

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

EVANDRO ALBERTO ZATTI

**GenIA: PLATAFORMA PARA CONSTRUÇÃO DE OBJETOS DE
APRENDIZAGEM DE MATEMÁTICA QUE FAZ USO DE PROGRAMAÇÃO
INTUITIVA E É ASSISTIDA POR INTELIGÊNCIA ARTIFICIAL**

CURITIBA

2023

EVANDRO ALBERTO ZATTI

**GenIA: PLATAFORMA PARA CONSTRUÇÃO DE OBJETOS DE
APRENDIZAGEM DE MATEMÁTICA QUE FAZ USO DE PROGRAMAÇÃO
INTUITIVA E É ASSISTIDA POR INTELIGÊNCIA ARTIFICIAL**

**GenIA: platform for constructing mathematics learning objects that makes use
of intuitive programming and is assisted by artificial intelligence**

Tese apresentada como requisito para obtenção do título de Doutor em Formação Científica, Educacional e Tecnológica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Prof. Dr. Marco Aurélio Kalinke

CURITIBA

2023



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



EVANDRO ALBERTO ZATTI

GENIA: PLATAFORMA PARA CONSTRUÇÃO DE OBJETOS DE APRENDIZAGEM DE MATEMÁTICA QUE FAZ USO DE PROGRAMAÇÃO INTUITIVA E É ASSISTIDA POR INTELIGÊNCIA ARTIFICIAL

Trabalho de pesquisa de doutorado apresentado como requisito para obtenção do título de Doutor Em Ensino De Ciências E Matemática da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Ensino, Aprendizagem E Mediações.

Data de aprovação: 03 de Agosto de 2023

Dr. Marco Aurelio Kalinke, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Gerson Pastre De Oliveira, Doutorado - Pontifícia Universidade Católica de São Paulo (Pucsp)

Dra. Ivanete Zuchi Siple, Doutorado - Fundação Universidade do Estado de Santa Catarina (Udesc)

Dr. Marcelo Souza Motta, Doutorado - Universidade Tecnológica Federal do Paraná

Dra. Viviane Dal Molin, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 03/08/2023.

Dedico esta pesquisa aos meus pais, Carlos e Ivete,
pelo constante reforço da importância
dos valores da Educação e da História.

AGRADECIMENTOS

Certamente estes parágrafos não irão acolher todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já, peço desculpas àquelas que não estão evidenciadas entre estas palavras, mas sabem que fazem parte do meu pensamento e de minha gratidão.

Agradeço ao meu orientador Prof. Dr. Marco Aurélio Kalinke, a quem carinhosamente chamo de “magnânimo”, pela condução impecável nesta jornada. Suas contribuições foram decisivas para a concretização deste feito.

Aos professores Doutores Gerson Pastre de Oliveira, Ivanete Zuchi Siple, Marcelo de Souza Motta e Viviane Dal Molin, que aceitaram participar das bancas de qualificação e defesa, contribuindo significativamente com o direcionamento do percurso e diretamente com a versão final deste texto.

Aos demais professores e funcionários da UTFPR, pelo aprendizado, cooperação e provocações, imprescindíveis ao processo.

Às integrantes do macroprojeto, Fabíola Stavny, Renata Balbino, Silvana Mattos e Sônia Silva, pela parceria nos momentos de angústia e de glória.

Aos colegas do GPTEM, pelas constantes contribuições e apoio na validação do produto e, em especial, à Flávia Suchek e a Taniele Loss pelo acolhimento e incentivo em ingressar no programa.

Gostaria de deixar registrado, também, incomensurável reconhecimento à minha família, cujo apoio e exímia paciência serviram de alicerce para transpor esse desafio. Carlos, Ivete, Angela, Carla e Mayara, amo vocês incondicionalmente.

Enfim, a todos os que de alguma forma contribuíram para a realização desta pesquisa.

*non lasciare andare un giorno
per ritrovar te stesso
perchè la vita è adesso.*

(BAGLIONI, 1985).

[não deixe passar um dia sequer
para se encontrar
porque a vida é agora]

RESUMO

Em constante evolução, a humanidade tem passado, sob o olhar de diferentes filósofos, por algumas eras ou marcos políticos, econômicos e sociais. A tecnologia acompanha esse processo, atuando como protagonista ou pivô em inúmeras tarefas do cotidiano. A sociedade estaria hoje vivenciando a era do conhecimento, na qual preconiza-se o uso das tecnologias digitais em diversas esferas, dos algoritmos triviais à inteligência artificial. Na Educação, observa-se o uso de diferentes técnicas envolvendo inteligência artificial, que vão desde o suporte a gestão educacional, até as contribuições com os processos de ensino e de aprendizagem. Na Educação Matemática, antes mesmo da popularização da inteligência artificial, o uso do computador já suportava os processos didáticos por meio de uma ampla gama recursos, como por exemplo: as linguagens de programação intuitivas, a internet, os objetos de aprendizagem e outros conteúdos multimídia, como áudio e vídeo. Diante desse cenário, o objetivo desta pesquisa é investigar como recursos de inteligência artificial, aliados à programação intuitiva, podem ser explorados na criação de uma plataforma destinada à construção de objetos de aprendizagem de Matemática. O produto educacional, vinculado à investigação, é uma plataforma para construção de objetos de aprendizagem de Matemática, que faz uso da programação intuitiva e é assistida por inteligência artificial. O movimento da pesquisa foi orientado por dois aportes metodológicos: a Pesquisa em Design Educacional, que considera projetos e intervenções no contexto educacional, e o processo de prototipação, que, sob as diretrizes da Engenharia de Software, orienta os aspectos técnicos da construção de software por meio de sucessivas iterações e versões para fins de experimentação e avaliação. A pesquisa integra um macroprojeto, o qual envolve outras pesquisas correlatas, que incluem compreensões sobre os aportes teórico-filosóficos das temáticas envolvidas e o estudo dos aspectos ergonômicos da plataforma. A plataforma, denominada GenIA, baseia-se nos pilares da programação intuitiva para possibilitar a construção de objetos de aprendizagem de Matemática. A construção dos objetos é feita por meio de fluxograma e assistida por algoritmos de inteligência artificial, fazendo uso de aprendizado de máquina. O produto educacional foi aplicado em diferentes contextos, tendo sido avaliado por professores de Matemática integrantes do grupo de pesquisa ao qual esta pesquisa está vinculada, por estudantes do curso de Bacharelado em Engenharia de Software de uma universidade particular e por estudantes concluintes do curso de Licenciatura em Matemática da Universidade Tecnológica Federal do Paraná (UTFPR). A partir dos resultados da pesquisa, concluiu-se que a GenIA, enquanto produto educacional concretizado, testado sob o ponto de vista técnico, avaliado por pesquisadores da Educação Matemática, e usado como recurso em ambiente educacional real, revela que a tese levantada é viável, sendo possível, sim, a criação de OA de Matemática com uso de programação intuitiva e técnicas de inteligência artificial. Percebeu-se, contudo, que a proposta da pesquisa, manifestando vertentes a serem exploradas no âmbito educacional, causou certo receio por parte dos pesquisadores, o que evidencia a importância de que se tenham mais pesquisas interdisciplinares envolvendo IA e Educação, para que alguns preceitos sejam desmistificados.

Palavras-chave: genia; inteligência artificial; aprendizado de máquina; objetos de aprendizagem; plataforma; programação intuitiva.

ABSTRACT

In constant evolution, humanity has passed under the eyes of different philosophers through some political, economic, and social eras or milestones. Technology follows this process, acting as a protagonist or supporting player in countless everyday tasks. Society is now experiencing the age of knowledge, in which the use of digital technologies in various spheres is advocated, from trivial algorithms to artificial intelligence. In Education, the use of different artificial intelligence techniques is observed, ranging from supporting educational management to contributions to the teaching and learning processes. In Mathematics Education, even before the popularization of artificial intelligence, the use of the computer already supported the didactic processes through a wide range of resources, such as: intuitive programming languages, the internet, learning objects and other multimedia content, such as audio and video. Given this scenario, the objective of this research is to investigate how artificial intelligence resources, combined with intuitive programming, can be explored in the creation of a platform for the construction of Mathematics learning objects. The educational product linked to the investigation is a platform for building mathematics learning objects, which makes use of intuitive programming and is assisted by artificial intelligence. The research movement is guided by two methodological contributions: Educational Design Research, which considers projects and interventions in the educational context, and the prototyping process, which, under the directives of Software Engineering, guides the technical aspects for the construction of software through successive iterations and versions for experimentation and evaluation purposes. The research is part of a macro-project, which involves other related researches, that include understanding of the theoretical-philosophical contributions of the themes involved and the study of the ergonomic aspects of the platform. The platform, called GenIA, is based on the pillars of intuitive programming to enable the construction of Mathematics learning objects. The construction of objects is done using flowchart and is assisted by artificial intelligence algorithms with machine learning. The educational product was applied in different contexts, having been evaluated by Mathematics professors who are members of the research group to which this research is linked, by students of the Bachelor's degree in Software Engineering at a private university and by students completing the Bachelor's degree in Mathematics from the Federal Technological University of Paraná (UTFPR). From the research results, it was concluded that GenIA, as a realized educational product, tested from a technical point of view, evaluated by Mathematics Education researchers, and used as a resource in a real educational environment, reveals that the thesis raised is viable, and it is possible to create Mathematics OA using intuitive programming and AI techniques. It was noticed, however, that the research proposal, expressing aspects to be explored in the educational sphere, caused some concern on the part of researchers, which highlights the importance of having more interdisciplinary research involving AI and Education, so that some precepts be demystified.

Keywords: genia; artificial intelligence; machine learning; learning objects; platform; intuitive programming.

LISTA DE ILUSTRAÇÕES

Figura 1 – GenIA: tela principal	26
Figura 2 – Pesquisas sobre Uso de TD na Educação Matemática	28
Figura 3 – Iterações de ciclos de design sistemáticos	31
Figura 4 – Etapas da prototipação	38
Figura 5 – Modelo de prototipação	40
Figura 6 – Histórico de uso de linguagens	46
Figura 7 – Quadrado desenhado por meio de programação Logo.....	70
Figura 8 – Blocos gráficos no Scratch.....	73
Figura 9 – Blocos gráficos e telas no App Inventor	74
Figura 10 – Blocos que indicam a ação de somar dois números	76
Figura 11 – Interface do Scratch	77
Figura 12 – Interface da GenIA	78
Figura 13 – Evidências do pilar da similaridade na GenIA	79
Figura 14 – Projeto Scratch para somar dois números	80
Figura 15 – Comparativo entre blocos no Scratch e trecho de código em C#	81
Figura 16 – Exemplo de fluxograma.....	84
Figura 17 – Fluxograma na GenIA	85
Figura 18 – Evidências do pilar da visualização na GenIA.....	86
Figura 19 – Apresentação do Pi e seus comandos	87
Figura 20 – Analogia entre neurônio biológico e modelo de neurônio artificial	89
Figura 21 – Um dos neurônios da SNARC.....	90
Figura 22 – Exemplo de RNA com duas camadas interconectadas.....	90
Figura 23 – Modelo de RNA de aprendizagem profunda	92
Figura 24 – Prospecção das tendências em IA na Educação até 2030	100
Figura 25 – Criação de contexto para treinamento da IA	106
Figura 26 – Treinamento e uso da IA	107

LISTA DE QUADROS

Quadro 1 – Características das pesquisas que utilizam PDE	33
Quadro 2 – Fatores considerados na escolha da solução	44
Quadro 3 – Índice TIOBE para junho de 2023	45
Quadro 4 – Principais características dos OA	54
Quadro 5 – Características da dimensão educacional	55
Quadro 6 – Características da dimensão técnica	56
Quadro 7 – Características de qualidade de um OA	57
Quadro 8 – Características adicionais de qualidade de um OA	58
Quadro 9 – Metodologias para a produção de OA	59
Quadro 10 – Fases e etapas da MPEDUC	59
Quadro 11 – Classificação dos OA de acordo com os objetivos pedagógicos.....	61
Quadro 12 – Ambientes digitais para construção de OA.....	62
Quadro 13 – Formas de notação de um algoritmo	65
Quadro 14 – Os diferentes paradigmas de programação	67
Quadro 15 – Algumas áreas de estudo envolvendo IA	95
Quadro 16 – AIEd e o <i>framework</i> SAMR.....	97
Quadro 17 – Três paradigmas da AIEd.....	98
Quadro 18 – Tendências tecnológicas da IA na Educação.....	100
Quadro 19 – Algoritmos para treinamento do ML.NET	107

LISTA DE ABREVIATURAS E SIGLAS

AIEd	- <i>Artificial Intelligence in Education</i> (Inteligência Artificial na Educação)
AVA	- Ambiente(s) Virtual(is) de Aprendizagem
BASIC	- <i>Beginner's All-purpose Symbolic Instruction Code</i> (Código de Instrução Simbólica para Qualquer fim dos Iniciantes)
Capes	- Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CEdMA	- <i>Computer Education Management Association</i> (Associação de Gestão de Ensino de Informática)
GPINTEDUC	- Grupo de Pesquisa em Inovação e Tecnologias na Educação
GPTEM	- Grupo de Pesquisas sobre Tecnologias na Educação Matemática
HTML	- <i>Hypertext Markup Language</i> (Linguagem de Marcação de Hipertexto)
IA	- Inteligência Artificial
IDE	- <i>Integrated Development Environment</i> (Ambiente de Desenvolvimento Integrado)
IEC	- <i>International Electrotechnical Commission</i> (Comissão Eletrotécnica Internacional)
IEEE	- <i>Institute of Electrical and Electronics Engineers</i> (Instituto de Engenheiros Elétricos e Eletrônicos)
INPI	- Instituto Nacional da Propriedade Intelectual
ISO	- <i>International Organization for Standardization</i> (Organização Internacional para Padronização)
KDD	- <i>Knowledge Discovery in Databases</i> (Descoberta de Conhecimento em Bases de Dados)
LORI	- <i>Learning Object Review Instrument</i> (Instrumento de Revisão de Objeto de Aprendizagem)
MIT	- <i>Massachusetts Institute of Technology</i> (Instituto de Tecnologia de Massachusetts)
ML	- <i>Machine Learning</i> (Aprendizado de Máquina)
MOOC	- <i>Massive Open Online Course</i> (Curso Online Aberto Massivo)
MPEDUC	- Metodologia de Produção de um Objeto de Aprendizagem na Dimensão Educacional
MS	- Mapeamento Sistemático

OA	- Objeto(s) de Aprendizagem
PDE	- Pesquisa em Design Educacional
PLN	- Processamento de Linguagem Natural
PPGECM	- Programa de Pós-Graduação em Educação em Ciências e em Matemática
PPGFCET	- Programa de Pós-Graduação em Formação Científica, Educacional e Tecnológica
RBC	- Raciocínio Baseado em Casos
RIUT	- Repositório Institucional da Universidade Tecnológica Federal do Paraná
RNA	- Rede(s) Neural(is) Artificial(is)
Seed	- Secretaria Estadual de Educação do Estado do Paraná
SGBD	- Sistema Gerenciador de Banco de Dados
SIT	- <i>System Integration Test</i> (Teste de Integração de Sistema)
SNARC	- <i>Stochastic Neural Analog Reinforcement Calculator</i> (Calculadora de Reforço Analógico Neural Estocástico)
STI	- Sistema(s) Tutorial(is) Inteligente(s)
TD	- Tecnologia(s) Digital(is)
TI	- Tecnologia da Informação
TIC	- Tecnologia(s) da Informação e Comunicação
TIOBE	- <i>The Importance Of Being Earnest</i> (A Importância de Ser Sincero)
UAT	- <i>User Acceptance Test</i> (Teste de Aceitação do Usuário)
UFPR	- Universidade Federal do Paraná
UFSC	- Universidade Federal de Santa Catarina
USP	- Universidade de São Paulo
UT	- <i>Unity Test</i> (Teste de Unidade)
UTFPR	- Universidade Tecnológica Federal do Paraná

SUMÁRIO

AGRADECIMENTOS	5
1 PRIMEIRA PESSOA: DO SINGULAR AO PLURAL	14
2 TERCEIRA PESSOA: UM OLHAR SOBRE O PROBLEMA	20
2.1 Tecnologias Digitais e Educação Matemática.....	22
2.2 Questão norteadora.....	24
3 ASPECTOS METODOLÓGICOS	27
3.1 O Macroprojeto	28
3.2 Pesquisa em Design Educacional (PDE)	30
3.3 Processo de software.....	34
3.4 Seguindo o fluxo.....	42
3.5 Aplicação do produto.....	49
4 OBJETOS DE APRENDIZAGEM	53
4.1 Dimensões e características dos OA.....	55
4.2 Metodologias para produção de OA	58
4.3 Classificação dos OA	60
4.4 Ambientes para construção e compartilhamento de OA	62
5 PROGRAMAÇÃO INTUITIVA	64
5.1 Programação de Computadores.....	65
5.2 Programação e fases das TD no contexto educacional	69
5.3 Software para programação na Educação	72
5.4 Os pilares da programação intuitiva	75
5.5 Programação intuitiva na GenIA.....	76
6 INTELIGÊNCIA ARTIFICIAL	88
6.1 IA Conexionista.....	89
6.2 IA Simbólica	93
6.3 Outras áreas de estudo envolvendo IA.....	94
6.4 IA na Educação	96
6.5 IA na GenIA.....	103
7 CONSIDERAÇÕES	109
7.1 Recomendações para trabalhos futuros	112
REFERÊNCIAS	115

1 PRIMEIRA PESSOA: DO SINGULAR AO PLURAL

Considero esta pesquisa um arcano maior. Não no sentido de ser um grande mistério, mas em ser um dos propulsores de grandes transformações na minha trajetória enquanto ser humano, e que não é um acaso, mas sim, uma predestinação.

Dentre os inúmeros valores que herdei de meus pais, penso que um deles possa ajudar a esclarecer essa dedução: a história. Meu pai, um incessante pesquisador dos lugares onde viveu, deixou em mim a estampa do quão importante é conhecer o contexto no qual estamos inseridos, olhando para nossas origens e percursos, antes de traçar objetivos ou de explorar possibilidades.

Por isso, a cada novo desafio que a vida me apresenta, e que me parece ter uma importância considerável, procuro responder à pergunta “por que isso agora?” lançando um olhar minucioso sobre o caminho já trilhado. Convido o leitor a acompanhar um pouco de minha história até este importante passo que estou dando.

Desde a infância tenho fascínio por elétrica, eletrônica, computação e áreas correlatas, que na época eu chamava simplesmente de tecnologia. Atualmente, por razões óbvias, não posso reduzir tecnologia a aparatos automatizados. É fato que o interesse sempre esteve ali, constantemente denunciado pelo brilho dos meus olhos, quando via qualquer aparelho em funcionamento. Do liquidificador ao aspirador, a cada desmonte e remonte, devo confessar que sobravam peças ou parafusos, o que me impulsiona trazer uma pessoa muito importante para o contexto: minha mãe.

De filhos, éramos eu e duas irmãs, a quem raramente era dada a opção do ócio. Tenho muito a agradecer à minha mãe, que munida de uma sensibilidade ímpar, foi capaz de aproximar com maestria a vontade e habilidade dos filhos com as possibilidades de atuação profissional. Aos dez anos de idade fui matriculado concomitantemente em uma escola de Inglês e em outra de Informática. Nesta última, aprendi a programar na Linguagem Logo, época em que sequer sonhava com a existência de Piaget ou de sua teoria construtivista. Mas o Papert, com seu construcionismo, era meu ídolo. Assim como meu pai, que me presenteou com um computador pessoal, contribuindo significativamente para o aumento contínuo de meu interesse pelo universo do hardware e software.

