Research ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

GENIA PLATFORM: A PROPOSAL FOR USING ARTIFICIAL INTELLIGENCE AND INTUITIVE PROGRAMMING IN THE CREATION OF LEARNING OBJECTS

PLATAFORMA GENIA: UMA PROPOSTA DE USO DA INTELIGÊNCIA ARTIFICIAL E DA PROGRAMAÇÃO INTUITIVA NA CRIAÇÃO DE OBJETOS DE APRENDIZAGEM

Evandro Alberto Zatti[1]
Marco Aurélio Kalinke[2]

Abstract: In Education, the use of artificial intelligence is observed, albeit timidly, in various situations ranging from management support to contributions to teaching and learning processes. This article presents completed doctoral research that investigated how artificial intelligence resources, combined with intuitive programming, could be explored in creating a platform for building Mathematics learning objects. Methodologically, prototyping and educational design research were employed. As part of a professional doctoral program, the research yielded an educational product called GenIA, which relies on the pillars of intuitive programming to construct Mathematics learning objects using flowcharts, assisted by artificial intelligence algorithms and machine learning. The platform was validated in different contexts and demonstrated its viability for use in Mathematics classes.

Keywords: Mathematics Education; Artificial intelligence; Learning objects; GenIA platform; Intuitive programming.

Resumo: Na Educação, observa-se, ainda que de forma tímida, o uso de inteligência artificial em diferentes situações que vão desde o suporte à gestão, até as contribuições com os processos de ensino e de aprendizagem. Este artigo apresenta uma pesquisa de doutorado já concluída, que investigou como recursos de inteligência artificial, aliados à programação intuitiva, poderiam ser explorados na criação de uma plataforma destinada à construção de objetos de aprendizagem de Matemática. Para tanto, metodologicamente foram utilizadas a prototipação e a pesquisa em design educacional. Por estar inserida em um programa de doutorado profissional, a pesquisa apresentou, como produto educacional, uma plataforma denominada GenIA, que se baseia nos pilares da programação intuitiva para construir, usando fluxogramas, objetos de aprendizagem de Matemática, assistida por algoritmos de inteligência artificial, fazendo uso de aprendizado de máquina. A plataforma foi validada em diferentes contextos, e se mostrou viável de ser explorada em aulas de Matemática.

Palavras-chave: Educação Matemática; Inteligência artificial; Objetos de aprendizagem; Plataforma GenIA; Programação intuitiva.

[1] Doctor in Science and Mathematics Education, Federal Technological University of Paraná (UTFPR). Pontifical Catholic University of Paraná (PUC-PR), Curitiba, Paraná, Brazil. E-mail: evandro.zatti@live.com.
[2] Doctor in Mathematics Education, Pontifical Catholic University of São Paulo (PUC-SP). Faculty member at the Federal Technological University of Paraná (UTFPR), Curitiba, Paraná, Brazil. E-mail: kalinke@utfpr.edu.br.

 ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

1 Introduction

In a continuous process of development, society is constantly exploring a variety of approaches and models in areas such as politics, economy, education, and culture. Some phenomena arise gradually, transforming over time, while others emerge suddenly and disruptively.

The analysis and recording of these transformations are often addressed in the literature (Toffler, 1980; Gates, 1995; Lévy, 2016), making it possible to identify various perspectives for classifying society. The presence of technology as a central factor or catalyst in this evolutionary process is notable and constant. The application of technology in the automation of tasks covers a wide range of examples, from the creation of the wheel to the use of digital personal assistants.

The opportunities and challenges that technology offers are supported, among others, by the theories of Tikhomirov (1981), which suggest that the use of technology can promote the reorganization of the individual’s creative mental processes.

> One of Vygotsky’s central theses is that mental processes in human beings change as their practical activity processes change (*i.e.*, mental processes become mediated). […] In the use of auxiliary means and signs (for example, making a notch on a stick to remember), humans produce changes in external things; but these subsequent changes then affect their internal mental processes (Tikhomirov, 1981, p. 264).

Currently, people live in a collective environment, sharing information in real time, thanks to the ubiquity of mobile networks and the internet. This has led to a redefinition of the concepts of time and space. Zatti and Kalinke (2021) consider that people are continuously in search of tools that enable time-saving, transferring to machines tasks that require physical effort, allowing them to focus their attention on personal intellectual growth and improvement.

Digital technologies (DT) enable not only the automation of tasks involving physical effort but also play a fundamental role from the spread of personal computers to the constant use of mobile devices in daily activities, enhanced by the sharing of information through the internet. These technologies have accompanied several generations over decades. Lévy (2007) already considered the possibilities of knowledge and understanding in the face of the reality of the great network.

> Our material relationship with the world is maintained through a formidable epistemic and software infrastructure: institutions of education and training, communication circuits, intellectual technologies with digital support, continuous updating, and dissemination of *savoir-faire*… Everything rests, in

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

> the long term, on the flexibility and vitality of our networks of production, commerce, and exchange of knowledge. (Lévy, 2007, p. 19).

Aligned with digital culture, DTs enter the educational scene offering the possibility to enhance teaching and learning processes through innovative didactic approaches. In this sense, Motta, Kalinke, and Mocrosky (2018, p. 67) point out that "[…] the insertion of technologies in the educational context presents a series of challenges, considering that new ways of relating and interacting with the world become evident." Regarding the use of technology and its impacts in the educational context, Borba, Scucuglia, and Gadanidis (2018, p. 21) make a transposition specifically for Mathematics Education: "[…] the dimensions of technological innovation allow the exploration and emergence of alternative scenarios for education, and especially for the teaching and learning of Mathematics".

The possibilities of using DTs in Mathematics Education are the subject of study by the Research Group on Technologies in Mathematics Education (GPTEM)[3], which explores the use of DT as a support tool for the teaching and learning processes of Mathematics. Among the various themes addressed by the group's researchers are learning objects (LO) and Artificial Intelligence (AI), which, along with intuitive programming, consolidate the triad that comprises the theoretical foundation of the research presented in this article.

