Sociedade Brasileira de Educação Matemática

RIPEM

Intelligence technologies in Mathematics Education: AnyLogic for the production of learning objects

Deive Barbosa Alves
Universidade Federal do Norte do Tocantins
Araguaína, TO — Brasil
✉ deive@uft.edu.br
iD 0000-0002-0850-7362

Arlindo Jose de Souza Junior
Universidade Federal de Uberlândia
Uberlândia, MG — Brasil
✉ arlindo@ufu.br
iD 0000-0002-5175-6129

Rosana Sueli da Motta Jafelice
Universidade Federal de Uberlândia
Uberlândia, MG — Brasil
✉ rmotta@ufu.br
iD 0000-0001-8489-3974

2238-0345 ISSN

10.37001/ripem.v13i3.3537 doi

Received • 28/12/2022
Approved • 21/02/2023
Published • 10/09/2023

Editor • Gilberto Januario iD

Abstract: This work aims to present a tool for authoring learning objects (LO) for mathematics education in digital culture. This context reveals simulation as the new way of knowing the world through three modeling methods: discrete events, agent-based and system dynamics. Convergence culture was used as a selection criterion, resulting in *AnyLogic* software as the only authoring tool capable of simultaneously performing these three types of modeling. To analyze it, we built an object that simulated the availability of care in a hospital during an epidemic. The results point to a code block programming culture in *AnyLogic* when creating the LO. In the simulation, we observed that depending on the number of vacancies in the hospital, we have the severity of the disease and, consequently, the level of concern of those infected, compatible with the situation of the epidemic.

Keywords: Modeling. Simulation. Hybrid Models. Digital Technology in Education.

Tecnologías de inteligencia en la Educación Matemática: AnyLogic para la producción de objetos de aprendizaje

Resumen: El objetivo, en este trabajo, es presentar una herramienta para la creación de objetos de aprendizaje (OA) para la educación matemática en la cultura digital. Este contexto revela la simulación como la nueva forma de conocer el mundo, a través de tres métodos de modelado: eventos discretos, basado en agentes y dinámica de sistemas. La cultura de convergencia se utilizó como criterio de selección, lo que resultó en el *software AnyLogic* como la única herramienta de autor capaz de realizar estos tres tipos de modelado simultáneamente. Para analizarlo, construimos un objeto que simulaba la disponibilidad de atención en un hospital durante una epidemia. Los resultados apuntan a una cultura de programación de bloques de código en *AnyLogic* al crear el OA. En la simulación se observa que, dependiendo del número de vacantes en el hospital, tenemos la gravedad de la enfermedad y consecuentemente el nivel de preocupación de los contagiados, compatible con la situación de la epidemia.

Palabras clave: Modelado. Simulación. Modelos Híbridos. Tecnología Digital en la Educación.

Tecnologias da inteligência na Educação Matemática: O AnyLogic para a produção de objetos de aprendizagem

Resumo: O objetivo, neste trabalho, é apresentar uma ferramenta para a autoria de objetos de aprendizagem (OA) para a educação matemática na cultura digital. O referido contexto revela a simulação como a nova forma de conhecer o mundo por três métodos de modelagem: eventos discretos, baseada em agentes e dinâmica de sistemas. Usou-se a cultura da convergência como critério de escolha, resultando no *software AnyLogic* como a única ferramenta de autoria capaz realizar esses três tipos de modelagem simultaneamente. Para analisá-la, construímos um objeto que simulou a disponibilidade de atendimento em um hospital durante uma epidemia. Os resultados apontam para uma cultura de programação por blocos de códigos no *AnyLogic* ao criar o OA. Na simulação, nota-se que, dependendo do número de vagas no hospital, temos a gravidade da doença e, consequentemente, o nível de preocupação dos infectados compatível com a situação da epidemia.

Palavras-chave: Modelagem. Simulação. Modelos Híbridos. Tecnologia Digital na Educação.

1 Introduction

The Center for Research in Media in Education at the Federal University of Uberlândia (Núcleo de Pesquisa em Mídias na Educação — NUPEME-UFU) was created in 2004. Since its foundation, one of the lines of research has been learning objects (LO). To Rebouças, Maia, and Scaico (2021, p. 4),

> some other variations of the term have been proposed to highlight sometimes the aspect of virtuality, sometimes the expansion beyond student learning. Denominations such as virtual learning objects (VLO), digital learning objects (DLO) and, more recently, digital educational resources (DER) are some of the examples created as an alternative to the vagueness of the concept and foundation of LO.

Although these authors base such distinctions on conceptual imprecision, it appears that most — if not all — use the initial concept created by Wiley (2000, p. 3): “Learning objects are [...] any entity, digital or not, that can be used, reused, or referenced during technology-mediated learning [...]”. Several criticisms were made about it over the years, which led to several improvements and metaphors of this concept, which can be read in the collection organized by Pimental, Sampaio, and Santos (2021). Although such discussions indicate one or another advance, they are part of the same historical perspective: “Learning objects (LOs) emerged as a result of the success of object-oriented programming [(POO)]” (Tarouco, Bulegon, & Àvila, 2021, p. 1), which demarcates a territory[1] created, maintained, extended, and under the ownership of the

> Institute of Electrical and Electronic Engineers (IEEE) [which] was created in 1884 and has as its mission to foster technological innovation and excellence for the benefit of humanity. The IEEE has a Technology Standards Committee, the IEEE Learning Technology Standards Committee (LTSC), whose role is to develop technical standards, recommended practices, and guides for learning technology that are internationally accredited (Braga & Menezes, 2014, p. 21).

In this same perspective are the records of academic achievements, prepared by researchers from NUPEME-UFU until the present article, as shown in the dissertations, in chronological order: Rodrigues (2006); Fonseca (2009); Nunes (2010); Cintra (2010); Alves (2012); Souza (2017). Also, the books published by Souza Júnior, Fernandes, Lopes and Silva

[1]For this study, *territory* “is the set of projects and representations into which a whole series of behaviors and investments will pragmatically land in times and social, cultural, aesthetic, and cognitive spaces” (Guattari & Rolnik, 1986, p. 323).

(2007) and Souza Júnior, Fernandes, Lopes, and Silva (2010). Book chapters: Silva, Fernandes, Lopes, and Souza Júnior (2007); Souza Júnior, Rodrigues, and Lopes (2007); Souza Júnior and Lopes (2007); Souza Júnior and Moura (2010), and Alves et al. (2015). Finally, the complete articles published in journals by Moura, Alves, and Souza Júnior (2013) and Moura, Souza Júnior, Carvalho, and Alves (2018). These works demarcate the NUPEME-IEEE territory, which is permeated with stories of LO technologies driven by the government project of the Virtual Interactive Education Network (Rede Interativa Virtual de Educação — RIVED):

> [...] in June 2004, the long-awaited opportunity happened: the MEC was setting up university groups to study and develop virtual learning objects through the project entitled RIVED (Rede Interativa Virtual de Educação). From that moment on, the team of the Federal University of Uberlândia, UFU, [...] [which constituted] a government initiative to introduce and use information and communication technologies in national public education (Rodrigues, 2006, pp. 10;35).

Cintra (2010) and Rebouças, Maia, and Scaico (2021) clarify that from 1999 to 2004, RIVED was the acronym for Rede Internacional Virtual de Educação [International Virtual Education Network], an international initiative involving Brazil, Peru, and Venezuela. It aimed to

> develop digital educational modules, composed of technical-pedagogical documentation and activities to be done by students in a computerized environment mediated by a teacher in high school biology, physics, chemistry, and mathematics. It was a pioneering initiative that used computer technology, through the creation of didactic material in the form of an LO, to help teaching and learning in the areas of science and mathematics. (Cintra, 2010, p. 20)

Those researchers claim that over 100 LOs were developed during that period. After 2004, the production process was transferred to universities by the Secretariat for Distance Learning (Secretaria de Educação a Distância — SEED), constituting the RIVED virtual factory, including, in addition to high school content, content for elementary school, vocational education and special education. “Based on these changes, RIVED is no longer an International Virtual Education Network, but an Interactive Virtual Education Network. The project ended in the mid-2010s [...]” (Rebouças, Maia, & Scaico, 2021, p. 9).

Although the government project ended in 2010, the works above verify the continuity of production in universities, besides presenting the importance of still discussing the LOs. What became more evident with the effects that the Covid-19 pandemic brought to education, as evidenced by the research by Santos, Lopes, Fagundes, Falavigna, and Andreatta-da-Costa (2022) with the work of 571 teachers during the pandemic, was the extreme urgency for teachers to know and produce LOs, so that the educational process does not drift in contexts such as the one presented in remote teaching.

Another important historical fact is that, concomitantly with the production of Learning Objects in universities, the SEED/MEC launched three notices for LO production awards, the first edition in 2005, the second in 2006, and the third in 2007. “Under and postgraduate students from national higher education institutions and teachers of basic and vocational education, including multipliers from the educational technology centers of the states [...]” could participate in the public notices. (Brasil, 2006, p. 1). In the first, seven were awarded, and in the second, they were “44 learning objects out of the 74 received by the competition organization” (Brasil, 2006, p. 1). Finally, in the last one, 80 were awarded, each receiving R$

5,000. “In this edition, approximately 200 objects were entered in the arts, sciences, history, geography, Portuguese language, mathematics, chemistry, biology, and physics” (Brasil, 2007, p. 1).

We did not find a precise number of how many LOs were built from 1999 to 2010, when the federal government promoted the project. However, RIVED taught and propagated the authoring tool *Adobe Flash* to create such objects. This product was viewed and reproduced on the internet through the software *Flash Player* which, unfortunately, for security reasons, has been discontinued from the web browsers on December 31, 2020. This fact meant that all learning objects constructed with this tool were no longer accessed by the typical user from that date on, causing researchers in the area and who produced with this tool to investigate other forms of authorship. From this gap, we created our research problem: *What authoring tool can fit the production of learning objects in the context of mathematics education*? Thus, the general objective of the work is to present a tool for producing learning objects for the mathematics degree students in the context of the digital culture.