Certa vez, diante de minha resposta a uma atividade escolar que foi solicitada, fui convidado a comparecer à sala da coordenação pedagógica. Eram meados da

década de 1980 e eu estava na quinta série primária (hoje sexto ano do Ensino Fundamental). A professora havia solicitado que informássemos a profissão que gostaríamos de seguir, a quem eu respondi com “mecatrônica”. Quando cheguei na antessala da coordenação, outra colega estava apavorada, se questionando qual o problema em querer ser astronauta. Para ela, explicaram que astronauta não era uma profissão que ela poderia simplesmente cursar uma graduação e passar a exercer quando se formasse. Para mim, perguntaram o que era aquela palavra esdrúxula. Respondi que era uma faculdade (é como nos referimos à graduação nessa idade) que estavam ofertando na Universidade de São Paulo (USP), e que o nome derivava da combinação de mecânica com eletrônica, e que nela eu poderia aprender a construir robôs. Diante da minha explicação, a expressão no rosto da orientadora virou um misto de interrogação com alívio, enquanto eu pensava “como pode ela não saber o que é?”. Saí da sala exalando certa arrogância, tendo a certeza de que eu estava à frente do meu tempo.

Depois de concluir o ensino fundamental, cursei apenas seis meses do segundo grau (atual Ensino Médio) em Educação Geral, em um colégio próximo de minha residência. Antes de concluir o primeiro ano, solicitei transferência para outro colégio que ofertava curso técnico de processamento de dados. O país vivia um momento de tecnicismo educacional, e eu aproveitei para me aproximar dos robôs.

Até este ponto da leitura, imagino que o leitor já tenha conseguido visualizar a Informática em meus passos. E a Educação Matemática? De alguma forma, ela já se fazia presente. Eram tantos os colegas com dificuldade em resolver polinômios e equações, e minha facilidade com números era tão evidente, que durante o ensino médio comecei a ministrar aulas particulares de Matemática. Porém, mesmo tendo um bom desempenho na disciplina, naquele momento, eu já acumulava certa inquietação com os processos de ensino e de aprendizagem de Matemática. Os professores com os quais tive contato apresentavam a teoria, ensinavam os algoritmos, mas pouco demonstravam sobre sua aplicabilidade. Em paralelo, comecei a ministrar aulas de inglês, particulares, e posteriormente em cursos de curta duração, fatos que contribuíram com meu olhar para perspectivas pedagógicas.

Ainda antes de concluir o ensino médio, eu havia abandonado a ideia de cursar mecatrônica na graduação, por valor familiar materno (prefiro colocar assim), que rezava que eu deveria cursar o ensino superior na cidade onde estava morando. Naquela época, Curitiba não tinha o curso que me apresentaria diretamente a

robótica, então prestei concurso vestibular para diferentes cursos relacionados à Tecnologia da Informação. Optei por seguir com Tecnologia em Processamento de Dados, o que me fez perceber também que eu não tinha me distanciado tanto assim dos meus ideais mecatrônicos.

Se, antes, minhas inquietações tinham origem na forma com que os professores apresentavam a aplicabilidade dos conceitos, a graduação trouxe um novo viés. Nela, os professores aproximavam a teoria da prática, porém de uma maneira que me provocava um novo ponto de reflexão: a computação quase sempre aplicada à... computação. Era como se os algoritmos precisassem ser aprimorados para serem incorporados a soluções computacionais já existentes. E as outras áreas?

Enquanto cursava a graduação, fui estagiário no laboratório de informática da instituição onde estudava. Eu era um dos responsáveis por deixar os computadores em pleno funcionamento para as aulas. Naquele momento, também, a instituição recebia seus primeiros computadores, interconectados em rede, para auxiliar nas atividades administrativas: secretaria, tesouraria, coordenações. Eventualmente, entre o preparo de um equipamento e outro, eu comecei a ser acionado para prestar suporte às atividades dos funcionários, que estavam engatinhando no uso do computador. Mais uma vez, minha preocupação em ensinar se apresentava, e eu fui convidado a organizar pequenas turmas de ensino de microinformática para parte dos funcionários da instituição. Em uma das aulas, o dono da instituição, e posteriormente reitor da universidade que viria a se tornar, proferiu-me uma frase que direcionou minha carreira. Foi mais ou menos assim: “você tem jeito pra coisa, deveria investir no seu futuro como professor”. Vale mencionar que, à época, em paralelo, eu ministrava cursos de microinformática em escolas especializadas.

Me formei em 1998 e, ainda antes disso, havia decidido que seguiria carreira como professor universitário. Por esse motivo, não me preocupei em cursar qualquer especialização, e comecei a busca por programas de mestrado que tivessem boa avaliação pela Capes. Foi então que, no ano 2000, ingressei no Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina (UFSC). Era uma área um pouco diferente da minha, mas ainda no âmbito das Ciências Exatas, e era uma oportunidade que eu tinha e que não perderia.

Ainda com aquela inquietação em mente, de não usar a computação para si própria, resolvi aplicar meus conhecimentos de programação em uma área diferente. Obtive, em 2002, o título de mestre em engenharia de produção, com um trabalho

que, para a época, causou um certo alvoroço. Além da dissertação, eu havia criado um software que fazia uso de realidade virtual, para que o estudante de graduação em odontologia pudesse visualizar e manipular dentes em três dimensões, fazendo uso de luva e capacete. Eu havia criado meu primeiro objeto de aprendizagem (OA), ainda que não conhecesse o termo e sequer uma possível definição.

Com o título de mestre, fui convidado a lecionar em disciplinas diversas dos cursos de graduação em Ciência da Computação, Engenharia da Computação e Tecnologia em Processamento de Dados, que naquele ano já dava espaço para o curso de Bacharelado em Sistemas de Informação. Dei continuidade às pesquisas com realidade virtual por mais algum tempo, mas sem perspectivas de evoluir, pois os equipamentos tinham custo elevado e a instituição optou por não investir em atualizações. A partir daí, durante quase 20 anos, dediquei-me primordialmente à sala de aula, com alguma contribuição em outro campo de estudo que viria a conquistar minha atenção: a Inteligência Artificial (IA).

Penso que o universo nos coloca em situações recorrentes para que tenhamos a oportunidade de optar por trilhar caminhos diferentes diante de desafios semelhantes. Esse ir e devir não se faz diferente com pessoas. Por razões que desconhecemos, há pessoas que entram em nossas vidas, permanecem por um tempo e se vão, para nunca mais voltarem. Em contrapartida há pessoas que, por mais que pareçam sair definitivamente de nossa jornada, em dado momento ressurgem por ocasiões inusitadas. Aqui, falo em especial do Prof. Dr. Marco Aurélio Kalinke, hoje meu orientador e amigo.

Em 2004, ingressei como professor em uma faculdade privada local, ministrando aulas para o curso de bacharelado em Sistemas de Informação. Passados três anos que eu lecionava naquela instituição, o prof. Kalinke assumiu a direção da faculdade. Em 2011, ele deixava o posto em busca de novos horizontes. Poderia ter sido aquele momento em que nossos caminhos se distanciariam, sem perspectivas de volta. Mas não foi assim.

Em 2017, o prof. Kalinke estava concluindo seu pós-doutorado na Itália e, por meio de uma rede social, publicamente solicitou auxílio de alguém que pudesse contribuir com aspectos técnicos na tradução de um software, especificamente na linguagem de programação Visual Basic. Contribuí com a tradução, o prof. Kalinke concluiu seu pós-doutorado e, tão logo retornou ao Brasil, me convidou para integrar

o Grupo de Pesquisa sobre Tecnologias na Educação Matemática (GPTeM¹). E lá estava eu novamente em contato não só com o prof. Kalinke, mas com a Educação Matemática.

Participando das reuniões do GPTeM, tive contato com pessoas maravilhosas, em sua maioria professores de Matemática, que me abriram oficialmente as portas desse universo. Com alguns, estabeleci laços de amizade, e tive a oportunidade de compreender suas motivações e anseios, e me senti em casa para, mais uma vez, trazer a semente da Computação para germinar em novo solo fértil.

E foi essa aproximação, esse acolhimento, juntamente com outros diversos fatores e objetivos de vida, que me motivaram a elaborar um projeto de pesquisa visando ingressar no doutorado do PPGFCET. Dentre as temáticas de pesquisa do GPTeM, uma, em especial, chamou minha atenção: os OA. Não era a primeira vez que me via empolgado com a ideia de que os estudantes poderiam interagir com conteúdo digital que pudesse proporcionar aprendizagem personalizada. Foi então que, em conjunto com quem viria a ser meu orientador, emergiu a ideia de que a IA poderia contribuir com essa personalização, mas que, antes disso, ela poderia assistir o processo de construção dos OA.

O processo seletivo aconteceu, a vaga foi ofertada, o projeto estava em consonância com os horizontes de pesquisa do GPTeM e do PPGFCET, o que me levou a ser aceito no programa. Sob a linha de Educação Matemática, iniciei mais uma trajetória de vida e de parceria com o prof. Kalinke.

Antes de adentrar os aspectos acadêmicos da pesquisa, não posso deixar de mencionar que a vida tentou nos aproximar em outros momentos. Na época em que eu pleiteava ingressar no ensino superior, ele era professor de Matemática no ensino médio, em destaque no tão conhecido “terceirão”, termo que se referia ao terceiro ano do antigo Segundo Grau com enfoque preparatório para o concurso vestibular. Porém, tal modalidade se fazia presente em escolas que ofereciam cursos de Educação Geral, mas isso não era uma prática no ensino técnico que eu cursava.

Não fosse o bastante, recentemente descobri que o prof. Kalinke e eu cursamos graduação na mesma instituição de ensino. Concluí o curso de Tecnologia

¹ GPTeM – Grupo de Pesquisa sobre Tecnologias na Educação Matemática, atualmente vinculado ao PPGFCET da UTFPR e ao Programa de Pós-Graduação em Educação em Ciências e em Matemática (PPGECM) da UFPR. <https://gptem5.wixsite.com/gptem>.

em Processamento de Dados em 1997, ano em que ele ingressou no curso de Matemática. No período em que o colega estudava, até o ano 2000, eu ainda trabalhava na universidade, e no mesmo campus (!), porém não chegamos a nos encontrar por lá. Reza a lenda que também foram grandes as chances de que iríamos fazer mestrado no mesmo programa de pós-graduação da UFSC, mas aparentemente ainda não era o momento para nossos caminhos se cruzarem.

Porém, como já antecipei, a vida me concedeu o privilégio de ter uma nova oportunidade de trabalhar com o prof. Kalinke, e reforço que não acredito ser um acaso. Minha trajetória, com raízes fortes na Computação e alguns devaneios pela Matemática precisava, sim, trespassar a jornada desse importante professor pesquisador da Educação Matemática e grande adepto do uso de tecnologias digitais (TD).

A conciliação dessas temáticas dá início às discussões e inquietações que pautam esta pesquisa, as quais convido o leitor a participar a partir do próximo capítulo.

2 TERCEIRA PESSOA: UM OLHAR SOBRE O PROBLEMA

Em constante evolução, a humanidade articula diferentes estratégias e paradigmas nas esferas política, econômica, social e cultural. Alguns fenômenos são decorrentes de transmutação gradativa, enquanto outros se apresentam de forma disruptiva. A observação e documentação dessas mudanças são recorrentes na literatura, sendo possível encontrar classificações da sociedade sob diferentes ópticas, percebendo-se, notória e incessantemente, a tecnologia como protagonista ou pivô desse processo evolucionário. O uso de tecnologia na automação de tarefas vai, por exemplo, desde a invenção da roda até o atual e cada vez mais comum uso de assistentes pessoais.

A indústria, por exemplo, teve sua primeira revolução no final do século XVIII, tendo como base tecnológica o carvão, fonte de energia para as máquinas a vapor e locomotivas. Por ter sido a primeira, hoje é referenciada como Indústria 1.0. No início do século XX surgem novas bases tecnológicas em substituição ao carvão, como a eletricidade e o petróleo, aumentando a produção em massa e colocando a Indústria em sua versão 2.0. Com a chegada dos anos 1970 surgem os computadores e a automação robótica, elevando a indústria para sua terceira versão.

Atualmente a sociedade presencia a Indústria 4.0, tendo como principal base tecnológica a conectividade, suportada pela internet e computação em nuvem, nos quais são utilizados dispositivos móveis e vestíveis para interação. O momento industrial estaria em transição para uma quinta versão, a Indústria 5.0, na qual humanos e robôs possam trabalhar juntos em uma sociedade mais saudável, inclusiva e igualitária, voltada para a customização. Esta seria uma perspectiva otimista sobre os resultados que poderiam advir de uma união harmônica entre indivíduo e máquina. Porém, existem adversidades que serão oportunamente comentadas neste trabalho.

Toffler (1980) já lançava um olhar para além da indústria, versando sobre períodos que a história registrou, sugerindo alguns marcos evolucionários, aos quais chamou de ondas. A primeira onda, por ele denominada Sociedade dos Músculos ou civilização agrícola, deu espaço à segunda onda, a que classificou como Sociedade da Riqueza ou civilização industrial. Por fim, segundo o autor, atualmente vive-se a terceira onda, a Sociedade do Conhecimento, que preconiza o armazenamento, transmissão e otimização da informação, em busca de sua melhor aplicabilidade.

As possibilidades e desafios proporcionados pela tecnologia encontram apoio nas ideias de Tikhomirov (1981), ao indicar que o indivíduo reorganiza os processos mentais criativos pelo uso da tecnologia:

Uma das teses centrais de Vygotsky é a de que os processos mentais nos seres humanos mudam na medida em que seus processos de atividade prática mudam (i.e., os processos mentais tornam-se mediados). [...] No uso de meios e sinais auxiliares (por exemplo, no fazer um entalhe numa vara para se lembrar), os humanos produzem mudanças nas coisas externas; mas estas mudanças subsequentemente têm um efeito nos seus processos mentais internos. (TIKHOMIROV, 1981, p. 264).

Atualmente, o indivíduo vive o coletivo, compartilhando informações em tempo real, permeado por redes móveis e internet. Com isso, referenciais de tempo e espaço foram ressignificados. Para Kenski (2003), o uso de software e a internet proporcionam uma interação entre pessoas e instituições para a construção de espaços de inteligência pessoal e coletiva.

Do analógico ao digital, o indivíduo busca constantemente por recursos que lhe permitam otimizar o tempo, deixando para a máquina a realização de atividades que demandem esforço físico. Desta maneira, sua atenção pode se voltar ao seu desenvolvimento e aprimoramento intelectual. (ZATTI; KALINKE, 2021, p. 75).

As TD possibilitam o desempenho não somente das tarefas que demandam esforço físico. Desde a popularização dos computadores pessoais até o uso constante dos dispositivos móveis nas atividades do cotidiano, as TD vêm acompanhando gerações por décadas. Em consonância com a terceira onda de Toffler (1980), elas adentram a esfera do conhecimento, estando cada vez mais presentes em uma sociedade cujo cotidiano é palco de incessante compartilhamento de informações.

Lévy (2007) considera sobre as possibilidades do saber e do conhecimento ante a realidade da grande rede:

Nossa relação material com o mundo se mantém por meio de uma formidável infraestrutura epistêmica e de software: instituições de educação e formação, circuitos de comunicação, tecnologias intelectuais com apoio digital, atualização e difusão contínua dos savoir-faire... Tudo repousa, a longo prazo, na flexibilidade e vitalidade de nossas redes de produção, comércio e troca de saberes (LÉVY, 2007, p. 19).

Lévy (2016) apresenta uma trajetória da comunicação e transmissão de informação, propondo uma divisão da civilização em quatro momentos. A primeira seria a civilização escriba, organizada politicamente nos palácios-templos, baseada na agricultura e conhecimentos sistemáticos, e que buscava a autoconservação dos símbolos.

A civilização escriba teria dado espaço a uma civilização letrada, politicamente fundamentada nos impérios, em que aparecem as religiões universais, a filosofia e a moeda. Naquele momento a busca era pela otimização da manipulação dos símbolos. Em um terceiro momento, viria a civilização tipográfica, organizada politicamente em estados-nação, focada na indústria e nas ciências naturais. Essa terceira civilização teve como principal característica a busca pela transmissão e reprodução automática dos símbolos.

Por fim, atualmente a sociedade estaria vivendo o quarto momento, a chamada sociedade algorítmica, com uma economia baseada na informação e no conhecimento. Nutre-se um período de revolução nas ciências humanas, devido a grande quantidade de dados acumulados sobre a atividade humana, resultado da ampla conectividade sustentada pela internet. O processamento e cruzamento desses dados pode alavancar progresso em diversas áreas, como a sociologia, psicologia social, economia etc.

Kenski (2003, p. 91) afirmava que “estamos vivenciando um momento de transição social que se reflete em mudanças significativas na forma de pensar e de fazer educação”. A transição apontada por Keski, em uma constante de evolução da sociedade, ainda pode ser observada, duas décadas depois. Em consonância com a cultura digital, as TD adentram o contexto educacional, podendo potencializar os processos de ensino e de aprendizagem por meio de perspectivas didáticas diferenciadas.

Esta pesquisa tem como fio condutor a utilização de TD no contexto educacional, em especial na Educação Matemática, um diálogo que será explorado a seguir.

2.1 Tecnologias Digitais e Educação Matemática

Kalinke, Mocrosky e Motta (2018, p. 67) pontuam que “A inserção das tecnologias no contexto educacional apresenta consigo uma série de desafios,

considerando que novas maneiras de se relacionar e interagir com o mundo ficam evidenciadas”.

Sobre o uso de tecnologia e seus impactos no contexto educacional, Borba, Scucuglia e Gadanidis (2018, p. 21) fazem uma transposição especificamente para a Educação Matemática: “As dimensões da inovação tecnológica permitem a exploração e o surgimento de cenários alternativos para a educação e, em especial, para o ensino e aprendizagem de Matemática”.

Possibilidades do uso de TD na Educação Matemática são objeto de estudo do GPTEM que, em seus trabalhos, explora o uso de TD como ferramentas de apoio aos processos de ensino e de aprendizagem de Matemática. As pesquisas vão desde a lousa digital até os OA, abrangendo outras áreas, tais como a Robótica Educacional e a IA.

À medida em que o mundo se conecta e se virtualiza cada vez mais, consolidando o ciberespaço discutido por Lévy (2011, 2018), o coletivo revela a diversidade, valorizando o indivíduo. Kalinke e Balbino (2016) pontuam sobre a individualidade nos processos educativos e a importância dos OA neste contexto:

Diante do cenário tecnológico que se apresenta no âmbito educacional, a utilização de tecnologias interativas aplicadas a educação permite ampliar a diversidade de abordagens, atender às diversas formas de aprendizagens e favorecer a construção de conhecimentos. Nesta perspectiva, o uso de objetos de aprendizagem como um recurso didático pode auxiliar nos processos de aprendizagem. (KALINKE; BALBINO, 2016, p. 24).

Nesse universo digital, em que cada vez mais informações pessoais e perfis de comportamento individuais são compartilhados e processados por servidores conectados em nuvem, emerge um ente que pode ser apontado como protagonista ou corresponsável: a IA.

Com seus primeiros passos em meados de 1950, a IA consolidou-se como um campo de estudo que tem contribuído com diversas outras áreas do conhecimento. Dos veículos autônomos ao reconhecimento de voz dos assistentes pessoais, pode-se perceber a sua presença em um número cada vez mais expressivo de atividades do cotidiano.

Percorrendo uma trajetória longa e compassada, a IA tem ganhado notoriedade exponencial nos últimos anos, em parte devido ao aumento substancial

da capacidade de processamento dos computadores. Tal feito possibilita que dados individuais sejam obtidos e relacionados na coletividade, identificando padrões de comportamento e consumo, trazendo para os sistemas a capacidade de sugerir produtos, serviços, conteúdos etc.

Na Educação, a conversa com a IA ainda é tímida, porém nesse contexto, ela suporta desde aplicações mais simples, como os mecanismos de busca, até soluções mais robustas, como os tutores inteligentes. Se em alguns casos a padronização ou direcionamento propostos pelos algoritmos pode se revelar interessante, por outro lado, os algoritmos de IA podem ser treinados quando executados repetidamente por um mesmo indivíduo, direcionando para a personalização. Esta segunda abordagem conversa diretamente com a proposta de aprendizagem diferenciada possibilitada pelos OA. Revela-se, aqui, uma dicotomia: padronizar para personalizar?

2.2 Questão norteadora

Ao refletir a respeito das possíveis contribuições da IA para a Educação Matemática, manifestam-se algumas inquietações: Como a IA pode contribuir com os processos de ensino e de aprendizagem de Matemática? Quais outras TD podem ser associadas à IA para potencializar tais contribuições? Que desafios podem emergir do trabalho conjunto entre a Educação Matemática e a Ciência da Computação?