The research movement can promote a deeper understanding of some of these concerns when it seeks to answer the following guiding question: How can AI resources, combined with intuitive programming, be explored in the creation of a platform designed for the construction of Mathematical LOs? It is relevant to emphasize that the research is associated with a professional doctorate program, and one of the distinctive characteristics of these programs in the field of teaching is the development of an educational product directly related to the research conducted on the topic in question. The educational product that served as the basis for exploring the possibilities presented by the research was a platform for the construction of Mathematical LOs that make use of intuitive programming and is assisted by AI, named GenIA[4].

There are other software and platforms commonly used for the construction of Mathematical LOs, such as GeoGebra[5] and Scratch[6]. However, the GenIA platform

---

[3] https://gptem5.wixsite.com/gptem
[4] http://plataformagenia.com
[5] https://www.geogebra.org/
[6] https://scratch.mit.edu/

 ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

differs from the existing resources in two main aspects: programming occurs through flowcharts, and to assist the user in constructing these, the platform implements AI algorithms.

The research sought to defend the thesis that AI can be an ally in the construction of Mathematical LOs with the use of intuitive programming, with an approach different from the visual programming adopted by other platforms. The theoretical studies that culminated in the creation of the educational product linked to the thesis will be presented in the next section.

2 In search of a theoretical basis

The path that led to the creation of GenIA was essentially based on three pillars: the LOs, intuitive programming, and AI. Therefore, it is important to go through its main theoretical contributions, which will be explored next.

2.1 Learning objects

LOs have been the subject of investigation in the educational field for a significant period, however, there is no consensus among researchers regarding their definition. The research in question adopts the definition presented by GPTEM, which considers an LO as "[…] any virtual multimedia resource, which can be used and reused to support the learning of a specific content, through interactive activity, presented in the form of animation or simulation" (Kalinke; Balbino, 2016, p. 25).

According to Meireles (2017), the phases involved in the creation of learning objects can become a challenging task if the fundamental characteristics of the objects are not adequately considered, especially interactivity and autonomy. Stavny *et al*. (2021, p. 3) regarding the challenges in constructing LOs consider that "[…] constructing an LO is not a simple task, as it involves different specificities, such as programming, design, pedagogical, methodological, and ergonomic aspects, among others. Because of this, it tends to become a multidisciplinary work."

Considering that GenIA is a product intended for the construction of LOs, it was considered important to visit methodologies that addressed the production process of this type of resource. Balbino and Mattos (2021) make considerations regarding the balance between pedagogical and computational aspects when elaborating an LO.

> When combining technical and pedagogical characteristics, there is a need to organize and standardize work methods that can contribute to the process of

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

> constructing LOs, that meet educational needs. […] Based on these propositions, we indicate the necessity of adopting a methodology that aims for a balance between the technical and pedagogical areas (Balbino; Mattos, 2021, p. 60).

In this perspective, the development of the platform was guided by a methodology proposed by Motta and Kalinke (2019), which "[…] associates the ideas of Cordeiro *et al*. (2007) about the RIVED method and of Kemczinski *et al*. (2012) on the instruments of production of an LO and the use of accessible software or virtual environments that utilize intuitive programming." (Motta; Kalinke, 2019, p. 208). The methodology is called "Methodology of Production of an LO in the Educational Dimension (MPEDUC)", and a compilation of its phases is presented in Table 1.

Table 1: Phases and steps of the MPEDUC

Phase	**Step**	**Goal**
Planning	LO general information	Identify the basic elements of the LO: objectives, specific content, and target audience.
	Conceptual map	Represent the content that will be proposed, structuring it sequentially and logically.
	Script	Structure a script of the object's screens, to present the sequence of use of the LO.
	Scenario map	Contextualize the LO, presenting a situation that is relevant to the reality of the target audience.
	Navigational map	Understand the direction, allowing to verify if the navigability represents the course of the proposed learnings.
Production	Tool choice	Choose the tool that best matches the teacher's competencies when manipulating.
	LO development	Develop the LO using the chosen intuitive tool.
	Preparation of the teacher's guide	Prepare material that serves as a support for teachers to take advantage of the LO's potential.
Validation	Feasibility test	Verify if the LO meets the parameters foreseen in the educational dimension.
Disclosure	Repository choice	Identify a virtual environment in which the LO can be shared.
	Preparation of metadata	Prepare the metadata of the LO, according to the repository, to enhance its organization and reuse.
	Making LO available	Allow other teachers to make use of the LO, adapting it to their context.

Source: Motta; Kalinke (2019, p. 204, adapted)

It should be noted that the platform created in conjunction with the research is intended for the development of the LO, one of the phases of the MPEDUC that integrates the production stage. In line with the assertion that development should be through an intuitive tool, it was considered necessary to have a way of programming that was, in a way, democratic, accessible to groups of Mathematics teachers who do not necessarily have computer programming in their training base. That is, the programming would need to be intuitive.

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

As for the LOs, there is no consensus definition for intuitive programming, so it is believed to be important to understand how the research considered this way of programming so that it could be implemented in GenIA, which will be addressed next.

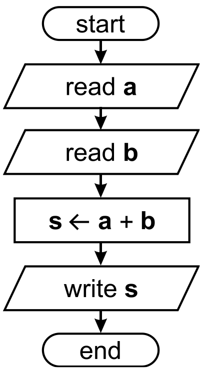
2.2 Intuitive programming

The definition of LOs used in the research, which describes digital resources aimed at learning through interactive activities, implies some conclusions. Highlighting the terms "digital," "learning," and "interactive," one can understand that the LO will be programmed by a teacher. Teachers in the field of Computing have in their training the mastery of computer programming, but for educators from other areas, developing such a skill may require specific training courses. As an alternative, intuitive programming can be used. Balbino *et al*. (2021), in search of a definition for intuitive programming in the educational context, point out that

> [...] one can propose a definition for intuitive programming as being a programming language intended for the construction of educational projects in computer environments that do not require mastery of a specific programming language and that present characteristics of similarity, visualization, and accessibility (Balbino *et al.* 2021, p. 19).

