this category, we considered the creative thinking skills - fluency, flexibility, and originality - as subcategories. We conducted a qualitative analysis based on the obtained responses (programs), considering that the authors/students should minimally describe the number of evaluations, assign grades, calculate the average, and define the minimum grade for student approval or disapproval. The prompt does not specify the need for recoveries or exams, leaving the students the freedom to choose this approach.

All presented programs are valid solutions for the proposed question, highlighting a characteristic of open-ended questions. Additionally, the students were not asked to generate more than one response to the question, which led us to analyze the different solutions generated as a whole.

We observed that the fluency skill was developed, as five distinct ideas were generated, all constituting suitable solutions for the question. However, we identified some mistakes in some solutions that do not globally compromise the programs. An example is in Program 2, where it is not specified what happens when a student obtains a grade equal to 5.0. In Program 3, the authors did not realize that the grade for the assignment works as a bonus in the student’s final grade. The authors of the programs should have revisited their solutions and conducted more simulations with different sets of test data for program testing and debugging.

To assess the flexibility skill and classify the variety of approaches in the generated solutions, we identified seven distinct groups. These groups were developed based on the strategies employed in the solutions, encompassing the number of evaluations/grades, the average, the possibility of an exam/recovery, and its grade. Table 1 presents this classification.

DOI: 10.20396/zet.v31i00.8672180

Table 1 - Classification of open-ended question solutions

	Two grades	More than two grades	Arithmetic mean	Weighted mean	Other type of mean	With exam/recovery	Considers the exam/recovery grade
Program 1	X			X		X	
Program 2	X		X				
Program 3		X			X	X	
Program 4	X	X	X				
Program 5	X		X			X	X

Source: Developed by the authors (2022)

We observed that Programs 1 and 2 specify a condition for the student to take an exam/recovery, but do not establish criteria for approval or disapproval after its completion. We assume that the grade from the exam/recovery can replace the previous grade, considering the values already assigned for approval or disapproval.

We observed that Program 5 encompasses elements from the seven groups, as highlighted in Table 1, making it a comprehensive solution for various possible situations in a real discipline context. The other programs involve elements from three groups, except for Program 2, which includes only two. Although all programs meet the open-ended question prompt, some have limited additional elements. We conclude that the flexibility skill has been developed.

Regarding the originality skill, we highlight Programs 1, 4, and 5 as having more original elements compared to the others. Program 1 proposes a weighted average - a less frequent approach compared to the other proposed averages. In Program 4, the number of evaluations is flexible, depending on the number of grades directly inserted into the program's code, and the final grade is calculated using an arithmetic mean, unlike the other solutions with a fixed number of evaluations. We consider this an element of originality, although reading data directly from the screen is more convenient for the user. Program 5 also presents original elements compared to the others, considering the calculation of the student's final grade in the case of an exam/recovery and displaying the approval or disapproval message. The other solutions have common elements.

In this regard, we understand that the originality skill is present in the presented programs.

In light of the above, we conclude the proposed activity, which is based on an open-ended question, promoting divergent thinking by enabling five distinct solutions. Additionally, the produced data reflects the creative thinking skills - fluency, flexibility, and originality - contributing to the development of creativity in mathematics.

(ii) Contributions of computer programming to creativity in mathematics

We begin by describing perceptions about Computational Thinking skills in the

DOI: 10.20396/zet.v31i00.8672180

conducted research, considering its pillars.

Upon analyzing the presented programs, we identified these pillars, with an emphasis on abstraction and algorithm. In the context of abstraction, there was data filtering and classification. To solve the proposed open-ended question, a strategy was devised, composed of a set of instructions implemented in Python.

Programs 1, 2, and 3 exhibit simpler structures, while the others present more elaborate approaches. For example, Program 4 employs commands for list manipulation, eliminating possible difficulties in handling different quantities of evaluations. Program 5 demonstrates the use of conditional statements if and else, highlighting a good understanding of logic on the part of the student-programmers. This originality in the solutions contributes to creativity in mathematics, as discussed earlier.

It is evident that when coding and testing a program, the author follows the entire process executed by the machine. This allows for simulation of situations, debugging, reflection on the solution, enhancement, innovation, and provides interactive and rapid feedback. This practice helps enhance the fluency and flexibility skills of creative thinking. This process of discovery, research, and reflection values different ways of thinking, not restricted to right or wrong, but unfolding in a continuous manner characteristic of research activities.

While programming in Python, the students demonstrated motivation, engagement in problem-solving, comfort with the trial-and-error approach, and the ability to work in groups to achieve common goals. These aspects reflect motivational, emotional, and attitudinal elements described by Barr and Stephenson (2011, p. 51), who also identify some elements of motivational, emotional, and attitudinal aspects, possible for the manifestation of Computational Thinking. These include:

> Confidence in dealing with complexity; Persistence in working with difficult problems; The ability to handle ambiguity; The ability to deal with open-ended problems; Setting aside differences to work with others to achieve a common goal or solution [...].