To answer the question and reach the objective, this article proposes a vanishing line from[2] the NUPEME-IEE territory to another: NUPEME-LÉVY, a new territory under creation that seeks to shape the authoring tool for the creation of LOs. So, in the section *Theoretical Foundation* we established the theoretical framework, which outlines a path for choosing this one. In the section *Investigation of Modeling Paradigms in the Choice of Authoring Tool*, the existing modeling methods that allow creating LOs are analyzed. In the section *Problematizing the Use of the Authoring Tool*, the problem situation of an epidemic was formulated, building a simulator with different types of modeling to shape the object for mathematics undergraduates. In the last section, *Results and Conclusions*, the usability of the LO was discussed, answering the question.

2 Theoretical Framework

This section will approach some concepts for constructing LOs in mathematics education. There are several possibilities for this discussion, so we chose to start with what we understand by technology, believing that, from there, other concepts will be concatenated and clarified.

So, when we say *technology*, we want to discuss it in the precepts of Foucault (1982, p. 323-324):

> [...] we must understand that there are four major types of these “technologies,” each a matrix of practical reason: (I) technologies of production, which permit us to produce, transform, or manipulate things; (2) technologies of sign systems, which permit us to use signs, meanings, symbols, or signification; (3) technologies of power, which determine the conduct of individuals and submit them to certain ends or domination, an objectivizing of the subject; (4) technologies of the self, which permit individuals to effect by their own means or with the help of others a certain number of operations on their own bodies and souls, thoughts, conduct, and way of being, so as to transform them in order to attain a certain state of happiness, purity, wisdom,

[2] It is “a deterritorialization. The French do not really know what that is. Of course, they run away like everyone else, but they think that running away is leaving the world, the mystical or the art, or else something cowardly, because one escapes commitments and responsibilities. Running away is not renouncing actions. There is nothing more active than an escape. It is the opposite of the imaginary. It is also making others flee, not necessarily others, but making something flee, making a system leak as if we pierced a pipe. […] To flee is to draw a line, lines, an entire cartography. One only discovers worlds through a long broken flight” (Deleuze & Parnet, 1998, p. 30).

> perfection, or immortality.

These four types of technologies hardly ever function separately [...] Each implies certain modes of training and modification of individuals, not only in the obvious sense of acquiring certain skills but also in the sense of acquiring certain attitudes" (Foucault, 1982, p. 324).

From this point of view, discussing a possible technology for teaching mathematics is, rather, understanding what it is and how it is practiced. In this regard, Devlin (2012, p. 4, our translation) clarifies that the

> [...] dramatic growth of mathematics led, in the 1980s, to the emergence of a new definition of mathematics as the science of patterns. According to this description, the mathematician identifies and analyzes abstract patterns — number, shape, movement, behavior, voting patterns in a population, repetition patterns of random events, and so on. These patterns can be real or imaginary, visual or mental, static or dynamic, qualitative or quantitative, utilitarian or recreational. They can arise from the world around us, the pursuit of science, or the inner workings of the human mind. Different types of patterns give rise to different branches of mathematics.

With this British mathematician, we started this path, understanding that mathematics is the science of abstract patterns and that its practical knowledge lies in identifying, analyzing, and using it to have a life worth living. Indeed, this led us to narrow down our interest, to building LOs to abstract real patterns that arise from the world around us. To Levy (2001, p. 71), "abstraction [...], as cognitive activities, has, therefore, an eminently practical origin [...]"

> in terms of their relations to intellectual technologies. Any problem beyond our immediate manipulation and recognition capacities is abstract. Thanks to systems of external representations, abstract problems can be translated or reformulated in such a way that we can solve them by executing a series of simple and concrete operations that use our operative and perceptive faculties. To be correctly conducted, these manipulations of representations must be objects of learning and training, like any other activity. [...] This is where the third cognitive faculty comes into play: our ability to "run" mental models of our environment. [...] Thanks to the simulation of mental models, the cognitive system partially introjects the representation systems and operating algorithms whose use it has acquired. Intellectual technologies, even though they belong to the "exterior" sensible world, also participate fundamentally in the cognitive process. They embody one of the objective dimensions of knowing subjectivity. Intellectual processes do not only involve the mind, they bring into play complex technical things and objects with a representative function and the operational automatisms that accompany them. [...] the creation of new models of representation and manipulation of information mark important stages in the human intellectual adventure. And the history of thought is not identified here with the series of products of human intelligence, but rather with the transformations of the intellectual process itself, this mix of subjective and objective activities (p. 159-160).

Lévy (2001) started from technical data to define that the way of knowing that comes from computer-based technologies is simulation: "our ability to 'run' mental models of our environment" (Lévy, 2001, p. 159). A way of knowing about the abstraction that

> it is not read or interpreted like a classic text, it is usually explored interactively. Contrary to most functional descriptions on paper or reduced analogue models, the computer model is essentially plastic, dynamic, endowed with a certain autonomy of

> action and reaction. (...) Knowledge by simulation is undoubtedly one of the new genres of knowledge that computerized cognitive ecology transports. [... In it, it is possible to manipulate the parameters and simulate all possible circumstances [which] give the program user a kind of intuition about the cause and effect relationships present in the model. They acquire knowledge by simulating the modeled system, which resembles neither theoretical knowledge, nor practical experience, nor the accumulation of an oral tradition (p. 121-122).

From such sayings, we understand that, in abstraction, there are distinct levels (layers) that are neither similar to real nor to theoretical knowledge. This way of knowing through simulation has already been discussed by Valente (1999) when analyzing the different types of software used in education. If, in Lévy's (2001) words, we have the perception that the user-student is a manipulator of the simulation parameters created by another, Valente (1999) makes a succinct differentiation:

> A particular phenomenon can be simulated in the computer. For that, it is enough that a model of this phenomenon is implemented in the machine. The simulation user is responsible for changing specific parameters and observing the behavior of the phenomenon, according to the assigned values. In modeling, the model of the phenomenon is created by the learner, who uses resources of a computational system to implement it. Once implemented, the learner can use it as if it were a simulation.
>
> Therefore, the difference between the simulation software and that of modeling is in who chooses the phenomenon and who develops its model. In the case of simulation, this is done a priori and provided to the learner. In the case of modeling, it is the learner who chooses the phenomenon, develops their model and implements it on the computer. In this sense, modeling requires some degree of engagement in the definition and computational representation of the phenomenon and, therefore, creates a situation quite similar to the programming activity, and the same phases of the description-execution-reflection-debug-description cycle take place (p. 95).

NUPEME-UFU research has moved in the direction that the student-user makes simulations using modeling software, as they believe in the precepts that people who create their own LOs learn best, as envisioned by the RIVED program (Prata & Nascimento, 2007; Alves, 2012, 2017).

As we follow this line, it is important to read Lima's (2021, P. 15) warning: from this perspective, each user-student "describes an abstraction taking into account the particularities that are important to them". We are aware of this criticism, but the results of the NUPEME-UFU research corroborate the ideas of Levy (2001), in which knowledge is given as the

> result of complex networks where a large number of human, biological, and technical elements interact. It is not "I" who am intelligent, but "I" with the human group of which I am a member, with my language, with a whole legacy of intellectual methods and technologies (among which, the writing). (...) The so-called intelligent subject is nothing more than one of the micro authors of a cognitive ecology that encompasses and restricts them". (...) Thinking takes place in a network in which neurons, cognitive modules, humans, educational institutions, languages, writing systems, books, and computers interconnect, transform, and translate representations (p.135).

Thus, Lévy (2001) understands knowledge by simulation as an open, plurivocal, and distributed modality. Therefore, the search for an LO authoring tool is reduced to choosing a simulation modeling software, which, according to Devlin (2012), should enable different abstractions of reality.

3 Investigation of Modeling Paradigms in the Choice of Authoring Tool

In the foundation of this article, we arrive at the modeling software in the words of Lévy (2001). In our experience of producing LOs for RIVED, we concluded that, for mathematics education, modeling was the most indicated methodology to create them (Alves & Souza Junior, 2013). According to Borshchev (2014), the modeler can adopt three different points of view when mapping the real-world system to their abstraction of the world in models: system dynamics modeling (SDM), discrete event modeling (DEM) and agent-based modeling (ABM).

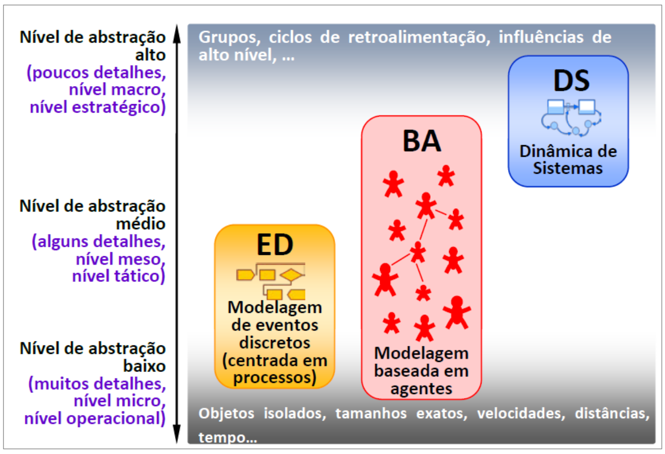
The SDM paradigm, according to Filho (2001, P. 30), "is a technique in which social, non-linear, dynamic, and complex systems can be understood and analyzed through interactions. In addition, new policies and structures can be designed to improve the behavior of the system". As for the DEM paradigm, according to Almeida (2016, p. 6), in "a system of discrete events, one or more phenomena of interest change their value, or state, at discrete points (instead of continuously) in time".

According to Emrich, Suslov, and Judex (2007), agent-based modeling (or multi-agent systems) is a generalization of cellular automata (CA) since both methods describe complex systems by definition of local interactions. In it, "the modeler describes the system from the point of view of individual objects that can interact with each other and with the environment" (Borshchev, 2014, p. 1). Fan, Bravo, and Collischonn (2016, p. 2) explain that the

> agents are autonomous components with particular behaviors that interrelate with each other and with the environment. Basically, the philosophy of an agent-based model is that the singular behavior of a single agent does not faithfully represent reality, but that the sum of all local behaviors corresponds to a good representation of the real behavior of the group. Thus, it is possible to verify that applying an agent-oriented approach to solving a problem means breaking it down into multiple autonomous components with interrelated objectives.