É possível ainda, estender tais reflexões ao encontro da perspectiva de Pierre Lévy (2019):

Qual é o papel do indivíduo ou do pequeno grupo na transformação da memória coletiva? Haverá sistemas para devolver às pessoas uma imagem de suas operações cognitivas e assim por diante. Isso é o que chamo de perspectiva de uma inteligência coletiva reflexiva. [...] Assim, nossa ideia de inteligência coletiva não é mais uma inteligência universal emanando para os seres humanos, mas justamente o contrário, os seres humanos contribuindo para a construção de um intelecto comum e, claro, se beneficiando dele. (LOLLINI; FARLEY; LÉVY, 2019, pp. 6-7, tradução nossa)

O movimento desta pesquisa pode promover aprofundamento em algumas dessas inquietações quando busca responder à seguinte questão norteadora: Como recursos de IA, aliados à programação intuitiva, podem ser explorados na criação de uma plataforma destinada à construção de OA de Matemática?

Cabe, neste momento, destacar que esta pesquisa está vinculada a um programa de doutorado profissional. Uma das principais características dos programas de pós-graduação profissionais da área de ensino é a criação de um produto educacional que se relacione com uma pesquisa relativa à temática nele abordada.

A Área de Ensino é, portanto, essencialmente de pesquisa translacional, que transita entre a ciência básica e a aplicação do conhecimento produzido. Desse modo, busca construir pontes entre conhecimentos acadêmicos gerados na pesquisa em educação e ensino para sua aplicação em produtos e processos educativos voltados às demandas da sociedade e às necessidades regionais e nacionais. (BRASIL, 2019, p. 3).

De acordo com o relatório de avaliação quadrienal da Capes (BRASIL, 2017), os produtos possuem uma tipologia: mídias educacionais; protótipos educacionais e materiais para atividades experimentais; propostas de ensino; material textual (livros didáticos ou paradidáticos e outros); materiais interativos; atividades de extensão (cursos, oficinas e outros); desenvolvimento de aplicativos.

Valendo-se de liberdades que são conferidas a pesquisas dessa natureza, contudo orientadas pelo ineditismo que lhe é exigido, para este trabalho foi adotada uma concepção diversa do que é comumente praticado. Normalmente, pesquisas de doutorado profissional apresentam discussões teóricas e investigações, a partir das quais tem-se um produto como resultado. Entretanto, no caso deste trabalho, assume-se que o resultado principal é o produto, em formato de software, e que a ele se associa um trabalho teórico.

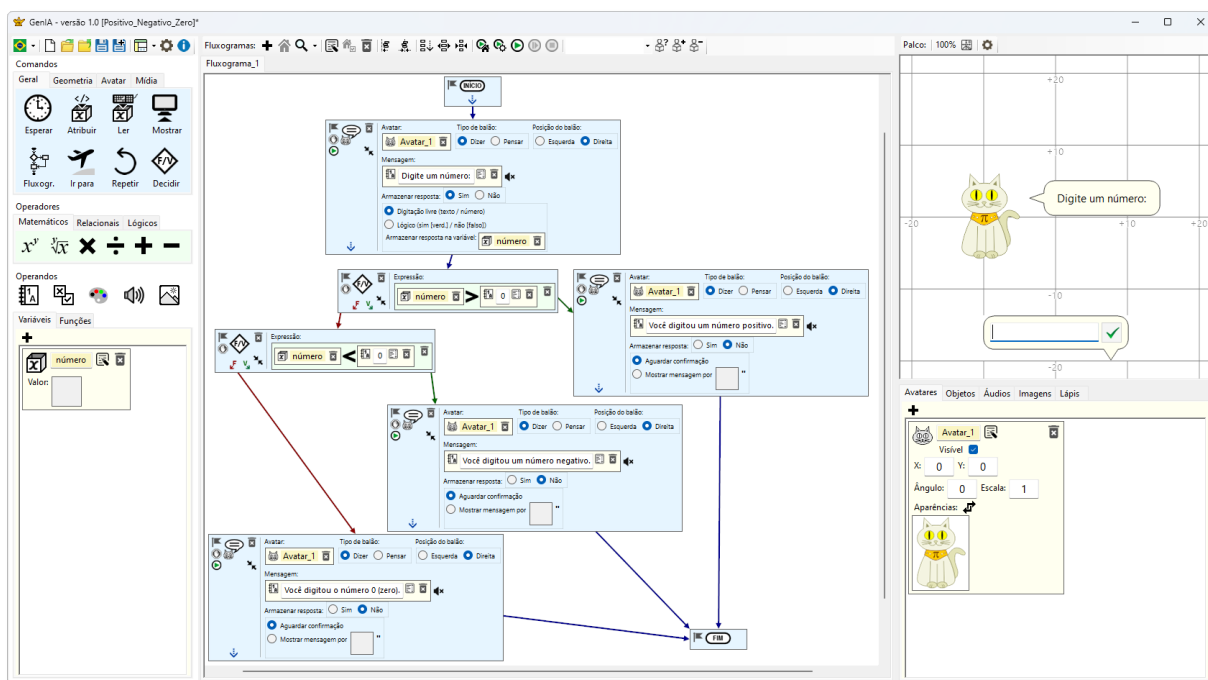
O produto educacional, que serviu como base para desvendar possibilidades a partir das inquietações apresentadas, foi uma plataforma para construção de OA de Matemática que faz uso de programação intuitiva e é assistida por IA. A plataforma, denominada GenIA², foi registrada no INPI³, sob o número BR512023001822-8, com certificado emitido em julho de 2023. Além disso, possui website próprio, com domínio registrado (plataformagenia.com), sob o qual estão hospedados, com acesso público e gratuito, além do software, outros artefatos resultantes desta pesquisa. Detalhes a

² GenIA é um acrônimo formado por Gen (iniciais de gênese) e IA (sigla para Inteligência Artificial).

³ Criado em 1970, o Instituto Nacional da Propriedade Industrial (INPI) é uma autarquia federal vinculada ao Ministério da Economia, responsável pelo aperfeiçoamento, disseminação e gestão do sistema brasileiro de concessão e garantia de direitos de propriedade intelectual para a indústria. Disponível em: <https://www.gov.br/inpi/pt-br>. Acesso em: 17 jun. 2023.

respeito do funcionamento dos comandos e de outros elementos de utilização estão disponíveis no documento de apresentação do produto. Recursos como tutoriais, vídeos e arquivos com exemplos de uso poderão ser obtidos no website da plataforma. A Figura 1 apresenta a tela principal da plataforma.

Figura 1 – GenIA: tela principal



Fonte: Autoria própria (2023)

Com esta pesquisa, busca-se defender a tese de que a IA pode ser uma aliada na construção de OA de Matemática com uso de programação intuitiva. Busca-se, também, uma alternativa de programação intuitiva diferente da programação visual adotada por outras plataformas. Não menos importante, busca-se relatar os desafios que podem emergir da conversa entre a Educação Matemática e a Ciência da Computação.

A seguir, são apresentados os aspectos metodológicos que serviram de direcionamento para tais buscas e compreensões.

3 ASPECTOS METODOLÓGICOS

Da mesma maneira que a delimitação do objeto de pesquisa é resultado de um percurso, a busca por compreensões teóricas e o desenvolvimento do produto também precisam ter elementos norteadores. O roteiro de viagem pode ser seguido rigorosamente como planejado, ou o aporte em novas terras pode trazer deslumbres e inquietações que sugerem desvios ou novas rotas. É importante que o caminho a ser explorado seja orientado por, mas não limitado a, trilhos da metodologia adotada. Seguir um fluxo pode levar o pesquisador a pontos de consolidação ou de divergência de ideias. A respeito desse caminhar, Bicudo e Borba (2005, p. 8) consideram que “O movimento da pesquisa gera ambiguidades internas e inerentes a todo processo em ação: modifica, diferencia e mantém.”

A pesquisa aqui apresentada é de caráter qualitativo. Segundo D’Ambrósio (2004, p. 10) “a pesquisa qualitativa, também chamada pesquisa naturalística, tem como foco entender e interpretar dados e discursos, mesmo quando envolve grupos de participantes.”

Para Bicudo e Costa (2019), essa modalidade de pesquisa considera que não é suficiente um relato de “como se faz”. Os autores apontam que se trata de um trabalho discursivo no qual são explicitadas também as dificuldades e controvérsias.

O discurso articula, de modo inteligível, o enredo – as perguntas formuladas – e evidencia o pensado, o materializado pela linguagem, ficando à disposição para ser retomado e vivificado pelas indagações formuladas mediante a dialética hermenêutica do pensar que se estabelece entre texto e leitor. Novas perguntas, outras compreensões e interpretações vão sendo produzidas e expostas. (BICUDO; COSTA, 2019, p. 15).

Esta pesquisa está inserida em um macroprojeto, que considera outras pesquisas de mestrado e doutorado envolvendo TD na Educação Matemática, todas sob a orientação do prof. Dr. Marco Aurélio Kalinke.

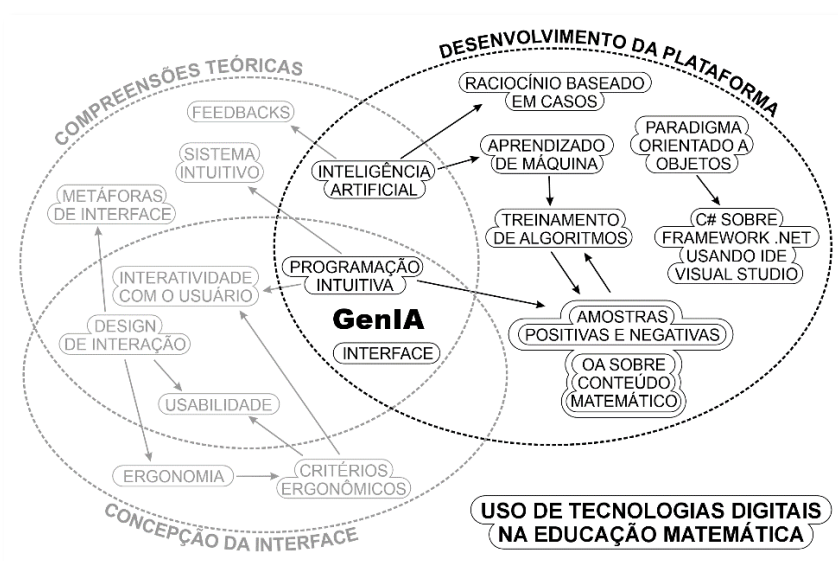
3.1 O Macroprojeto

Dentre as diferentes abordagens envolvendo as pesquisas integrantes do macroprojeto, as que mais se aproximam ao desenvolvimento da GenIA, são objeto de estudo de teses de doutorado vinculadas ao PPGECM da UFPR.

Uma das pesquisas de doutorado teve como objetivo estabelecer uma base teórica entre alguns dos assuntos abordados nesta pesquisa. Aquela pesquisa, intitulada “Em Busca de Compreensões sobre Inteligência Artificial e Programação Intuitiva na Educação Matemática”⁴, foi conduzida por Silvana Gogolla de Mattos, tendo sido defendida em novembro de 2022.

A outra pesquisa, ainda em andamento, visa a seleção de critérios ergonômicos que possam ser implementados em interfaces de plataformas assistidas por IA, voltadas para a construção de OA de Matemática. Tal pesquisa, inicialmente intitulada “Uma Proposta para Concepção de Interfaces para Plataformas Assistidas por Inteligência Artificial Fundamentada no Design de Interação e na Ergonomia”, está sendo conduzida por Renata Oliveira Balbino, tendo sido qualificada em fevereiro de 2023. A Figura 2 ilustra os aspectos de interesse das pesquisas de doutorado componentes do macroprojeto. Na imagem, a pesquisa aqui relatada é indicada pelo conjunto destacado em negrito.

Figura 2 – Pesquisas sobre Uso de TD na Educação Matemática



Fonte: Zatti et al. (2022, p. 262, adaptado)

⁴ Disponível em: <https://acervodigital.ufpr.br/handle/1884/80496>. Acesso em: 17 jun. 2023.

Além das pesquisas de doutorado, o macroprojeto integrou duas pesquisas de mestrado profissional, ambas vinculadas ao PPGFCET da UTFPR.

A primeira delas buscava identificar concepções de um grupo de professores que ensinam Matemática acerca da construção de OA de um determinado conteúdo matemático. Com dissertação intitulada “Um Olhar para a Concepção de Professores na Construção de Objetos de Aprendizagem de Matemática”⁵, a pesquisa foi conduzida por Fabiola Martins Stavny, com defesa em agosto de 2022. O produto educacional resultante dessa pesquisa foi um guia metodológico de utilização de OA sobre função polinomial do primeiro grau.

A outra pesquisa de mestrado tinha por objetivo buscar compreensões teórico-filosóficas a respeito das contribuições da IA para Educação Matemática. A dissertação dessa pesquisa, com viés filosófico, é intitulada “Perspectivas Teórico-Filosóficas sobre a Inteligência Artificial à Luz de Pierre Lévy: Ontologia, Desenvolvimento e Possibilidades em Processos Educacionais” e foi conduzida por Sonia Martins de Souza e Silva. A pesquisa teve como produto educacional resultante um glossário, em formato de e-book, abordando termos e conceitos de Lévy relacionados à pesquisa sobre IA nos processos educacionais. Esta última pesquisa foi defendida em maio de 2023 e o acesso público ainda está em processo de disponibilização pela biblioteca da UTFPR.

Os elementos que dão movimento às pesquisas integrantes do macroprojeto estão inseridos em um contexto educacional, especificamente a Educação Matemática. Por isso, sob uma perspectiva geral comum, buscou-se por metodologias de pesquisa voltadas a processos educativos.

Especificamente para a pesquisa aqui relatada, era necessário que a metodologia a ser adotada pudesse contribuir com o processo de avaliação do produto educacional, sob a óptica dos professores de Matemática. Vale ressaltar que, sob este viés, o olhar crítico do avaliador deveria concentrar-se nas potencialidades da GenIA sob a perspectiva de educador, e não de um usuário comum de software.

Era necessário, também, que a metodologia em questão possibilitasse revisitar itens já avaliados à medida em que tais itens fossem aprimorados, bem como ampliar as compreensões para novos itens quando adicionados.

⁵ Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/30187>. Acesso em 17 jun. 2023.

Diante das necessidades elencadas, e em consonância com outras pesquisas que integram o macroprojeto, foi adotada a Pesquisa em Design Educacional (PDE), apresentada a seguir. Depois de apresentadas as características da metodologia, será detalhado como ela serviu de fio condutor para a pesquisa.

3.2 Pesquisa em Design Educacional (PDE)

A PDE diz respeito ao estudo sistemático de concepção, desenvolvimento e avaliação de intervenções educacionais e, segundo Plomp (2013), é indicada para:

projetar e desenvolver uma intervenção (como programas, estratégias e materiais de ensino-aprendizagem, produtos e sistemas) como solução para um problema educacional complexo, bem como avançar nosso conhecimento sobre as características dessas intervenções e os processos para projetar e desenvolver ou, alternativamente, para projetar e desenvolver intervenções educacionais (sobre, por exemplo, processos de aprendizagem, ambientes de aprendizagem e similares) com o objetivo de desenvolver ou validar teorias. (PLOMP, 2013, p. 15, tradução nossa).

A adoção da PDE como metodologia pressupõe um caminho consolidado por três fases: preliminar, de prototipagem e avaliação.

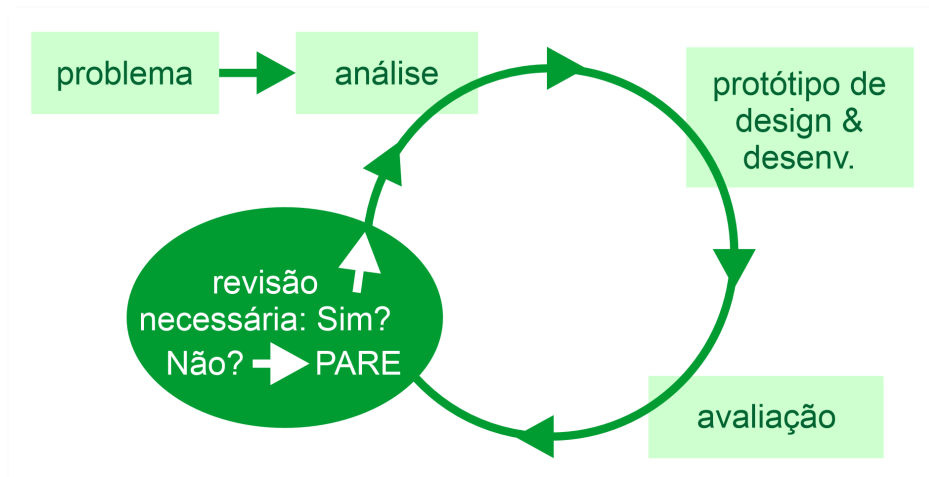
A fase preliminar prevê o cumprimento das seguintes tarefas: análise do contexto no qual a intervenção educacional será desenvolvida; identificação do problema educacional que busca ser solucionado; e revisão bibliográfica. Esta última é apontada como uma etapa importante no percurso das pesquisas qualitativas. “Ela é importante não só para que ‘não se reinvente a roda’, refazendo o que já está feito, mas também porque o exercício de encontrar lacunas em trabalhos realizados ajuda na ‘focalização da lente’ do pesquisador.” (ARAÚJO; BORBA, 2019, p. 45)

Na fase de prototipagem são elaborados protótipos (ou modelos) que irão sofrer iterações no decorrer da pesquisa. Espera-se que as iterações promovam aprimoramentos em tais protótipos.

A fase de avaliação considera que o protótipo deve passar por uma verificação, na qual é observado se ele cumpriu (ou não) os critérios estabelecidos. As iterações previstas na PDE são reforçadas por McKenney e Reeves (2012), ao afirmarem que os *insights* e as intervenções da PDE evoluem ao longo do tempo, por

meio de múltiplas iterações de investigação, desenvolvimento, testes e refinamento. A evolução por meio das iterações pode ser observada na Figura 3.

Figura 3 – Iterações de ciclos de design sistemáticos



Fonte: PLOMP (2013, p. 17, tradução nossa)

A PDE pode ser classificada em dois tipos de estudo, de acordo com seu propósito de investigação: estudo de desenvolvimento e estudo de validação. O estudo de desenvolvimento tem por objetivo desenvolver soluções para os problemas da prática educacional. O estudo de validação diz respeito à elaboração ou validação de uma teoria ou compreensão teórica.

Para Plomp (2013), esses dois tipos de estudo não são excludentes, e é possível desenvolver pesquisa combinando ambos. De um lado podem existir investigações das intervenções educacionais, e do outro pode emergir o conhecimento sobre tais intervenções.

A partir de um problema complexo e persistente, por exemplo na Educação Matemática, o grupo de pesquisa pode decidir aplicar os princípios de concepção (teorias locais⁶) resultantes de outros estudos na sua pesquisa. Ao fazer isso, não estão apenas desenvolvendo uma intervenção, mas ao mesmo tempo investigando a validade dos princípios de concepção (teoria) desenvolvidos em outro contexto para seu próprio contexto problemático. (PLOMP, 2013, p. 26, tradução nossa).

A produção científica pode ser, dependendo do objeto de estudo, multidisciplinar ou interdisciplinar. Sem adentrar nas nuances da transdisciplinaridade,

⁶ Na física, o princípio de localidade afirma que um objeto é influenciado diretamente pelo seu entorno. Uma teoria que considera o princípio de localidade é chamada de teoria local.

diferentes áreas de conhecimento podem se unir em prol de um objetivo, para resolver um problema, para responder a um questionamento.

Algumas dessas áreas, como a Física e a Biologia, são mais tradicionais, com uma base mais sólida, comprovada pela constância dos resultados frutíferos que tem produzido historicamente (CHALMERS, 1999). Estas são, indiscutivelmente, chamadas de ciências, pois a epistemologia se dá por rigoroso método científico. Outras áreas, por sua vez, são mais recentes, ou mais recentemente consideradas ciências, como por exemplo, a Ciência Política e a Ciência da Computação.