Programming a computer involves specifying a logical and finite sequence of commands intended to solve a particular problem or perform a task. This sequence, called an algorithm, according to Zatti (2017), can be noted in three ways, namely: narrative, graphic, or pseudocode. Table 2 presents the same algorithm in different forms.

Table 2: Ways to notate an algorithm

<table>
<tr><th>Narrative (natural language)</th><th>Graphic (flowchart)</th><th>Pseudocode (structured English)</th></tr>
<tr><td>Let a and b be two numbers, provided by the user, calculate their sum, and present the result</td><td>


</td><td>

```
Program "sumoftwonumbers"
Var
   a, b, s: integer
Start
   read (a)
   read (b)
   s <- a + b
   write (s)
End
```

</td></tr>
</table>

Source: Balbino *et al.* (2021, p. 4, adapted)

Currently, it is common for computers and digital devices to be capable of receiving commands in natural (human) language, whether by text or voice, and perform tasks related to image recognition, which suggests that it would be possible to program a

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

computer in any of the presented forms. However, in the exercise of their profession, a programmer needs to express commands in a programming language intended for the system they wish to program, such as C++, Java, C#, or Python.

When choosing a language, commands must be recorded in a text file known as source code, and it is essential to strictly adhere to the specified syntax.

The first computers were programmed through direct actuation of switches and buttons that combined to form sequences of electrical pulses inherent to their operation. Instructions in this primary form are known as machine language. In its evolutionary process, combinations were initially associated with mnemonics[7] and later with words increasingly closer to human writing. Because of this, one way to classify programming languages is by their degree of abstraction (Zatti, 2017). Languages closer to binary sequences and the use of symbols are called low-level languages, while languages with syntax closer to natural language are classified as high-level.

For Balbino *et al*. (2021), even when using a high-level language, a professional programmer needs to employ specific skills, requiring logical reasoning and the ability to construct effective algorithms to create programs using different languages, rules, levels of abstraction, and paradigms. Considering this perspective, intuitive programming reaches a higher level compared to languages traditionally considered high-level, as it needs to abstract the complexities related to software and hardware, coming even closer to natural language.

In search of elements that could underpin what would be intuitive in computational terms, Balbino *et al*. (2021) highlight three characteristics: similarity, visualization, and accessibility. They suggest that intuitive programming makes use of visual elements (graphics), a feature that Lévy (2004) considers an advantage, given the possibility of avoiding syntax errors by eliminating esoteric code.

Accessibility suggests that basic interface commands should be available in buttons and icons with clear meanings for the user. Visualization is achieved by combining, for example, colored blocks to perform programming, as seen in resources like Scratch and App Inventor. The fittings in these blocks prevent syntactically incompatible function elements from being combined. Finally, similarity suggests that the main focus should be on creating the LO, not learning the environment.

[7] "The symbols, known as mnemonics, are associations of binary sequences with short sequences of characters that are intelligible and, therefore, easier to memorize" (Zatti, 2017, p. 15).

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

Finally, we can move on to the third and last pillar of the research reported here, namely AI.

2.3 Artificial intelligence

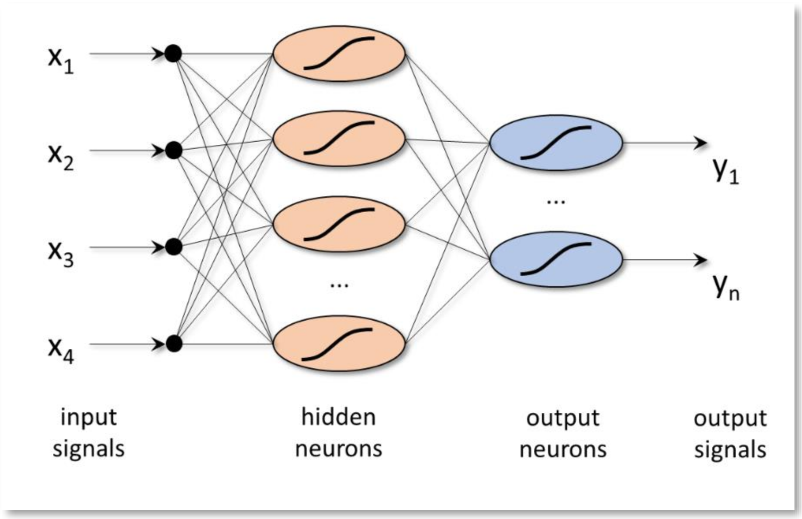
AI is a multidisciplinary field of study that has consistently attracted the attention of researchers from various domains. Several authors (Searle, 1980; Minsky; Papert, 1988; Russell, 2021) agree that AI represents a way for computers to perform tasks that would otherwise be carried out by humans. The term "artificial intelligence" was introduced in the 1950s and is credited to John McCarthy (1955), who coined it while referring to a proposal for studying how machines could learn.

However, experiments in this area began as early as the 1940s when McCulloch and Pitts (1943) presented the idea of simulating the functioning of neurons through mathematical models. Based on the model proposed by these authors, the first artificial neural network (ANN) was created. In the late 1950s, Minsky and Papert (1988) constructed a network of devices they called the "Perceptron".

According to Haykin (2001), an ANN bears a basic resemblance to the human brain in two aspects: 1) knowledge is acquired by the network through the learning process, and 2) synaptic weights (connection strengths) are used to store acquired knowledge.

Artificial neurons are referred to as the cells of the ANN. These cells are distributed in layers and can be classified based on their activation function. Figure 1 illustrates this structure.

Figure 1: Example of an ANN with two interconnected layers



Fonte: Kubat (2017, p. 92, adapted)

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

The functioning of machine learning (ML) in an ANN essentially occurs as follows: input values (signals) are provided to each neuron in the first layer, known as the input layer. These values are multiplied by pre-defined weights (bias[8] weight); by the input and activation functions, each neuron in the input layer produces an output, which serves as input for the next layer of neurons, called the hidden layer; this process continues successively until reaching the last layer of the network, known as the output layer.