These three methods for simulation have a "specific range of levels of abstraction. System dynamics presupposes a high level of abstraction [...]. Discrete events support low-level abstractions. In the middle are agent-based models [...]" (Grigoryev, 2015, p. 13), Figure 1.

Figure 1: Levels of Abstraction of Modeling Paradigms



Source: Grigoryev (2015, p. 13)

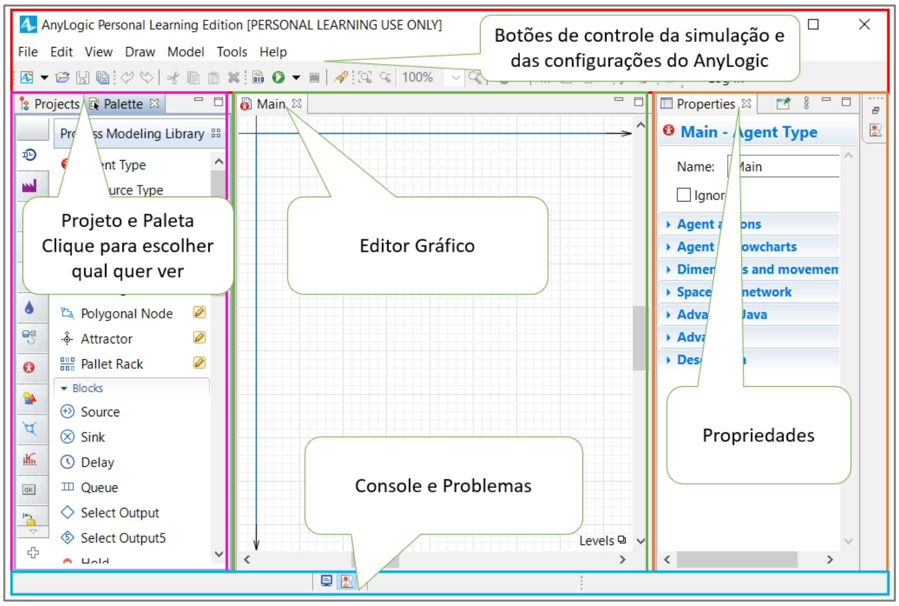
From these abstractions, the argument of the convergence of modeling paradigms was used to choose a single authoring tool that student-users can, in a single environment, follow three paths to abstract models from the real world (Jenkins, 2015). To Anylogic (2022a, p. 1), this convergence takes place in the *AnyLogic* because "it is the only software of multi-method [hybrid] simulation modeling available on the market. The Personal Learning Edition (PLE) license is provided at no charge for educational and self-learning purposes." Emrich, Suslov, and Judex (2007) explain that

> it is a programming and simulation environment, aimed mainly at modeling hybrid systems, based on JAVA. It allows the user to combine different techniques and approaches, such as differential equations, discrete events, and agent-based systems. These combination possibilities make it a very interesting tool for simulating complex systems (p. 2).

The software can be installed on *Microsoft Windows*, *Apple Mac*, and Linux operational systems. Hardware with 1.5 Gigabyte (GB) free disk space, random access memory (RAM) greater than 4 GB, and a processor with at least two cores are recommended (Anylogic, 2022b, P. 1). The free version of the tool "has all the basic features and functionality of the *AnyLogic*, including support to all three modeling methods, all standard libraries, integration with GIS maps, 3D animations [...]" (Anylogic, 2022a, p. 1). The restriction is on the number of distinct types of agents that can be created. To download the software, one must fill in a form with some personal data, see Anylogic (2022a). The installation can be read in Grigoryev (2015).

AnyLogic is composed of five regions (Figure 2). The first, from top to bottom, brings the control buttons for the software simulation and settings. Next, from left to right, are the *Projects* and *Palette* tabs. At the center is the Graphic Editor, and on the right is the *Properties* tab. Below is the tab that returns errors in case they occur during the simulation.

Figure 2: Work area of *AnyLogic*



Source: Research Data

Grigoryev (2015, p. 26) explains that the *Projects* tab "allows you to access models that are open in the workspace, and the project tree helps you easily navigate them." In the *Palette* is the "list of objects grouped in palettes. To add an element to your model, drag it from the *Palette* for the graphic editor" (Grigoryev, 2015, p. 26, emphasis added). The Graphic Editor "allows you to edit the agent diagram, and you can add model elements by dragging them from the *Palette* for the diagram and placing them on the editor's canvas. The elements placed in *blue frame* will appear in the model window when it is executed" (Grigoryev, 2015, p. 26, emphasis added). The *Properties* tab "allows you to view and modify the properties of the selected item" (Grigoryev, 2015, p. 26).

The creation logic resembles other softwares applied to mathematics education, for example *Scratch* (Marji, 2014) and/or *App Inventor* (Alves & Pereira, 2016). In them, the idea is to drag the code blocks from the *Palette* tab, drop them in the graphic editors and configure them in the *Properties* tab. The differential, perhaps, is the amount of programming blocks that the *AnyLogic* has and the possibility to program directly in Java language, using object-oriented programming. From that perspective, our method for using it was to create agents in *Projects*; in *Palette*, choose the programming blocks necessary for the simulation to work; and, finally, configure them in *Properties*.

After choosing the tool, we started problematizing a real-world situation for constructing an LO. The following planning documents were related to the creations of the RIVED project: Instructional design; General design; Storyboards; Teacher and Student Guide (Rodrigues, 2006). Such documents were not constructed, as they were not within the scope of this work. We summarized the problem situation and built the simulator so that student-users could reflect on a question we left on the simulator's homepage (although research by NUPEME-UFU points out that teachers and students use them in classroom practice based on their own questions).

4 Problematizing the Use of the Authoring Tool

The problematization that gave rise to the simulation learning object is a modification of the works carried out by Borshchev (2014) and Anylogic (2022d). The problem situation was: as a viral, contagious, but non-lethal disease progresses in a community, the level of concern of an infected person increases when this variable reaches a threshold value. The patient seeks treatment in a hospital that has limited capacity. We suggest that the student-user investigate the following problem: *How does the limitation of hospital care affect the dynamics of the disease and the level of concern of those infected?*

The construction of the simulation was divided into three tasks, each with a different modeling paradigm: the modeling of the population's behavior was performed using the ABM; the number of vacancies in a hospital was calculated using the DEM; and, finally, the variation in the infected person's level of concern about whether or not to go to the hospital was performed using the SDM.

In the first, we created a population of agents, an abstraction of the population to be modeled. The initial number is 400 people, seven of whom are infected. These people (agents) stay in a rectangular area of 4,000 square meters (m^2), that is, 100m width and 40 height. In this region, we assume that people are in contact with each other, connected, if they are ten meters away, thus forming a network of interactions. This information is needed to configure the *AnyLogic*. The computational programming for people's contamination is carried out based on the non-linear ordinary differential equations (ODE) model proposed by Kermack and McKendrick (1927):

$$S'(t) = -\beta S(t)I(t)$$

$$I'(t) = \beta S(t)I(t) - \gamma I(t)$$

$$R'(t) = \gamma I(t)$$

This model became known as SIR, in which S(t) denotes the number of susceptibles at time t, I(t) the number of infected and R(t) the number of recovered. The variables β and γ

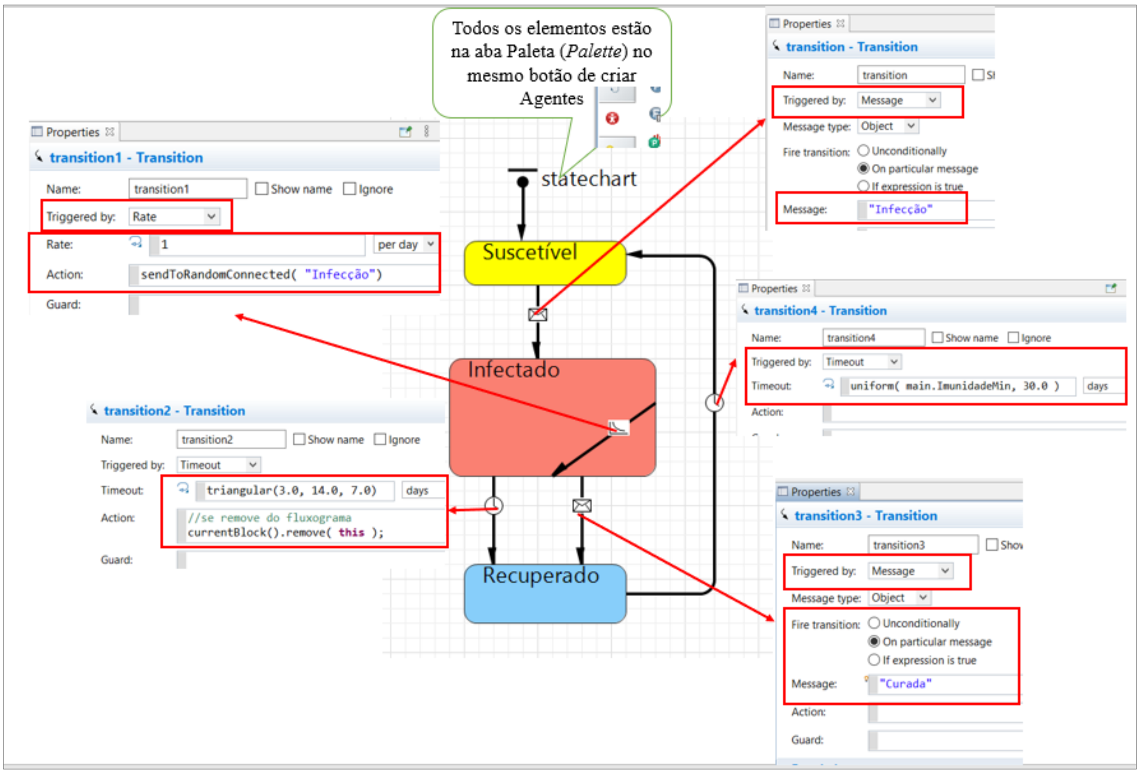
represent the infection and recovery rates, respectively, both greater than zero (Emrich, Suslov, & Judex, 2007). So, we have three states and the transitions between them.