Esta pesquisa busca aproximar a Educação Matemática da Ciência da Computação, por laços que incluem a programação intuitiva e a IA. As teorias que embasam as diferentes áreas estariam, dessa maneira, sendo validadas de forma concomitante. Por consequência, o produto educacional (a plataforma) pode ser considerado a intervenção a que Plomp (2013) se refere.

Sob uma perspectiva epistemológica, a conversa entre as áreas estaria colocando o pesquisador no que Fleck (2010) denominou “coletivo de pensamento”:

Se definirmos “coletivo de pensamento” como a comunidade das pessoas que trocam pensamentos ou se encontram numa situação de influência recíproca de pensamentos, temos, em cada uma dessas pessoas, um portador do desenvolvimento histórico de uma área de pensamento, de um determinado estado do saber e da cultura, ou seja, de um estilo específico de pensamento. [...] Quanto mais nos aprofundarmos numa área do saber, tanto maior se torna o vínculo a um estilo de pensamento. (FLECK, 2010, pp. 82, 131).

Pesquisas que utilizam a PDE podem auxiliar na transposição das lacunas entre prática e teoria no contexto educacional (POWELL; ALI, 2018). Segundo Plomp (2013), diferentes autores concordam em algumas características que estariam presentes nesse tipo de pesquisa. O Quadro 1 apresenta uma compilação das principais características:

Quadro 1 – Características das pesquisas que utilizam PDE

Intervencionista	a pesquisa visa projetar uma intervenção em um cenário do mundo real.
Iterativa	a pesquisa incorpora ciclos de análise, concepção e desenvolvimento, avaliação e revisão.
Orientada a processo	o foco está na compreensão e melhoria das intervenções (um modelo de caixa preta de medição de entrada-saída é evitado).
Orientada a utilidade	o mérito de uma concepção é medido, em parte, por sua praticidade para usuários em contextos reais.
Orientada a teoria	a concepção é (pelo menos parcialmente) baseada em um arcabouço conceitual e em proposições teóricas, enquanto a avaliação sistemática de protótipos consecutivos da intervenção contribui para a construção da teoria.
Envolvimento de profissionais	a pesquisa envolve participação ativa ou colaboração com profissionais nas várias etapas e atividades da pesquisa – isso aumentará a chance de que a intervenção se torne de fato relevante e prática para o contexto educacional, o que aumenta a probabilidade de uma implementação bem-sucedida.

Fonte: PLOMP (2013, p. 20, tradução nossa, adaptado)

É possível avaliar se a PDE poderia ser, de fato, norteadora desta pesquisa ao se considerar se ela possui, de forma total ou parcial, cada uma das características citadas. No sentido de identificar se esta pesquisa possui características concordantes com a aplicação da PDE, surge o primeiro questionamento: é uma pesquisa intervencionista?

Uma vez que o principal resultado da pesquisa é uma plataforma para construção de OA, pode-se entender que é intervencionista, pois pretende-se que futuramente a plataforma seja utilizada para que professores possam criar conteúdo para ser utilizado por estudantes, tanto em ambientes virtuais quanto em salas de aula ou laboratórios com aprendizagem mediada por tecnologia.

A pesquisa é iterativa, pois ela incorpora os ciclos de análise, concepção e desenvolvimento, avaliação e revisão, que são repetidamente revisitados.

Além dos direcionamentos apresentados pela PDE, a criação da plataforma também seguiu diretrizes do desenvolvimento de software, que sugerem etapas análogas a tal metodologia. Atividades de verificação e validação propostas pela PDE, também presentes no desenvolvimento de software, sugerem o acompanhamento detalhado da execução de tarefas sob múltiplas entradas e gerando diferentes saídas, o que caracteriza a pesquisa como orientada a processo.

O uso de programação intuitiva revela que a pesquisa é orientada à utilidade, sobretudo por trazer praticidade na construção dos OA.

O objetivo de identificar as possíveis contribuições da IA para a Educação Matemática, sendo este o principal vínculo entre as diferentes áreas, coloca a pesquisa na condição de orientada a teoria. Isto porque, a cada novo protótipo gerado, foram feitas avaliações no sentido de observar tais contribuições.

Por fim, quem cria os OA e colabora com o aporte teórico para o desenvolvimento da pesquisa são os pesquisadores do GPTEM, em sua maioria professores de Matemática e pedagogos. Suas contribuições são de valor substancial para a pesquisa, revelando a sua importância no contexto da Educação Matemática. Esse suporte profissional deixa evidente que a pesquisa não apenas promove, mas é dependente do envolvimento de profissionais.

Além das contribuições proporcionadas pela PDE, tendo em vista que esta pesquisa tem como produto educacional a criação de um software, julgou-se necessário, também, considerar aspectos metodológicos do desenvolvimento de software.

Enquanto a PDE norteava as sucessivas avaliações e aprimoramentos do produto sob uma perspectiva educacional, também era fundamental que fossem observados aspectos técnicos do desenvolvimento. Tais aspectos incluem, por exemplo, identificar erros de programação ou necessidades de aprimoramento das funcionalidades, situações em que o professor contribuiu desempenhando o papel de um usuário comum de software.

A seção a seguir busca elucidar os principais aspectos que devem ser observados no desenvolvimento de software, estabelecendo, quando oportuno, analogia com a PDE. Por fim, é apresentado o processo de software que foi adotado para o desenvolvimento da GenIA e como, ao trilhar esse caminho, ele conversa com a PDE.

3.3 Processo de software

Segundo Sommerville (2016), o software é abstrato e intangível, não é limitado ao uso de materiais nem sofre limitação física, e, por isso, pode se tornar extremamente complexo e de difícil compreensão. Para tanto, a Engenharia de Software, uma área pertencente à Computação, discorre sobre métodos para especificação, desenvolvimento, verificação e validação, implantação, e manutenção de software. “A Engenharia de Software engloba um processo, uma coleção de

métodos (práticas) e uma matriz de ferramentas que permitem que profissionais construam software de computador de alta qualidade.” (PRESSMAN e MAXIM, 2015, p. 14, tradução nossa).

Cabe, neste momento, uma breve discussão a respeito do uso da palavra software. Na língua portuguesa, software é definido como “coleção de programas, procedimentos e documentação que controla ou desempenha alguma tarefa em um sistema de computação” (SOFTWARE, 2021). Curiosamente, mesmo tendo sua definição como já sendo uma coleção, é comum o uso da palavra flexionada no plural: softwares. Seriam softwares, então, o coletivo de uma coleção de programas? Talvez. Acontece que o termo software, de uma forma geral, quando traduzido para a língua portuguesa passou a ser sinônimo de programa ou de aplicativo.

A título de ilustração: o mesmo não acontece com hardware, que se refere ao conjunto de componentes físicos do computador e muito raramente encontra-se um texto que faça uso da flexão hardwares (no plural), como sinônimo de peças.

Ainda deve-se considerar que, sob a perspectiva da Engenharia de Software, Sommerville (2016) pontua que o termo software não fica limitado a um programa ou um conjunto de programas:

Muitas pessoas pensam que software é simplesmente outra palavra para programas de computador. No entanto, quando falamos de engenharia de software, o software não é apenas os próprios programas, mas também toda a documentação associada, bibliotecas, sites de suporte e dados de configuração necessários para tornar esses programas úteis. (SOMMERVILLE, 2016, p. 19, tradução nossa).

A partir da afirmação do autor, observa-se que a GenIA pode ser entendida como um software. Porém, considerando que esta pesquisa possui raízes em outra área de conhecimento (Educação Matemática), apresentá-la como “o software GenIA” poderia direcionar o leitor para uma interpretação reduzida de suas potencialidades. Isto porque a GenIA vai além do programa para construção de OA, incluindo, a título de exemplo, os módulos de IA e a documentação de utilização, juntamente com outros recursos que serão disponibilizados em seu website.

Portanto, é importante deixar claro ao leitor que o termo software pode ter conotação diferente dependendo do contexto e/ou área de conhecimento, e que esta pesquisa irá adotar preferencialmente sua forma original, sem flexão, já referindo-se

ao coletivo. Quando oportuno, serão adotados termos específicos, como programa, aplicativo, plataforma etc., que podem aparecer flexionados no plural. Por fim, vale ainda ressaltar que, sobretudo nesta seção, quando for mencionado o termo software, pode ser estabelecida uma relação direta com a plataforma que foi desenvolvida.

Segundo Pressman e Maxim (2015), a metodologia voltada ao desenvolvimento de software é chamada de processo de software:

Quando você trabalha para construir um produto ou sistema, é importante seguir uma série de passos pré-definidos – um roteiro que ajuda você a criar um resultado oportuno e de alta qualidade. O roteiro que você segue é chamado de processo de software. (PRESSMAN; MAXIM, 2015, p. 30, tradução nossa).

Sommerville (2016) afirma que há quatro atividades fundamentais que são comuns a qualquer processo de software:

1. Especificação de software, na qual clientes e engenheiros definem o software a ser produzido e as restrições à sua operação;
2. Desenvolvimento de software, na qual o software é projetado e programado;
3. Validação de software, na qual o software é verificado para garantir que é o que o cliente exige;
4. Evolução do software, na qual o software é modificado para refletir as mudanças nos requisitos do cliente e do mercado.

Porém, há diferentes roteiros e a inserção dessas atividades pode variar de acordo com o tipo de software que será criado:

Por exemplo, o software em tempo real em uma aeronave deve ser completamente especificado antes do início do desenvolvimento. Em sistemas de comércio eletrônico, a especificação e o programa geralmente são desenvolvidos em conjunto. Consequentemente, essas atividades genéricas podem ser organizadas de diferentes maneiras e descritas em diferentes níveis de detalhes, dependendo do tipo de software que está sendo desenvolvido. (SOMMERVILLE, 2016, p. 23, tradução nossa).

Os processos de software podem seguir diferentes modelos, agrupados de acordo com o sequenciamento das atividades previstas para cada etapa. No grupo dos modelos de “processo prescritivo”, as atividades e tarefas são realizadas em

sequência, obedecendo diretrizes pré-definidas. Durante muito tempo este foi o grupo mais utilizado, e será comentado na sequência. Os modelos de “processo especializado” são baseados nos modelos prescritivos, porém aplicam-se a situações bem específicas de desenvolvimento de software. Por fim, o “processo unificado” é o mais utilizado em sistemas comerciais, pois baseia-se em casos de uso e no paradigma da análise orientada a objetos (PRESSMAN; MAXIM, 2015; SOMMERVILLE, 2016).

Antes de dar sequência aos aspectos metodológicos do desenvolvimento de software, cabe um esclarecimento ao leitor a respeito de alguns termos (jargões) utilizados na Computação. A Engenharia de Software reconhece diferentes papéis no que tange às pessoas envolvidas no desenvolvimento de um software. O cliente solicita o desenvolvimento e esclarece suas necessidades (requisitos). A partir disso, o analista de sistemas irá fazer a concepção e modelagem. Na sequência, o programador irá transcrever o que foi projetado em código. E, por fim, o usuário é quem vai interagir diretamente com o software. Este último pode ser o próprio cliente que solicitou o desenvolvimento. Todos os papéis citados são chamados genericamente de “interessados”, sob a perspectiva da gestão de projetos.

Acontece que o software no contexto desta pesquisa é um produto educacional, e que promove uma conversa entre diferentes áreas de conhecimento. O cliente, que define os objetivos, pode ser tanto o pesquisador quanto o orientador, ou até mesmo uma terceira pessoa que contribua com o direcionamento da pesquisa. Os papéis de analista e programador (engenheiros de software) são desempenhados pelo pesquisador. O usuário pode ser um professor de Matemática ou pedagogo, por exemplo, ou até mesmo um estudante de licenciatura. Nesta perspectiva, pode-se entender que eles também sejam clientes, sob a óptica da Engenharia de Software.

Para evitar desconforto durante a leitura, fica estabelecido, portanto, que os termos serão apresentados como eles aparecem na Computação, porém quando o relato da pesquisa adentrar o contexto da Educação Matemática, os termos irão especificar as atividades de pesquisador, professor ou estudante.

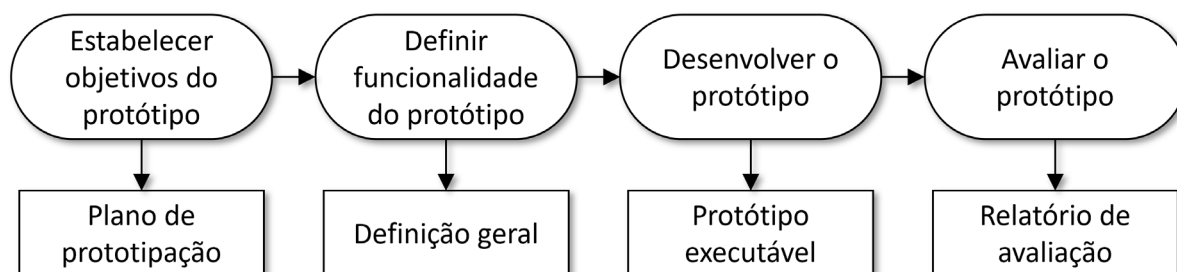
Como já foi comentado, a adoção da PDE teve como principal elemento motivador o fato de considerar momentos avaliativos em sucessivas iterações, com perspectiva educacional. Neste viés, o processo de software adotado para desenvolvimento da GenIA precisaria estar em consonância com a PDE, no que tange

a importância de sucessivas avaliações, porém direcionando o olhar para os aspectos técnicos.

Por se tratar de uma investigação, os modelos voltados ao desenvolvimento de software comercial⁷ não são aplicáveis, uma vez que se baseiam na premissa de requisitos estáveis. Também não foram considerados *frameworks* de desenvolvimento ágil⁸, pois estes visam otimização de um processo produtivo de software, caminhando em sentido oposto ao de uma abordagem exploratória. Sendo assim, as possibilidades e restrições encaminham a pesquisa para a adoção do modelo de prototipação, pertencente ao grupo de modelos de processo prescritivo, cujas fases conversam com as fases da PDE.

“Um protótipo é uma versão inicial de um sistema de software que é usado para demonstrar conceitos, experimentar opções de design e descobrir mais sobre o problema e suas possíveis soluções.” (SOMMERVILLE, 2016, p. 62, tradução nossa). O desenvolvimento de software por prototipação acontece, segundo Sommerville (2016), em quatro etapas, evidenciadas na Figura 4.

Figura 4 – Etapas da prototipação



Fonte: SOMMERVILLE (2016, p. 63, tradução nossa)

Os objetivos do protótipo devem ser explicitados desde o início do processo, podendo focar na criação da interface ou nas funcionalidades. Uma única iteração (versão) de um protótipo geralmente não consegue atender a todos os objetivos. Portanto, são criadas sucessivas iterações do protótipo, atendendo gradativamente aos objetivos estabelecidos.

⁷ “Software comercial” é o termo utilizado para se referir a aplicativos que fazem uso de banco de dados, normalmente auxiliando em tarefas administrativas, nos níveis operacional e de gestão.

⁸ Um *framework* (arcabouço) de desenvolvimento ágil é um conjunto de valores, princípios e práticas que são utilizados em conjunto com metodologias de desenvolvimento de software com o intuito de otimizar o processo de desenvolvimento no cotidiano da prática profissional.

Atualmente, o controle das iterações de um software segue uma prática conhecida como versionamento semântico. “Como uma solução para este problema proponho um conjunto simples de regras e requisitos que ditam como os números das versões são atribuídos e incrementados.” (PRESTON-WERNER, [s. d.]).

Seguindo o padrão semântico, a versão de um software é identificada por um número sequencial, subdividido em três níveis, respeitando as seguintes regras:

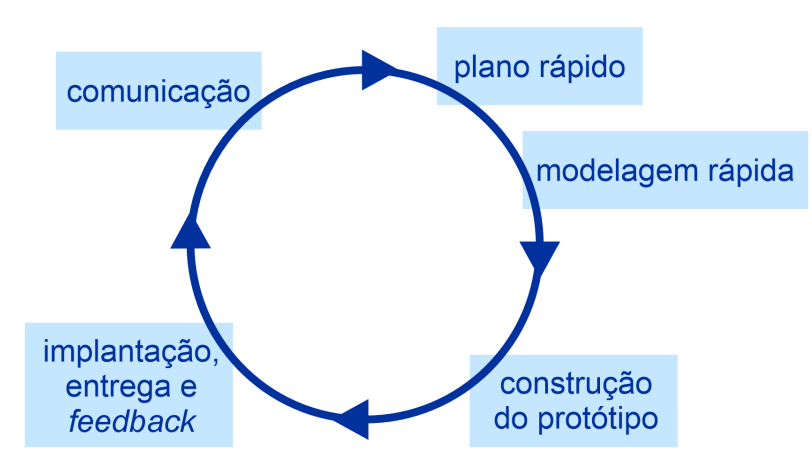
Dado um número de versão MAJOR.MINOR.PATCH, incremente a:
versão Maior (MAJOR): quando fizer mudanças incompatíveis na API,
versão Menor (MINOR): quando adicionar funcionalidades mantendo compatibilidade, e
versão de Correção (PATCH): quando corrigir falhas mantendo compatibilidade.
Rótulos adicionais para pré-lançamento (pre-release) e metadados de construção (build) estão disponíveis como extensão ao formato MAJOR.MINOR.PATCH. (PRESTON-WERNER, [s. d.])

De acordo com as regras, infere-se que a versão inicial da GenIA seja a de número 1.*n.n*, isto é, uma versão maior inicial (1) e, depois de algumas iterações, com versão menor (e possivelmente de correção) superior a 1. Porém, por se tratar de um protótipo resultante de uma pesquisa, a GenIA irá permanecer em sua versão 1.0 até a defesa deste trabalho, versão sob a qual será publicada no Repositório Institucional da Universidade Tecnológica Federal do Paraná (RIUT) e no endereço web já citado.

Pressman e Maxim (2015) consideram que o modelo de prototipação é utilizado quando existe um direcionamento para os objetivos a serem cumpridos com o software, porém não há requisitos detalhados por parte do cliente.

A necessidade de adequações no que já foi implementado ou a necessidade de desenvolvimento de novas funcionalidades revela-se à medida em que o software seja desenvolvido e validado. Além disso, o desenvolvedor pode não estar confortável com a eficiência dos algoritmos, a adaptabilidade ao sistema operacional ou a maneira como o usuário deve interagir com o sistema (PRESSMAN; MAXIM, 2015). A Figura 5 ilustra o modelo de prototipação.

Figura 5 – Modelo de prototipação



Fonte: PRESSMAN E MAXIM (2015, p. 46, tradução nossa)

O modelo de prototipação inicia com a comunicação. Nessa fase, há uma reunião com os interessados para definir os objetivos gerais do software. São identificados os requisitos do que deve ser desenvolvido e outras áreas que podem ser afetadas. A fase de comunicação da prototipação conversa diretamente com a fase preliminar da PDE, na qual o problema é proposto e analisado.

Na sequência, acontecem o plano rápido e a modelagem rápida, cujo foco é identificar os aspectos do software que são visíveis ao cliente, como telas, botões, e outras formas de interação. Estas atividades levam à construção do protótipo. Em analogia à PDE, estas três atividades estão ligadas à fase de prototipação.

Por fim, chega o momento de verificar se o que foi construído atende às expectativas. Na fase de implantação, entrega e *feedback* os interessados avaliam o que foi construído, identificando pontos de aceitação ou de melhoria, que serão implementados na próxima iteração. Para ampliar compreensões nesta fase, abordando aspectos que vão além do desenvolvimento do software, foi utilizada a PDE.

É importante ressaltar que a avaliação de um software, seja ele um produto comercial ou apenas um protótipo, acontece de maneira contínua, por meio de testes sob diferentes níveis e estratégias.

Processos de verificação e validação (V&V) são utilizados para determinar se o desenvolvimento de produtos de uma certa atividade está em conformidade com os requisitos daquela atividade e se o produto satisfaz sua intenção de uso e as necessidades do usuário. (IEEE, 2012, n.p., tradução nossa).

Os processos de verificação têm por objetivo garantir que o software seja construído sob boas práticas de análise e desenvolvimento. Para tanto, são realizados testes de verificação estática, como por exemplo a inspeção de código e o teste de mesa; e verificação dinâmica, por meio da execução de casos de teste e análise de resultados, dentre outras técnicas (MYERS; BADGETT; SANDLER, 2011). Os processos de validação, por sua vez, preocupam-se em garantir que o software satisfaz sua intenção de uso e as necessidades do usuário.

