

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

### Deive Barbosa Alves

Universidade Federal do Norte do Tocantins  
Araguaína, TO — Brasil

✉ [deive@uft.edu.br](mailto:deive@uft.edu.br)

🆔 0000-0002-0850-7362

### Arlindo Jose de Souza Junior

Universidade Federal de Uberlândia  
Uberlândia, MG — Brasil

✉ [arlindo@ufu.br](mailto:arlindo@ufu.br)

🆔 0000-0002-5175-6129


### Rosana Sueli da Motta Jafelice


Universidade Federal de Uberlândia  
Uberlândia, MG — Brasil

✉ [rmotta@ufu.br](mailto:rmotta@ufu.br)

🆔 0000-0001-8489-3974



2238-0345 

10.37001/ripem.v13i3.3537 

Recebido • 28/12/2022

Aprovado • 21/02/2023

Publicado • 10/09/2023

Editor • Gilberto Januario 

**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 OA. Na simulação, nota-se que, dependendo do número de vagas no hospital, temos a gravidade da doença e, conseqüentemente, o nível de preocupação dos infectados compatível com a situação da epidemia.

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

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

**Abstract:** The objective, in this work, is 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 performing these three types of modeling simultaneously. 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 OA. In the simulation, it is noted 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 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.

## 1 Introdução

O Núcleo de Pesquisa em Mídias na Educação da Universidade Federal de Uberlândia (NUPEME-UFU) foi criado em 2004. Desde sua fundação uma das linhas de pesquisa são Objetos de Aprendizagem (OA). Para Rebouças, Maia e Scaico (2021, p. 4),

algumas outras variações do termo têm sido propostas para evidenciar ora o aspecto da virtualidade, ora o da ampliação para além da aprendizagem discente. Denominações como Objetos Virtuais de Aprendizagem (OVA), Objetos Digitais de Aprendizagem (ODA) e, mais recentemente, Recursos Educacionais Digitais (RED) são alguns dos exemplos criados como alternativa para a imprecisão do conceito e fundamentação de OA.

Embora esses autores pautem tais distinções na imprecisão conceitual, verifica-se que boa parte — se não todos — usam a conceituação inicial criada por Wiley (2000, p. 3): “Objetos de Aprendizagem são [...] qualquer entidade, digital ou não, que pode ser usado, reutilizado ou referenciado durante a aprendizagem mediada por tecnologia [...]”. Diversas críticas foram tecidas sobre ela ao longo dos anos, o que fez com que vários aprimoramentos e metáforas desse conceito fossem feitas, as quais podem ser lidas na coletânea organizada por Pimental, Sampaio e Santos (2021). Ainda que tais discussões assinalam um ou outro avanço, elas fazem parte da mesma perspectiva histórica: os “Objetos de Aprendizagem (OA) surgiram como uma decorrência do sucesso da programação orientada a objeto [(POO)]” (Tarouco, Bulegon & Ávila, 2021, p. 1), o que demarca um território<sup>1</sup> criado, mantido, ampliado e sob a posse do

Instituto de Engenheiros Eletrônicos e Eletricistas (Institute of Electrical and Electronic Engineers — IEEE) [que] foi criado em 1884 e possui como missão fomentar a inovação tecnológica e a excelência para o benefício da humanidade. O IEEE possui um Comitê de Padrões para a Tecnologia, o IEEE Learning Technology Standards Committee (LTSC), que possui a função de desenvolver padrões técnicos, práticas recomendadas e guias para a tecnologia da aprendizagem que sejam internacionalmente creditados (Braga & Menezes, 2014, p. 21).

Nessa mesma perspectiva estão os registros das realizações acadêmicas, elaborados pelos pesquisadores do NUPEME-UFU até o presente artigo, como mostram as dissertações,

<sup>1</sup> Para este estudo, *território* “é o conjunto de projetos e representações nos quais vai desembocar, pragmaticamente, toda uma série de comportamentos, de investimentos, nos tempos e nos espaços sociais, culturais, estéticos, cognitivos” (Guattari & Rolnik, 1986, p. 323).

por ordem cronológica: Rodrigues (2006); Fonseca (2009); Nunes (2010); Cintra (2010); Alves (2012); Souza (2017). Também, os livros publicados por Souza Júnior, Fernandes, Lopes e Silva (2007) e Souza Júnior, Fernandes, Lopes e Silva (2010). Os capítulos de livros: Silva, Fernandes, Lopes e Souza Júnior (2007); Souza Júnior, Rodrigues e Lopes (2007); Souza Júnior e Lopes (2007); Souza Júnior e Moura (2010) e Alves *et al.* (2015). Por fim, os artigos completos publicados em periódicos: Moura, Alves, Souza Júnior (2013) e Moura, Souza Júnior, Carvalho e Alves (2018). Esses trabalhos demarcam o território NUPEME-IEEE, que é permeado de histórias das tecnologias do OA impulsionadas pelo projeto governamental da Rede Interativa Virtual de Educação (RIVED):

[...] em junho de 2004, aconteceu a oportunidade tão esperada: o MEC estava constituindo grupos de universidade para estudarem e desenvolverem objetos de aprendizagem virtuais através do projeto intitulado RIVED (Rede Interativa Virtual de Educação). A partir desse momento, foi constituída a equipe da Universidade Federal de Uberlândia, UFU [...] [que constituiu] uma iniciativa do governo de introduzir e utilizar as Tecnologias da Informação e Comunicação no Ensino Público Nacional (Rodrigues, 2006, p. 10 e 35).

A pesquisa de Cintra (2010) e a de Rebouças, Maia e Scaico (2021) esclarecem que nos anos de 1999 a 2004 RIVED foi o acrônimo para Rede Internacional Virtual de Educação, iniciativa internacional que envolviam Brasil, Peru e Venezuela. Ele tinha

o propósito de desenvolver módulos educacionais digitais, os quais foram compostos por documentação técnico-pedagógica e atividades a serem desenvolvidas por alunos em ambiente informatizado mediado por professor nas áreas de Biologia, Física, Química e Matemática do Ensino Médio. Foi uma iniciativa pioneira que utilizava a tecnologia do computador, por meio da criação de material didático na forma de OA, para auxiliar o ensino e a aprendizagem nas áreas de Ciências e Matemática (Cintra, 2010, p. 20).

Os referidos pesquisadores afirmam que foram desenvolvidos mais 100 OA nesse período. Depois de 2004, o processo de produção foi transferido para as universidades pela Secretaria de Educação a Distância (SEED), constituindo a Fábrica Virtual da RIVED, incluindo, além do Ensino Médio, conteúdos para o Ensino Fundamental, Educação Profissional e Educação Especial. “A partir dessas mudanças, a RIVED deixa de ser Rede Internacional Virtual de Educação para Rede Interativa Virtual de Educação. O projeto foi encerrado em meados dos anos 2010 [...]” (Rebouças, Maia & Scaico, 2021, p. 9).

Embora o projeto governamental tenha se encerrado em 2010, as obras supracitadas constatarem a continuidade de produção nas universidades, além de apresentarem a importância de ainda se discutir os OA. O que ficou mais evidente com os efeitos que a pandemia da Covid-19 trouxe à educação, como constata a pesquisa de Santos, Lopes, Fagundes, Falavigna e Andreatta-da-Costa (2022), com o trabalho de 571 docentes na pandemia, foi a extrema urgência de os professores conhecerem e produzirem OA, para que o processo educativo não fique à deriva em contextos como o apresentado no ensino remoto.

Outro fato histórico importante a ser dito é que, concomitante à produção de Objetos de Aprendizagem nas universidades, a SEED/MEC lançou três editais para prêmios de produção de OA, sendo a primeira edição em 2005, a segunda em 2006 e a terceira em 2007. Poderiam participar dos editais “alunos de graduação e pós-graduação de Instituições de Ensino Superior nacionais e professores da Educação Básica e profissionalizante, incluindo multiplicadores dos

Núcleos de Tecnologia Educacional dos estados [...]” (Brasil, 2006, p. 1). No primeiro, foram premiados sete, no segundo “44 objetos de aprendizagem dos 74 recebidos pela organização do concurso” (Brasil, 2006, p. 1). No último, premiou-se 80, em que cada um deles recebeu R\$ 5 mil. “Nesta edição, aproximadamente 200 objetos foram inscritos nas áreas de artes, ciências, história, geografia, língua portuguesa, matemática, química, biologia e física” (Brasil, 2007, p. 1).

Não encontramos um número preciso de quantos OA foram construídos de 1999 a 2010, período em que o Governo Federal fomentava o projeto. Mas sabe-se que a RIVED ensinava e propagava a ferramenta de autoria *Adobe Flash* para criar tais objetos. Esse produto era visualizado e reproduzido na internet por meio do *software Flash Player* que, por questões de segurança, foi descontinuado dos navegadores *web* em 31 de dezembro de 2020. Fato que fez com que todos os Objetos de Aprendizagem construídos com essa ferramenta não fossem mais acessados pelo usuário comum a partir da referida data, fazendo com que os pesquisadores da área e que produziam com tal ferramenta investigassem outras formas de autoria. É dessa lacuna que criamos nosso problema de pesquisa: *que ferramenta de autoria é essa que se mostra para a produção de Objetos de Aprendizagem no contexto da Educação Matemática?* Assim, o objetivo geral do trabalho é apresentar uma ferramenta para a produção de Objetos de Aprendizagem para estudantes da Licenciatura em Matemática no contexto da Cultura Digital.

Para responder ao questionamento e chegar ao objetivo, foi proposto, neste artigo, uma linha de fuga<sup>2</sup> do território NUPEME-IEE para outro: NUPEME-LEVY, um novo território em criação que busca dar forma à ferramenta de autoria da criação de OA. Assim, na seção *Fundamentação Teórica* estabelecemos o arcabouço teórico, que delinea um caminho para a escolha dessa. Na seção *Investigação dos Paradigmas de Modelagem na Escolha da Ferramenta de Autoria*, analisa-se os métodos de modelagem existentes e que permitem criar OA. Na seção *Problematização para Uso da Ferramenta de Autoria*, formulou-se a situação-problema de epidemia, construindo um simulador com diferentes tipos de modelagem para dar forma ao objeto para licenciandos em Matemática. Na última seção, *Resultados e Conclusões*, discutiu-se a usabilidade do OA, respondendo à questão.