The behavior in the ABM starts with 393 people in the *Susceptible* status, who can be infected by the virus. Disease transmission is modeled by the "Infection" message sent from one person to another. Having received such a message, the individual becomes infected. This subject sends, daily, an "Infection" message to the people who are in the interaction network. When that other one receives the message and is in the *Susceptible* status, that person goes to the *Infected* status and starts infecting other people as well.

The individual leaves the *Infected* status and enters the *Recovered* status in two ways: by time of infection or by treatment. We modeled the first through a probabilistic distribution of the triangular type, having the interval [3.0, 14.0] as the base of the triangle and the point (7.0, 1.0) outside this base, with 7.0 being the mode value in the interval. At the software, we configure the triangular function (3.0, 14.0, 7.0). The software draws the time of infection in the range from 3.0 to 14.0 days, and the value that will be repeated the most will be 7.0. The unit of time is in days. At the *AnyLogic*, there are 39 types of distribution for the draw. In the second form, the hospital sends a message that the person is "Cured", showing that they have recovered.

This disease gives immunity for a while, i.e., after a random period, the agent leaves the *Recovered* status and returns to the *Susceptible* status. This time is drawn based on a uniform probability distribution over the interval [MinImmunity, 30). With the software, we encode the *Timeout: uniform (main.MinImmunity, 30.0)*. In the programming interface (agent *Main*) there is a parameter that is the minimum immunity of the person (MinImmunity), which was set to start at five but can be changed. If the user does not change it, a number of days is drawn, which can take any value between 5 and 29, interval [5, 30). They leave the *Recovered* status and return to the *Susceptible* status. Figure 3 shows how agents behaved in the *AnyLogic*.

Figure 3: Agents' Behaviors

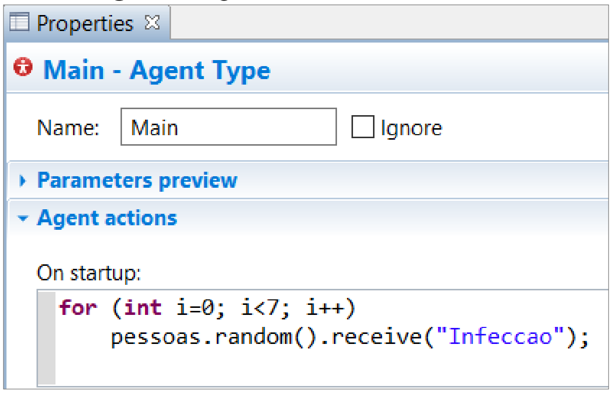


Source: Research Data

To visualize each occurring status, we create circles with different colors, representing people. In addition, we produce a boolean variable "BeingTreated", that tells us if it is (*true*) or not (*false*) in the hospital undergoing treatment. Thus, we build on *AnyLogic* a yellow circle to represent the *Susceptible* status, a red circle to represent the Infected state, and a blue circle to represent the *Recovered* status. Finally, a green rectangle representing the infected person, but undergoing treatment.

The number of individuals that start infected is defined in the agent *Main*, in the *Properties* tab, which appear in the agent configuration windows. In *Agent actions*, in the field *On startup*, we write a repeating loop *for*, starting in *1* and ending in *7*. So the software draws seven of the 400 people to stay in the *Infected* status on the *AnyLogic*: *pessoas.random().receive("Infecção" [infection])*, Figure 4.

Figure 4: Agents that start contaminated



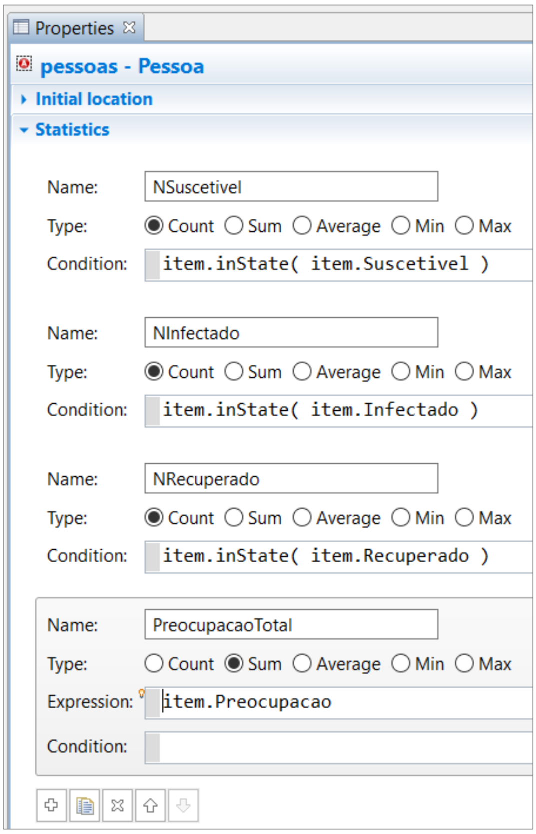
Source: Research Data

With that, we already have the agent-based simulation finished, so what is missing is to obtain statistical data from it. In this context, in *AnyLogic*, it is necessary to search, inside *Main*, the *Agents* class, because in it, we have the Object *pessoas* [people]. In it, we double-click to select it; in the *Properties* tab, we choose the window *Statistics*; we click on the plus symbol (+) to add four fields from which we will obtain statistical data: *Nsuscetivel* [*Nsusceptible*], *Ninfectado [Ninfected], Nrecuperado [Nrecovered],* and *PreocupaçaoTotal [TotalConcern]*. These variables accumulate, respectively, the number of people in the *Susceptible*, *Infected*, *Recovered* status and the level of concern about the disease. Figure 5 shows the settings in the software.

As the ABM is already finalized, graphs are created to show the number of individuals in each status. In the *Palette* tab, in the eleventh code block *Analyses*, there are several chart builders. We drag and drop the *Time Plot* (time graph) block in the Graph Editor, then, in the *Properties* tab, we insert "grafico_estados" [graph_status]. In the *Scale* window, we configure the axis of the abscissas, and in the *Time window*, insert 120 days and choose *model time units* to be the axis unit. On the ordinate axis, we write to start at zero and end at 400, with *Vertical scale* marked on *Fixed*. For that axis, in the *Data* window, the functions to be plotted are: *Suscetivel [Susceptible]*, *Infectado [Infected[*, and *Recuperado [Recovered]*, and the colors are *gold*, *red*, and *dodgerBlue*, respectively, as shown in Figure 6.

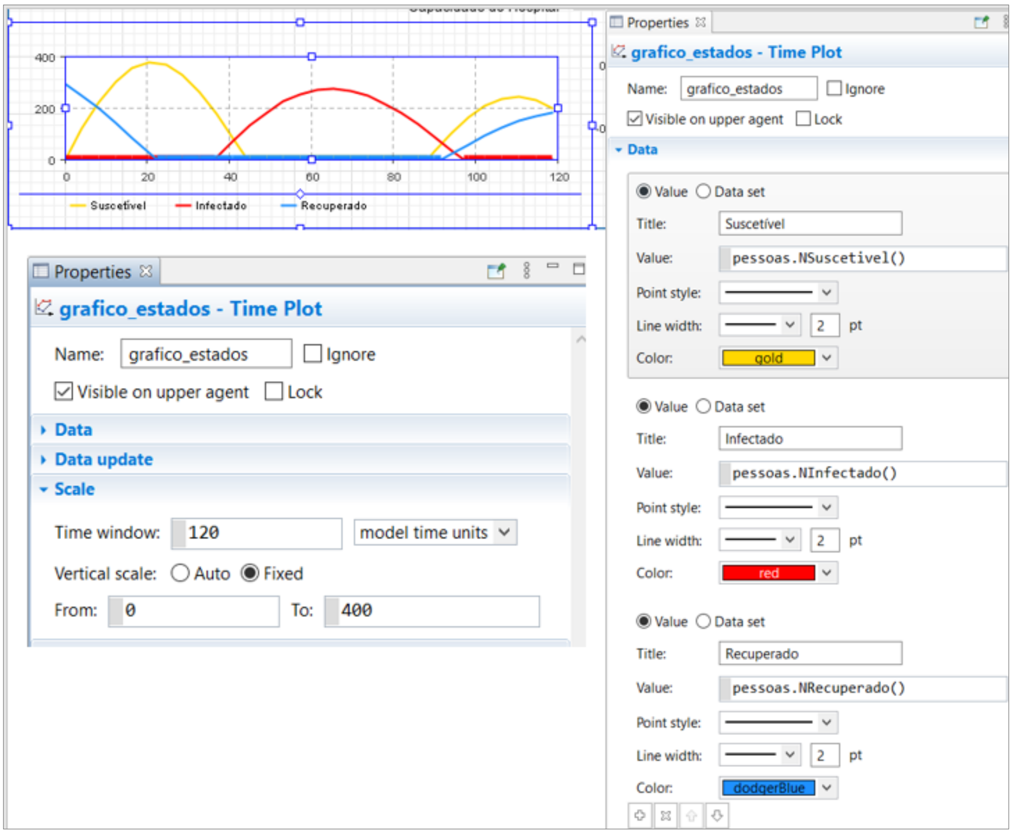
With the representation of people and the infection behaviors built, we started to create the representation of the hospital with the discrete event modeling. We select *Main* agent to add the code blocks that symbolize it. This hospital is represented by a waiting line and the person's treatment time in that place. In the hypothesis of modeling, once in treatment, they will be cured.