The network undergoes successive executions (referred to as "generations"). In each generation, the network's output is analyzed, and if it does not yield the expected result, the bias weights of the neurons are adjusted, and the entire process is repeated. The more generations the network is subjected to, the more refined the generated result becomes.

The branch of AI that deals with ANNs is currently referred to as "connectionist". GenIA implements ML algorithms for the recognition and classification of LOs, and it is within this context that its AI resources fit.

Next, we present the methodological aspects that guided the conception and development of the platform, along with the milestones achieved during the process.

3 Methodological aspects

The research presented in this article is qualitative. According to D'Ambrósio (2004, p. 10), "[…] qualitative research, also known as naturalistic research, focuses on understanding and interpreting data and discourses, even when it involves participant groups." For Bicudo and Costa (2019), this type of research considers that it is not sufficient to provide a mere account of "how things are done," but rather involves a discursive work in which difficulties and controversies are also explicitly addressed.

To conduct the research, the Educational Design Research (EDR) approach was chosen. EDR pertains to the systematic study of the conception, development, and evaluation of educational interventions. According to Plomp (2013), it is particularly suitable for:

> [...] to design and develop an intervention (such as programs, teaching-learning strategies and materials, products, and systems) as a solution to a complex educational problem as well as to advance our knowledge about the

[8] In artificial neural networks, the term "bias" refers to an input value associated with a weight in a neuron, with the purpose of altering its activation function.

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

> characteristics of these interventions and the processes to design and develop them, or alternatively to design and develop educational interventions (about for example, learning processes, learning environments and the like) with the purpose to develop or validate theories (Plomp, 2013, p. 15).

The EDR assumes a path consolidated by three phases: preliminary, prototyping, and evaluation. The preliminary phase involves the following tasks: analyzing the context in which the educational intervention will be developed, identifying the educational problem to be solved, and conducting a literature review. In the prototyping phase, prototypes or models are created that will undergo iterations throughout the research process. It is expected that these iterations will lead to improvements in these prototypes. The evaluation phase considers that the prototype must undergo verification, during which it is observed whether it has fulfilled (or not) the established criteria.

EDR can be categorized into two types of studies, depending on their investigative objectives: development studies and validation studies. Development studies aim to create solutions for challenges encountered in educational practice, while validation studies are related to formulating or confirming a theory or theoretical understanding. According to Plomp (2013), these two types of studies are not mutually exclusive, and it is possible to combine both in an intervention work. The research presented here sought to strengthen the relationship between Mathematics Education and Computer Science, establishing connections that encompass intuitive programming and AI. In this process, the theories underlying these diverse areas were simultaneously validated. Therefore, the educational product (the platform) can be considered the intervention referred by Plomp (2013).

Considering that the research's educational product is a platform, it was also necessary to consider methodological aspects of software development. While EDR guided the successive evaluations and improvements of the product from an educational perspective, it was essential to concurrently observe technical development aspects, such as identifying programming errors or enhancement needs for the platform's functionalities.

In parallel with the use of EDR, the software development process adopted for the development of GenIA also needed to follow the iterations, focusing on the technical aspects of the product. In this regard, the prototyping model was adopted, as outlined in the literature on Software Engineering: "A prototype is an early version of a software system that is used to demonstrate concepts, try out design options, and find out more about the problem and its possible solutions" (Sommerville, 2016, p. 62).

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

Pressman and Maxim (2015) consider that the prototyping model follows five phases: 1) communication, in which stakeholders meet to define the overall objectives of the software; 2) quick design e 3) rapid modeling, which focuses on identifying aspects of the software that are visible to the customer, such as screens, buttons and other forms of interaction; 4) prototype construction; e 5) delivery and *feedback*, moment when stakeholders evaluate what was built, identifying points of acceptance or improvement, which will be implemented in the next iteration. A careful examination of the prototyping stages reveals their alignment with the phases of EDR[9].

To navigate the paths of EDR and the software development process through prototyping, we initially designed a theoretical technological solution and sought to determine if it would meet the needs for creating GenIA. The initial intention was to adopt a solution based on the Microsoft ecosystem, specifically: programming in the C# language using the .NET framework, utilizing the Visual Studio IDE, with collaboration and data sharing via cloud services.

Through searches conducted on the manufacturer's website and developer communities, it was found that the tools would indeed address the listed factors. Additionally, in the pursuit of software longevity, it was also identified that as of March 2022, and confirmed in June 2023, the C# language ranked among the top ten most used languages in applications across various countries (Tiobe, 2023).

To assess the feasibility of the proposal, several factors that the ecosystem should address were considered: development under the object-oriented paradigm; creation of applications with graphical interfaces; implementation of ML algorithms; execution in different environments (desktop, web, mobile devices); Data sharing in the cloud; Execution across various operating environments (desirable); Tracking development metrics through the IDE (desirable).

The platform, named GenIA[10], has been registered with the INPI[11], under the number BR512023001822-8, with a certificate issued in July 2023. Additionally, it has its website with the registered domain (plataformagenia.com), where other research artifacts, in addition to the software, are hosted and accessible to the public free of charge.

---

[9] The consonance between the prototyping and PDE stages is detailed in the author's doctoral thesis.
[10] GenIA is an acronym formed by Gen (initials for genesis) and IA (initials of Artificial Intelligence).
[11] The National Institute of Industrial Property (INPI), created in 1970, is a federal agency linked to the Ministry of Economy. It is responsible for the improvement, dissemination, and management of the Brazilian system for granting and guaranteeing intellectual property rights for the industry. Available at: <https://www.gov.br/inpi/pt-br>. Last access: Sep, 24th, 2023.

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

The registered version of the software with the INPI comprises approximately 45,000 lines of source code.

4 Searching for research data

The prototyping phase began in the second half of 2020 and continued through various iterations until the end of the research. At the end of each iteration, the following were evaluated: the graphical feasibility of the adopted solutions; the compatibility of metadata from other similar systems, especially Scratch, for feeding the platform and subsequent training of AI algorithms; the possibilities of using AI algorithms with ML found in the Microsoft ML.NET libraries; the feasibility of using intuitive programming within the platform, under different approaches; the evolution of ML algorithms and the inference in the operation of the interface by the teacher user.