## 2 Fundamentação Teórica

Nesta seção abordaremos alguns conceitos para a construção de OA na Educação Matemática. Há várias possibilidades para esta discussão, de modo que escolhemos iniciar pelo nosso entendimento do que seja tecnologia, acreditando que, a partir daí, outros conceitos sejam concatenados e esclarecidos.

Assim, quando dizemos *tecnologia*, queremos discuti-la nos preceitos de Foucault (1982, p. 323-324) em que

devemos entender que há quatro grupos principais de “tecnologias”, cada um deles uma matriz de razão prática: (1) tecnologias de produção, que permitem produzir, transformar ou manipular as coisas; (2) tecnologias dos sistemas de signos, que permitem utilizar signos, sentidos, símbolos ou significação; (3) tecnologias de poder, que determinam a conduta dos indivíduos e os submetem a certos fins ou dominação, objetivando o sujeito; (4) tecnologias de si, que permitem aos indivíduos efetuar, com

<sup>2</sup> É “uma desterritorialização. Os franceses não sabem bem o que é isso. É claro que eles fogem como todo mundo, mas eles pensam que fugir é sair do mundo, místico ou arte, ou então alguma coisa covarde, porque se escapa dos engajamentos e das responsabilidades. Fugir não é renunciar às ações, nada mais ativo que uma fuga. É o contrário do imaginário. É também fazer fugir, não necessariamente os outros, mas fazer alguma coisa fugir, fazer um sistema vazar como se fura um cano. [...] Fugir é traçar uma linha, linhas, toda uma cartografia. Só se descobre mundos através de uma longa fuga quebrada” (Deleuze & Parnet, 1998, p. 30).

seus próprios meios ou com a ajuda de outros, um certo número de operações em seus próprios corpos, almas, pensamentos, conduta e modo de ser, de modo a transformá-los com o objetivo de alcançar um certo estado de felicidade, pureza, sabedoria, perfeição ou imortalidade.

Esses saberes consolidados dos práticos, que “dificilmente operam separadamente, [...] Cada um implica certos modos de treinamento e modificação dos indivíduos, não apenas no sentido óbvio de aquisição de certas habilidades, mas também de aquisição de certas atitudes” (Foucault, 1982, p. 324).

Desse ponto de vista, discutir uma possível tecnologia para o ensino da Matemática é, antes, compreender o que ela é e como se pratica. A esse respeito, Devlin (2012, p. 4, tradução nossa) esclarece que o

crescimento dramático da matemática levou, na década de 1980, ao surgimento de uma nova definição de matemática como a ciência dos padrões. De acordo com essa descrição, o matemático identifica e analisa padrões abstratos — padrões numéricos, padrões de forma, padrões de movimento, padrões de comportamento, padrões de votação em uma população, padrões de repetição de eventos aleatórios e assim por diante. Esses padrões podem ser reais ou imaginários, visuais ou mentais, estáticos ou dinâmicos, qualitativos ou quantitativos, utilitários ou recreativos. Eles podem surgir do mundo ao nosso redor, da busca pela ciência ou do funcionamento interno da mente humana. Diferentes tipos de padrões dão origem a diferentes ramos da matemática.

É com esse matemático britânico que passamos a caminhar, compreendendo que a Matemática é a ciência dos padrões abstratos e que seu saber prático está em identificar, analisar e usá-la para se ter uma vida digna de ser vivida, o que nos levou a afunilarmos nosso interesse, na construção de OA para abstração de padrões reais que surgem do mundo ao nosso redor. Para Lévy (2001, p. 71) a “abstração [...], enquanto atividades cognitivas, tem, portanto, uma origem eminentemente prática [...]”,

em termos de suas relações com as tecnologias intelectuais. É abstrato todo o problema fora de nossas capacidades de manipulação e de reconhecimento imediatos. Graças a sistemas de representações externas, problemas abstratos podem ser traduzidos ou reformulados de tal forma que possamos resolvê-los através da execução de uma série de operações simples e concretas, que façam uso de nossas faculdades operativas e perceptivas. Para serem corretamente efetuadas, estas manipulações de representações devem ser objetos de um aprendizado e treinamento, como qualquer outra atividade. [...] É aqui que entra em jogo a terceira faculdade cognitiva: nossa capacidade de “rodar” modelos mentais de nosso ambiente. [...] Graças à simulação de modelos mentais, o sistema cognitivo introjeta parcialmente os sistemas de representação e os algoritmos operativos cujo uso foi adquirido por ele. As tecnologias intelectuais, ainda que pertençam ao mundo sensível “exterior”, também participam de forma fundamental no processo cognitivo. Encarnam uma das dimensões objetivas da subjetividade cognoscente. Os processos intelectuais não envolvem apenas a mente, colocam em jogo coisas e objetos técnicos complexos de função representativa e os automatismos operatórios que os acompanham. [...] as criações de novos modelos de representação e de manipulação da informação marcam etapas importantes na aventura intelectual humana. E a história do pensamento não se encontra identificada, aqui, com a série dos produtos da inteligência humana, mas sim com as transformações do processo intelectual em si, este misto de atividade subjetivas e objetivas (Lévy, 2001, p. 159-160).

Lévy (2001) partiu de dados técnicos para definir que o modo de conhecer que vem das

tecnologias baseadas na informática é a simulação: “nossa capacidade de ‘rodar’ modelos mentais de nosso ambiente” (Lévy, 2001, p. 159). Uma forma de conhecer da abstração que

não é lido ou interpretado como um texto clássico, ele geralmente é explorado de forma interativa. Contrariamente à maioria das descrições funcionais sobre papel ou aos modelos reduzidos analógicos, o modelo informático é essencialmente plástico, dinâmico, dotado de uma certa autonomia de ação e reação. (...) O conhecimento por simulação é sem dúvida um dos novos gêneros de saber que a ecologia cognitiva informatizada transporta. [... Nela é possível fazer a] manipulação dos parâmetros e a simulação de todas as circunstâncias possíveis [que] dão ao usuário do programa uma espécie de intuição sobre as relações de causa e efeito presentes no modelo. Ele adquire um conhecimento por simulação do sistema modelado, que não se assemelha nem a um conhecimento teórico, nem a uma experiência prática, nem ao acúmulo de uma tradição oral (Lévy, 2001, p. 121-122).

A partir de tais dizeres, compreendemos que, na abstração, há níveis (camadas) distintas que não se assemelham nem ao real e nem ao conhecimento teórico. Essa forma de conhecer por simulação já foi discutida por Valente (1999), ao analisar os diferentes tipos de *softwares* usados na educação. Se nos dizeres de Lévy (2001) temos a percepção de que o usuário-estudante é um manipulador dos parâmetros da simulação criados por outro, Valente (1999) faz uma sucinta diferenciação:

Um determinado fenômeno pode ser simulado no computador, bastando para isso que um modelo desse fenômeno seja implementado na máquina. Ao usuário da simulação, cabe a alteração de certos parâmetros e a observação do comportamento do fenômeno, de acordo com os valores atribuídos. Na modelagem, o modelo do fenômeno é criado pelo aprendiz, que utiliza recursos de um sistema computacional para implementá-lo. Uma vez implementado, o aprendiz pode utilizá-lo como se fosse uma simulação.

Portanto, a diferença entre o *software* de simulação e o de modelagem está em quem escolhe o fenômeno e em quem desenvolve o seu modelo. No caso da simulação, isso é feito a priori e fornecido ao aprendiz. No caso da modelagem, é o aprendiz quem escolhe o fenômeno, desenvolve o seu modelo e implementa-o no computador. Nesse sentido, a modelagem exige um certo grau de envolvimento na definição e representação computacional do fenômeno e, portanto, cria uma situação bastante semelhante à atividade de programação e acontecem as mesmas fases do ciclo descrição-execução-reflexão-depuração-descrição (Valente, 1999, p. 95).

As pesquisas do NUPEME-UFU têm caminhado no sentido de que o usuário-estudante faça simulações com uso de *softwares* de modelagem, pois acreditam nos preceitos de que se aprende melhor quem cria seus próprios OA, como vislumbrou o programa RIVED (Prata & Nascimento, 2007) (Alves, 2012, 2017).

Ao seguirmos essa linha, é importante ler o alerta que Lima (2021, p. 15) faz: nessa perspectiva, cada usuário-estudante “[...] descreve uma abstração levando em conta as particularidades importantes para ele”. Sabemos dessa crítica, mas os resultados das pesquisas do NUPEME-UFU corroboram as ideias de Levy (2001), em que o conhecimento se dá como

o resultado de redes complexas onde interagem um grande número de atores humanos, biológicos e técnicos. Não sou “eu” que sou inteligente, mas “eu” com o grupo humano do qual sou membro, com minha língua, com toda uma herança de métodos e tecnologias intelectuais (dentre as quais, o uso da escrita). (...) O pretenso sujeito inteligente nada mais é que um dos micros autores de uma ecologia cognitiva que o

engloba e restringe”. (...) O pensamento se dá em uma rede na qual neurônios, módulos cognitivos, humanos, instituições de ensino, línguas, sistema de escrita, livros e computadores se interconectam, transformam e traduzem as representações (Lévy, 2001, p.135).

Assim, Lévy (2001) compreende o conhecimento por simulação como uma modalidade aberta, plurívoca e distribuída. Logo, a busca por uma ferramenta de autoria de OA se reduz à escolha de um *software* de modelagem de simulação, que, segundo Devlin (2012), deve possibilitar diferentes abstrações da realidade.