Os testes devem ser executados em diferentes fases (níveis) do processo de desenvolvimento de software, sendo que podem ser utilizadas diferentes estratégias, conforme a metodologia adotada. De acordo com Pressman e Maxim (2015), os níveis mais comuns para realização dos testes são:

- UT – *Unity Test* (teste de unidade): consiste em testar se uma funcionalidade específica apresenta os resultados esperados;
- SIT – *System Integration Test* (teste de integração de sistema): é executado para garantir que um módulo que está sendo construído converse adequadamente com outros módulos já existentes no sistema;
- UAT – *User Acceptance Test* (teste de aceitação do usuário): é o momento em que o usuário atesta que uma funcionalidade ou o software como um todo foi desenvolvido de acordo com o que foi solicitado.

Além das estratégias, existem diferentes técnicas de testes. Pressman e Maxim (2015) consideram que as técnicas de teste seguem duas filosofias:

- Técnicas estruturais (caixa branca): conhecendo-se o funcionamento interno de um produto, testes podem ser realizados para garantir que “todas as engrenagens se encaixam”, ou seja, que a operação interna do produto tem um desempenho de acordo com as especificações e que os componentes internos foram adequadamente postos à prova;
- Técnica funcional (caixa preta): Conhecendo-se a função específica que um produto projetado deve executar, testes podem ser realizados para demonstrar que cada função é totalmente operacional.

Em suma, enquanto o programador cria um código para atender a uma funcionalidade, ele executa inúmeras vezes os testes de unidade, por diferentes técnicas estruturais, para verificar se as linhas de código recém inseridas estão funcionando. Na sequência, ele executa o software acionando funcionalidades já

existentes para garantir que a integração aconteceu a contento, normalmente pela técnica funcional. Por fim, o software chega às mãos do usuário, que irá fazer a verificação e validação exclusivamente por meio da técnica funcional.

Conhecendo-se a PDE e o processo de software, apresenta-se a seguir como a pesquisa transitou pelos caminhos de tais metodologias.

3.4 Seguindo o fluxo

Seguindo os passos da PDE, a fase preliminar iniciou no segundo semestre de 2019 e perdurou até o primeiro semestre de 2021. Durante o referido período, além das habituais leituras realizadas nos encontros do GPTM e revisão bibliográfica voltada à pesquisa, três disciplinas do PPGFCET, cursadas pelo pesquisador, contribuíram significativamente para essa fase da pesquisa: Seminários de Projetos – Análise e Produção de Artigos Científicos (FCET35); Tendências de Pesquisas sobre Tecnologias da Informação e Comunicação no Ensino de Ciências e Matemática (FCET26); e Bases Epistemológicas para Pesquisa em Ensino de Ciências e Matemática (FCET39).

Durante o curso das disciplinas mencionadas buscou-se, em diferentes bases de dados, produções que pudessem sustentar o referencial teórico desta pesquisa. Especificamente na disciplina FCET35, no segundo semestre de 2019, iniciou-se um mapeamento sistemático (MS) com o objetivo de identificar publicações que abordassem, de forma concomitante, os mesmos objetos de estudo desta pesquisa.

Segundo Motta, Kalinke e Mocosky (2018, p. 69), um MS possibilita “contribuir de forma efetiva com os estudos teóricos de uma área de conhecimento específica, permitindo a identificação dos aspectos conceituais envolvidos no estudo, limitações e potencialidades, além de possibilitar a categorização das informações.”

Motta, Basso e Kalinke (2019, p. 206) pontuam que um MS permite classificar e esquematizar uma certa área de interesse, possibilitando que os dados quantifiquem as publicações ou trabalhos por meio de categorias, sendo organizado em quatro etapas distintas: planejamento, condução, descrição e portfólio.

Na fase de planejamento, definiu-se que as bases consultadas seriam: Portal de Periódicos da Capes, Catálogo de Teses e Dissertações da Capes, Google Acadêmico e Sci-Hub.

Na condução do MS, realizada em agosto de 2019, foram utilizados os descritores “educação matemática”, “objetos de aprendizagem”, “inteligência artificial”, “programação intuitiva” e “programação visual”. Os termos também foram utilizados na língua inglesa: “*math education*”, “*mathematics education*”, “*learning objects*”, “*artificial intelligence*”, “*intuitive programming*” e “*visual programming*”. Foram montadas expressões com diferentes combinações por meio dos operadores booleanos “*and*” e “*or*”. Sendo assim, as expressões utilizadas nas ferramentas de busca das bases consultadas foram:

- (“inteligência artificial” or “*artificial intelligence*”) and (“educação matemática” or “*math education*” or “*mathematics education*”) and (“objetos de aprendizagem” or “*learning objects*”)
- (“inteligência artificial” or “*artificial intelligence*”) and (“educação matemática” or “*math education*” or “*mathematics education*”) and (“programação intuitiva” or “*intuitive programming*”)
- (“inteligência artificial” or “*artificial intelligence*”) and (“educação matemática” or “*math education*” or “*mathematics education*”) and (“programação visual” or “*visual programming*”)

Em nenhuma das bases, sob nenhuma das expressões fornecidas, foram encontrados resultados. Por conta disso, entendendo-se que, naquele momento, não existiam publicações que visassem o mesmo objetivo desta pesquisa, o MS foi interrompido, não seguindo com as fases subsequentes. Diante de tal situação, observa-se que existe um campo a ser explorado e carente de investigações e produções a respeito da temática em questão, revelando o ineditismo desta pesquisa.

Em paralelo à condução do MS, buscou-se identificar se a solução tecnológica proposta pelo pesquisador atenderia às necessidades de criação da GenIA. No contexto da Computação, entende-se por solução tecnológica a combinação de diferentes ferramentas, como por exemplo: linguagem de programação; ambiente de desenvolvimento integrado (*IDE – integrated development environment*); sistema gerenciador de banco de dados (SGBD); computação em nuvem; dentre outras. Ao se iniciar um projeto de desenvolvimento de software, é importante escolher uma solução que ofereça suporte às especialidades do que se está construindo.

A ideia inicial, motivada pelo conhecimento técnico prévio do pesquisador, era adotar uma solução baseada em algumas ferramentas da Microsoft, a saber:

programação na linguagem C# sobre o *framework* .NET⁹, utilizando a IDE do Visual Studio, com colaboração e compartilhamento de dados por meio de serviços de nuvem.

O processo de investigação e validação levou em consideração fatores diversos, sendo alguns apontados como necessários e outros desejáveis. Oportuno destacar que a importância atribuída a cada fator seria no sentido de que o software estivesse apto para futuras expansões, além desta pesquisa, mas que não necessariamente se traduziriam em recursos que deveriam ser incorporados sob a versão inicial da plataforma até a defesa do trabalho. O Quadro 2 apresenta os fatores considerados para a criação da GenIA.

Quadro 2 – Fatores considerados na escolha da solução

Fator	Importância
Possibilitar o desenvolvimento sob o paradigma orientado a objetos	necessário
Possibilitar a criação de aplicativos com interface gráfica	necessário
Possibilitar a implementação de algoritmos de aprendizado de máquina	necessário
Possibilitar a execução em diferentes ambientes (<i>desktop</i> , <i>web</i> , dispositivo móvel)	necessário
Possibilitar o compartilhamento de dados em nuvem	necessário
Possibilitar a execução em diferentes plataformas computacionais (Windows, Linux, MacOS etc)	desejável
Possibilitar que a IDE contabilize métricas de desenvolvimento	desejável

Fonte: autoria própria (2023)

A respeito da solução proposta pelo pesquisador, foram realizadas buscas no site de documentação do fabricante (Microsoft¹⁰) e em uma das principais comunidades de desenvolvedores (Stack Overflow¹¹), revelando que as ferramentas atenderiam às necessidades elencadas.

Além dos fatores citados, outro aspecto importante que deve ser considerado na escolha de uma solução tecnológica é a sua longevidade. Para tanto, buscou-se identificar se a solução proposta vem se consolidando ao longo do tempo ou se poderia apresentar indícios de obsolescência. Neste viés, uma empresa de consultoria norte-

⁹ O *Framework* .NET é uma plataforma de desenvolvimento, isto é, um conjunto de linguagens de programação, ambiente de desenvolvimento integrado e bibliotecas, que possibilitam que sejam gerados aplicativos para serem executados em diferentes sistemas e ambientes operacionais.











¹⁰ Disponível em: <https://docs.microsoft.com>. Acesso em: 17 jun. 2023.

¹¹ Disponível em: <https://stackoverflow.com>. Acesso em: 17 jun. 2023.

americana, chamada TIOBE¹² (2023) publica periodicamente um *ranking* de uso e popularidade de linguagens e ambientes de programação. A medição é feita com base em mais de um bilhão de linhas de código de aplicativos criados para atender diferentes áreas, desde aplicações médicas até financeiras. No índice TIOBE, conhecido da comunidade de software, são listadas as cem primeiras linguagens de programação, por ordem de incidência, sendo que a primeira da lista é a mais utilizada.

Uma consulta ao *ranking* da TIOBE em março de 2022 revelou que a linguagem C# figurava entre as dez mais utilizadas, ocupando a quinta posição. Uma nova consulta ao *ranking* da TIOBE em junho de 2023 revelou que a linguagem C# mantinha-se na quinta posição, conforme pode ser observado no Quadro 3.

Quadro 3 – Índice TIOBE para junho de 2023

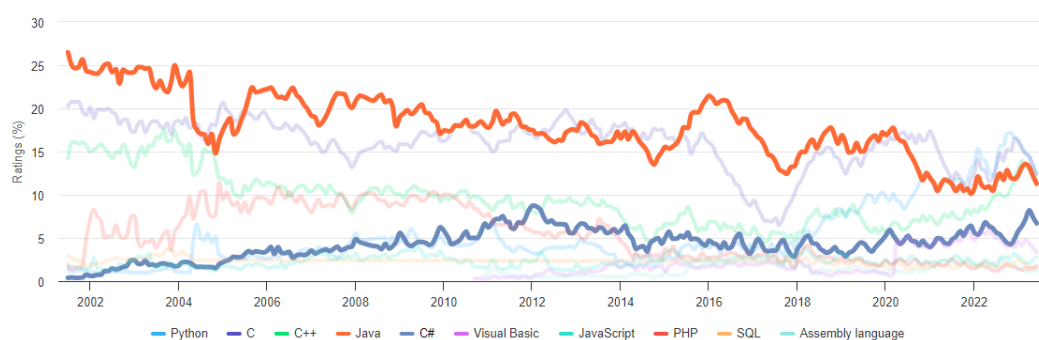
Jun 2023	Jun 2022	Change	Programming Language	Ratings	Change
1	1		 Python	12.46%	+0.26%
2	2		 C	12.37%	+0.46%
3	4	▲	 C++	11.36%	+1.73%
4	3	▼	 Java	11.28%	+0.81%
5	5		 C#	6.71%	+0.59%
6	6		 Visual Basic	3.34%	-2.08%
7	7		 JavaScript	2.82%	+0.73%
8	13	▲▲	 PHP	1.74%	+0.49%
9	8	▼	 SQL	1.47%	-0.47%
10	9	▼	 Assembly language	1.29%	-0.56%

Fonte: TIOBE (2023)

É possível observar, também, pelo histórico de uso da linguagem C#, apresentado na Figura 6, que a ela se mantém constante no *ranking* desde seu lançamento no final dos anos 2001, o que sugere uma estabilidade de seu uso.

¹² TIOBE é grafado em maiúsculas porque a origem do nome vem das iniciais de uma frase: The importance of being earnest (do inglês: A importância de ser sincero).

Figura 6 – Histórico de uso de linguagens



Fonte: TIOBE (2023)

A linguagem Java, bastante popular e com estrutura e recursos similares à linguagem C#, foi considerada, naquele momento, como uma opção para o desenvolvimento da GenIA. Porém, um olhar detalhado sobre o histórico de uso das linguagens revelou um ponto de atenção: observou-se que, com o passar do tempo, seu uso vem diminuindo gradativa e continuamente, o que sugere que, possivelmente, tal linguagem esteja em uma tendência de descontinuidade. Por esse motivo, a linguagem Java, mesmo ainda sendo bem aceita pela comunidade de programadores, foi descartada como alternativa para a condução desta pesquisa.

Considerando-se que as ferramentas investigadas atenderiam aos fatores necessários à criação da GenIA e que o histórico da linguagem sugere longevidade, optou-se por iniciar a programação adotando-se a solução inicialmente proposta, dando início à próxima fase da PDE.

A fase de prototipagem iniciou no segundo semestre de 2020 e perdurou por diversas iterações até o final da pesquisa. Para esta fase, buscou-se elencar objetivos gerais com relação à construção do protótipo, listados a seguir:

- testar a viabilidade gráfica do *framework* Microsoft .NET e da linguagem C# em diferentes ambientes;
- testar a compatibilidade dos metadados advindos de outros sistemas similares, em especial o Scratch¹³, para alimentação da plataforma e posterior treinamento dos algoritmos de IA;
- testar as possibilidades de uso de algoritmos de IA com aprendizado de máquina presentes nas bibliotecas do Microsoft ML.NET;

¹³ O Scratch (<https://scratch.mit.edu/>) é um software criado no MediaLab do Instituto de Tecnologia de Massachussetts (MIT), projetado para criar conteúdo interativo por meio de uma forma de programação que se baseia no arrastar e soltar de blocos coloridos.

- testar a viabilidade do uso de programação intuitiva dentro da plataforma, sob diferentes abordagens;
- avaliar a evolução dos algoritmos de aprendizado de máquina e inferência na operação da interface por parte do professor utilizador.

Em uma primeira iteração, buscou-se verificar a viabilidade gráfica da linguagem no ambiente Windows (outros ambientes poderão ser contemplados em versões futuras). Também nesta iteração foi testada a viabilidade de uso da programação intuitiva dentro da plataforma. Neste caso, optou-se por utilizar uma programação baseada em fluxograma, assunto abordado com detalhes na seção 5.5.

Para a criação do protótipo, em cada iteração foram respeitados os passos do processo de software, com plano e modelagem mínimos, tendo produzido documentação técnica do software, bem como um protótipo executável.

Sob orientação da PDE, do ponto de vista educativo, o protótipo executável foi observado por duas pesquisadoras do macroprojeto, sendo que uma é professora atuante do ensino fundamental em escolas da Rede do Estado do Paraná e a outra desempenha atividades na Secretaria de Educação do Estado do Paraná (Seed). Ambas fizeram considerações diversas de usabilidade, desde as imagens representativas dos ícones até melhorias visuais na execução dos OA criados na plataforma. Também identificaram alguns erros de programação. Os apontamentos eram registrados em arquivo compartilhado na nuvem, para acompanhamento e tratativa pelo pesquisador. Pode-se considerar tais apontamentos como execução de testes no nível de UAT e, dessa maneira, também foi contemplada a fase de avaliação desta iteração sob a perspectiva da PDE.

Em uma segunda iteração, mais uma parcela dos objetivos iniciais foi considerada. Desta vez, buscava-se uma maneira de registrar em arquivo os fluxogramas criados na GenIA, em conjunto com outros elementos que iriam compor os OA. Considerando a forte presença do Scratch na Educação Matemática, optou-se por verificar como tal software abordava esta questão e observou-se que os metadados dos projetos nele criados eram registrados utilizando-se o formato Json¹⁴. Com base nesta constatação, entendendo que tal estratégia seria tecnicamente viável para ser adotada por esta pesquisa e visando uma futura compatibilidade para

¹⁴ Json é um acrônimo para *JavaScript Object Notation* (Notação de Objetos em JavaScript). Disponível em: <https://json.org>. Acesso em: 17 jun. 2023.

intercâmbio de projetos entre a GenIA e o Scratch, optou-se por utilizar também o formato Json para registro dos projetos da GenIA. Assim como aconteceu na iteração anterior, foi gerada uma nova versão executável do protótipo para ser avaliada.

A partir das observações registradas sobre as iterações iniciais, iniciou-se uma constante de expansão de funcionalidades da GenIA, gerando, de forma contínua, novas versões executáveis. Até a banca de qualificação da pesquisa, em março de 2022, tais iterações não passaram mais por testes de UAT, ficando a avaliação limitada a testes unitários e de integração.

Depois de qualificada a pesquisa, a GenIA continuou em processo de refinamento técnico das funcionalidades para que pudesse ser submetida a avaliação pelos integrantes do GPTEM que são, em sua maioria, professores de Matemática de diferentes níveis de ensino. Entretanto, também fazem parte do grupo professores e pesquisadores atuantes em outras áreas. Destaca-se que o projeto foi aprovado pelo comitê de ética da instituição, estando, portanto, apto para ser submetido à avaliação pelos integrantes.

Optou-se por validar o software com os pesquisadores do GPTEM porque eles já têm familiaridade com a construção de OA em outras plataformas. Esse é um fator importante para o treinamento dos algoritmos de IA da GenIA, assunto que será explicado com detalhes na seção 6.5.

A GenIA foi disponibilizada para os integrantes do grupo em agosto de 2022. Enquanto eles validavam as questões de usabilidade e a programação intuitiva, deu-se continuidade ao desenvolvimento, desta vez explorando as possibilidades de implementação dos módulos de IA. A partir daquele momento, optou-se por não mais criar comandos e funcionalidades, ficando a programação limitada, sob este aspecto, a correção de erros, para que o pesquisador pudesse focar nos algoritmos de IA.

Seguindo o proposto pela PDE, em harmonia com o processo de software, foram geradas novas iterações do protótipo e avaliações subsequentes. Detalhes das avaliações e da aplicação do produto são apresentados na seção 3.5, pois considera-se impreterível um entendimento de suas possibilidades para que se tenha uma melhor compreensão das devolutivas.

À medida em que eram recebidas e consideradas as devolutivas a respeito da usabilidade da GenIA, sob a perspectiva da construção dos fluxogramas, foi-se consolidando a implementação da IA. Essa fase iniciou com a criação de amostras,

pelo próprio pesquisador, com o objetivo de estudar as possibilidades das bibliotecas do ML.NET, sempre com validação calcada na execução de testes de unidade.

Posteriormente, os fluxogramas criados pelos integrantes do GPTEM foram sendo gradativamente inseridos nos testes de treinamento dos algoritmos. Detalhes das estratégias e recursos técnicos adotados nesta fase podem ser observados na seção 6.5.

Os testes iniciais feitos a partir dos OA criados pelo grupo de pesquisa revelaram que é possível o treinamento dos algoritmos de IA para atender os anseios da pesquisa, ainda que a quantidade de amostras produzidas não seja suficiente para atingir a acurácia¹⁵ necessária dentro do prazo previsto para a conclusão do doutorado. Contudo, os algoritmos estão implementados na plataforma e permanecerão em constante treinamento a partir de OA criados pelo grupo de pesquisa, podendo receber contribuições de outras esferas da comunidade acadêmica e servir de base para futuras pesquisas sob temáticas correlatas.

3.5 Aplicação do produto

O produto foi utilizado em processos educacionais em dois momentos. A primeira aplicação aconteceu no segundo semestre de 2022, quando a GenIA ainda não incorporava os módulos de IA. A plataforma foi utilizada por um grupo de estudantes do curso de Bacharelado em Engenharia de Software, de uma universidade particular de Curitiba. Naquele momento, os estudantes estavam cursando a disciplina de Algoritmos de Programação e entendeu-se que seria uma boa oportunidade para testes de interface e construção de fluxogramas na GenIA. Cabe destacar que, ainda que a plataforma seja orientada à construção de OA de Matemática, ela possibilita expressar a lógica e construir conteúdo para outras disciplinas.