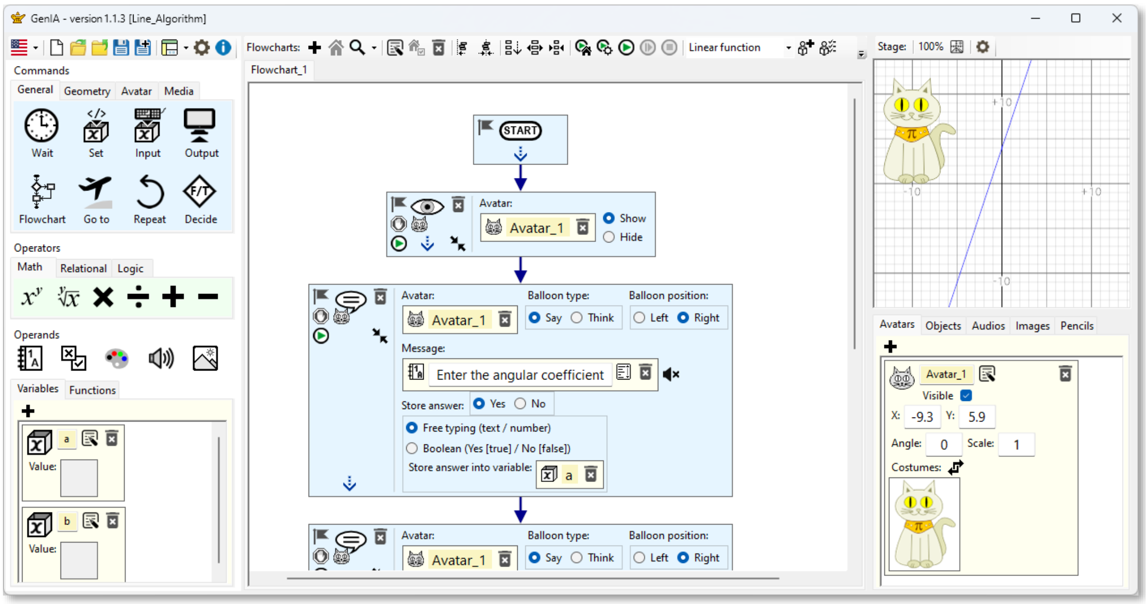
In the development of the prototype, in each iteration cycle, the stages of the software development process were followed, with minimal planning and modeling; resulting in the creation of comprehensive technical documentation of the software, for the responsible researcher, along with the executable prototype itself, which was made available to the public.

One of the challenges that needed to be overcome in the first iteration was how GenIA could provide intuitive programming. In the search for answers to this question, some programming environments considered intuitive were observed. Considering the history of the Logo language and its contributions to the educational context, it was decided to analyze Scratch because: 1) it is considered an evolution of Logo; 2) it is present in a significant number of research and published works that investigate programming in the educational field, with special attention to the publications of GPTEM.

Considering that intuitive programming is one of the pillars of the research, it was imperative that the design of the GenIA interface, presented in Figure 2, also considered these aspects.

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

Figure 2: GenIA interface



Source: Own authorship (2023)

As it is a platform aimed at creating Mathematical LOs, the icons and the organization of the commands were designed considering the prior knowledge of teachers in the field, such as the specific commands for geometry and the presence of formal symbols for operators. These characteristics reveal an initial concern with similarity, since the texts and messages from GenIA can be viewed in both Portuguese and English and can be easily translated into other languages.

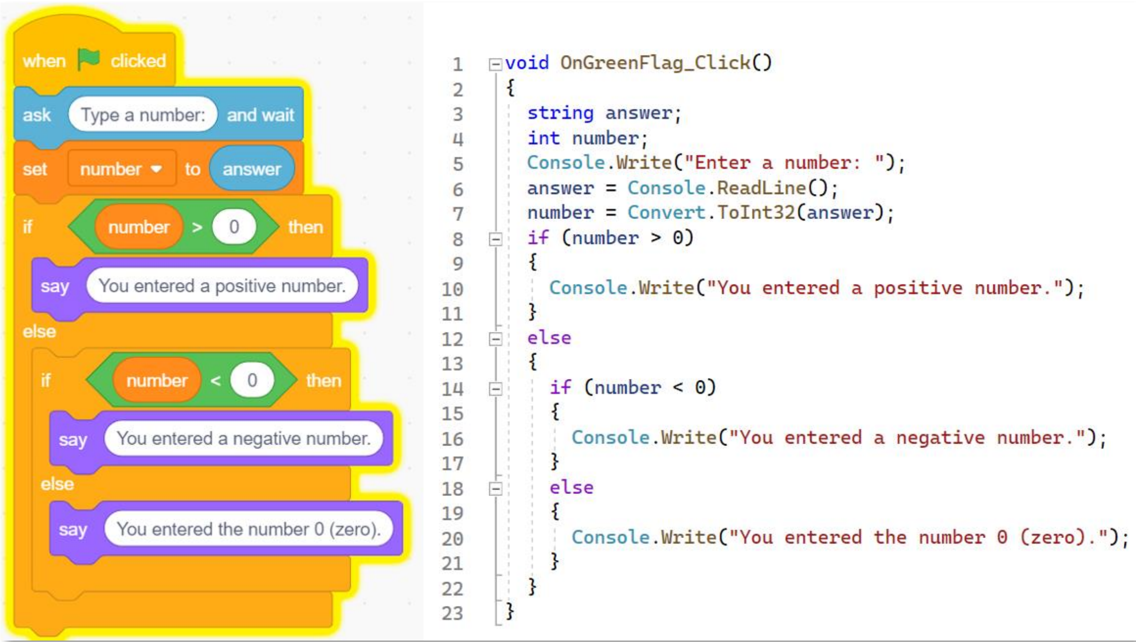
The attention to accessibility is immediately evident as the commands are represented by buttons and icons, organized by context, as well as by functionality, and offer graphical tips to the user.