### 3 Investigação dos Paradigmas de Modelagem na Escolha da Ferramenta de Autoria

Na fundamentação deste artigo, chegamos aos *softwares* de modelagem pelos dizeres de Lévy (2001). Em nossa experiência de produzir OA para a RIVED, concluímos que, para a Educação Matemática, a modelagem era a metodologia mais indicada para criá-los (Alves & Souza Júnior, 2013). Segundo Borshchev (2014), existem três pontos de vista diferentes que o modelador pode adotar ao mapear o sistema do mundo real para sua abstração do mundo em modelos, são eles: Modelagem de Dinâmica de Sistemas (MDS), Modelagem de Eventos Discretos (MED) e a Modelagem Baseada em Agentes (MBA).

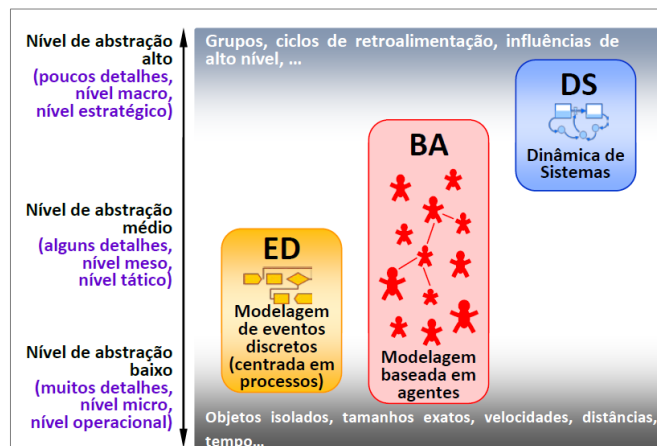
O paradigma da MDS, segundo Filho (2001, p. 30), “é uma técnica, na qual sistemas sociais, não lineares, dinâmicos e complexos, podem ser entendidos e analisados, através de interações. Além disso, novas políticas e estruturas podem ser desenhadas para melhorar o comportamento do sistema”. Já o paradigma MED, segundo Almeida (2016, p. 6), em “um sistema de eventos discretos, um ou mais fenômenos de interesse mudam seu valor, ou estado, em pontos discretos (ao invés de continuamente) no tempo”.

De acordo com Emrich, Suslov e Judex (2007), a Modelagem Baseada em Agentes (ou Sistemas Multiagentes) é uma generalização dos autômatos celulares (AC), uma vez que ambos os métodos descrevem sistemas complexos por definição de interações locais. Nele, “o modelador descreve o sistema do ponto de vista de objetos individuais que podem interagir entre si e com o ambiente” (Borshchev, 2014, p. 1). Fan, Bravo e Collischonn (2016, p. 2) explicam que os

agentes são componentes autônomos com comportamentos particulares que se inter-relacionam entre si e com o ambiente. Basicamente, a filosofia de um modelo baseado em agentes é que o comportamento singular de um único agente não representa fielmente a realidade, mas que a soma de todos os comportamentos locais corresponde a uma boa representação do comportamento real do grupo. Dessa forma, é possível constatar que aplicar uma abordagem orientada a agentes para a resolução de um problema significa decompô-lo em múltiplos componentes autônomos com objetivos particulares e que se inter-relacionam.

Esses três métodos para simulação têm uma “gama específica de níveis de abstração. A dinâmica de sistemas pressupõe um nível de abstração alto [...]. Eventos discretos suportam abstrações de níveis [...] baixo. No meio, estão os modelos baseados em agentes [...]” (Grigoryev, 2015, p. 13), Figura 1.

**Figura 1:** Níveis de Abstração dos paradigmas de modelagem



Fonte: Grigoryev (2015, p. 13).

A partir dessas abstrações, usou-se o argumento da convergência dos paradigmas de modelagem para escolher uma única ferramenta de autoria que os usuários-estudantes possam, em um único ambiente, seguir três caminhos para abstrair do mundo real modelos (Jenkins, 2015). Segundo Anylogic (2022a, p. 1), essa convergência se dá no *AnyLogic*, pois ele “é o único *software* de modelagem de simulação multi-método [híbrido] disponível no mercado. A licença Personal Learning Edition (PLE) é fornecida sem custo para fins educacionais e autoaprendizado”. Emrich, Suslov e Judex (2007) explicam que ele

é um ambiente de programação e simulação, voltado principalmente para modelagem de sistemas híbridos, baseado em JAVA. Ele permite ao usuário combinar diferentes técnicas e abordagens, como equações diferenciais, eventos discretos e sistemas baseados em agentes. Essas possibilidades de combinação o tornam uma ferramenta muito interessante para simulação de sistemas complexos, (Emrich, Suslov & Judex, 2007, p. 2).

O *software* pode ser instalado nos sistemas operacionais *Microsoft Windows*; *Apple Mac* e *Linux*. Recomenda-se um *hardware* de 1,5 *Gigabyte* (GB) de espaço livre em disco; memória de acesso aleatório (RAM) superior a 4 GB; e, por fim, um processador com pelo menos 2 núcleos (Anylogic, 2022b, p. 1). A versão gratuita da ferramenta “possui todos os recursos e funcionalidades básicos do *AnyLogic*, incluindo suporte aos três métodos de modelagem, todas bibliotecas padrão, integração com mapas GIS, animações 3D [...]” (Anylogic, 2022a, p. 1). A restrição está no número de tipos distintos de agentes que pode ser criado. Para fazer o *download* do *software* é preciso preencher um formulário com alguns dados pessoais do usuário, consulte o Anylogic (2022a). A instalação pode ser lida em Grigoryev (2015).

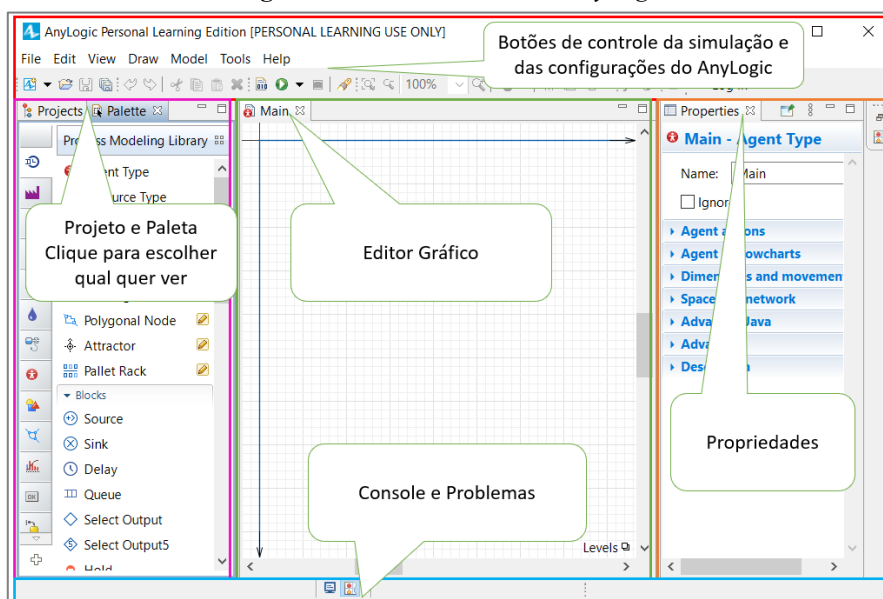
O *AnyLogic* é composto por cinco regiões (Figura 2). Na primeira, de cima para baixo, estão os botões de controle da simulação e das configurações do *software*. Em seguida, da esquerda para a direita, estão as abas do Projeto e da Paleta. No centro, está o Editor Gráfico e a aba de Propriedades. Embaixo, estão a aba que retorna erros, caso ocorram na execução da simulação.

Grigoryev (2015, p. 26) explica que a aba Projetos (*Projects*) “permite acessar os modelos abertos na área de trabalho, e a árvore de projetos ajuda a navegar facilmente por eles”. Na Paleta (*Palette*), fica a “lista os objetos agrupados em paletas. Para adicionar um elemento ao seu modelo, arraste-o do **Palette** para o editor gráfico” (Grigoryev, 2015, p. 26, grifos do autor). O Editor Gráfico “permite editar o diagrama do agente, e você pode adicionar elementos



do modelo arrastando-os da **Palette** para o diagrama e colocando-os no canvas do editor. Os elementos colocados no *frame azul* aparecerão na janela do modelo, quando ele for executado” (Grigoryev, 2015, p. 26, grifos do autor). A aba Propriedade (*Properties*) “permite visualizar e modificar as propriedades do item selecionado” (Grigoryev, 2015, p. 26).

**Figura 2:** Área de trabalho do *AnyLogic*



**Fonte:** Dados da Pesquisa

A lógica de criação se assemelha com outros *softwares* aplicados à Educação Matemática, por exemplo: *Scratch* (Marji, 2014) e/ou *App Inventor* (Alves & Pereira, 2016). Neles, a ideia é arrastar os blocos de código da aba Paleta, soltá-los nos editores gráficos e configurar na aba Propriedades. O diferencial, talvez, seja a quantidade de blocos de programação que o *AnyLogic* tem e a possibilidade de programar direto na linguagem Java, fazendo uso da Programação Orientada a Objetos. Dessa perspectiva, nosso método para o seu uso foi criar agentes na aba Projetos; eleger na aba Paleta quais blocos de programação seriam necessários para que a simulação funcionasse; e, por fim, configurá-los na aba Propriedade.

Com a escolha da ferramenta, passamos a problematizar uma situação do mundo real para a construção de um OA. Nas criações do projeto RIVED, estavam relacionados os documentos de planejamento: *Design* instrucional; *General design*; *Storyboards*; Guia do professor e do estudante (Rodrigues, 2006). Tais documentos não foram construídos, pois não estavam no escopo deste trabalho. Fizemos o resumo da situação-problema e construímos o simulador para que os usuários-estudantes refletissem sobre um questionamento que deixamos na página inicial do simulador (embora as pesquisas do NUPEME-UFU apontem que, na prática de sala de aula, os professores e alunos os usam a partir de questionamentos próprios).