Figure 5: Configuring simulation statistics



Source: Research Data

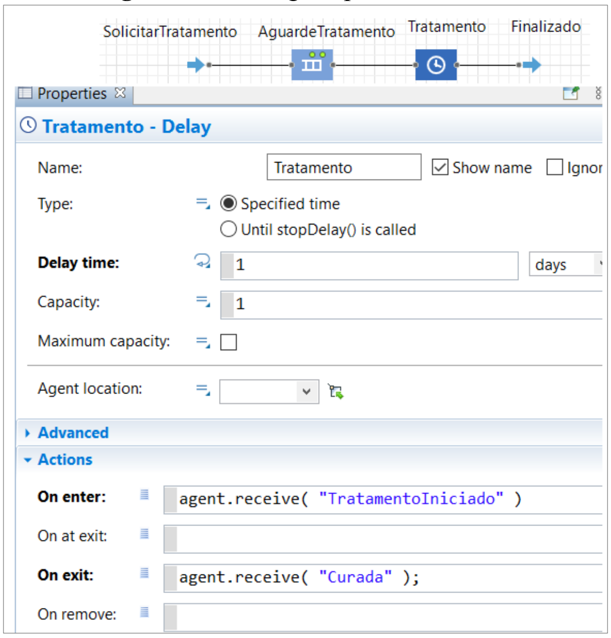
Figure 6: The graphical representation of the ABM



Source: Research Data

In the *Palette* tab, in the first code block, *Process Modeling Library*, we drag and drop four blocks: *Enter; Queue; Delay* and *Exit*, which we named, respectively: *SolicitarTratamento [RequestTreatment]*; *AguardeTratamento [WaitTreatment]*; *Treatment [Tratamento]* e *Finished [Finalizado]*. In *Treatment*, in the *Properties* tab, we insert *1 days* in the field *Delay time*, and in *Capacity* (vacant places in the hospital), we defined 1, initially. In the *Actions* window, in the field *On enter*, we write the code *agent.receive("TratamentoIniciado [TreatmentInitiated])* and *On exit*, we add agent.receive*("Curada"[Cured])*, Figure 7.

Figure 7: Building hospital abstraction



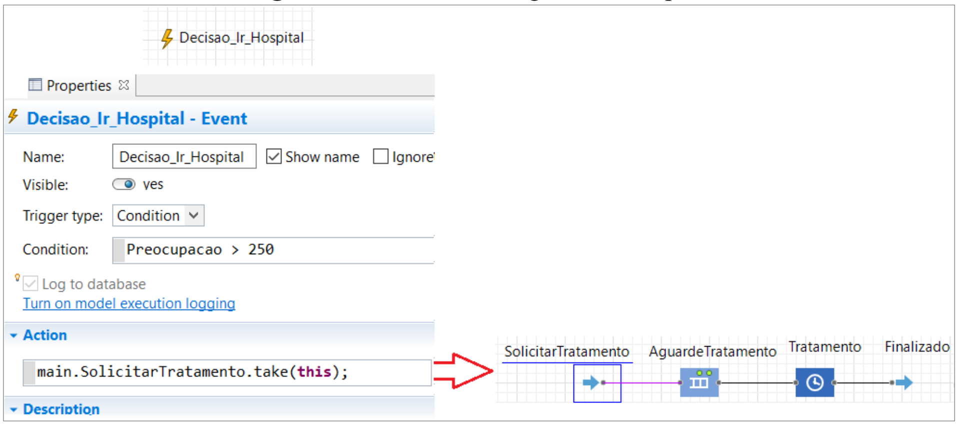
Source: Research Data

With the representation of the hospital built, it was necessary to model people's level of concern to decide whether to go to the hospital or not. This process takes place with SDM. To do so, we go back to the Projects tab and double-click on the Person agent to select it. Then, in the *Palette*, in the eighth code block, *Agent*, we drag and drop a block called in the Graph Editor *Event*, named *Decisao_Ir_Hospital* [*Decision_Go_Hospital]*.

Event is responsible for monitoring a compartment called *Preocupacao* [*Concern]* and applying a condition to go to the hospital and start treatment. For this, in the *Properties* tab, we choose in *Trigger type*, *Condition*. We define that if *Concern > 250*, then infected people go to the hospital. In the *Action* window, we write *main.RequestTreatement.take(this);*. The *main.SolicitarTratamento* refers to the person's admission to hospital, which we put in the *Main* agent, with discrete event modeling. This is where the interaction between the different types of modeling takes place, as shown in Figure 8.

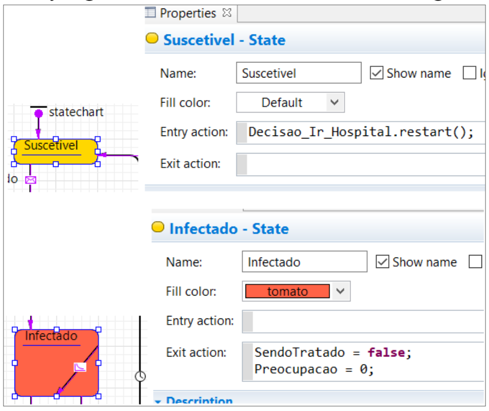
In the Person agent, we changed two behavioral statuses and included the SDM elements in the ABM to add up the person's level of concern in relation to the disease. The first status to be modified is *Suscetivel [Susceptible]*, so we add the command *Decisao_Ir_Hospital.restart();* in the field *Entry action* so that, every time the person returns to that status, the decision to go to the hospital is restarted, considering that they can be ill again. In the *Infectado [Infected]* status, we modify the field *Exit action*, placing the codes*: SendoTratado [BeingTreated[=false*; *Preocupacao [Concern] = 0;*. These two configurations are necessary because when exiting the *Recovered* status, there is a random immunity time to the disease (Figure 9).

Figure 8: The decision to go to the hospital



Source: Research Data

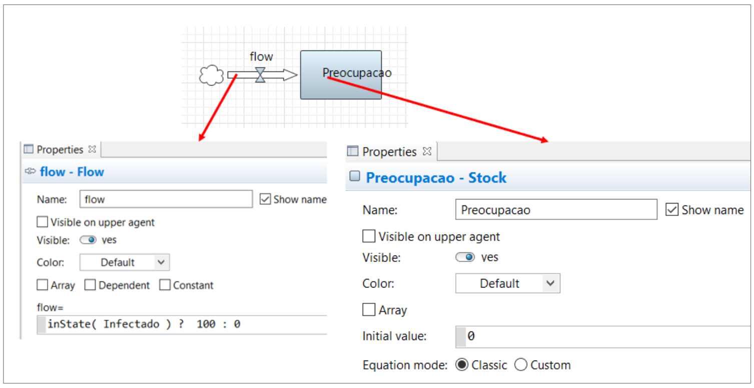
Figure 9: Modifying the ABM to decide whether to go to the hospital



Source: Research Data

With the changes made, we started the SDM. In the Palette tab, in the seventh code block *System Dynamics*, we drag and drop *Stock* and the *Flow* block. In the compartment, in *Properties*, in the field *Name*, we change it to *Preocupacao [Concern]* and, in the field *Initial value*, we put zero. Then, we connect the Flow to the compartment *Concern*. In *Properties*, in *Flow*, we write the condition: *inState(Infectado) ? 100: 0*, having the function of stating that, for each day that the person is sick, 100 units are added to their *Concern* (Compartment). Figure 10 shows these configurations.

Figure 10: Configuring whether to go to the hospital

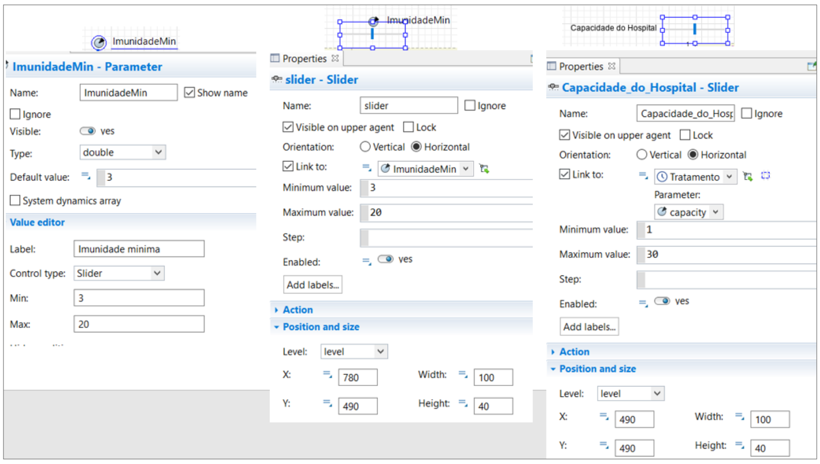


Source: Research Data

To finish modeling, we go back to the *Projects* tab and double-click with the left button on *Main* to select it. We will add two parameters: the hospital's capacity to care and the people's minimum immunity to the disease. To each, we associate a *Slider* control, for the student-user to change to the desired value within a predetermined range. In *Palette*, we drag and drop a parameter and two sliders. The second parameter is associated with the DEM, which is the automatically created capacity of the hospital.

We select the parameter that was added in the graphical part, with a click of the left mouse button, and in the *Properties* tab, in the field *Name*, we write *ImunidadeMin [MinImmunity[* and check *Show name*. In the field *Type*, we choose *double* (number in the set of the real numbers). We set *Default value* in 3; in *Label*, we add *imunidade mínima [minimum immunity]*; in *Control type*, we choose *Slider*, and for the minimum and maximum values, we write 3 and 20 days, respectively. Figure 11 presents such configurations.

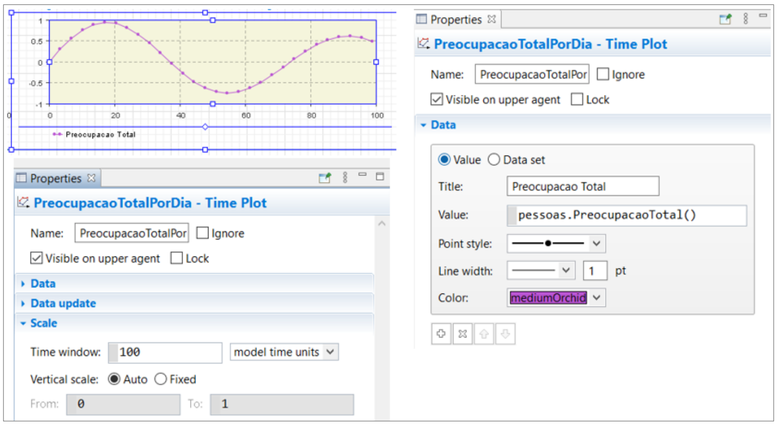
Figure 11: Configuring the parameters



Source: Research Data

Finally, we put a graph in the *Main* agent to show the evolution of people's level of concern about the disease. In the *Palette* tab, in the code block *Analyses*, we drag and drop a *Time Plot* block into the Graph Editor. In *Properties*, we change the name to *PreocupacaoTotalPorDia [TotalConcernPerDay]*; in the *Data* window, we add the title *PreocupacaoTotal [Total Concern]*, and in the value, we put *pessoas.PreocupacaoTotal() [people.TotalConcern()[*. In the *Time* window, we write 100 units of time and in the vertical scale we leave it automatic (Figure 12).