Finally, visualization is addressed in different aspects. Firstly, graphic elements such as avatars and objects can be positioned directly on the Cartesian plane with the usual drag-and-drop feature of graphic environments, an action that directly reflects in the change of the Cartesian coordinates values of the corresponding component.

The second evidence of the characteristic regarding visualization as one of GenIA's differentials concerns the way programming occurs, through flowcharts. Although used by applications such as Scratch and App Inventor, block programming was, for the researchers involved with GenIA, a point of discomfort, since the set of blocks follows a visual structure analogous to structured programming. Figure 3 seeks to elucidate this association, placing a sequence of Scratch blocks and the analogous code snippet written in the C# programming language.

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

Figure 3: Comparison between blocks in Scratch and code snippet in C#



Fonte: Own authorship (2023)

Comparing the set of colored blocks with the C# code snippet, one can infer that, from Lévy's (2004) perspective, block programming could be classified as "image-dressed" programming:

> We can divide visual programming languages into two categories: 1. "Image-dressed" programming languages are not intrinsically visual but have overlaid visual representations. They are traditional languages with a visual interface or graphical type assistance to computer engineering. 2. Intrinsically visual programming languages seek to develop new programming paradigms whose expressions are naturally visual and would not even have textual equivalents (Lévy, 2004, p. 209).

To implement in GenIA an intuitive form of programming that could approach what Lévy called "intrinsically visual", the flowchart was used, understood as a diagram in which commands are interconnected by arrows indicating the direction of the flow, that is, the sequence according to which the commands will be executed. Flow representations are commonly used in different areas of knowledge, for content organization and problem-solving, such as the water cycle in Science education, the classification of quadrilaterals in Mathematical Education, organograms, and conceptual maps.

The development of the platform followed a constant expansion of functionalities, generating new executable versions, totalizing five versions in a continuous process of technical refinement until it could be submitted for user evaluation. At this stage, an initial version was tested by members of GPTEM, who are mostly Mathematics teachers from different educational levels.

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

While the group members validated usability issues and intuitive programming, the development proceeded to another expansion, this time exploring the possibilities of implementing AI modules. This phase began with samples created by the researcher himself, intending to study the possibilities of the ML.NET libraries, and later, the flowcharts created by the GPTEM members were gradually inserted into the training tests of the algorithms.

Based on the results obtained from the evaluations conducted and the available technical solutions, it was decided to adopt supervised ML. This is because, from the beginning, the proposal involved creating a system where, when a teacher developed a flowchart, the platform would be able to determine if that sequence of commands corresponded to an LO related to specific content. In this scenario, the input-output pairs would consist of the LO flowchart as input and the assignment of a “yes” or “no” label as output, indicating whether it is related to the trained content or not.

It should be noted that, according to Microsoft’s technical documentation[12], the implementation of ML in ML.NET should follow several steps: 1) Creation of an ML context; 2) Data loading; 3) Data transformation; 4) Algorithm selection; 5) Model training; 6) Model evaluation; 7) Model usage.

The creation of the context refers to the content that is being trained. To meet this step, GenIA has implemented an option where the teacher can create labels to identify different contents and later select them for training. Data loading is done by reading GenIA’s file, which contains the flowcharts. Data transformation represents an additional step that can be useful if there is a need to specify which content attributes should be processed by the algorithms. The choice of algorithm is a task exclusively assigned by the programmer, and the algorithms available for use by ML.NET are presented in Table 3.

Table 3: Algorithms for ML.NET training

ML Task	Algorithms
Binary classification	AveragedPerceptronTrainer, SdcaLogisticRegressionBinaryTrainer
Multi-class classification	LightGbmMulticlassTrainer, OneVersusAllTrainer
Regression	LbfgsPoissonRegressionTrainer, FastTreeRegressionTrainer
Clustering	KMeansTrainer
Anomaly Detection	RandomizedPcaTrainer
Recommendation	MatrixFactorizationTrainer
Ranking	LightGbmRankingTrainer, FastTreeRankingTrainer

Source: ML.NET (2023)

[12] Available at: https://dotnet.microsoft.com/en-us/learn/ml-dotnet/what-is-mldotnet. Last access: Sep, 24th, 2023.

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

For classification and identification of the content built in GenIA, a model based on the task of Binary Classification was used, employing a simple linear classification through the AveragedPerceptronTrainer algorithm, based on RNA Perceptrons. To feed the algorithm, a path traversal (graph) is performed, passing through all the components of the created flowchart, generating a sequence analogous to a sentence, which is evaluated by the sentiment analysis technique[13], a well-known technique of content classification by AI.

To expand usability testing, the platform was once again explored by two new groups of users[14]. The first application took place in the second half of 2022 when AI components were not yet included. The platform was used by a class of students enrolled in the bachelor's degree in Software Engineering at a private university in Curitiba, in the subject of Programming Algorithms, for interface testing and flowchart construction. The subject aims to develop skills in programming logic and is usually taught using textual commands and pseudocode for program development, but it also employs techniques for constructing flowcharts. During the classes taught by the author of the doctoral thesis and his advisor with the support of GenIA, the students were able to transpose pseudocodes into flowcharts and make considerations about the functioning of the commands, including interface and usability issues.

In the first semester of 2023, GenIA was tested again, this time in the subject Technologies in Mathematics Teaching, with students from the eighth period of the Mathematics Teaching degree at UTFPR, Curitiba campus. During the activity with the class, which consisted of nine students, the research objectives were explained and a general demonstration of the platform's functionalities was carried out, to contextualize its use.

The students had the opportunity to experience the creation of LO through GenIA, with the guidance of the researcher and the subject teacher. The intention was that, at the end of the dynamic, the students could consider the technical aspects of its use, but mainly, present a perspective of future use in educational processes.

[13] Sentiment analysis is the use of natural language processing, text analysis, computational linguistics, and biometrics to identify, extract, quantify, and systematically study affective states and subjective information.
[14] Regarding ethical aspects, the two applications with the students are supported by item VII of Circular Letter No. 17/2022/CONEP/SECNS/MS, which provides guidance on article 1 of CNS Resolution No. 510, from April 7, 2016.