#### 4 Problematização para Uso da Ferramenta de Autoria

A problematização que deu origem ao Objeto de Aprendizagem de simulação é uma modificação dos trabalhos realizados por Borshchev (2014) e Anylogic (2022d). A situação-problema foi: à medida que uma doença viral, contagiosa, mas não letal progride em uma comunidade, o nível de preocupação de uma pessoa infectada aumenta quando essa variável atinge um valor limite. O doente busca por tratamento em um hospital que tem capacidade limitada. Sugere-se que o usuário-estudante busque investigar o seguinte problema: *como a limitação de atendimento do hospital afeta a dinâmica da doença e o nível preocupação dos*

*infectados?*

A construção da simulação foi dividida em três tarefas, cada uma com um paradigma de modelagem diferente: a modelagem do comportamento da população foi realizada por meio da MBA; já a quantidade de vagas em um hospital foi realizada por meio da MED; e, por fim, a variação do nível de preocupação do infectado para ir ou não ao hospital foi realizada mediante a MDS.

Na primeira, criamos uma população de agentes, uma abstração da população a ser modelada. O número inicial é de 400 pessoas, sendo sete infectadas. Essas pessoas (agentes) ficam em uma área retangular de 4000 metros quadrados ( $m^2$ ), ou seja, com 100m de largura (*width*) e 40 altura (*height*). Nessa região, supomos que as pessoas têm contato umas com as outras, conectadas, se estão a dez metros de distância, formando, assim, uma rede de interações. Essas informações são necessárias para configurar o *AnyLogic*. A programação computacional da contaminação das pessoas é realizada baseada no modelo de Equações Diferenciais Ordinárias (EDO) não linear, proposto por Kermack e McKendrick (1927):

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

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

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

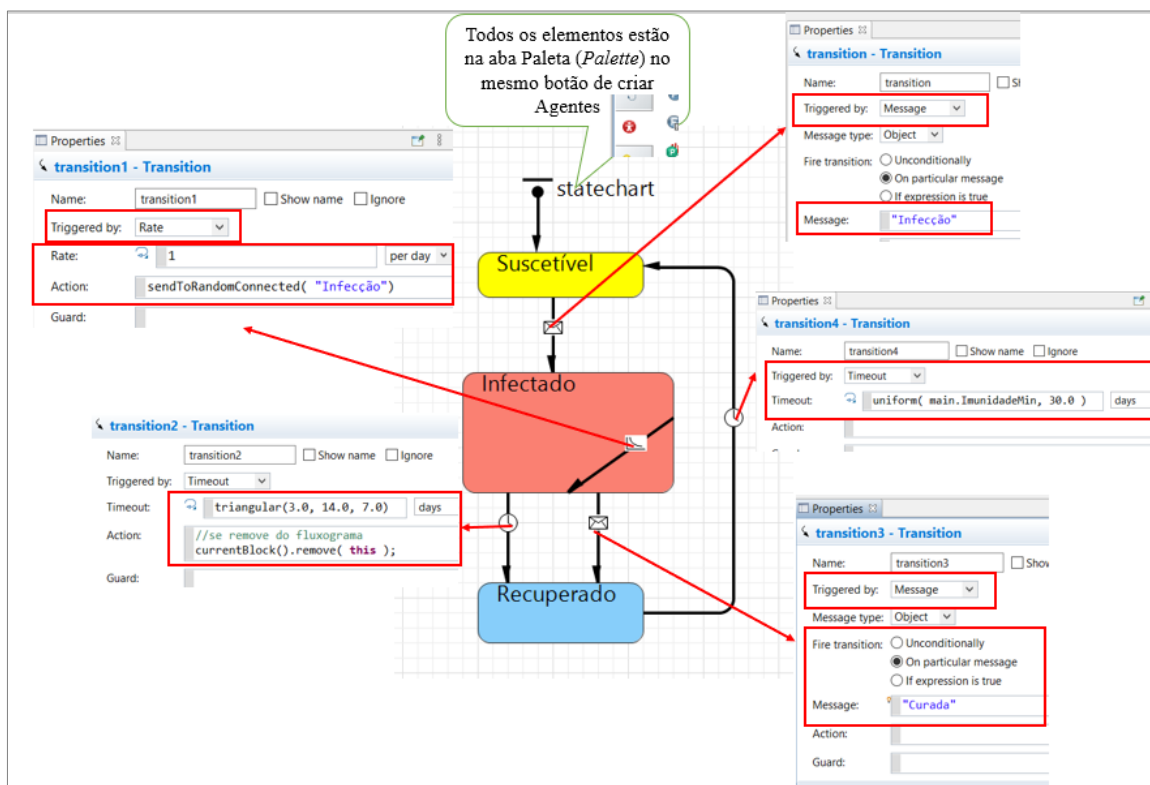
Esse modelo ficou conhecido como SIR, em que  $S(t)$  denomina o número de susceptíveis no tempo  $t$ ,  $I(t)$  o número de infectados e  $R(t)$  o número de recuperados. As variáveis  $\beta$  e  $\gamma$  representam as taxas de infecção e recuperação, respectivamente, ambas maiores do que zero (Emrich, Suslov & Judex, 2007). Assim, temos três estados e as transições entre estes.

O comportamento na MBA inicia-se com 393 pessoas no estado Suscetível que podem ser infectadas pelo vírus. A transmissão da doença é modelada pela mensagem “Infecção” enviada de uma pessoa para outra. Tendo recebido tal mensagem, o indivíduo se torna infectado. Esse sujeito envia, diariamente, uma mensagem de “Infecção” para as pessoas que estão na rede de interação. Quando essa outra recebe a mensagem e está no estado Suscetível, vai para o estado Infectado e começa a infectar outras pessoas também.

O indivíduo sai do estado Infectado e entra no Recuperado de duas maneiras: por tempo de infecção ou por tratamento. Modelamos a primeira por uma distribuição probabilística do tipo triangular tendo como base do triângulo o intervalo [3.0, 14.0] e com vértice fora dessa base o ponto (7.0, 1.0), sendo 7.0 o valor da moda no intervalo. No *software*, configuramos a função triangular (3.0, 14.0, 7.0). O *software* faz um sorteio do tempo de infecção no intervalo de 3.0 a 14.0 dias, sendo que o valor que mais se repetirá será o sete, a unidade de tempo é em dias. No *AnyLogic*, há 39 tipos de distribuição para sorteio. Na segunda maneira, o hospital passa uma mensagem dizendo que a pessoa está “Curada”, mostrando que ela está recuperada.

Essa doença dá imunidade por um tempo, ou seja, depois de um período aleatório, o agente sai do estado Recuperado e volta para o Suscetível. Esse tempo é sorteado com base em uma distribuição probabilística uniforme no intervalo [ImunidadeMin, 30). No *software*, codifica-se *Timeout: uniform (main.ImunidadeMin, 30.0)*. Na interface de programação (agente *Main*) tem-se um parâmetro que é a imunidade mínima da pessoa (*ImunidadeMin*), que foi colocado para iniciar em cinco, mas pode ser alterado. Se o usuário não alterar, sorteia-se um número de dias, podendo assumir qualquer valor entre 5 e 29, intervalo [5, 30). Sai do estado Recuperado e volta ao Suscetível. A Figura 3 apresenta como ficou o comportamento dos agentes no *AnyLogic*.

**Figura 3:** Comportamentos dos Agentes

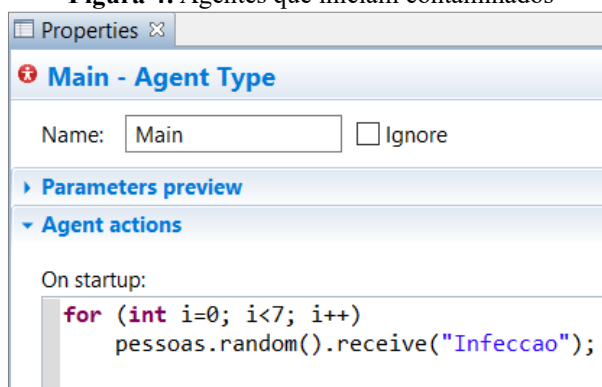


Fonte: Dados da Pesquisa

Para visualizar cada estado que está ocorrendo, criamos círculos com cores diferentes, representando as pessoas. Além disso, produzimos uma variável booleana “SendoTratado” que nos diz se está (*true*) ou não (*false*) no hospital em tratamento. Assim, construímos no *AnyLogic* um círculo amarelo para representar o estado Suscetível, um vermelho para representar o estado Infectado e um azul para representar o estado Recuperado. Por fim, um retângulo verde que representa a pessoa infectada, mas em tratamento.

A quantidade de indivíduos que iniciam contaminados é definida no agente *Main*, na aba Propriedades, que aparecem nas janelas de configuração do agente. Em *Agent actions* (Ações do agente), no campo *On startup* (Ao iniciar), escrevemos um laço de repetição (*loop For*), iniciando em *um* e terminando em *sete*. Assim, o *software* sorteia sete das 400 pessoas para ficarem no estado de Infectado, no *AnyLogic*: `peessoas.random().receive("Infeccao")`, Figura 4.