In this analysis, we also emphasize that problem-solving with the aim of exploring Computational Thinking and its benefits for learning in mathematics values diversity and creative formulation of solutions. This contributes to the understanding of phenomena through programming and encourages individual and collective decision-making, corroborating Azevedo and Maltempi (2020). Additionally, emotional, motivational, and group work aspects enrich creativity in mathematics, as already referenced. Thus, we perceive a strong connection between computer programming in Python and creativity in mathematics, an aspect previously observed by Resnick in his work with programming in Scratch.

DOI: 10.20396/zet.v31i00.8672180

Therefore, computer programming, especially through open-ended problem-solving in Python, demonstrates the potential to develop creative thinking skills - flexibility, fluency, and originality. This approach provides an opportunity to compare one's own thoughts (translated into a programming language) with those of other learners who also created solutions for the same problem through programming, and to study one's own learning process by comparing different developed programs.

Final Considerations

We have presented the results of research aimed at identifying and analyzing indications of mathematical creativity in programming activities developed in the Python language.

In the data analysis, it becomes evident, through the described elements, that computer programming has the potential to develop creative thinking skills - flexibility, fluency, and originality - through solving open-ended problems and implementing their solutions in a programming language, such as Python, for example. This language is functional and dynamic, easy to handle in Google Colab, and has proposals for its use in K-12 Education to work on aspects of Computational Thinking. We observe that it is possible to work on Computational Thinking skills and creative thinking since they share common elements, as previously mentioned. In this regard, we propose its use to promote creative thinking and/or creativity in mathematics in the school environment.

This proposal aligns with the works of Resnick (2007, 2020) and his movement to combine technologies - computer programming - with creative thinking and creative learning. According to this author, one of the possibilities of creating opportunities in schools to work with creativity is the use of technology, as it has "great potential to expand the scope of what people can create and what they can learn in this process. With the computer, they can create dynamic, interactive things that move, make their animations, simulations, and videos" (Resnick, 2022), with technology not replacing other forms of creation.

According to Azevedo and Maltempi (2020, p. 87), with computer programming, it is "possible to encourage and support students' creativity in their productions, envisioning new possibilities for imagination, investigation, curiosity, scientific argumentation, and invention of artifacts for problems encountered in society." Furthermore, for these authors, "creativity in the field of mathematics and computer science is unified as a purpose of mathematical education that transforms contexts and materials useful for science" (Azevedo & Maltempi, 2020, 87), and contributes to the civic education of young people.

This study presented research conducted in a continuing education course involving teachers and students of a Mathematics degree program. They solved a programming activity in Python based on an open-ended problem. While the mathematical content covered was relatively simple compared to the participants' level of knowledge, the difficulties

DOI: 10.20396/zet.v31i00.8672180

encountered were more related to the programming language. The goal was to demonstrate examples of open-ended mathematical problems that could be worked on in K-12 Education, combining programming and mathematical creativity. We hope these problems inspire the creation of other activities for teaching practice in the future.

Finally, we want to emphasize the importance of working with teachers in initial and continuing education in an extension course, especially when addressing innovative topics for K-12 Education, such as computer programming, Computational Thinking, and creative thinking. This equips teachers for their future teaching practices and allows for exploration of new approaches to learning, aligning with the trends proposed by Resnick (2007, 2020) and Azevedo and Maltempi (2020).

Acknowledgment

The second author has the support of The National Council for Scientific and Technological Development (CNPq) at Brazil. Grant 305264/2022-2.

References

Aiken, L. R. (1973). Ability and creativity in mathematics. *Review of educational research*, 43 (4), 405-432.

Alencar, E. M. L. S. (1990). *Como desenvolver o potencial criador*: um guia para a liberação da criatividade em sala de aula. Petrópolis: Vozes.

Azevedo, G. T., & Maltempi, M. V. (2020). Processo Formativo em Matemática e Robótica: Construcionismo, Pensamento Computacional e Aprendizagem Criativa. *Tecnologias, sociedade e conhecimento*, 7 (2), 85-107. https://doi.org/10.20396/tsc.v7i2.14857.

Barbosa, L. L. S., & Maltempi, M. V. (2020). Matemática, Pensamento Computacional e BNCC: desafios e potencialidades dos projetos de ensino e das tecnologias na formação inicial de professores. *Revista Brasileira de Ensino de Ciências e Matemática*, 3 (3), 748-776. https://doi.org/10.5335/rbecm.v3i3.11841.

Bardin, L. (2016). *Análise de conteúdo*. São Paulo: Edições 70.

Barichello, L. (2021). Pensamento Computacional. São Paulo: Instituto Nacional de Matemática Pura e Aplicada.

Barr, V., & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*, 2 (1), 48-54.