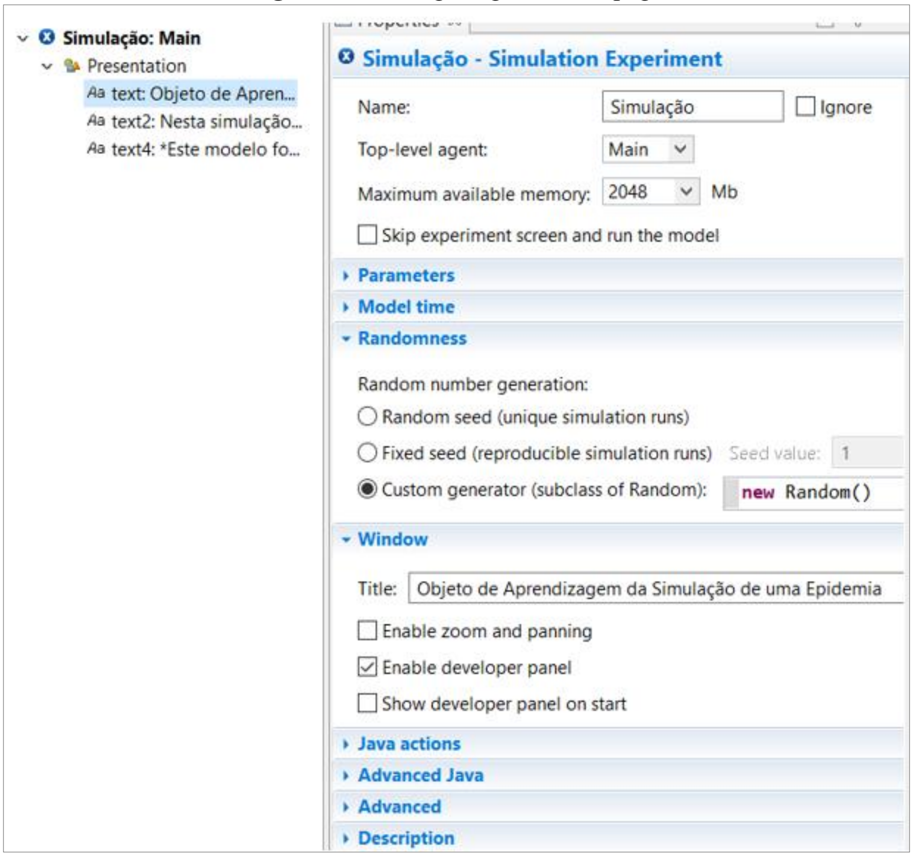
Figure 12: Graphical representation of people's concern



Source: Research Data

With these configurations, the LO is finished, all that remains is to add and configure a homepage to explain to the student-user how the object works and how the assessment takes place. This homepage is configured in a class called *Simulation Experiment*, located on the *Projects* tab, selected by double-clicking the left mouse button. In the *Properties* tab, we change the name to *Simulação [Simulation]* and the memory that the computer allows for simulation, we put 2048 MB; in the *Randomness* window, we choose *Custom generator*. In *Window*, we change the title to *Objeto de Aprendizagem de Simulação de uma Pandemia [Learning Object of the Simulation of an Epidemic]* and check the option of *Enable developer panel* (Figure 13).

Figure 13: Configuring the Homepage



Source: Research Data

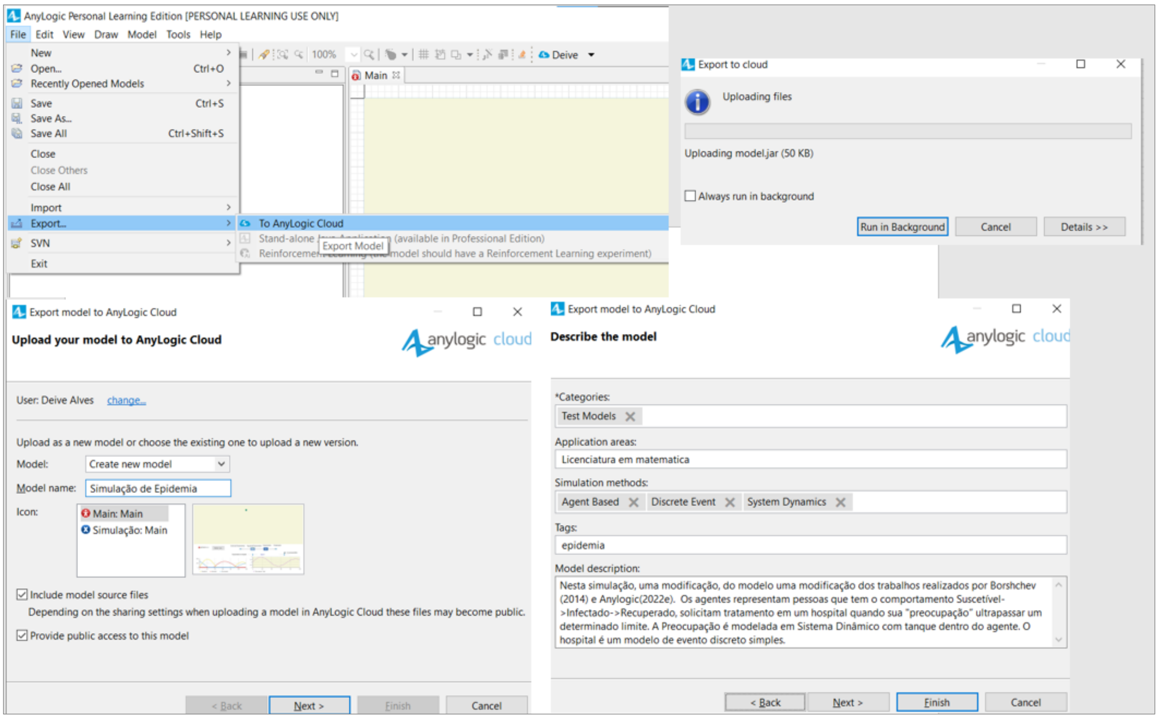
We added text fields to write the title, one with the summary of the simulation history and another with the observation question of the simulation. To run the application for the first time, two tasks must be performed. The first is to look for errors, which *AnyLogic* does through the computer keyboard by clicking on *F7* to compile the object. The second is to run the application by clicking on *F5* key. This is how to test and use the simulation for those who have installed the software on the computer.

5 Findings and Conclusions

In the free version, we cannot export an executable version of the object. The only option, besides having the software installed on the computer, is to export to the repository of *AnyLogic*. To do this, we must have an account on *AnyLogic Cloud*. We pressed the button that leads to Gmail accounts, but it did not work. So, we used Hotmail, the email service from Microsoft. The registration generated a confirmation message that went directly to the spam box.

With the registration done to export the learning object, we must go to *File*, *Export*, and *To AnyLogic Cloud*. After a few seconds, a window will open, asking for the name of the simulation. By clicking *Next>*, some other metadata must be filled in: *Categories*, *Application* *areas*, *Simulation Methods*, *Tags*, and *Model description* (Figure 14).

Figure 14: Exporting the LO



Source: Research Data

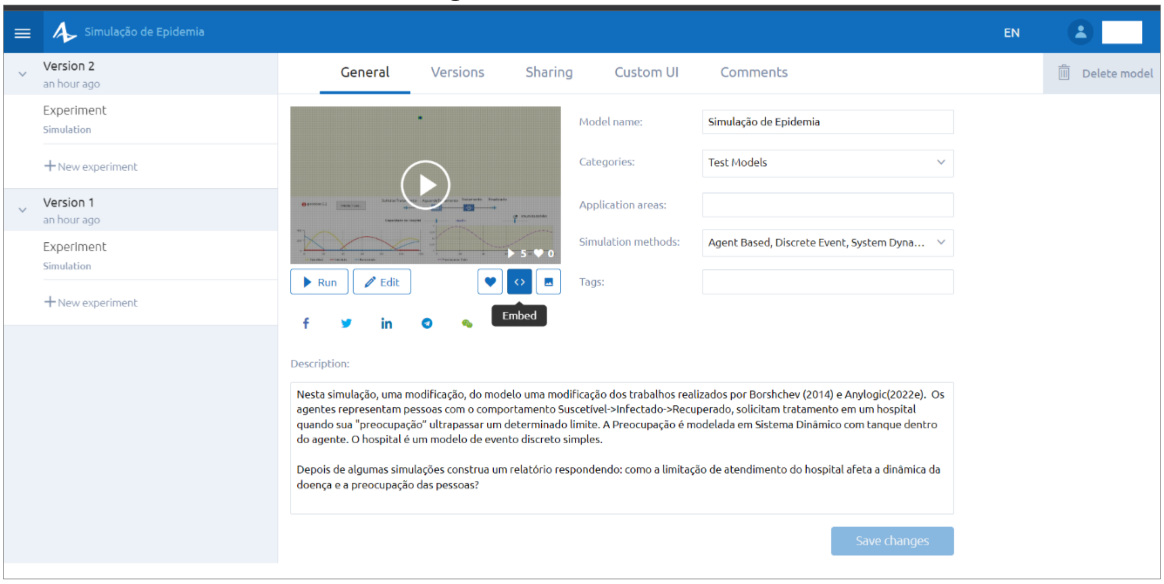
The author can edit the metadata information on the site (Anylogic, 2022c). When logging in, we are sent to a webpage containing the author and public models. When hovering the mouse over the simulation, the description and the symbol of a trash can appear on the left side to delete the model. If we click on the simulation, we are taken to another page dedicated to experimenting with the simulator.

To edit it, just click on the model name, located on the upper left side of the page, next to the *AnyLogic* logo. There, we see: *General*, *Versions*, *Sharing*, *Custom UI*, *Comments* and *Delete model*. The *General*, *Sharing*, and *Comments* tabs are the most used (Figure 15).