**Figura 4:** Agentes que iniciam contaminados

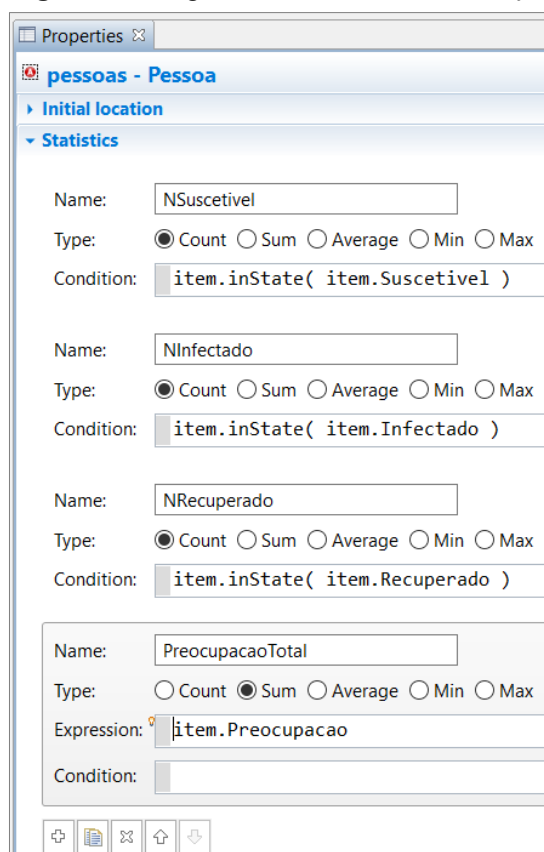


Fonte: Dados da Pesquisa

Com isso, já temos a simulação baseada em agentes finalizada, restando obter dados

estatísticos dela. Neste contexto, no *AnyLogic*, é necessário buscar, dentro do agente *Main*, a classe *Agents*, pois nela temos o Objeto pessoas. Nele, damos um duplo clique para selecioná-lo; na aba Propriedades escolhemos a janela *Statistics* (Estatística); clicamos no símbolo de mais (+) para adicionar quatro campos que obteremos dados estatístico: *NSuscetivel*, *NInfectado*, *NRecuperado* e *PreocupacaoTotal*. Essas variáveis acumulam, respectivamente, o número de pessoas no estado Suscetível, Infectado, Recuperado e o nível de preocupação com a doença. A Figura 5 apresenta as configurações no *software*.

**Figura 5:** Configurando a estatística da simulação

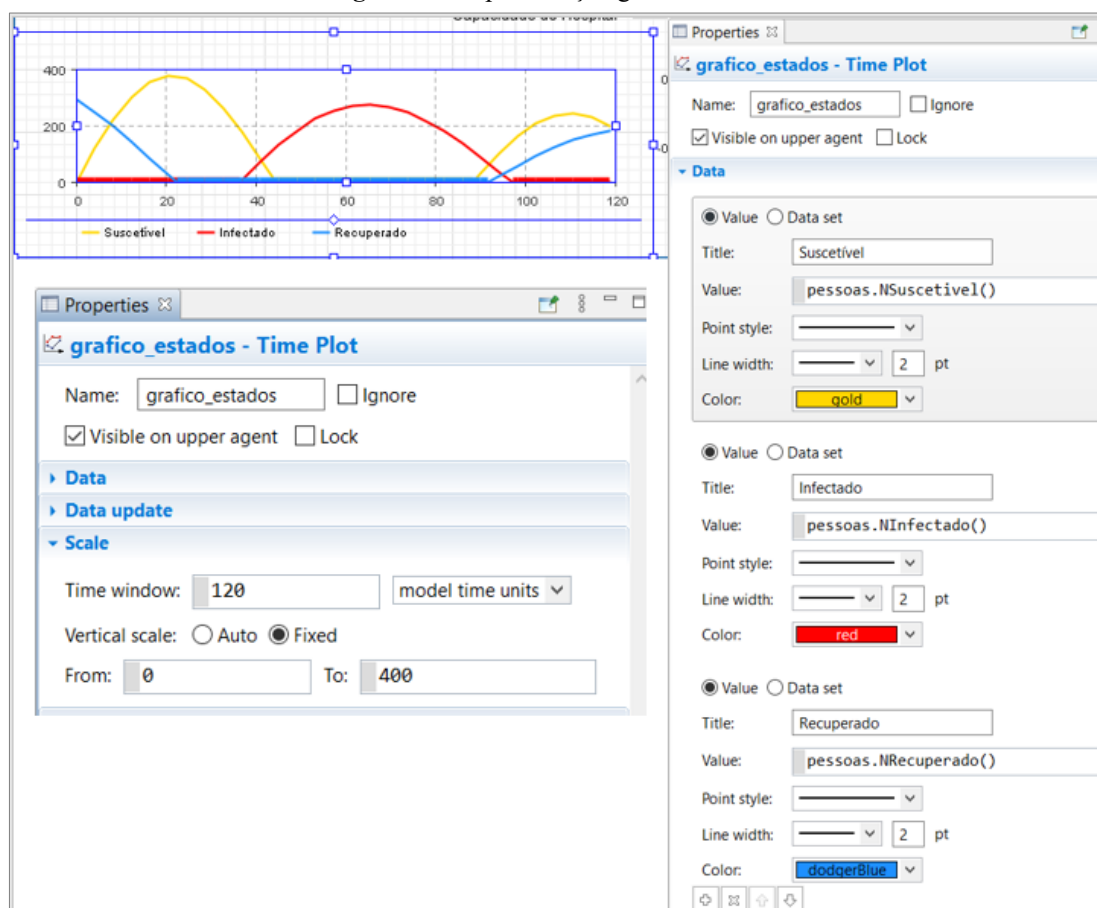


**Fonte:** Dados da Pesquisa

Como a MBA já está finalizada, elaboram-se gráficos para exibir o número de indivíduos em cada estado. Na aba Paleta, no décimo primeiro bloco de códigos *Analysis* (Análises), há vários construtores de gráficos. Arrastamos e soltamos no Editor Gráfico o bloco *Time Plot* (Gráfico de tempo), em seguida, na aba Propriedades, colocamos o nome de “grafico\_estados”. Na janela *Scale* (Escala), configuramos o eixo das abscissas, o tamanho da janela (*Time window*), colocamos 120 dias e escolhemos *model time units* (unidades de tempo modelo) para ser a unidade do eixo. No eixo das ordenadas, escrevemos para iniciar em zero e terminar em 400, com *Vertical scale* (escala vertical) marcado em *Fixed* (Fixo). Ainda para esse eixo, na janela *Data* (dados) as funções a serem plotadas: Suscetível, Infectado e Recuperado, as cores são, respectivamente, ouro, vermelho e azul, conforme a Figura 6.

Com a representação das pessoas e os comportamentos de infecção construídos, passamos a criar a representação do hospital com a Modelagem de Eventos Discretos. Selecionamos o agente *Main* para adicionarmos os blocos de códigos que o simboliza. Esse hospital é representado por uma fila de espera e um tempo de tratamento que a pessoa fica no local. Na hipótese da modelagem, uma vez em tratamento ficará curada.

**Figura 6:** A representação gráfica da MBA



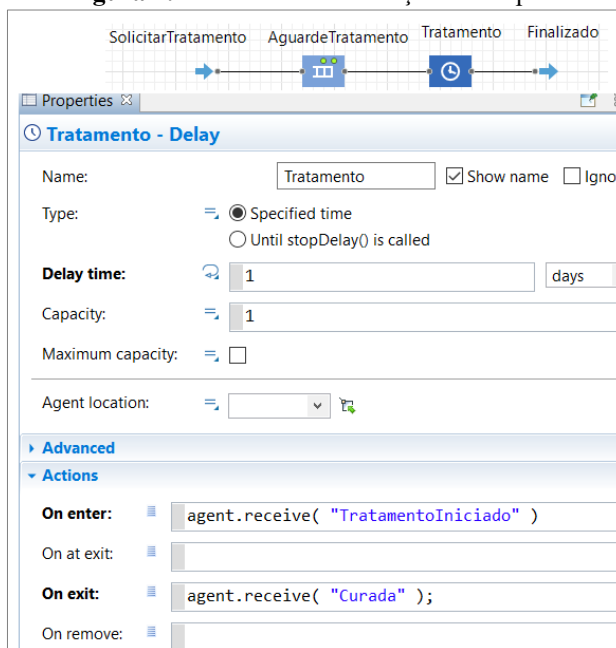
**Fonte:** Dados da Pesquisa

Na aba Paleta, no primeiro bloco de códigos, *Process Modeling Library* (Biblioteca de Modelagem de Processos), arrastamos e soltamos quatro blocos: *Enter* (Entrar); *Queue* (Fila); *Delay* (Atraso) e *Exit* (Sair), os quais nomeamos, respectivamente: *SolicitarTratamento*; *AguardeTratamento*; *Tratamento* e *Finalizado*. Em *Tratamento*, na aba Propriedades, colocamos no campo *Delay time* (Tempo de atraso) em um dia, e em *Capacity* (Capacidade de vagas no hospital) definimos um, inicialmente. Na janela *Action* (Ações), no campo *On enter* (Ao entrar) escrevemos o código *agent.receive("TratamentoIniciado")* e em *On exit* (Ao sair) acrescentamos *agent.receive("Curada")*, Figura 7.

Com a representação do hospital construída, foi necessário modelar o nível de preocupação das pessoas, para estas tomarem a decisão de ir ou não ao hospital. Esse processo se dá com a MDS. Para isso, voltamos à aba Projetos e damos um duplo clique no agente Pessoa para selecioná-lo. Depois, na Paleta, no oitavo bloco de códigos, *Agent*, arrastamos e soltamos no Editor Gráfico um bloco chamado *Event*, nomeado de *Decisao\_Ir\_Hospital*.

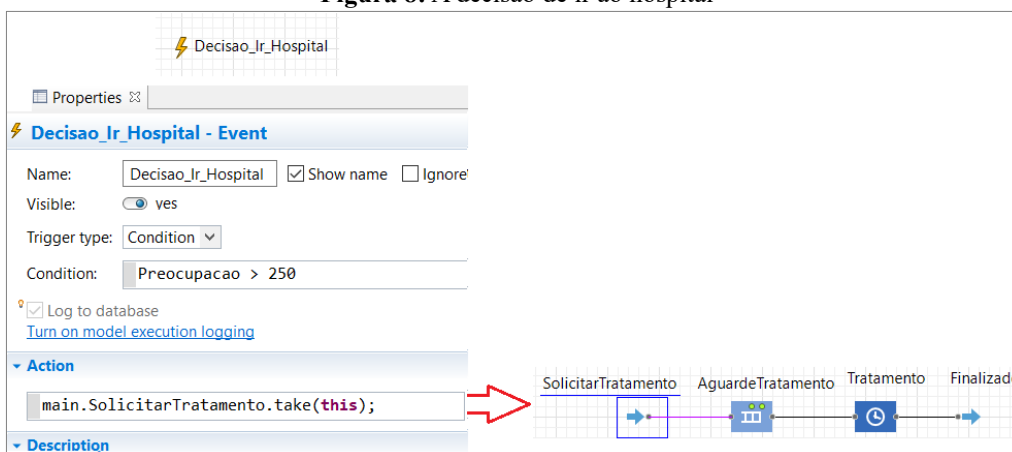
O *Event* é responsável por monitorar um compartimento chamado *Preocupacao* e aplicar uma condição para ir ao hospital e iniciar o tratamento. Para isso, na aba Propriedades, escolhemos em *Trigger type* (Tipo de disparo), *Condition* (Condição). Definimos que, se a *Preocupacao > 250*, então as pessoas infectadas vão ao hospital. Na janela *Action* (Ações), escrevemos *main.SolitarTratamento.take(this)*; Esse *main.SolitarTratamento* refere-se à entrada da pessoa no hospital, que colocamos no agente *Main*, com a Modelagem de Eventos Discretos. É aqui que se tem a interação entre os diferentes tipos de modelagem, conforme a Figura 8.