Bicudo, M. A. V. (2019). Pesquisa qualitativa e pesquisa qualitativa segundo a abordagem fenomenológica. In: M. C. Borba & J. L. Araújo (Orgs.), *Pesquisa Qualitativa em Educação Matemática.* (pp. 107-119). São Paulo: Autêntica.

Bogdan, R., & Biklen, S. (1997). *Investigação qualitativa em educação*: uma introdução à teoria e aos métodos. Porto: Porto Editora.

DOI: 10.20396/zet.v31i00.8672180

Brackmann, C. P. (2017). *Desenvolvimento do Pensamento Computacional Através de Atividades Desplugadas na Educação Básica*. [Tese de Doutorado em Informática na Educação – Cinted]. Porto Alegre: Universidade Federal do Rio Grande do Sul.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *In Proceedings of the 2012 annual meeting of the American Educational Research Association*, 1-25.

Denning, P. J. (2017). Remaining Trouble Spots with Computational Thinking. *Communications of the ACM*, 60 (6), 33-39. https://doi.org/10.1145/2998438.

Fleith, D. S. (2007). A promoção da criatividade no contexto escolar. In: A. M. R. Virgolim. (Org.), *Talento criativo*: expressão em múltiplos contextos (pp. 143-157). Brasília: Editora Universidade de Brasília.

Gontijo, C. H. (2007). *Relações entre criatividade, criatividade em matemática e motivação em matemática de alunos do ensino médio*. [Tese de Doutorado em Psicologia - Instituto de Psicologia]. Brasília: Universidade de Brasília.

Gontijo, C. H., Carvalho, A. T., Fonseca, M. G., & Farias, M. P. (2019). *Criatividade em matemática:* conceitos, metodologias e avaliação. Brasília: Editora da UnB.

Gontijo, C., Fonseca, M., Carvalho, A. & Bezerra, W. (2021). Criatividade em Matemática: alguns elementos históricos na constituição do campo de pesquisa e de intervenção pedagógica. *REnCiMa*, 12 (5), 1-24. https://doi.org/10.26843/rencima.v12n5a20.

Hadamard, J. (2009). Psicologia da invenção matemática. Rio de Janeiro: Contraponto.

Haylock, D. W. (1987). Mathematical creativity in schoolchildren. *The Journal of Creative Behavior*, 21 (1), 48-59.

Leonardo, F. M. (2020). *Grandezas, álgebra e algoritmos*. São Paulo: Moderna. Coleção Conexões Matemática e Suas Tecnologias.

Maltempi, M. V. (2005). Novas Tecnologias e Construção de Conhecimento: Reflexões e Perspectivas. In: *V Congresso Ibero-americano de Educação Matemática (CIBEM)*. Porto, Portugal.

Martindale, C. (1999). Biological bases of creativity. In: R. Sternberg, (Org.), Handbook of Creativity (pp. 137-152). Cambridge: University Press.

Ministério da Educação. (2000). *Parâmetros Curriculares Nacionais*: ensino médio: ciências da natureza, matemática e suas tecnologias. República Federativa do Brasil, Brasília.

Oliveira, A. N. (2016). *Projetos de conhecimento acoplados as tecnologias digitais para promover a criatividade em matemática.* [Tese de Doutorado em Informática na Educação - Centro Interdisciplinar de Novas Tecnologias em Educação]. Porto Alegre: Universidade Federal do Rio Grande do Sul.

Papert, S. (1994). *A máquina das crianças*: repensando a escola na era da informática. Porto Alegre: Artes Médicas.

Papert, S. (1997). *A família em rede*: ultrapassando a barreira digital entre gerações. Lisboa: Relógio D'Água Editores.

DOI: 10.20396/zet.v31i00.8672180

Pereira, E. (2008). *A modelagem matemática e suas implicações para o desenvolvimento da criatividade*. [Dissertação de Mestrado em Educação]. Ponta Grossa: Universidade Estadual de Ponta Grossa.

Resnick, M. (2007). Sowing the Seeds for a More Creative Society. *Learning and Leading with Technology*.

Resnick, M. (2020). *Jardim de infância para a vida toda*: por uma aprendizagem criativa, mão na massa e relevante para todos. Porto Alegre: Penso Editora.

Resnick, M. (2022). Entrevista para a *VI Jornada de Educação Infantil* - Unisinos. Disponível em: https://poseducacao.unisinos.br/blog/pensamento-criativo.

Valente, J. A. (2016). Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. *Revista e-Curriculum*, 14 (3), 864-897. https://doi.org/10.23925/1809-3876.2016v14i3p0864.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49 (3), 33-35. Available in: https://www.cs.cmu.edu/afs/cs/Web/People/15110-s13/Wing06-ct.pdf

Wing, J. M. (2011). *Computational Thinking*: *What and Why*. Cambridge. Available in: https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why.