We verified that the metadata filled in without being in the pre-established list does not appear on the webpage. The greater and lesser symbol, known as *Embed* using HTML (hypertext markup language) tag *<iframe> </iframe>*, allows inclusion and visualization in another HTML document, helping with the interoperability of the object. To test it, we included the simulation in a blog of *Google Blogger* (Figure 16). All buttons worked correctly in the blog, but the simulation was embedded without the homepage, present in the two other environments.

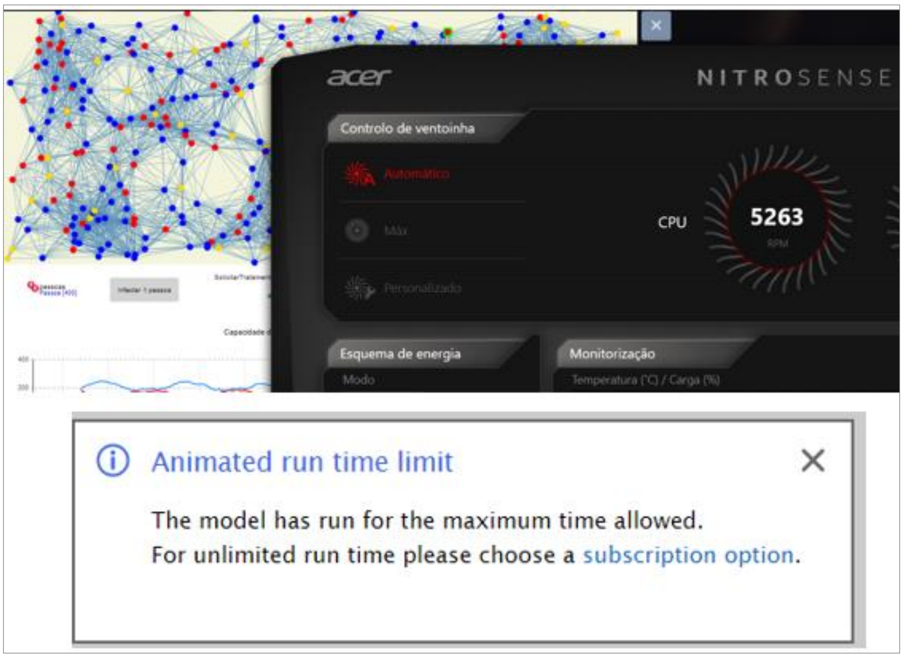
In the three environments: installed software, *AnyLogic* repository and in the blog, at the time the simulation was running, the laptop temperature increased, changing the speed of the fans, which, in monitoring the *Acer Nitrosense*, raised from 2,800 to 5,263 rpm (Figure 16). Apart from this heating, after the online simulation had run for around five minutes, the following message appeared: *Animated runtime limit*. This did not happen in the simulation with the software installed. The message disappears when we refresh the webpage, allowing us to start the simulation again (Figure 16).

Figure 15: LO online edition



Source: Research Data

Figure 16: Tests with the online simulation



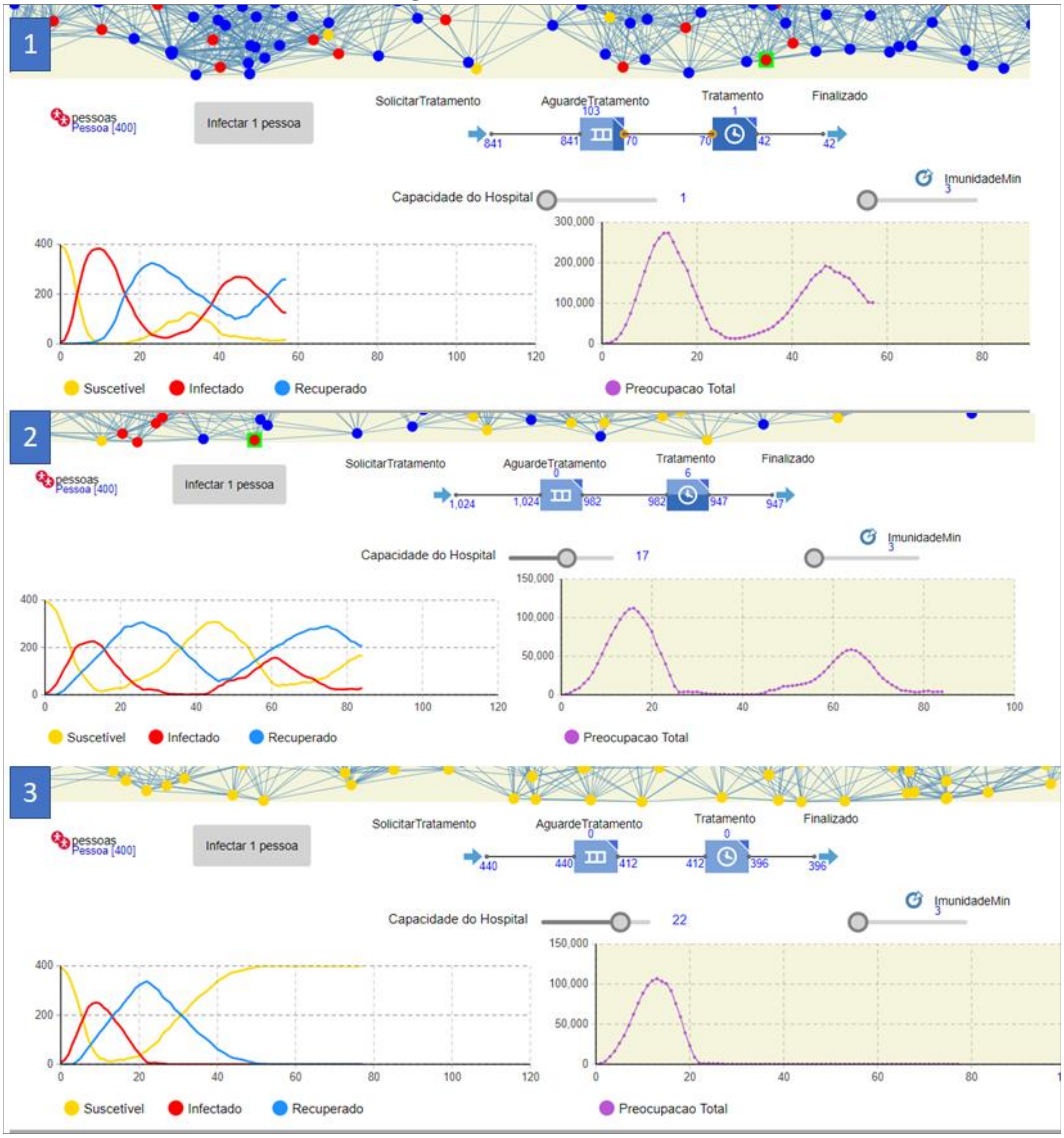
Source: Research Data

The simulation in all environments had the same behavior, maintaining immunity and changing the service capacity. We found that people's level of concern decreases and the disease tends to disappear. This factor can be seen in Figure 17.

In the simulation in Figure 17 (1), with immunity defined in three days, we set the hospital's capacity to serve one person. We verified that there is an oscillation both in the statuses and in the level of concern. We altered to 17 beds in the hospital (Figure 17 [2]) so that there was a reduction in the level of concern, and in 22 beds (Figure 17 [3]), after the first wave, there were no more waves of the disease, and the concern level was reduced to zero.

Thus, we can conclude that the authoring tool *AnyLogic* is an intelligence technology that channels a culture of convergence of three ways of knowing the world, with different levels of abstractions: low (DEM), medium (ABM) and high (SDM) (Grigoryev, 2015; Devlin, 2012). There is no doubt that this methodology facilitates the work of building LOs. However, we must be careful so that the mathematics is not hidden in code blocks, or even from the perspective of the ABM and the rules of behavior of the agents.

Figure 17: Simulator screens



Source: Research Data

An answer to such precautions was presented in research by Costa (2017), who worked with elementary school nine graders, investigating the deforestation of the Amazon with the software NetLogo, a well-known tool for working with agent-based modeling. However, the research used the simulation perspective to obtain data that would allow the group to discuss mathematical concepts of that teaching stage. Another possibility is in Emrich, Suslov, and Judex's (2007) work, when validating the ABM using differential equations.

The fact is that we are at the beginning of these discussions in mathematics education, as highlighted by Devlin (2012). However, the Covid-19 pandemic has shown us that we must progress with teachers' and students' digital authorships. In that sense, the *AnyLogic* has shown potential for developing mathematical and computational heuristic learning.

This tool showed to be possible to model the level of concern of infected individuals using characteristics of an epidemiological model in a community. This methodology can be motivating for mathematics degree students in the sense of interpreting an epidemic, which is a present issue. Furthermore, due to the ease of use of the software, degree students may become multipliers of this technology for basic education students.

References

Almeida, J. F. F. (2016). *Simulação por eventos Discretos: teoria & prática*. (1a. ed.). Bambuí, MG: IFMG.

Alves, D. B. & Souza Júnior, A. J. (2013). O papel da modelagem matemática na construção de Objetos de Aprendizagem. In: *Anais do 11º Encontro Nacional de Educação Matemática* (pp. 56-71). Curitiba, PR.

Alves, D. B. (2012). *O Processo de Autoria na Cultura Digital: a perspectiva dos licenciandos de Matemática*. 2012. 172f. Dissertação (Mestrado em Educação). Universidade Federal de Uberlândia. Uberlândia, MG.

Alves, D. B. (2017). *Modelagem Matemática no Contexto da Cultura Digital: uma perspectiva de educação pela pesquisa no curso de técnico em Meio Ambiente integrado ao Ensino Médio*. 2017. 281f. Tese (Doutorado em Educação). Universidade Federal de Uberlândia. Uberlândia, MG.

Alves, F. J. C. & Pereira, C. C. M. (2016). *Aplicativos para o ensino de matemática em App Inventor*. (1a ed.) São Paulo, SP: CRV.