**Figura 7: Construindo a abstração do hospital**



Fonte: Dados da Pesquisa

**Figura 8: A decisão de ir ao hospital**



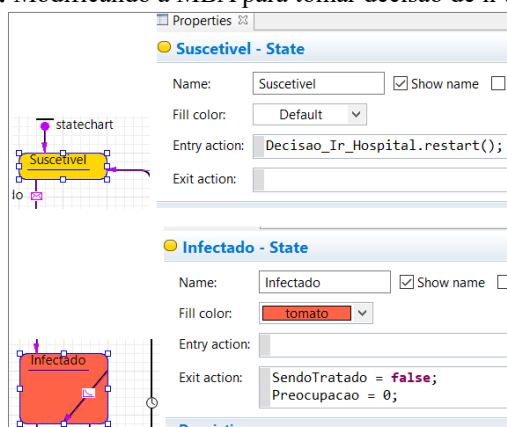
Fonte: Dados da Pesquisa

Ainda no agente Pessoa, alteramos dois estados de comportamento e incluímos os elementos da MDS na MBA para somar o nível de preocupação da pessoa em relação à doença. O primeiro estado a ser modificado é o Suscetível, de modo que acrescentamos no campo *Entry action* o comando *Decisao\_Ir\_Hospital.restart();*, para que, toda vez que a pessoa retorne a esse estado, seja reiniciado a decisão de ir ao hospital, haja vista que pode contrair novamente a doença. No estado Infectado, modificamos o campo *Exit action* (Ação de saída) colocando os códigos: *SendoTratado = false;*, e *Preocupacao = 0;*. Essas duas configurações são necessárias, pois na saída do estado de Recuperado tem um tempo aleatório de imunidade da doença (Figura 9).

Com as alterações realizadas, iniciamos a MDS. Na aba Paleta, no sétimo bloco de códigos *System Dynamics* (Sistemas Dinâmicos), arrastamos e soltamos um *Stock* (compartimento) e bloco *Flow* (Fluxo). No compartimento, na aba Propriedades, no campo *Name* (Nome), alteramos para *Preocupacao* e, no campo *Initial value* (Valor inicial), colocamos zero. Em seguida, conectamos o Fluxo ao compartimento *Preocupacao*. Nas propriedades, no

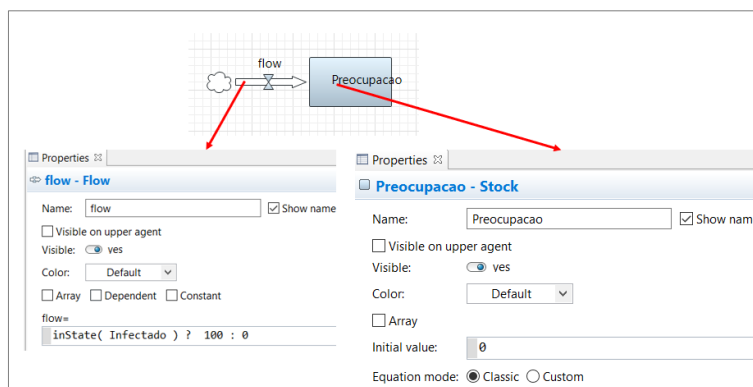
campo *Flow*, escrevemos a condição:  $inState(Infestado) ? 100 : 0$ , tendo a função de afirmar que, para cada dia que a pessoa fica doente, é adicionada à sua Preocupação (Compartimento) 100 unidades. A Figura 10 apresenta essas configurações.

**Figura 9:** Modificando a MBA para tomar decisão de ir ao hospital



Fonte: Dados da Pesquisa

**Figura 10:** Configurando a decisão de ir ao hospital

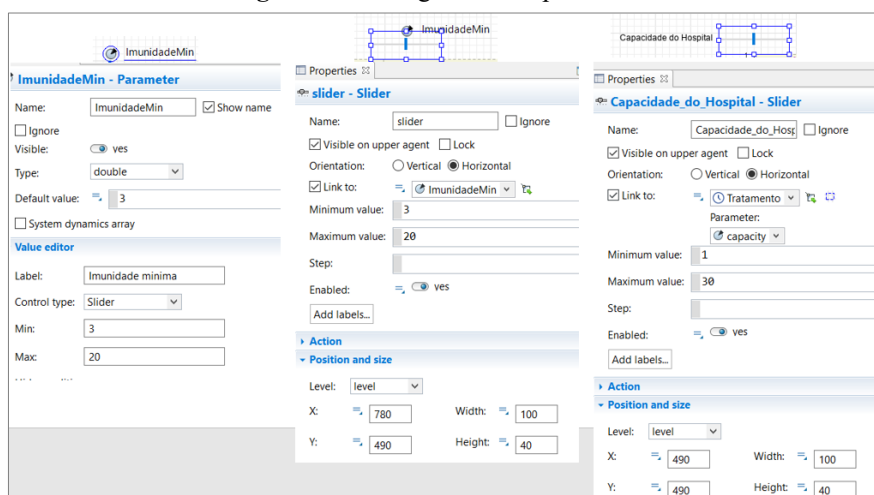


Fonte: Dados da Pesquisa

Para terminar a modelagem, voltamos à aba Projetos e damos um duplo clique com o botão esquerdo no agente *Main* para selecioná-lo. Nele, adicionaremos dois parâmetros: a capacidade de atendimento do hospital e a imunidade mínima das pessoas com relação à doença. A cada um deles, associamos um *Slider* (Controle deslizante), para o usuário-estudante alterar para o valor que desejar dentro de um intervalo predeterminado. Na aba Paleta, arrastamos e soltamos um parâmetro e dois controles deslizantes. O segundo parâmetro está associado à MED, que é a capacidade de atendimento do hospital criado automaticamente.

Selecionamos o parâmetro que foi adicionado na parte gráfica, com um clique do botão esquerdo do mouse, e na aba Propriedades, no campo *Name*, escrevemos *ImunidadeMin* e marcamos para mostrar o nome dele (*Show name*). No campo *Type* (Tipo), escolhemos *Double* (número no conjunto dos Reais). Em *Default value* (Valor padrão), colocamos para iniciar em três, no campo *Label* (rótulo) acrescentamos Imunidade mínima, em *Control type* escolhemos *Slider*, para os valores mínimo e máximo escrevemos três e 20 dias, respectivamente. Na Figura 11 apresentamos tais configurações.

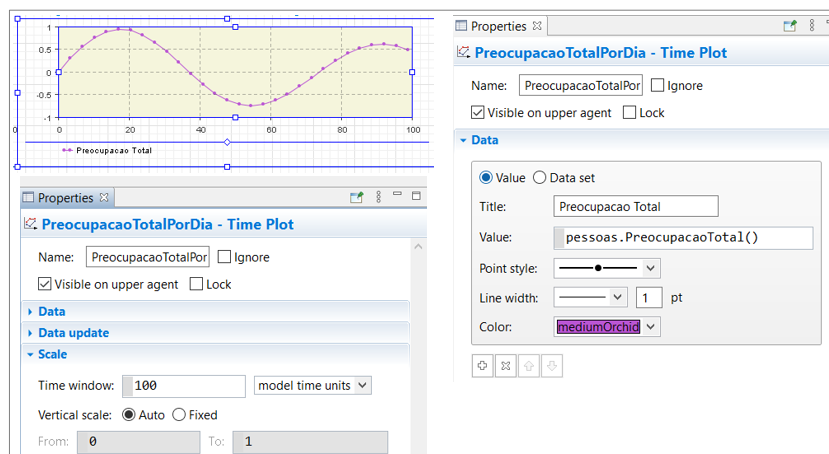
**Figura 11:** Configurando os parâmetros



**Fonte:** Dados da Pesquisa

Para finalizar, colocamos no agente *Main* um gráfico para apresentar a evolução do nível de preocupação das pessoas em relação à doença. Na Paleta, no bloco de código *Analysis*, arrastamos e soltamos para o Editor Gráfico um bloco *Time Plot*. Nas propriedades, alteramos o nome para *PreocupacaoTotalPorDia*, na janela *Data* adicionamos o título de *Preocupacao Total* e no valor colocamos *pessoas.PreocupacaoTotal()*. Na janela escala em *Time window*, escrevemos 100 unidades de tempo e na escala vertical deixamos automática (Figura 12).