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

The students expressed favorable opinions about the platform in their speeches, highlighting that the creation of LO with it proved to be simpler compared to other platforms they had used before, such as Scratch and App Inventor. However, they made criticisms regarding the mastery of the commands necessary to put into practice the ideas they had for the construction of the LOs. According to them, GenIA would need to have a repository of tutorials to facilitate the understanding of its possibilities. This led the researchers involved with GenIA to create these materials, to be made available to users along with the installation files. Such materials are available on the platform's website[15], organized into specific sections, in the tutorials tab.

Knowing the work resulting from the thesis of the first author, the following section presents considerations, with suggestions for future work based on a look at the path taken.

4 Some considerations

The adopted methodology followed two distinct phases: the development of the software with a focus on intuitive programming; and the incorporation of AI features. Concurrently with the evolution of the first stage, following technical requirements, verifications and validations were conducted at various levels, following the adopted software development process. In the educational context, taking into account usability and intuitive programming, the teachers who are part of GPTEM carried out continuous evaluations, following the iterations guided by EDR. Based on the data collected in this phase, about 40 compilations of GenIA were produced, incorporating new functionalities and corrections.

Along with the validation carried out by the members of GPTEM, the platform was implemented in real academic scenarios in two different situations: with students from the Bachelor's degree in Software Engineering, who provided technical considerations on the creation of flowcharts, and later, as a tool used in a subject in the Mathematics Teaching degree course. This use generated data on any specific difficulties users had in exploring specific features of the platform, which led to the construction of support materials and tutorials.

Reflecting on the path taken and, equally important, on GenIA as a tangible educational product, subject to tests under technical perspectives and evaluations by

[15] http://plataformagenia.com

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

researchers in Mathematics Education, it is understood that the proposed thesis is feasible and that the presented question was answered. The AI resources, combined with intuitive programming, were explored on a platform for building Mathematical OAs through the use of flowcharts and AI algorithms that use supervised ML.

The initial tests conducted with the LOs created by the users revealed that it is possible to train the AI algorithms, even though the number of samples produced is not sufficient to achieve the necessary accuracy within the timeframe anticipated for the completion of the doctorate. However, since the algorithms are implemented on the platform and will remain in constant training with new LOs created, they may receive contributions from other spheres of the academic community and serve as a basis for future research on related themes. To this end, the platform's website accepts uploads of files of the LOs constructed by the community.

Regarding the movements that surrounded the research, it was noticed that the insertion of AI among the study and research themes of GPTEM has revealed quite unusual path. As new proposals emerge, manifesting aspects to be explored in the educational field, there is also apprehension on the part of new researchers, which highlights the importance of having more interdisciplinary research involving AI and Education, so that some precepts can be demystified.

Nevertheless, several researchers are willing to develop new research on the use of AI in Mathematics classes, from the perspective of Mathematics Education, and many of these studies are underway within GPTEM. Many of them are dedicated to exploring GenIA with different audiences and expanding their understanding of its potentialities and weaknesses.

4.1 Recommendations for future work

Considering the technical aspects of the platform, various paths can still be explored. Not due to limitation, but by choice, the initial version of GenIA was developed exclusively for execution in a Windows environment. A proposal for future research is to port GenIA to a web environment, enabling its use on a broader range of equipment, including mobile devices. Usability-related issues can also be improved, as there are still various graphic resources and layout features that can be inserted or enhanced.

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

Redirecting attention to broader issues, for example, the possibility of integrating GenIA with kits aimed at Robotics, in solutions that combine with Arduino or similar, is envisioned.

GenIA has in its design a special concern for serving diverse audiences, so much so that its initial version allows use in two languages (Portuguese and English), which immediately suggests the possibility of research on transcultural translations into other languages.

There is also a need to explore other AI algorithm possibilities for implementation in GenIA. Its initial version works only with content classification. It is understood that it is possible, for example, to employ algorithms that allow the automatic completion of graphic elements during the construction of flowcharts. Still in the vein of AI alternatives, the implementation of personalized messages is suggested, according to the teacher's direction in the construction of the LOs.

Expanding the vision to an academic-scientific perspective, it is suggested that GenIA be used for various audiences (teachers, undergraduate academics, elementary and high school students, etc.), accompanied by researchers who seek to understand how GenIA can be explored in educational activities and what its potential is in activities or subjects that investigate Computational Thinking. In parallel, the development of research on flowchart programming concerning those carried out by blocks may bring a greater understanding of this specific theme.

Finally, in the social context, it is further suggested that there be extensive discussions about the ethical aspects surrounding both the use of GenIA itself and research that explores AI as an educational resource.

Acknowledgments

We thank the National Council for Scientific and Technological Development – CNPq for their support.

References

BALBINO, R. O.; KALINKE, M. A.; ZATTI, E. A.; MATTOS, S. G. DE; LOSS, T.; MOTTA, M. S. Programação Intuitiva: em Busca de Compreensões. **Perspectivas da Educação Matemática**, Campo Grande, v. 14, n. 36, p. 1-22, dez. 2021. https://doi.org/10.46312/pem.v14i36.12121.

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

BALBINO, R. O.; MATTOS, S. G. Uma proposta de utilização da MPEDUC para a construção de objetos de aprendizagem. In: MOTTA, M. S.; KALINKE, M. A. (org.). **Inovações e Tecnologias Digitais na Educação**: uma busca por definições e compreensões. Campo Grande, MS: Life Editora, 2021, p. 57-74.

BICUDO, M. A.; COSTA, A. P. (org.). **Leituras em Pesquisa Qualitativa**. São Paulo: Editora Livraria da Física, 2019.

BORBA, M. C.; SCUCUGLIA, R. R. S.; GADANIDIS, G. **Fases das tecnologias digitais em Educação Matemática**: sala de aula e internet em movimento. 2. ed.; 2. reimp. Belo Horizonte: Autêntica, 2018.