Alves, D. B.; Souza Júnior, A. J.; Meyer, J. F. C. A.; Barbosa, F. C.; Alexandre, M. L.; Menezes, D. C. & Marin, D. (2015). Collective Production with Mathematical Modelling in Digital Culture. In: G. A. Stillman; W. Blum; , M. S. Biembengut (Org.). *Mathematical Modelling in Education Research and Practice*. (1a ed. pp. 465-474). Melbourne, AU: Springer.

AnyLogic Personal Learning Edition [Anylogic] (2022a). *Software de Simulação Gratuito para Educação*.

AnyLogic Personal Learning Edition [Anylogic] (2022b). *System requirements*.

AnyLogic Personal Learning Edition [Anylogic] (2022c). *Anylogic Cloud: login*.

AnyLogic Personal Learning Edition [Anylogic] (2022d). *Epidemic and Clinic with Accumulating Concern*.

Borshchev, A. (2014). Multi-method modeling: AnyLogic. In: S. Brailsford; L. Churilov & B. Dangerfield (Ed.). *Discrete-Event Simulation and System Dynamics for Management Decision Making*. (1a. ed., pp. 248-279). Nova Jersey, USA: Wiley.

Braga, J. & Menezes, L. (2014). Introdução aos Objetos de Aprendizagem. In: J. Braga (Org.). *Objetos de aprendizagem, volume 1: introdução e fundamentos*. (1a ed., pp. 19-40). Santo André, SP: UFABC.

Brasil. Ministério da Educação. (2006). *Ministério premia 44 objetos de aprendizagem*. Brasília, DF.

Brasil. Ministério da Educação. (2007). *Concurso MEC-SEED/PNUD 3ª Edição/2007: Prêmio Concurso Objetos de Aprendizagem*. Brasília, DF.

Cintra, V. P. (2010). *O projeto rived: a aprendizagem dos participantes*. 2010. 106f. Dissertação (Mestrado em Educação Matemática). Universidade Estadual Paulista. Rio Claro, SP.

Costa, D. (2017). *Educação Ambiental com modelagem matemática no ensino fundamental*. 2017. 210f. Dissertação (Mestrado em Desenvolvimento Regional). Universidade Tecnológica Federal do Paraná. Pato Branco, PR.

Deleuze, G. & Parnet, C. (1998). *Diálogos*. Tradução de E. A. Ribeiro. (1a. ed.). São Paulo, SP: Escuta.

Devlin, K. (2012). *Introduction to Mathematical Thinking*. (1a, ed.). Palo Alto, USA: Stanford.

Emrich, Š.; Suslov, S. & Judex, F. (2007). Fully Agent Based Modellings of Epidemic Spread Using Anylogic. In: *Anais da 6º Federation of European Simulation Societies [EUROSIM] Congress on Modelling and Simulation* (pp. 1-7). Ljubljana, SI.

Fan, F. M.; Bravo, J. M. & Collischonn, W. (2016). *Modelagem baseada em agentes para a simulação da dispersão de poluentes em cursos d'água. Revista Engenharia Sanitária e Ambiental*, 21(04), 739-746.

Filho, J. B. (2001). *Simulação Dinâmica de Modelos operacionais com enfoque aplicado à Engenharia de Projetos*. 2001. 134f. Dissertação (Mestrado em Engenharia de Produção). Universidade Federal de Santa Catarina. Florianópolis, SC.

Fonseca, D. S. (2009). *Ambiente de Aprendizagem na Escola Norturna: ensinando e aprendendo Matemática com Tecnologias da Informação e Comunicação*. 2009. 123f. Dissertação (Mestrado em Educação). Universidade Federal de Uberlândia. Uberlândia, MG.

Foucault, M. (2004). Tecnologias de si. *Revista do Núcleo de Sociabilidade Libertária [NU-SOL]*, 6(2004), 321-360.

Guattari, E. & Rolnik, S. (1986). *Micropolítica: cartografias do desejo*. (10a. ed.). Petrópolis, RJ: Vozes.

Grigoryev, I. (2015). *Anylogic 7 em três dias: um curso rápido de modelagem de simulação*. (2a, ed.). South Carolina, USA: Createspace Independent.

Jenkins, H. (2015). *Cultura da convergência*. Tradução de S. L. Alexandria. (2a. ed.). São Paulo, SP: Goya.

Kermack, W. O.; McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Royal College of Physicians*, 1(1), 700–721.

Lévy, P. (2001). *As Tecnologias da Inteligência: o futuro do pensamento na era da informática*. Tradução de C. I. C. (10a. ed.). Rio de Janeiro, RJ: Editora 34.

Lima, W. D. (2021). *Pensamento computacional: abstrações*. (1a. ed.). Gama, DF: UNICEPLAC.

Marji, M. (2014). *Aprenda a Programar com Scratch: uma introdução visual à programação com jogos, arte, ciência e matemática*. (1a. ed.). São Paulo, SP: Novatec.

Moura, E. M.; Souza Júnior, A. J.; Carvalho, A. M. & Alves, D. B. (2018). *Building a Photovoltaic Charger at School. PBL for the next geration: Blending active learning, techonology and social justice*, 1, 1-12.

Moura, É. M.; Alves, D. B. & Souza Júnior, A. J. (2013). *Mathematics Education in a Digital Culture. Journal of Mathematical Modelling and Application*, *1*(5), 32-41.

Nunes. C. A. (2010). *Educação Matemática: processos formativos e a sua interface com as mídias*. 2010. 166f. Dissertação (Mestrado em Educação). Universidade Federal de Uberlândia. Uberlândia, MG.

Pimentel, M.; Sampaio, F. F. & Santos, E. O. (Org.). *Informática na Educambientes de aprendizagem, objetos de aprendizagem e empreendedorismo*. (1a. ed.). Porto Alegre: Sociedade Brasileira de Computação.

Prata, C. L. & Nascimento, A. C. A. (Org.) (2007). *Objetos de aprendizagem: uma proposta de recurso pedagógico*. (1a. ed.). Brasília, DF: MEC.

Rebouças, A. D.; Maia, D. L. & Scaico, P. D. (2021). Objetos de Aprendizagem: da Definição ao Desenvolvimento, Passando pela Sala de Aula. In: M. Pimentel; F. F. Sampaio & E. O. Santos (Org.). *Informática na Educação: ambientes de aprendizagem, objetos de aprendizagem e empreendedorismo*. (1a ed.). Porto Alegre, RS: Sociedade Brasileira de Computação.

Rodrigues, A (2006). *Produção coletiva de objeto de aprendizagem: o diálogo na universidade e na escola*. 2006. 113 f. Dissertação (Mestrado em Educação). Universidade Federal de Uberlândia, Uberlândia, MG.

Santos, F. D.; Lopes, M. M. F.; Fagundes, A. G. F.; Falavigna, G. & Andreatta-da-costa, L. (2022). Objetos de Aprendizagem Aplicados no Ensino Fundamental em Época de Pandemia: Relatos de Experiências com Dispositivos Móveis em Sala de Aula Remota. *Revista Brasileira de Ensino de Ciências e Matemática*, 5, 21-34.

Silva, R. M. G.; Fernandes, M. A.; Lopes, C. R. & Souza Júnior, A. J. (2007). Construção de Objetos de Aprendizagem: fundamentos de uma proposta. In: A. J. Souza Júnior, M. A. Fernandes, C. R. Lopes, R. M. G. Silva (Org.). *Informática na educação: elaboração de objetos de aprendizagem*. (1a ed. pp.65-88). Uberlândia, MG: EDUFU.

Souza, V. A. (2017). *O Trabalho Educativo com Software de Geometria Dinâmica no Quinto Ano do Ensino Fundamental*. 2017. 183f. Dissertação (Mestrado em Educação). Universidade Federal de Uberlândia. Uberlândia, MG.

Souza Júnior, A. J. & Lopes, C. R. (2007). Saberes Docentes e o Desenvolvimento de Objetos de Aprendizagem. In: C. L. Prata & A. C. A. Nascimento (Org.). *Objetos de Aprendizagem: Uma proposta de recurso pedagógico*. (1a ed. pp. 7-15). Brasília, DF: MEC.

Souza Júnior, A. J. & Moura, É. M. (2010). Constituição de Um Ambiente Virtual de Aprendizagem Com Objetos de Aprendizagem. In: V. Marim & C. C. Oliveira (Org). *Educação Matemática: contexto e práticas docentes*. (1a ed. pp. 179-192). Campinas, SP: Alínea.

Souza Júnior, A. J.; Fernandes, M. A.; Lopes, C. R. & Silva, R. M. G. (Org.) (2007). *Informática na Educação: elaboração de objetos de aprendizagem*. (1a. ed.). Uberlândia, MG: EDUFU.

Souza Júnior, A. J.; Fernandes, M. A.; Lopes, C. R. & Silva, R. M. G. (Org.) (2010). *Objetos de Aprendizagem: Aspectos Conceituais Empíricos e Metodológicos*. (1a. ed.). Uberlândia, MG: EDUFU.

Souza Júnior, A. J.; Rodrigues, A. & Lopes, C. R. (2007). Objeto de Aprendizagem na Escola Pública. In: A. J. Souza Júnior; M. A. Fernandes, C. R. & Lopes, R. M. G. da Silva (Org.). *Informática na educação: elaboração de objetos de aprendizagem.* (1a ed. pp.65-88). Uberlândia, MG: EDUFU.

Tarouco, L. M. R.; Bulegon, A. M. & Ávila, B. G. (2021). Objetos de aprendizagem:uso e intencionalidade pedagógica. In: M. Pimentel; F. F. Sampaio & E. O. Santos (Org.). *Informática na Educambientes de aprendizagem, objetos de aprendizagem e empreendedorismo*. (1a ed., pp. 1-28). Porto Alegre, RS: Sociedade Brasileira de Computação.

Valente, J. A. (1999). Análise dos diferentes tipos de *software*s usados na educação. In: J. A. Valente (Org.). *O computador na sociedade do conhecimento*. (1a. ed., pp. 89-99). Campinas, SP: UNICAMP.

Wiley, D. A. (2000). *Learning object design and sequencing theory*. 2000. 120f. Tese (Doutorado em Psicologia). Brigham Young University. Provo, UT.