**Figura 12:** Representação gráfica da Preocupação das pessoas

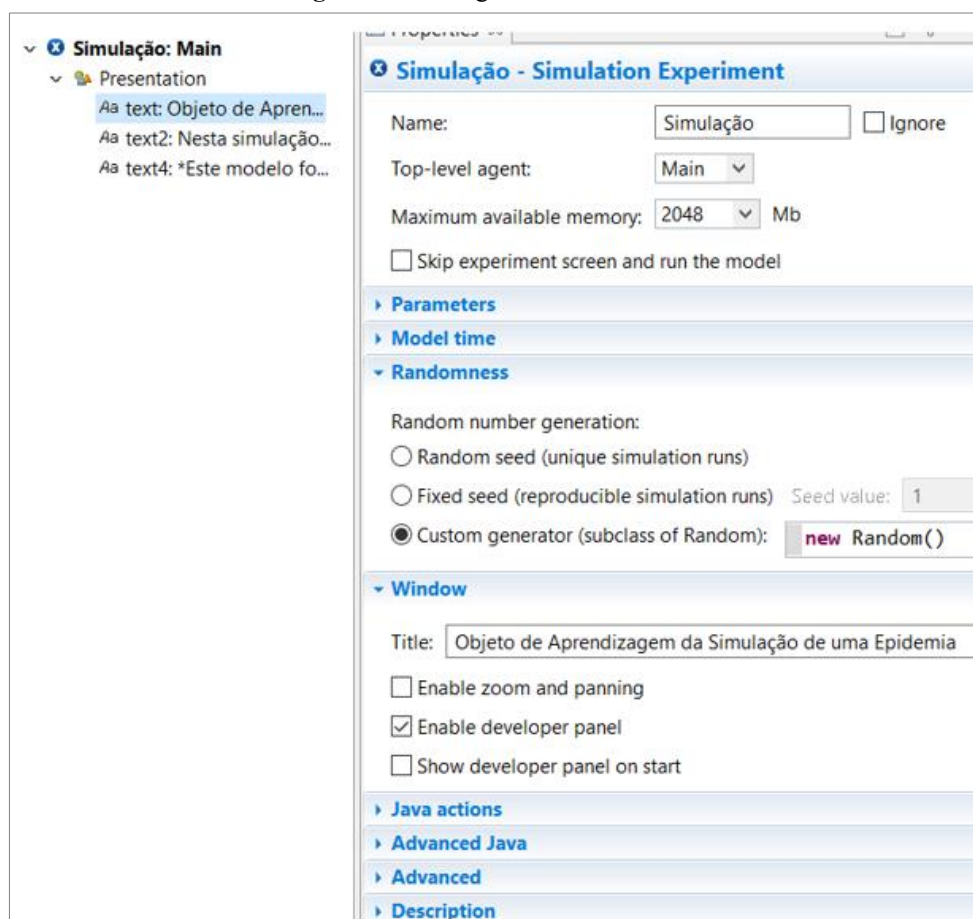


**Fonte:** Dados da Pesquisa.

Com essas configurações, o OA está finalizado, faltando apenas acrescentar e configurar uma tela inicial para explicar ao usuário-estudante como é o funcionamento do objeto e como se dá o processo avaliativo dele. Essa tela inicial configura-se em uma classe chamada *Simulation Experiment*, que fica na aba *Projetos*, selecionada com um duplo clique do botão esquerdo do mouse. Na aba *Propriedades*, modificamos o nome para *Simulação* e a memória que o computador permite para simulação, colocamos 2048 Megabyte (MB) e na janela *Randomness* escolhemos *Custom generator*. Em *Windows*, alteramos o título para *Objeto de Aprendizagem da Simulação de uma Epidemia* e marcamos a opção de *Enable developer panel* (Figura 13).



Figura 13: Configurando a Tela inicial



Fonte: Dados da Pesquisa

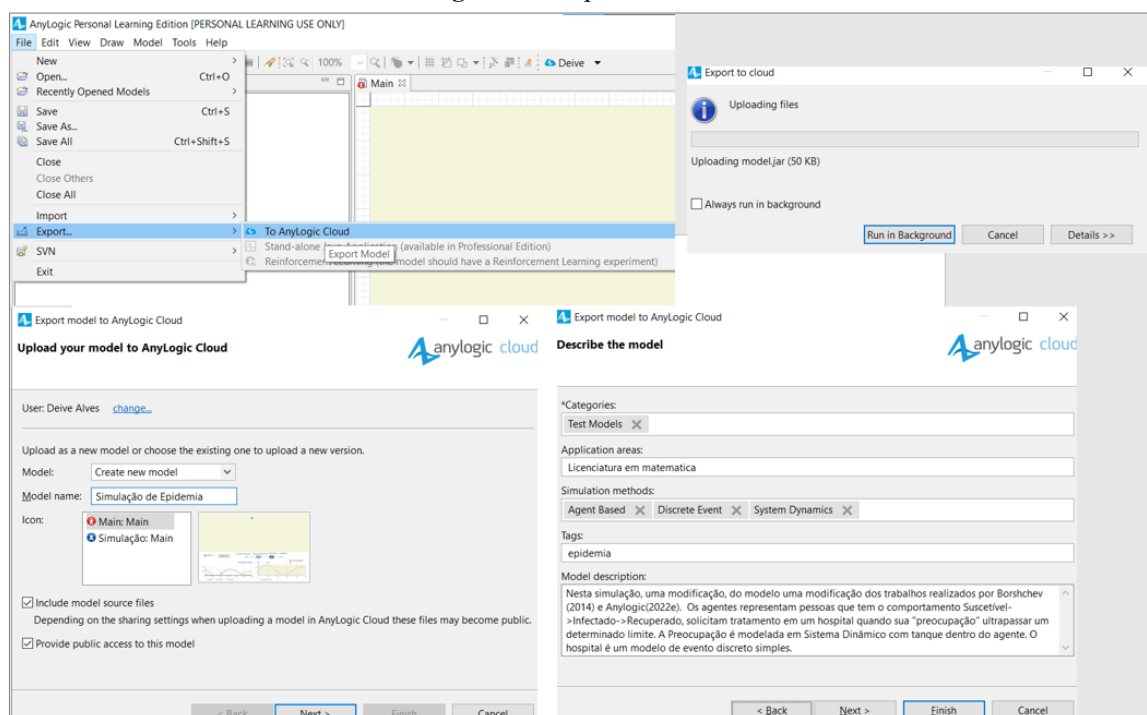
Adicionamos campos de textos para escrever o título, um com o resumo da história da simulação e outro com a pergunta de observação da simulação. Para executar pela primeira vez o aplicativo, há duas tarefas a serem realizadas. A primeira é procurar por erros que *AnyLogic* se faz pelo teclado do computador clicando na tecla *F7* para compilar o objeto. A segunda é executar o aplicativo clicando na tecla *F5*. Essa é a forma de testar e usar a simulação para quem tem instalado o *software* no computador.

## 5 Resultados e Conclusões

Na versão gratuita não se pode exportar uma versão executável do objeto. A única opção além de ter o *software* instalado no computador é exportar para o repositório do *AnyLogic*. Para isso, é necessário ter uma conta no *AnyLogic Cloud*. Acionamos o botão que direciona para contas do Gmail, mas não funcionou. Então, utilizamos o Hotmail, serviços de *e-mail* da Microsoft. O registro gerou um mensagem de confirmação que foi diretamente para a caixa de *spam* (lixo eletrônico).

Com o cadastro realizado para exportar o Objeto de Aprendizagem, é necessário ir em *File* (Arquivo), *Export...* (Exportar...) e *To AnyLogic Cloud* (Para *AnyLogic Cloud*). Depois de alguns segundos abrirá uma janela solicitando o nome da simulação; clicando em próximo (*Next*) há outros metadados a serem preenchidos: Categoria, Área de aplicação, Método de Simulação, *Tags* e Descrição da simulação (Figura 14).

**Figura 14: Exportando o OA**

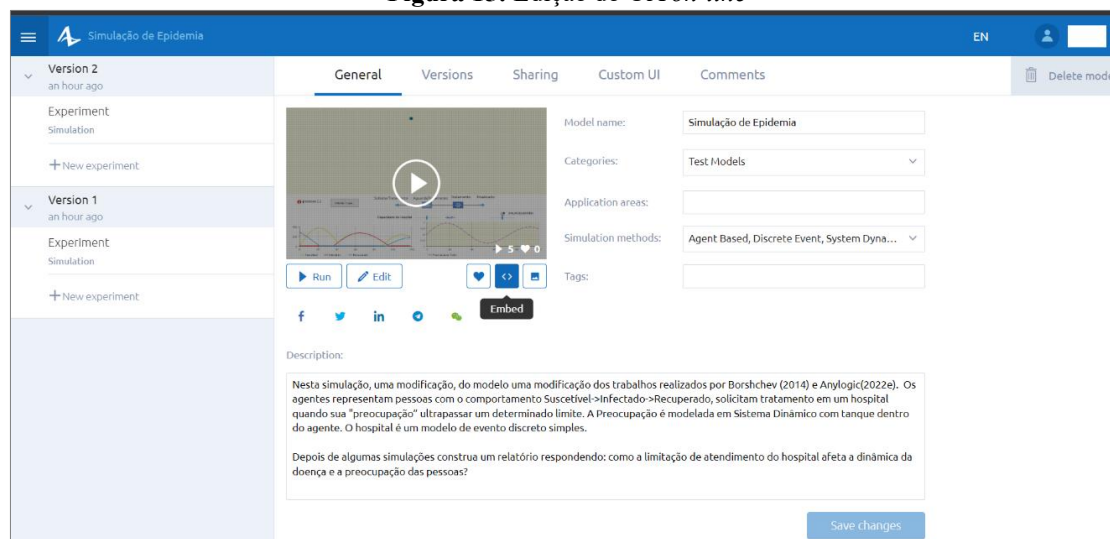


**Fonte:** Dados da Pesquisa

O autor pode editar as informações do metadado no *site* (Anylogic, 2022c). Ao fazer o *login*, é direcionado a uma página *web* que contém os modelos públicos e do autor. Ao passarmos o mouse sobre a simulação, aparece a descrição e o símbolo de uma lixeira do lado esquerdo para deletar o modelo. Se clicarmos na simulação, somos direcionados para outra página, que é dedicada a fazer experimentos com o simulador.

Para editar basta clicar no nome do modelo, localizado no lado esquerdo superior da página, próximo ao símbolo da *AnyLogic*. Nele há: *General* (Geral), *Versions* (Versões), *Sharing* (Compartilhamento), *Custom UI* (Upload personalizado), *Comments* (Comentários) e *Delete model* (Deletar modelo). As abas Geral, Compartilhamento e Comentários são as mais usadas (Figura 15).