A disciplina de Algoritmos de Programação, presente na matriz curricular de cursos da área da Computação, é voltada ao desenvolvimento das habilidades em lógica de programação. Tal disciplina costuma ser ministrada fazendo uso de comandos textuais e pseudocódigos para desenvolvimento dos programas e é nela

¹⁵ Em aprendizado de máquina, a acurácia é uma métrica comumente usada para avaliar o desempenho de um modelo de classificação; ela mede a taxa de acertos ou a precisão do modelo na classificação de exemplos.

que normalmente os estudantes aprendem técnicas para construção de fluxogramas. Durante a aula ministrada com apoio da GenIA, os estudantes puderam transpor para fluxogramas os algoritmos que estavam trabalhando anteriormente em pseudocódigos e fazer observações a respeito do funcionamento dos comandos, bem como da usabilidade em relação à interface.

Em junho de 2023, a GenIA foi novamente levada para a sala de aula. Desta vez, o produto foi utilizado na disciplina Tecnologias no Ensino da Matemática (MA78N), lecionada pelo Prof. Dr. Marco Aurelio Kalinke, para o oitavo período do curso de Licenciatura em Matemática da UTFPR. A turma era composta de nove estudantes e, durante a atividade, foram apresentados os objetivos da pesquisa e feita uma demonstração de funcionamento da plataforma, a fim de contextualizar sua aplicabilidade. A partir dessa dinâmica, os estudantes puderam experimentar a construção de OA por meio da GenIA, com acompanhamento tanto do Prof. Kalinke quanto do pesquisador, a fim de elucidar possíveis dúvidas de utilização. A proposta era que, ao final do processo, os alunos pudessem fornecer devolutivas, considerando a perspectiva de uso futuro em processos educacionais.

Durante o uso, os estudantes identificaram alguns erros de operação. Além disso, fizeram considerações a respeito do domínio dos comandos necessários para colocar em prática as ideias que tinham para a construção dos OA. Segundo eles, normalmente quando se utiliza um aplicativo e não se tem o conhecimento para realizar determinada atividade, costuma-se recorrer a algum repositório de tutoriais e, naquele momento, talvez fosse a maior dificuldade com a GenIA: não ter um repositório de tutoriais. Ainda assim, os estudantes teceram considerações positivas sobre a plataforma, apontando que a construção dos OA nela se revelou mais fácil em comparação a outras plataformas por eles anteriormente utilizadas, como por exemplo, o Scratch.

O projeto final avaliativo da disciplina em questão visa a construção de um OA de Matemática, o que pode ser realizado por qualquer aplicativo que seja de domínio dos estudantes. Dentre as opções de aplicativos que lhe são sugeridas pelo professor estão o Hot Potatoes, o Scratch, o App Inventor, ou até mesmo o pacote Microsoft Office. A partir da boa recepção por parte dos estudantes, identificada durante o uso da GenIA no encontro, o professor propôs que a plataforma pudesse integrar a lista de opções de aplicativos a serem utilizados como recurso para elaboração do projeto final, e um dos estudantes apresentou o OA construído na GenIA.

O assunto do OA apresentado trata de função linear (referenciada no trabalho como função afim ou função do 1° grau). Durante a execução do OA, um avatar interage com o usuário apresentando os elementos de uma função linear, como coeficientes angular e linear, demonstrando-os por meio do traçado de reta e pontos no plano cartesiano. Os elementos foram apresentados sob uma função específica criada pelo estudante, possibilitando que o usuário informe novos valores para os referidos coeficientes e compare o resultado do traçado de seus elementos geométricos com os propostos pelo OA.

No momento em que enviou o OA, o estudante apresentou, a pedido do pesquisador, suas considerações sobre o uso da GenIA. Ele indicou que o desenvolvimento no software é intuitivo e, após algum tempo pega-se a prática e compreensão dos seus elementos e funcionalidades. Destacou, entretanto, que ele apresenta algumas falhas pontuais.

Comentou a respeito da dificuldade de manipulação quando uma única expressão envolve muitos operandos, pois acaba se tornando um objeto muito grande, mais largo que a área de confecção do fluxograma, o que torna difícil o encaixe no comando correspondente. Salientou ainda a necessidade de se travar o zoom da escala de apresentação do plano.

Por fim, discorreu sobre aspectos gerais de usabilidade, afirmando que gostou muito de trabalhar com o aplicativo e acredita que com melhorias pontuais a GenIA pode se tornar uma excelente plataforma de construir objetos de aprendizagem, tanto de Matemática quanto de programação, porém que sentiu falta de uma biblioteca de exemplos e/ou manual de instruções com exemplos.

As considerações apresentadas pelo estudante revelaram alguns pontos de usabilidade que podem ser observados em futuras versões. Neste caso, é importante destacar que as correções de erro evidenciadas já foram incorporadas à plataforma. Contudo tratativas relacionadas a novas maneiras de manipular os objetos, ainda que melhores, demandariam maior esforço e não seriam incorporadas de imediato, pois o direcionamento do pesquisador naquele momento era para a implementação dos algoritmos de IA. Estes aspectos serão retomados nas recomendações para trabalhos futuros.

A trilha que resultou na criação da GenIA fundamentou-se basicamente em três pilares teóricos: os OA, a programação intuitiva e a IA. Em busca de contribuir com a compreensão do leitor a respeito de algumas ideias teóricas sobre as quais os

estudos desta pesquisa se apoiam, os capítulos subsequentes apresentam os aportes teóricos que embasam a criação da GenIA e, quando oportuno, indicando de que forma eles contribuíram para o seu desenvolvimento.

4 OBJETOS DE APRENDIZAGEM

Os OA tem sido foco de investigação no contexto educacional há algum tempo, porém ainda não há um consenso entre os pesquisadores a respeito de sua definição. Segundo Polsani (2003), o termo OA foi popularizado por Wayne Hodgins em 1994, quando ele nomeou o grupo de trabalho do CedMA (*Learning Architectures, APIs and Learning Objects – Arquiteturas de Aprendizagem, APIs e Objetos de Aprendizagem*).

Os termos Objetos de Aprendizagem (OA) e Objetos de Aprendizagem Reutilizáveis são frequentemente empregados de forma acrítica, reduzindo-os a meros slogans. A grave falta de clareza conceitual e reflexão é evidente na multiplicidade de definições e usos dos OA. (POLSANI, 2003, n.p., tradução nossa).

A partir daí, o termo começou a ser difundido, porém com diferentes definições. O *Learning Technology Standards Committee* (Comitê de Padrões de Tecnologia de Aprendizagem), um comitê do *Institute of Electrical and Electronics Engineers* (IEEE), definiu OA como “qualquer entidade, digital ou não-digital, que é utilizada para aprendizagem, educação ou treinamento” (IEEE, 2020, n.p, tradução nossa).

Wiley (2002) define OA como “qualquer recurso digital que pode ser reutilizado para suportar a aprendizagem” (WILEY, 2002, p. 6, tradução nossa). A definição de Gutierres (2004) é mais ampla, ao considerar que:

Um objeto de aprendizagem pode ser conceituado como sendo todo objeto que é utilizado como meio de ensino/aprendizagem. Um cartaz, uma maquete, uma canção, um ato teatral, uma apostila, um filme, um livro, um jornal, uma página na web, podem ser objetos de aprendizagem. (GUTIERRES, 2004, p. 6).

Observa-se que existe uma divergência de interpretação já na segregação entre digital e não-digital. A título de exemplo, um caderno poderia ser considerado um OA sob a óptica do IEEE (2020) e de Gutierres (2004), mas não sob a definição de Wiley (2002).

Munhoz (2013) considera que um OA é um recurso digital educacional, reutilizável, direcionado a aprendizagem, composto por simulações, animações, vídeos e outros elementos.

Para o Grupo de Pesquisa em Inovação e Tecnologias na Educação (GPINTEDECUC) OA “São recursos digitais para suporte à aprendizagem de um conteúdo específico, por meio da interatividade, que podem ser usados e reusados, em diferentes níveis e modalidades de ensino¹⁶”.

A definição adotada nesta pesquisa é aquela apresentada pelo GPTEM, que considera um OA como sendo “qualquer recurso virtual multimídia, que pode ser usado e reutilizado com o intuito de dar suporte à aprendizagem de um conteúdo específico, por meio de atividade interativa, apresentada na forma de animação ou simulação” (KALINKE; BALBINO, 2016, p. 25).

Tanto a definição do GPTEM quanto a do GPINTEDECUC corroboram com a definição de Munhoz (2013) ao considerarem que um OA é um recurso que possa ser reutilizado. Este entendimento comum entre os grupos de pesquisa se apresenta também em uma revisão sistemática realizada por Motta e Kalinke (2021), na qual identificaram que a maioria dos autores tem opinião convergente ao compreenderem que o OA deve ser “[...] reutilizável em diferentes contextos, sem a necessidade de ser remodelado e reestruturado” (MOTTA; KALINKE, 2021, p. 159).

A reusabilidade aparece no rol das principais características dos OA apontadas por Mendes, Souza e Caregnato (2007), listadas no Quadro 4.

Quadro 4 – Principais características dos OA

Característica	Atributos
Reusabilidade	Reutilizável diversas vezes em diversos ambientes de aprendizagem.
Adaptabilidade	Adaptável a qualquer ambiente de ensino.
Granularidade	Conteúdo em pedaços, para facilitar sua reusabilidade.
Acessibilidade	Acessível facilmente via Internet para ser usado em diversos locais.
Durabilidade	Possibilidade de continuar a ser usado, independente da mudança de tecnologia.
Interoperabilidade	Habilidade de operar através de uma variedade de hardware, sistemas operacionais e browsers, intercâmbio efetivo entre diversos sistemas.

Fonte: MENDES; SOUZA; CAREGNATO (2007, p. 3, adaptado)

¹⁶ Disponível em: <https://gpinteduc.wixsite.com/utfpr/definicoes-do-grupo>. Acesso em: 17 jun. 2023.

A respeito das características apontadas por Mendes, Souza e Caregnato (2007), é possível observar que discorrem sobre aspectos técnicos dos OA. Porém, um OA pode ser observado e caracterizado também por outras dimensões.

4.1 Dimensões e características dos OA

Motta e Kalinke (2019), ao examinarem trabalhos que abordavam metodologias de construção de OA, identificaram algumas características recorrentes ou similares e propuseram uma compilação de características sob a dimensão educacional, apresentadas no Quadro 5.

Quadro 5 – Características da dimensão educacional

Característica	Atributos
Interatividade	É a forma como o estudante irá interagir com o objeto digital e a maneira como este influenciará a sua cognição. É nessa troca entre homem e artefato digital que ocorre a construção ou ressignificação do conhecimento.
Cooperação	É a capacidade de permitir aos estudantes trabalhar em de forma colaborativa, facilitando a interatividade do sujeito com o recurso digital.
Autonomia	Proporciona ao aluno ser agente ativo de sua própria aprendizagem.
Cognição	É a maneira como o OA proporciona ao estudante outras formas de pensar e aprender um conteúdo específico, instigando o a buscar além do que solicitado, permitindo, em algumas situações, uma reflexão sobre a sua própria ação.
Afeto	São aspectos que favorecem e motivam a aprendizagem do aluno durante suas ações no recurso tecnológico.

Fonte: MOTTA; KALINKE (2019, p. 204)

A respeito da dimensão educacional, Meireles (2017) aponta que as etapas para criação de objetos de aprendizagem podem resultar em um trabalho árduo se não forem observadas as características essenciais dos objetos, sobretudo a interatividade e autonomia.

Na Educação Matemática, as características de interatividade, cooperação e cognição podem potencializar a aprendizagem, por exemplo, em contextos que buscam explorar as representações de um mesmo objeto matemático nos diferentes registros. Lima e Siple (2021; 2021) apresentam relatos a respeito de tais

características quando encontram, no GeoGebra Classroom¹⁷ e no GeoGebra Groups¹⁸, alternativas para explorar o raciocínio covariacional¹⁹ em aulas não presenciais durante a pandemia da COVID-19, por exemplo.

Tendo como base o trabalho realizado por Kemczinski *et al.* (2012), considerando a dimensão técnica dos OA, Motta e Kalinke (2019) identificaram que os conceitos das características técnicas necessitavam de adequações e complementações. A partir de tal estudo, propuseram uma nova gama de características sob a referida dimensão, que pode ser observada no Quadro 6.

Quadro 6 – Características da dimensão técnica

Característica	Atributos
Adaptabilidade	Possibilita que o objeto seja utilizado em diferentes níveis e ambientes de ensino.
Agregação	Possibilita que o objeto seja agrupado a outro, formando um conjunto maior de conteúdos.
Classificação	Apresenta informações (metadados) que permitem a sua identificação em repositórios ou mecanismos de buscas.
Digital	Deve ser utilizado e disponibilizado em ambiente virtual.
Durabilidade	Possibilita que o recurso não pare de ser manipulado, mesmo que a tecnologia utilizada em sua construção deixe de existir.
Interoperabilidade	Possibilita que o objeto seja utilizado em diferentes ambientes digitais e em diferentes sistemas operacionais.
Reusabilidade	Possibilita ao objeto ser utilizado em diferentes contextos e temáticas, podendo ser readaptado a diferentes propósitos não previstos em sua concepção.

Fonte: KEMCZINSKI *et al.* (2012, p. 2, adaptado)

Braga *et al.* (2012) pontuam que os OA podem ser considerados produtos de software, por se tratar de recursos digitais.

Sendo assim, a produção de OA poderia se beneficiar das boas práticas dos processos de desenvolvimento de software estudados na área de Engenharia de Software. De outro lado, o projeto de um OA precisa ser construído sob a égide de alguma teoria de aprendizagem que possa lhe conferir objetivos pedagógicos, formas de aplicação e avaliação claros. Seguindo esse raciocínio, poder-se-ia direcionar seu desenvolvimento para o almejado equilíbrio técnico e pedagógico, e, ainda, a garantia de sua reutilização. (BRAGA *et al.*, 2012, p. 2).

¹⁷ Disponível em: <https://www.geogebra.org/classroom>. Acesso em: 17 jun. 2023.

¹⁸ Disponível em: <https://www.geogebra.org/m/rQrbooeq>. Acesso em: 17 jun. 2023.

¹⁹ Variação dos valores de duas quantidades que ocorre simultaneamente.

Observa-se que, em seu trabalho, os pesquisadores também consideram ambas as dimensões, técnica e educacional. A partir desse entendimento, propõem características de qualidade para OA, baseando-se em três teorias: a) normas de qualidade de software ISO/IEC 9126-1; b) itens de avaliação sugeridos pela *Learning Object Review Instrument* (LORI); e c) índices de satisfação sugeridos pela *Computer Education Management Association* (CEdMA, 2001). O Quadro 7 apresenta uma compilação das características baseadas nas teorias observadas.

Quadro 7 – Características de qualidade de um OA

Característica	Atributos	Baseado em
Habilidades Didático Pedagógicas	O OA deve ser capaz de mostrar ao aluno o objetivo do aprendizado a que se propõe. Sendo esse objetivo alinhado às metas de aprendizagem e características dos alunos. É desejável que o OA forneça <i>feedback</i> suficiente para facilitar o aprendizado do aluno.	LORI
Disponibilidade	O OA deve ser indexado e armazenado de maneira que possa ser facilmente encontrado.	CEdMA
Acessibilidade	O OA pode ser acessado por diferentes dispositivos, diferentes contextos (ex.: velocidade de conexão diferente) e principalmente possuir versão adaptada para diferentes tipos de usuários (deficientes visuais, idosos etc.).	LORI, CEdMA
Precisão	O OA deve apresentar resultados precisos, dentro do esperado.	ISO/IEC 9126
Confiabilidade	O OA não deve possuir falhas técnicas.	ISO/IEC 9126
Facilidade de instalação	O OA deve ser fácil de ser instalado.	ISO/IEC 9126
Portabilidade	O OA deve funcionar em diversos cenários como: diferentes sistemas operacionais, diferentes ambientes virtuais de Avaliação, diferentes hardware etc.	ISO/IEC 9126
Interoperabilidade	O OA deve poder de interagir com outros OA ou sistemas.	ISO/IEC 9126
Usabilidade	O OA deve ser fácil de ser utilizado e estar de acordo com os padrões mais consagrados de usabilidade.	ISO/IEC 9126

Fonte: BRAGA et al. (2012, p. 2, adaptado)

A norma ISO/IEC 9126-1, publicada em 2001, foi substituída, em 2011, pela norma ISO/IEC 25010. Esta última agrega mais duas características que devem ser consideradas no sentido de garantir a qualidade de um software: segurança, que aborda aspectos de confidencialidade e integridade; e compatibilidade, voltando-se para a coexistência e a interoperabilidade. É possível fazer uma transposição para o contexto dos OA, conforme sugere o Quadro 8:

Quadro 8 – Características adicionais de qualidade de um OA

Característica	Atributos
Segurança	O OA deve possibilitar que dados referentes à sua utilização permaneçam restritos ao estudante, quando desejado, por meio de autenticação (senha, <i>token</i> , biometria etc); os dados de um OA não podem ser alterados indevidamente por outro usuário.
Compatibilidade	O OA deve ser compatível com outros OA dentro do mesmo repositório e sua utilização não pode comprometer o funcionamento dos outros OA.

Fonte: autoria própria (2023)

Tão importante quanto conhecer as características de um OA, sob as diferentes dimensões, é saber construí-lo. Para tanto, deve-se observar os aspectos metodológicos que orientam a produção do OA.

4.2 Metodologias para produção de OA

Tendo em vista que a GenIA implementa algoritmos de IA para assistir à construção dos OA, considerou-se imperativa uma busca por metodologias que orientassem o processo de produção de OA.

Braga (2015) afirma que a construção de OA pode considerar três abordagens em suas metodologias: as que consideram fortemente o design instrucional (aspectos pedagógicos), as que consideram somente o desenvolvimento de software (aspectos computacionais), e as que mesclam ambos os aspectos. Balbino e Mattos (2021) também tecem considerações a respeito da conversa entre tais abordagens:

Ao aliar as características técnicas e pedagógicas, há a necessidade de organizar e padronizar métodos de trabalho que possam contribuir com o processo de construção de OA, que atendam às necessidades educacionais. [...] A partir dessas proposições, indicamos a necessidade da adoção de uma metodologia que vise o equilíbrio entre as áreas técnica e pedagógica. (BALBINO; MATTOS, 2021, p. 60).

Nesse sentido, foram encontradas em Kemczinski *et al.* (2012) algumas metodologias, que estão compiladas no Quadro 9.

Quadro 9 – Metodologias para a produção de OA

Metodologia	Instituição	Particularidades
ADDIE	Univ. Porto	Utiliza eXe-Learning para autoria, atendendo ao padrão SCORM.
MACOBA	Univ. Politecnica de Aguascalientes	Atende ao padrão IMS-LD.
CETL	Univ. Metrop. de Londres; Univ. de Cambridge; Univ. de Nottingham	Utiliza o Reload Editor para autoria.
MEAMOA	Univ. Federal do Rio Grande do Sul (UFRGS)	Utiliza eXe-Learning para autoria, atendendo ao padrão SCORM.
Aprendizagem Significativa	Univ. Federal da Paraíba (UFPB)	Utiliza Adobe Flash, HTML e XML para autoria.
Própria	Univ. Estadual do Norte do Paraná (UENP)	Utiliza Adobe Flash para autoria.
RIVED	Seed / Ministério da Educação	Utiliza Adobe Flash para autoria

Fonte: KEMCZINSKI *et al.* (2012, p. 2, adaptado)

Motta e Kalinke (2019) apresentam uma metodologia, a qual denominaram “Metodologia de Produção de um OA na Dimensão Educacional (MPEDUC)”, que “associa as ideias de Cordeiro *et al.* (2007) sobre o método RIVED e de Kemczinski *et al.* (2012) sobre os instrumentos de produção de um OA e a utilização de softwares ou ambientes virtuais acessíveis ou que utilizam programação intuitiva.” (MOTTA; KALINKE, 2019, p. 208).

A MPEDUC preconiza a dimensão educacional, porém sem descartar aspectos da dimensão técnica, organizando a produção de OA em quatro fases: planejamento, produção, validação e divulgação. O Quadro 10 apresenta uma compilação de fases e etapas da MPEDUC.

Quadro 10 – Fases e etapas da MPEDUC

Fase	Etapas	Objetivo
Planejamento	Informações gerais do OA	Identificar os elementos básicos do OA: objetivos, conteúdo específico e público-alvo.
	Mapa conceitual	Representar o conteúdo que será proposto, estruturando-o de forma sequencial e lógica.
	Roteiro	Estruturar um roteiro das telas do objeto, de forma a apresentar a sequência de utilização do OA.
	Mapa de cenário	Contextualizar o OA, apresentando uma situação presente à realidade do público-alvo.
	Mapa navegacional	Compreender o direcionamento, permitindo verificar se a navegabilidade representa o percurso das aprendizagens propostas.