D'AMBROSIO, U. In: ARAÚJO, J. L., BORBA, M. C. **Pesquisa Qualitativa em Educação Matemática**. Belo Horizonte: Editora Autêntica, 2004. p. 9-21.

GATES, B. **The Road Ahead**. New York: Viking Penguin, 1995.

HAYKIN, S. **Redes Neurais -** Princípios e Práticas. 2. ed. São Paulo: Bookman, 2001.

KALINKE, M. A.; BALBINO, R. O. Lousas Digitais e Objetos de Aprendizagem. *In:* KALINKE, M. A.; MOCROSKY, L. F. (org.). **A Lousa Digital & Outras Tecnologias na Educação Matemática**. Curitiba, CRV, 2016. p. 13-32.

KUBAT, M. **An Introduction to Machine Learning**. 2 ed. Cham: Springer, 2017.

LÉVY, P. **A ideografia dinâmica**: Rumo a uma imaginação artificial? 2. ed. São Paulo: Edições Loyola, 2004.

LÉVY, P. **A inteligência coletiva**: por uma antropologia do ciberespaço. Tradução de Luiz Paulo Rouanet. 5. ed. São Paulo: Edições Loyola, 2007.

LÉVY, P. **O big data e a próxima revolução científica**. 2016. Disponível em: https://www.youtube.com/watch?v=W5hIcxKPVRw. Acesso em: 24 set. 2023.

MCCARTHY, MCCARTHY, J.; MINSKY, M. L.; ROCHESTER, N.; SHANNON, C. E. **A Proposal for The Dartmouth Summer Research Project on Artificial Intelligence**. 1955. Disponível em: http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html. Acesso em: 24 set. 2023.

MCCULLOCH, W. S.; PITTS, W. H. A Logical Calculus of The Ideas Immanent. **Nervous Activity. Bulletin of Mathematical Biophysics**, New York, v. 5, n. 01, p. 115-133, 1943.

MEIRELES, T. F. **Desenvolvimento de um Objeto de Aprendizagem de Matemática usando o Scratch**: da elaboração à construção. Curitiba, 2017. 217 f. Dissertação (Mestrado em Educação) – Universidade Federal do Paraná, Curitiba, 2017. Disponível em: https://acervodigital.ufpr.br/xmlui/handle/1884/56109. Acesso em: 20 mar. 2024.

MINSKY, M. L.; PAPERT, S. A. **Perceptrons**: An Introduction to Computational Geometry. Expanded Edition. 3 ed. Boston: MIT Press, 1988.

ML.NET. **What is ML.NET?** Disponível em: https://dotnet.microsoft.com/en-us/learn/ml-dotnet/what-is-mldotnet. Acesso em: 24 set. 2023.

MOTTA, M. S.; KALINKE, M. A. Uma proposta metodológica para a produção de objetos de aprendizagem na perspectiva da dimensão educacional. In: MOTTA, M. S.; KALINKE, M. A.

ISSN 2525-8222
DOI: http://dx.doi.org/10.33361/RPQ.2024.v.12.n.30.720

(org.). **Objetos de aprendizagem**: pesquisas e possibilidades na Educação Matemática. Campo Grande, MS: Life Editora, 2019, p. 203-218.

MOTTA, M. S.; KALINKE, M. A.; MOCROSKY, L. F. Mapeamento das dissertações que versam sobre o uso de tecnologias educacionais no ensino de Física. **Actio**: Docência em Ciências, Curitiba, v. 3, n. 3, p. 65-85, set./dez. 2018. http://dx.doi.org/10.3895/actio.v3n3.7591.

PLOMP, T. Educational design research: an introduction. In: PLOMP, T., NIEVEEN, N. (org.). **Educational design research** – Part A: an introduction. 3 ed. Enschede: Netherlands Institute for Curriculum Development (SLO), 2013. p. 10-51.

PRESSMAN, R. W; MAXIM B. R. **Software Engineering** – A Practitioner's Approach. 8th ed. New York: McGraw-Hill, 2015.

RUSSELL, S. **Inteligência artificial a nosso favor**: Como manter o controle sobre a tecnologia. Tradução de Berilo Vargas. São Paulo: Companhia das Letras, 2021.

SCRATCH. **Crie estórias, jogos e animações**. Disponível em: https://scratch.mit.edu/. Acesso em: 24 set. 2023.

SEARLE, J. Minds, brains, and programs. **The Behavioral and Brain Sciences**, Berkeley, v. 3, p. 417-457. 1980. Disponível em: http://cogprints.org/7150/1/10.1.1.83.5248.pdf. Acesso em: 24 set. 2023.

SOMMERVILLE, I. **Software Engineering**. 10th ed. Harlow: Pearson Education Limited, 2016.

STAVNY, F. M.; MATTOS, S. G.; BALBINO, R.; ZATTI, E. A.; KALINKE. M. A. Em busca de compreensões sobre utilização de recursos digitais na criação de objetos de aprendizagem de matemática. **REVEMAT**: Revista Eletrônica de Educação Matemática, Florianópolis, v. 16, p. 01-22, jan./dez. 2021. https://doi.org/10.5007/1981-1322.2021.e80312.

TIKHOMIROV, O. K. The Psychological Consequences of Computerization. In: WERTSCH, J. V. (org.). **The Concept of Activity in Soviet Psychology**. New York: M. E. Sharpe Inc., 1981. p. 256-278.

TIOBE. **TIOBE Programming Community Index**. 2023. Disponível em: https://www.tiobe.com/tiobe-index/. Acesso em: 24 set. 2023.

TOFFLER, A. **A Terceira Onda**. Rio de Janeiro: Record, 1980.

ZATTI, E. A. **Programação orientada a objetos**. Curitiba: Fael, 2017.

ZATTI, E. A.; KALINKE, M. A. Inteligência Artificial na Educação Matemática: tendências ou "entendências". In: KALINKE, M. A.; MOTTA, M. S. (org.). **Inovações e Tecnologias Digitais na Educação**: uma busca por definições e compreensões. Campo Grande, MS: Life Editora, 2021. p. 75-96.

Received on: October 6, 2023.

Accepted on: April 1, 2024.