**Figura 15: Edição do OA on-line**

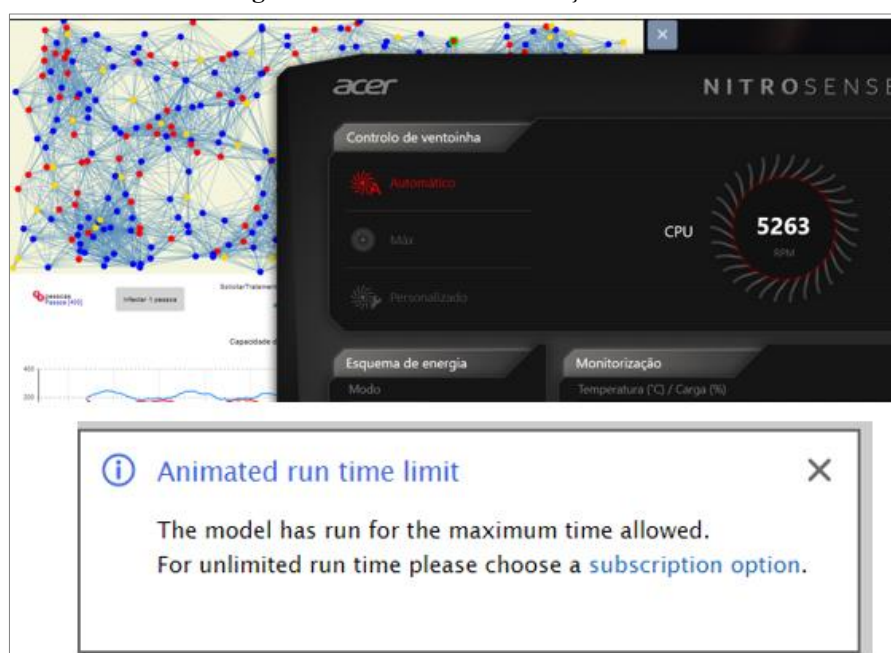


**Fonte:** Dados da Pesquisa

Verificamos que os metadados preenchidos sem estar na lista preestabelecida não aparecem na página *web*. O símbolo de maior e menor, conhecido como *Embed* (Embutir) que usa etiqueta (tag) HTML (Linguagem de Marcação de Hiper Texto) `<iframe> </iframe>`, permite a inclusão e visualização em outro documento HTML, ajudando na Interoperabilidade do objeto. Para testar, incluímos a simulação em um *blog* do *blogger* do *Google* (Figura 16). No *blog*, todos os botões funcionaram corretamente, mas foi embutida a simulação sem a tela inicial, presente nos dois outros ambientes.

Nos três ambientes: *software* instalado, repositório do *AnyLogic* e no *blog*, no momento que a simulação “rodava”, houve um aumento da temperatura do *notebook*, alterando a velocidade das ventoinhas, que no monitoramento do *Acer Nitrosense* passou de 2800 rotações por minuto (rpm) para 5263 (Figura 16). Além desse aquecimento, a simulação *on-line* ocorreu por, aproximadamente, cinco minutos, quando surgiu a mensagem: *Animated run time limit* (Limite de tempo de execução da animação). O que não ocorreu na simulação com o *software* instalado. A mensagem desaparece quando se atualiza a página *web*, permitindo inicializar novamente a simulação (Figura 16).

**Figura 16:** Testes com a simulação *on-line*



**Fonte:** Dados da Pesquisa

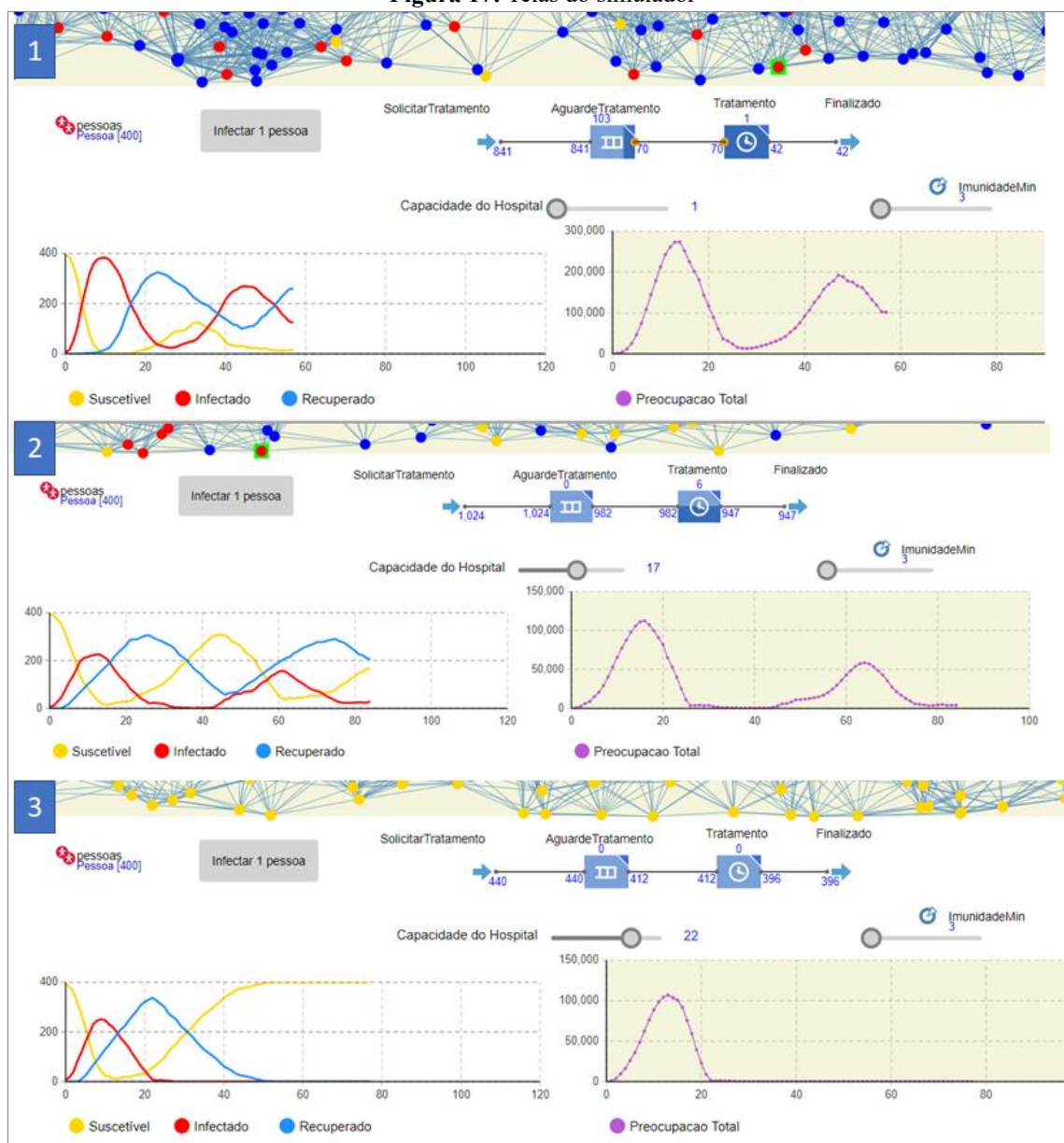
A simulação em todos os ambientes teve o mesmo comportamento, mantendo a imunidade e alterando a capacidade de atendimento. Verifica-se que o nível de preocupação das pessoas diminui e a doença tende a desaparecer. Esse fator pode ser visualizado na Figura 17.

Na simulação da Figura 17 (1), com a imunidade definida em três dias, colocamos a capacidade do hospital atendendo uma pessoa. Verifica-se que há uma oscilação tanto nos estados quanto no nível de preocupação. Alteramos para 17 vagas no hospital (Figura 17 [2]), de modo que houve uma redução no nível de preocupação e em 22 vagas, (Figura 17 [3]), depois da primeira onda, não houve mais ondas da doença e o nível de preocupação foi reduzido a zero.

Assim, podemos concluir que a ferramenta de autoria *AnyLogic* é uma tecnologia da inteligência que canaliza uma cultura da convergência de três modos de conhecer o mundo, com diferentes níveis de abstrações: baixo (MED), médio (MBA) e alto (MDS) (Grigoryev, 2015;

Devlin, 2012). Não há dúvida que essa metodologia facilita o trabalho da construção de OA, só é preciso ter cuidado para que a matemática não fique escondida nos blocos de códigos, ou mesmo na perspectiva do uso da MBA e nas regras de comportamento dos agentes.

**Figura 17:** Telas do simulador



**Fonte:** Dados da Pesquisa

Uma resposta a tais cuidados foi apresentada na pesquisa de Costa (2017), que trabalhou com alunos do 9º ano do Ensino Fundamental, investigando o desmatamento da Amazônia com o *software* NetLogo, ferramenta conhecida para se trabalhar a Modelagem Baseada em Agentes, mas a pesquisa usou a perspectiva da simulação para obter dados que permitissem ao grupo discutir conceitos matemáticos daquela etapa de ensino. Outra possibilidade está no trabalho de Emrich, Suslov e Judex (2007), ao validar a MBA com uso de Equações Diferenciais.

O fato é que estamos no início dessas discussões na Educação Matemática, como salientado por Devlin (2012), mas a pandemia da Covid-19 nos mostrou que precisamos avançar nas autorias digitais de professores e estudantes. Nesse sentido, o *AnyLogic* tem apresentado potencial para o desenvolvimento de uma aprendizagem heurística, tanto

matemática quanto computacional.

Essa ferramenta apresentou-se como uma possibilidade de modelar o nível de preocupação de indivíduos infectados utilizando características de um modelo epidemiológico em uma comunidade. Essa metodologia pode ser motivadora para os estudantes de Licenciatura em Matemática no sentido de interpretar uma epidemia que é tema da atualidade. Além disso, devido à facilidade de utilização do *software*, é possível que os licenciados sejam multiplicadores desta tecnologia para os estudantes da Educação Básica.

## Referências

- 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 6ª 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 Noturna: 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 [NUSOL]*, 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 generation: Blending active learning, technology 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, RS: 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 *softwares* 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.