Produção	Escolha da ferramenta	Escolher a ferramenta que mais se aproxima da das competências do professor ao manipular.
	Desenvolvimento do OA	Desenvolver o OA utilizando a ferramenta intuitiva escolhida.
	Elaboração da guia do professor	Elaborar um material que sirva de suporte para que os professores aproveitem as potencialidades do OA.
Validação	Teste de viabilidade	Verificar se o OA atende aos parâmetros previstos na dimensão educacional.
Divulgação	Escolha do repositório	Identificar um ambiente virtual no qual o OA possa ser compartilhado.
	Elaboração dos metadados	Elaborar os metadados do OA, de acordo com o repositório, para potencializar sua organização e reuso.
	Disponibilização do OA	Possibilitar que outros docentes possam fazer uso do OA, adaptando-o ao seu contexto.

Fonte: MOTTA; KALINKE (2019, adaptado)

A criação da GenIA, desde a concepção até a programação, considera os aspectos metodológicos da MPEDUC.

Na fase de divulgação de um OA, alguns fatores devem ser observados na elaboração de seus metadados, visando potencializar a sua organização e reuso. Um dos fatores considerado importantes, trata da classificação.

Em uma perspectiva orientada a objetos, classificar, por si só, já é uma maneira de organizar. Sendo assim, julga-se importante conhecer algumas formas de classificar um OA.

4.3 Classificação dos OA

Os OA podem ser classificados sob diferentes dimensões. Motta e Kalinke (2019) identificaram que os OA podem ser classificados pela maneira como são empregados ou de acordo com sua utilização em um contexto específico.

Apesar de considerarem relevantes ambas as classificações, Motta e Kalinke (*ibid.*) propõem uma classificação resultante de “uma adaptação da tipologia dos softwares educacionais adotada por Vieira (1999).” (MOTTA; KALINKE, 2019, p. 206). O Quadro 11 apresenta a classificação proposta.

Quadro 11 – Classificação dos OA de acordo com os objetivos pedagógicos

Classificação	Descrição	Exemplos
Informacionais	São objetos que transmitem informações de forma organizada e lógica.	Guias didáticos digitais, videoaulas, ebooks, apostilas, sequências didáticas instrucionais, sites, apresentações de conteúdos interativos, dentre outros.
Instrução Programada	São objetos que apresentam características instrucionais do tipo exercícios e prática.	Jogos digitais, aplicativos educacionais móveis, atividades virtuais de exercício e prática, dentre outros.
Simulação	São objetos que possibilitam a imersão do estudante de espaços e situações fora de seu contexto.	Ambientes virtuais, realidade aumentada, realidade virtual, jogos virtuais, simulações, aplicativos educacionais móveis, dentre outros.
Multimídia	São objetos que priorizam os sentidos, por meio da utilização de recursos auditivos e/ou visuais.	Videoaulas, <i>podcasts</i> , repositório de imagens, pinturas e mosaicos virtuais, dentre outros.
Aplicativo	São objetos desenvolvidos para serem utilizados em <i>tablets</i> , <i>smartphones</i> ou similares.	Jogos, atividades recreativas, exercícios online, tutoriais, dentre outros.
Programação	São objetos desenvolvidos com softwares de programação intuitiva, tais como: Scratch e App Inventor.	Jogos virtuais, exercícios e prática, atividades recreativas, simulações, dentre outros.
Jogos Digitais	São objetos que possuem como objetivo proporcionar a aprendizagem por meio de desafios, motivação e competição.	Jogos virtuais, exercícios e prática, aplicativos educacionais móveis, dentre outros.
Modelagem	São objetos que permitem a modelação, pelo próprio estudante, de situações não possíveis no contexto natural.	Ambientes virtuais, simulações, realidade aumentada, realidade virtual, jogos virtuais, dentre outros.

Fonte: MOTTA; KALINKE (2019, p. 206, adaptado)

Uma vez compreendidas as características dos OA, as possibilidades metodológicas para sua construção e suas diferentes classificações, julgou-se importante explorar ambientes e aplicativos que pudessem contribuir com a concepção e o desenvolvimento da GenIA.

Sendo a GenIA uma ferramenta para construção de OA de Matemática, recorreu-se a recursos existentes, que fossem utilizados pela comunidade da Educação Matemática em suas práticas pedagógicas, e que considerassem aspectos da programação intuitiva, pois este é um dos objetos de estudo desta pesquisa. A seção a seguir apresenta alguns dos recursos visitados.

4.4 Ambientes para construção e compartilhamento de OA

Além de compreender as concepções teóricas, incluindo aspectos técnicos e pedagógicos a respeito da produção de OA, é preciso escolher uma ferramenta digital para, de fato, construí-los, segundo a metodologia adotada. O Quadro 12 apresenta uma compilação de alguns dos ambientes que contribuíram com esta pesquisa.

Quadro 12 – Ambientes digitais para construção de OA

App Inventor²⁰	Ferramenta de desenvolvimento de aplicativos para dispositivos móveis. Possibilita o desenvolvimento de aplicativos que sejam adaptados de forma a explorar a interatividade com o usuário.
Canva²¹	Plataforma de design gráfico que permite a criação de conteúdos visuais como infográficos, vídeos e animações. Está disponível on-line e em dispositivos móveis.
Geogebra²²	Software de geometria dinâmica, multiplataforma, de código aberto que possibilita a construção de simulações, jogos e aplicativos interativos em páginas Web, além de combinar diferentes áreas da Matemática, como álgebra, geometria e estatística. É, também, uma comunidade virtual.
Hot Potatoes²³	É composto por seis aplicativos para a construção de atividades interativas: JClose (preenchimento de lacunas), JMatch (associação de pares), JQuiz (questionário de múltipla escolha ou resposta curta), JCross (palavras cruzadas), JMix (ordenação de frases) e The Masher (múltiplas atividades).
Scratch²⁴	Projetado para facilitar a manipulação de mídias pelo usuário pelo método de arrastar e soltar blocos coloridos. Possibilita a criação de histórias, jogos e animações, denominados projetos. Além de criar, o usuário pode compartilhar e reformular outros projetos contribuindo com a comunidade on-line Scratch.
Thinglink²⁵	Plataforma destinada à elaboração de imagens e vídeos interativos e integração de diferentes recursos digitais nos objetos desenvolvidos. É possível compartilhá-los em redes sociais e incorporá-los em páginas web. Está disponível on-line e em dispositivos móveis.

Fonte: BALBINO; MATTOS (2021, p. 66, adaptado)

Existem diversas questões a serem consideradas na escolha do software que será utilizado para construção dos OA. Alguns apresentam recursos que inferem um apelo para questões visuais e estéticas, contudo sem direcionamento específico para algum tipo de conteúdo ou área de conhecimento. É o caso, por exemplo, do Canva e do Thinglink. Em contrapartida, a exemplo do Geogebra, outros apresentam recursos voltados à criação de conteúdo para uma área de conhecimento específica.

²⁰ Disponível em: <https://appinventor.mit.edu/>. Acesso em: 17 jun. 2023.

²¹ Disponível em: <https://www.canva.com/>. Acesso em: 17 jun. 2023.

²² Disponível em: <https://www.geogebra.org/>. Acesso em: 17 jun. 2023.

²³ Disponível em: <https://hotpot.uvic.ca/>. Acesso em: 17 jun. 2023.

²⁴ Disponível em: <https://scratch.mit.edu/>. Acesso em: 17 jun. 2023.

²⁵ Disponível em: <https://www.thinglink.com/>. Acesso em: 17 jun. 2023.

Não excluindo totalmente os já mencionados, observa-se que o App Inventor, o Hot Potatoes e o Scratch apresentam fortemente uma característica comum, que é relevante para esta pesquisa: eles se preocupam em potencializar a interatividade por meio de recursos de programação.

Para criação da GenIA, buscou-se referências em vários aspectos dos aplicativos que foram apresentados no Quadro 12, porém o fio condutor sempre foi a programação. Porém, era necessária uma maneira de programar que fosse, de certa forma, democrática, acessível a grupos de professores de Matemática, que não necessariamente carregam em sua base de formação a programação de computadores. Nesta perspectiva, ela teria que ser intuitiva.

Não há uma definição consensual para a programação intuitiva, então é importante conhecer como esta pesquisa entende este recurso e como ele foi implementado na plataforma, o que será abordado no próximo capítulo.

5 PROGRAMAÇÃO INTUITIVA

A definição de OA adotada por esta pesquisa, que o considera como um recurso digital que dá suporte à aprendizagem por meio de atividade interativa, carrega em si algumas inferências. A interatividade pressupõe ação e resposta, por e para quem vai utilizar o OA. Reduzindo a definição de OA aos termos “digital”, “aprendizagem”, e “interativa”, pode-se entender que o OA será programado por um professor.

Um professor da área da Computação, especificamente sob o âmbito da programação, pode se valer de suas habilidades para produzir um OA, pelo menos no que diz respeito aos aspectos técnicos. Mas e os educadores de outras áreas de conhecimento, precisariam realizar um curso de formação específica em programação de computadores? A resposta para esta pergunta deita-se sobre a programação intuitiva.

Assim como acontece com os OA, não há uma única definição amplamente difundida sobre o que é a programação intuitiva. É muito comum que o termo seja utilizado para, de forma genérica, referir-se a uma forma de programar, por meio da manipulação de elementos visuais relacionados a tarefas específicas, que quando combinados resultam em uma ação que será executada pelo computador.

Balbino *et al.* (2021) propõem uma definição para a programação intuitiva no contexto educacional:

[...] pode-se propor uma definição para programação intuitiva como sendo uma linguagem de programação destinada à construção de projetos educacionais em ambientes computacionais que não necessitem o domínio de uma linguagem de programação específica e que apresentem características de similaridade, visualização e acessibilidade. (BALBINO *et al.*, 2021, p. 19)

Considerando que esta pesquisa promove uma aproximação da Ciência da Computação com a Educação Matemática, justifica-se uma visita aos diferentes contextos. O aporte teórico apresentado nas seções seguintes visa estabelecer algumas premissas necessárias à compreensão de como a programação intuitiva acontece na GenIA, o que será aprofundado na seção 5.5.

5.1 Programação de Computadores

A palavra programa refere-se, em aspectos gerais, a uma lista de itens organizados em sequência e de forma temporal. Para programar um computador, parte-se do mesmo princípio. Nesse contexto, uma sequência lógica e finita de passos, de comandos, destinados a resolver um determinado problema ou executar uma tarefa, é chamada de algoritmo. Zatti (2017) considera que um algoritmo pode ser notado de três formas, a saber: narrativa, gráfica ou pseudocódigo.

A forma narrativa de um algoritmo acontece em linguagem natural, expressão utilizada para se referenciar a linguagem humana. Esta linguagem é passível de interpretação e não oferece um grau aceitável de formalismo. Por outro lado, um algoritmo pode ter notação gráfica, sendo uma das formas o fluxograma²⁶, no qual diferentes figuras denotam funcionalidades específicas. Por fim, o pseudocódigo é a maneira que mais se aproxima de uma linguagem de programação, pois traz a sequência de comandos transcritos em uma linguagem próxima à linguagem natural, porém com alto grau de formalismo. Neste caso, são utilizadas palavras específicas para se referir aos comandos: compute, escreva, leia. O Quadro 13 apresenta um mesmo algoritmo sob as três diferentes formas de notação.

Quadro 13 – Formas de notação de um algoritmo

Narrativa (linguagem natural)	Gráfica (fluxograma)	Pseudocódigo (português estruturado)
Sejam a e b dois números, informados pelo usuário, calcule a soma entre eles e apresente o resultado	<pre> graph TD Inicio([início]) --> LeiaA[/leia a/] LeiaA --> LeiaB[/leia b/] LeiaB --> CalcS[s ← a + b] CalcS --> EscrevaS[/escreva s/] EscrevaS --> Fim([fim]) </pre>	<p>Algoritmo "somadoisnumeros"</p> <p>Var</p> <p>a, b, s: <u>inteiro</u></p> <p>Inicio</p> <p>leia (a)</p> <p>leia (b)</p> <p>s ← a + b</p> <p>escreva (s)</p> <p>Fimalgoritmo</p>

Fonte: BALBINO *et al.* (2021, p. 4, adaptado)

Até recentemente, nenhuma das formas mencionadas eram utilizadas diretamente para criar um programa de computador, sendo que a maneira mais

²⁶ Além do fluxograma, há outras formas de representação gráfica de um algoritmo, como, por exemplo, o diagrama de Chapin e o diagrama de Nassi-Shneiderman.

comum de se programar ainda é por meio de uma linguagem de programação específica, algo que será detalhado mais para frente, ainda nesta seção.

Quando um algoritmo é convertido para uma sequência que possa ser executada por um computador (ou equipamento digital), ele passa a ser um programa. Nesta perspectiva, a criação de uma sequência de comandos para um computador, utilizando uma linguagem específica para este fim, é denominada programação. (BALBINO *et al.*, 2021, p. 4).

Atualmente é comum que computadores ou equipamentos digitais recebam instruções em linguagem natural, tanto na forma escrita quanto por voz, e que também atuem em questões que envolvam o reconhecimento de imagens. Sob esta perspectiva, seria possível que um computador pudesse, sim, ser programado por qualquer das três notações apresentadas. Contudo, para melhor elucidar tais possibilidades, faz-se necessário uma breve visita às origens da computação.

Os primeiros computadores funcionavam de forma analógica. Como sua única funcionalidade, naquele momento, era a realização de cálculos (e daí vem a palavra computador), a programação se dava pela combinação de rodas dentadas ou outros componentes físicos.

A partir dos anos 1940, os computadores passaram a ser digitais. Esta denominação deve-se ao fato de seu funcionamento ser baseado em sequências de liga-desliga. Naquela época, a programação era feita por meio de chaves e botões, e a execução acontecia por relés eletromagnéticos, que alternavam o estado de conexão do circuito, permitindo ou bloqueando a passagem de corrente elétrica.

Do relé para a válvula, da válvula para os transístores, dos transístores para os circuitos integrados. Estes últimos ainda são a base dos computadores digitais atuais. Dessa maneira, os cálculos e a transmissão de dados são feitos no sistema binário de numeração (baseado em zeros e uns), e os comandos obedecem a decisões proposicionais (baseadas em verdadeiro e falso).

Mesmo com a evolução tecnológica, o princípio não mudou e os computadores ainda trabalham com sequências binárias, que são conhecidas como linguagem de máquina. Sendo assim, para programar um computador, é necessário converter as sequências de comandos que idealizamos em um algoritmo para a linguagem que o computador entende. É neste ponto que entram as linguagens de programação. (BALBINO *et al.*, 2021, p. 5).

Para programar um computador, a sequência de tarefas idealizada no algoritmo precisa ser expressa em comandos que possam ser convertidos em sequências binárias. Com isso, adota-se uma linguagem de programação destinada ao sistema que se deseja programar, como por exemplo o C++, Java, C# ou Python.

A partir da escolha da linguagem, os comandos devem ser escritos em um arquivo de texto, chamado de código fonte, e a sintaxe deve ser rigorosamente respeitada. Lévy (2004) pondera que as linguagens de programação não são línguas e deveriam ser chamadas “escritas de programação”, por conta de seu formalismo baseado na Lógica Matemática.

O código fonte precisa, portanto, ser convertido em sequências binárias. Isto pode ser feito de duas formas: interpretada ou compilada. Na primeira forma, um programa (denominado interpretador) irá converter os comandos e executá-los um a um. Na segunda forma, o código fonte inteiro passa por um processo denominado compilação. Na compilação, um programa (conhecido como compilador) irá fazer a conversão de todos os comandos para a linguagem de máquina, tendo como resultado as sequências binárias que serão executadas pelo equipamento. Ressalta-se que o compilador também é um programa que outrora foi compilado.

Sebesta (2016) sugere que as linguagens de programação podem ser categorizadas sob três paradigmas distintos: imperativo, funcional e lógico. No paradigma imperativo, a programação acontece por meio de comandos que devem ser “obedecidos”. O paradigma funcional abriga as linguagens que resolvem os problemas mediante aplicação de funções para parâmetros fornecidos. Por fim, no paradigma lógico, são estabelecidas regras e combinações para se chegar a um resultado ou inferência. O Quadro 14 apresenta um panorama dos paradigmas mencionados.

Quadro 14 – Os diferentes paradigmas de programação

Paradigma Imperativo	
Código do programa em C++:	Execução/uso do programa:
<pre>void somadoisnumeros() { int a, b, s; std::cin >> a; std::cin >> b; s = a + b; std::cout << s; }</pre>	<p>2 3 5</p>

Paradigma Funcional	
Código do programa em LISP:	Execução/uso do programa:
<pre>(defun dobro (N) "Calcula 2 vezes n" (* 2 N))</pre>	<pre>USER(1): (dobro(5)) 10</pre>
Paradigma Lógico	
Código do Programa em Prolog ²⁷ :	Execução/uso do programa:
<pre>conhece(joao,tecnologia). conhece(maria,tecnologia). conhece(joao,matematica). conhece(maria,matematica).</pre>	<pre>?- conhece(joao,tecnologia) yes. ?- conhece(joao,X) X = tecnologia X = matematica ?- conhece(Y,matematica) Y = joao Y = maria</pre>

Fonte: BALBINO *et al.* (2021, p. 6, adaptado)

O paradigma imperativo abriga a maioria das linguagens de programação comerciais, que são utilizadas no desenvolvimento de sistemas voltados ao uso de banco de dados. Sob este paradigma também estão os aplicativos de uso geral, incluindo os que são executados em dispositivos móveis.

Segundo Zatti (2017), também é possível classificar as linguagens de programação de acordo com seu grau de abstração. São denominadas linguagens de baixo nível aquelas que possuem um baixo grau de abstração, isto é, que fazem uso de símbolos mnemônicos e instruções muito próximas da linguagem de máquina.

Em contrapartida, as linguagens que possuem sintaxe mais próxima da linguagem natural são classificadas como linguagens de alto nível. Mesmo fazendo uso de uma linguagem de alto nível, um programador profissional precisa ter conhecimento específico da área de programação.

Sendo assim, é essencial partir do domínio das noções básicas de *hardware* e *software*, ter raciocínio lógico e saber construir algoritmos eficazes para que se possa criar programas utilizando diferentes linguagens, cada uma com as suas regras, em diferentes níveis de abstração e sob diversos paradigmas. A escolha da linguagem depende dos objetivos traçados para cada projeto. (BALBINO *et al.*, 2021, p. 6).

A partir do exposto, sugere-se que a programação intuitiva esteja em um nível ainda superior ao das linguagens classificadas como de alto nível, pois ela precisa

²⁷ No Prolog, o código não diz respeito a instruções, sendo, nesse caso, definições para construção de uma base de conhecimento; e as respostas, por sua vez, não são resultado direto de instruções, mas sim, resultados de inferências sobre a base construída.

abstrair as questões de software e hardware, posicionando-se em maior proximidade com a linguagem natural. Esta modalidade de programação seria, portanto, a mais adequada ao contexto educacional e, por ser o tipo de programação usada para construção dos OA na GenIA, o assunto será devidamente explorado nas próximas seções.

5.2 Programação e fases das TD no contexto educacional

Resnick *et al.* (2009) afirmam que a metodologia pautada na programação pode estimular o desenvolvimento cognitivo das crianças que atuam na busca de soluções mentais para problemas e desafios que lhe são propostos. Sendo assim, a programação pode ser considerada uma TD, cuja inserção nos processos educacionais encontra fundamentação teórica em Tikhomirov (1981), Lévy (2010, 2011) e Kenski (2011).

Um movimento de busca pela inserção de programação no âmbito educacional, com contribuições para a Educação Matemática, recai sobre o trabalho de Borba, Scucuglia e Gadanidis (2018), que propõem uma organização cronológica da utilização das TD no contexto da Educação Matemática, dividida em quatro fases²⁸.

A primeira fase é caracterizada pelo uso do Logo²⁹, o que coloca a programação como protagonista das TD daquele momento. A principal perspectiva teórica sobre seu uso pedagógico é a teoria do Construcionismo de Seymour Papert (1980), que enfatiza a construção do conhecimento baseada em ações concretas desenvolvidas por meio do uso do computador, que sejam de interesse do usuário.

“A teoria de Papert (1985) contribui para que sejam superadas as dificuldades das crianças no ensino de Matemática, em especial de geometria, de maneira que o aluno é o sujeito ativo no processo de construção do conhecimento.” (MOTTA, 2008, p. 62).

Balbino *et al.* (2021) consideram que a linguagem Logo poderia ser considerada uma precursora da programação intuitiva, ao afirmarem que:

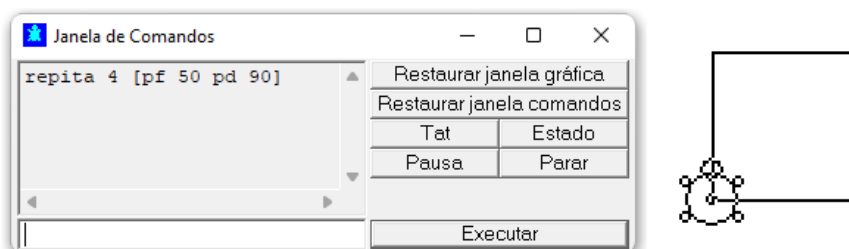
²⁸ Há um livro, intitulado “Vídeos na educação matemática: Paulo Freire e a quinta fase das tecnologias digitais”, que sugere uma quinta fase. Porém, trata-se de uma obra recentemente lançada (fev. 2022) e, portanto, com propostas ainda não consolidadas.

²⁹ “Linguagem de programação criada no final da década de 60, idealizada por Seymour Papert. A palavra ‘Logo’ é utilizada para se referir tanto à linguagem em si quanto ao software no qual a linguagem é empregada.” (BALBINO *et al.*, 2021, p. 7).

O Logo apresentou as primeiras características do que, futuramente, evoluiria em busca de se tornar uma linguagem de programação que pode remeter à noção de intuitiva, uma vez que para seu uso não eram necessários conhecimentos avançados de programação. (BALBINO *et al.*, 2021, p. 7).

O princípio de funcionamento do Logo é a existência de um ícone representando uma tartaruga, que recebe comandos destinados à sua movimentação, podendo assim desenhar imagens na tela do computador. Como exemplo, é possível criar uma imagem que represente a figura geométrica de um quadrado com a sequência de comandos: `repita 4 [pf 50 pd 90]`, conforme ilustra a Figura 7.

Figura 7 – Quadrado desenhado por meio de programação Logo



Fonte: BALBINO *et al.* (2021, p. 8, adaptado)

Os comandos “pf” e “pd” são as iniciais de “para frente” e “para direita”, respectivamente. O número 50, à direita do comando “pf”, corresponde à quantidade de passos que a tartaruga vai se deslocar; e o número 90, à direita do comando “pd”, corresponde à quantidade de graus que a tartaruga irá girar. A sequência de deslocamento e giro, quando repetida 4 vezes, resulta no desenho do quadrado.

A linguagem Logo propõe colocar a criança no comando de um robô, ou de uma representação dele por meio da tela do computador. A representação simbólica de uma tartaruga como modelo de robô pode ser compreendida como uma forma de utilização da arte na comunicação das ideias. Nesta proposta, estudantes e professores podem ser considerados artistas que elaboram suas próprias estratégias para criar ou resolver projetos de seu interesse. O Logo proporciona ao sujeito a imersão em um micromundo de interatividades com o software. (BALBINO *et al.*, 2021, p. 8).

O uso das iniciais para representação dos comandos infere que seu uso seja simplificado, principalmente pela facilidade de memorização, ainda que possa se discutir se é intuitivo. Tal característica é análoga ao princípio da programação de baixo nível, quando as sequências binárias começaram a ser substituídas pelos símbolos mnemônicos. Como exemplo, a linguagem Assembly utiliza o símbolo MOV

para movimentar um dado de uma posição de memória para outra, e à sua direita são informados os endereços de memória referentes à origem e destino para movimentação do dado (TANENBAUM, 2013).

Ainda que a popularidade do Logo tenha diminuído com o passar do tempo e com a inserção de novas linguagens de programação visuais no contexto educacional, a ideia central influenciou outros ambientes de programação intuitiva, como o Scratch, e a GenIA também incorpora algumas características do Logo, o que será explorado na seção 5.5.

A segunda fase do uso de tecnologias educacionais que estabelecem relação com a programação é caracterizada pela produção de software educacional que possibilitou explorar diferentes práticas pedagógicas.

Especificamente para a Matemática, destacam-se o Winplot, Cabri Géomètre, Maple, entre outros³⁰. Borba, Scucuglia e Gadanidis (2018, p. 27) afirmam que esses aplicativos “são caracterizados não apenas por suas interfaces amigáveis, que exigem pouca ou nenhuma familiaridade com linguagens de programação, mas principalmente pela natureza dinâmica, visual e experimental.”

Percebe-se, portanto, que essa natureza visual e as interfaces amigáveis, mais uma vez, elevam o nível de abstração no que tange a interatividade, distanciando o usuário dos detalhes intrínsecos à programação, ainda que tais recursos estivessem ali para serem explorados pelos que tivessem alguma experiência em programação.

A internet é a protagonista da terceira fase das tecnologias. Sua popularização trouxe um novo viés para a comunicação e manipulação de dados e informação. Consolidou-se o termo TIC, referente às Tecnologias da Informação e Comunicação, em um universo em que residia a sigla TI (Tecnologia da Informação). Borba, Scucuglia e Gadanidis (2018) apontam que a difusão de ambientes virtuais de aprendizagem (AVA) possibilitou interações coletivas, de forma síncrona e assíncrona.

Nesta fase, são disponibilizados diversos aplicativos que permitem a criação e publicação de conteúdo na internet, sem a necessidade de domínio sobre as *tags*³¹

³⁰ “WinPlot é um programa para gerar gráficos de 2D e 3D a partir de funções ou equações matemáticas. O Cabri Géomètre é um software de geometria dinâmica. Maple é um sistema algébrico computacional comercial de uso genérico.” (BALBINO *et al.*, 2021, p. 9).

³¹ Uma *tag* (em português: etiqueta, marcação) HTML é uma sequência de letras, análoga a um comando, utilizada para formatar o conteúdo que será visualizado por meio de um navegador de internet. A palavra também é um acrônimo para *task-action grammar* (gramática de ação-tarefa).

HTML (*Hypertext Markup Language* – Linguagem de Marcação de Hipertexto). Antes do advento desses aplicativos, quem iria disponibilizar conteúdo na internet precisava inserir manualmente o código com *tags* HTML.

Atualmente esse processo é abstraído pelos aplicativos editores de conteúdo, tornando a tarefa mais fácil para o usuário. Balbino *et al.* (2021) pontuam que se tornou tão corriqueiro que pode ser realizado por muitos usuários, incluindo estudantes da Educação Básica.

A quarta fase do uso de TD, vivida atualmente, iniciou com a melhoria das conexões e dos tipos de recursos utilizados nessa esfera, o que potencializou a comunicação online. O momento tem como principais características o uso massivo de redes sociais e *streaming*³² de vídeo, música ou *podcast*³³, sendo utilizados tanto para lazer quanto no âmbito educacional, em conjunto com software educacional e OA. Diante de tantas possibilidades, essa fase revela “um cenário exploratório, fértil ao desenvolvimento de investigações e à realização de pesquisas.” (BORBA; SCUCUGLIA; GADANIDIS, 2018, p. 41).

5.3 Software para programação na Educação

São diversas as possibilidades de uso de programação na Educação, no que tange os aplicativos, recursos e ambientes. Na busca por recursos que pudessem contribuir com a criação da GenIA, esta pesquisa lançou um olhar especial sobre aqueles que podem ser utilizados para a construção de OA. No Quadro 12 foram listados alguns ambientes, porém nem todos fazem uso de programação.

Dos ambientes já mencionados, dois merecem especial atenção, e serão abordados nesta seção: o Scratch e o App Inventor. O Scratch foi criado MediaLab do Instituto de Tecnologia de Massachussetts (MIT), berço do Logo. O App Inventor foi criado pela Google, porém atualmente também está sob a responsabilidade do MIT.

No que se refere à programação, Scratch e App Inventor são muito similares, pois a criação de um programa é feita por meio de blocos lógicos relativos a

³² O termo *streaming* refere-se à transmissão contínua de dados via internet para consumo de conteúdo multimídia, como assistir vídeo ou ouvir música.

³³ Um *podcast* é uma transmissão de áudio por meio de *streaming*, normalmente com conteúdo informativo, como notícias, entrevistas ou narrações de livros e outros conteúdos textuais.

funcionalidades específicas. Tais blocos, que são análogos aos elementos de programação (comandos, operadores, variáveis etc.), são agrupados por cores, conforme pode ser observado na Figura 8.

Figura 8 – Blocos gráficos no Scratch



Fonte: BALBINO et al. (2021, p. 8, adaptado)

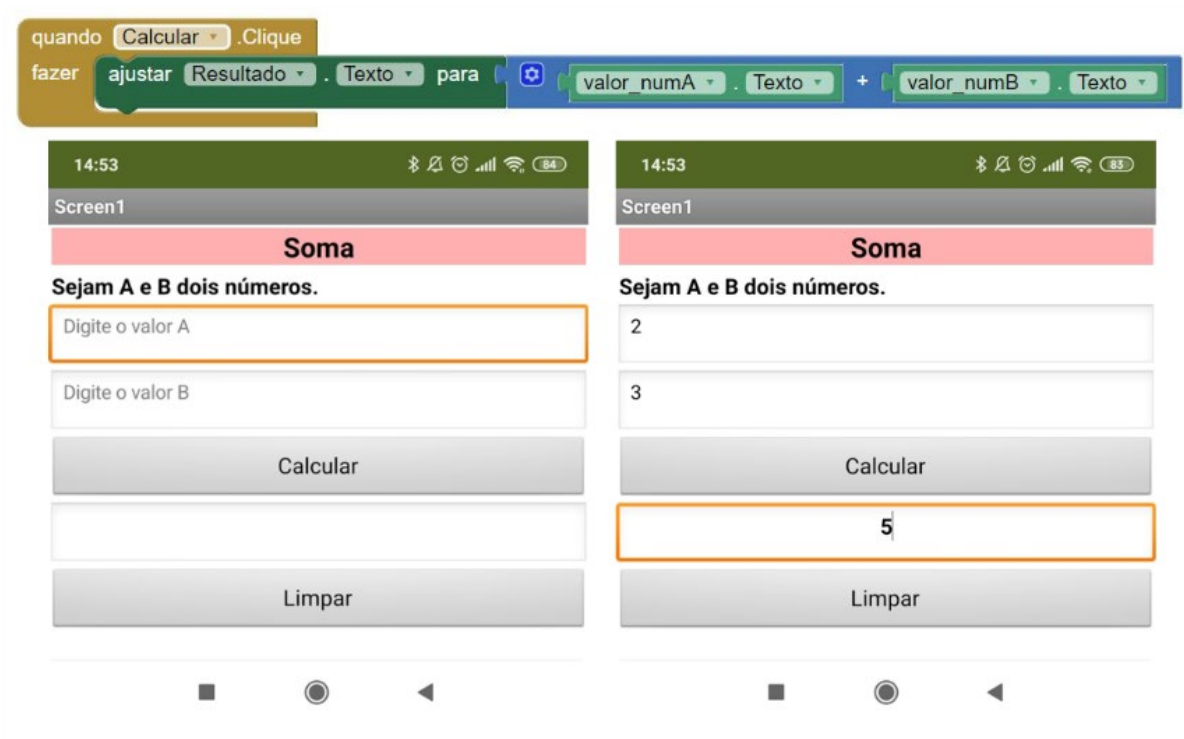
No exemplo apresentado, o programa solicita que o usuário digite dois números, que são armazenados em variáveis denominadas “Número A” e “Número B”, e guarda a soma dos números informados em uma terceira variável denominada “Resultado”.

A diferença fundamental entre os dois aplicativos mencionados é o ambiente e equipamento em que são executados: enquanto o Scratch é usado em computadores pessoais, o App Inventor é destinado à criação de *apps*³⁴ para dispositivos móveis.

³⁴ “O termo *app*, abreviatura de *application* (do inglês: aplicação ou aplicativo) é utilizado popularmente para se referir especificamente aos aplicativos para dispositivos móveis, tais como os *smartphones* e os *tablets*.” (BALBINO et al., 2018, p. 11)

Um programa similar ao apresentado anteriormente no Scratch pode ser visualizado na Figura 9.

Figura 9 – Blocos gráficos e telas no App Inventor



Fonte: BALBINO *et al.* (2021, p. 13, adaptado)

A programação também acontece por blocos, porém observa-se que, neste caso, há uma construção (desenho) prévia das telas que serão exibidas no dispositivo móvel, e o código, quando executado, opera com os valores dos campos já existentes.

Balbino *et al.* (2021, p. 14) consideram que “É importante que não se confunda programação por blocos com ‘programação intuitiva’. Certamente o modelo de programação por encaixe de blocos com programações predefinidas pode ser um dos modelos de programação intuitiva.”

O Hot Potatoes é um exemplo de aplicativo que possibilita a criação de conteúdo interativo, por meio da manipulação visual de elementos, e gera automaticamente o código HTML correspondente à sua execução, sem que o usuário precise sequer visualizá-lo.

Com objetivo de implementar na GenIA uma programação que poderia ser considerada intuitiva, buscou-se na literatura quais aspectos deveriam ser observados que pudessem caracterizar a ação de programar sob tal perspectiva.

5.4 Os pilares da programação intuitiva

No sentido de encontrar elementos que pudessem embasar o que seria intuitivo em termos computacionais, Balbino *et al.* (2021, p. 16) apontam que “uma ferramenta computacional baseada na intuição possui preocupações com três pilares: a similaridade, a visualização e a acessibilidade.”

Observa-se que tais pilares inferem que a programação intuitiva faça uso de elementos visuais (gráficos). A respeito de uma programação que faça uso de elementos gráficos, Lévy (2004) considera que

uma das principais vantagens da programação à base de imagens é evitar numerosos erros de sintaxe, pela eliminação do uso de um código esotérico. Aliás, é mais agradável programar criando e manipulando imagens do que prolongando linhas de código. (LÉVY, 2004, p. 208)

Antes de adentrar a discussão sobre cada um dos pilares, cabe destacar que houve uma preocupação com eles no desenvolvimento da GenIA. Porém, para que não haja uma compreensão reduzida de como a programação intuitiva se faz presente na plataforma, o detalhamento será comentado na seção 5.5.

O pilar da similaridade sugere que o ambiente deve apresentar ferramentas com estrutura de linguagem semelhante à do usuário. “Dessa forma, o foco principal da ação de programar se detém na criação e no desenvolvimento do produto e não na aprendizagem de uma nova linguagem.” (BALBINO *et al.*, 2021, p. 16).

Os ambientes mencionados nesta seção, além de possibilitarem o uso em diversos idiomas, trazem mensagens e indicativos gráficos atrelados aos comandos que estão sendo utilizados. A Figura 10 evidencia essa característica na programação do App Inventor.

Figura 10 – Blocos que indicam a ação de somar dois números



Fonte: BALBINO *et al.* (2021, p. 13, adaptado)

No Scratch e no App Inventor, a combinação dos blocos trabalha o pilar da visualização sob mais de um aspecto. Desde a organização dos blocos em grupos e cores até o momento do encaixe, em que o aplicativo impede que um bloco seja encaixado em uma posição na qual sua função sintática não faz sentido.

Mulholland (1997) aponta que a visualização de um programa deve utilizar-se de técnicas que buscam representar o próprio programa em execução. Nesse viés, o Scratch, por exemplo, traz em sua interface o *stage* (palco), região na qual observa-se, parcialmente, a execução do programa durante a sua criação.

Por fim, o pilar da acessibilidade considera que os comandos básicos da interface estejam disponíveis em botões e ícones. “A presença de ícones representativos, que apresentam uma estrutura de linguagem similar à do usuário, contribui com a acessibilidade do programa.” (BALBINO *et al.*, 2021, p. 18).

Tendo em vista os aspectos destacados a respeito da programação intuitiva, a próxima seção apresenta como ela foi abordada no desenvolvimento da GenIA.

5.5 Programação intuitiva na GenIA

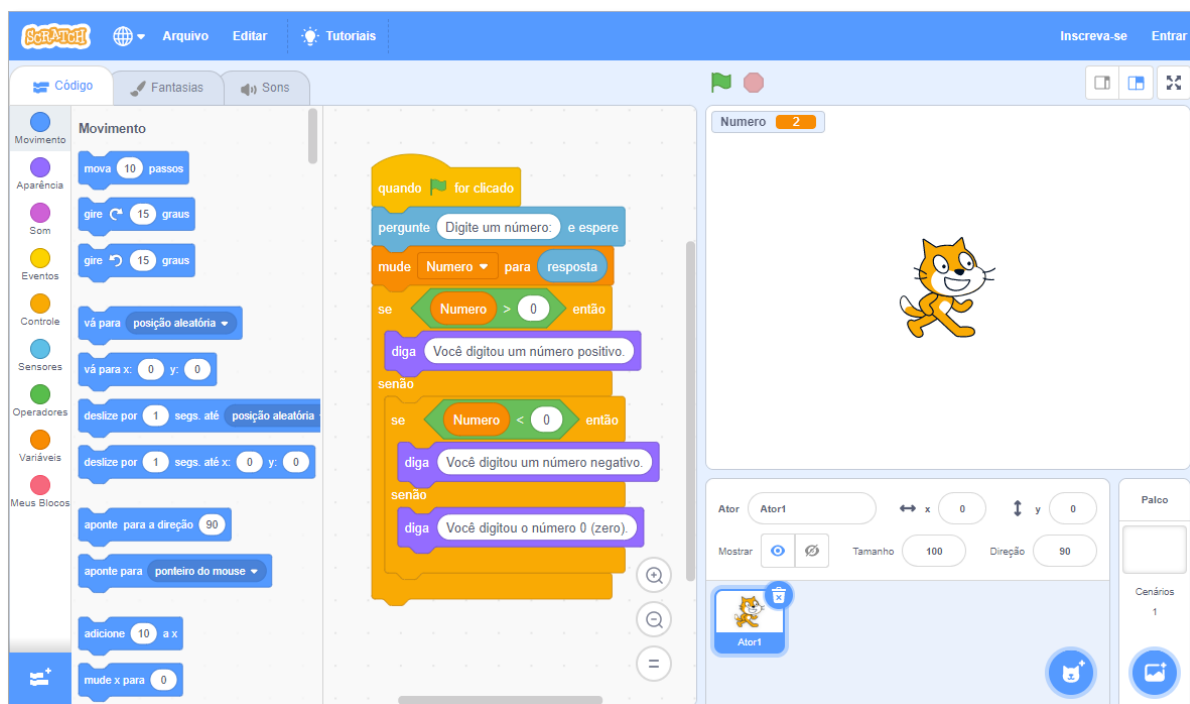
Um dos desafios que surgiram durante a concepção da GenIA foi no sentido de como ela poderia proporcionar uma programação que fosse intuitiva. Na busca de respostas para este questionamento, foram contemplados os ambientes de programação considerados intuitivos apresentados na seção 5.3.

A análise dos ambientes buscou observar dois aspectos, a saber: 1) como a intuição era tratada na interface do ambiente; 2) em que aspectos a programação intuitiva em tais ambientes era semelhante ou se distanciava da programação convencional.

Considerando-se o histórico do Logo e suas contribuições no contexto educacional, optou-se por analisar o Scratch. Tal escolha apoia-se em duas justificativas: 1) ele é considerado uma evolução do Logo; 2) ele se faz presente em um número expressivo de pesquisas e trabalhos publicados que investigam a programação no âmbito educacional, com atenção especial às publicações do GPTEM.

Iniciou-se a exploração pela interface do Scratch, apresentada na Figura 11.

Figura 11 – Interface do Scratch



Fonte: scratch.mit.edu

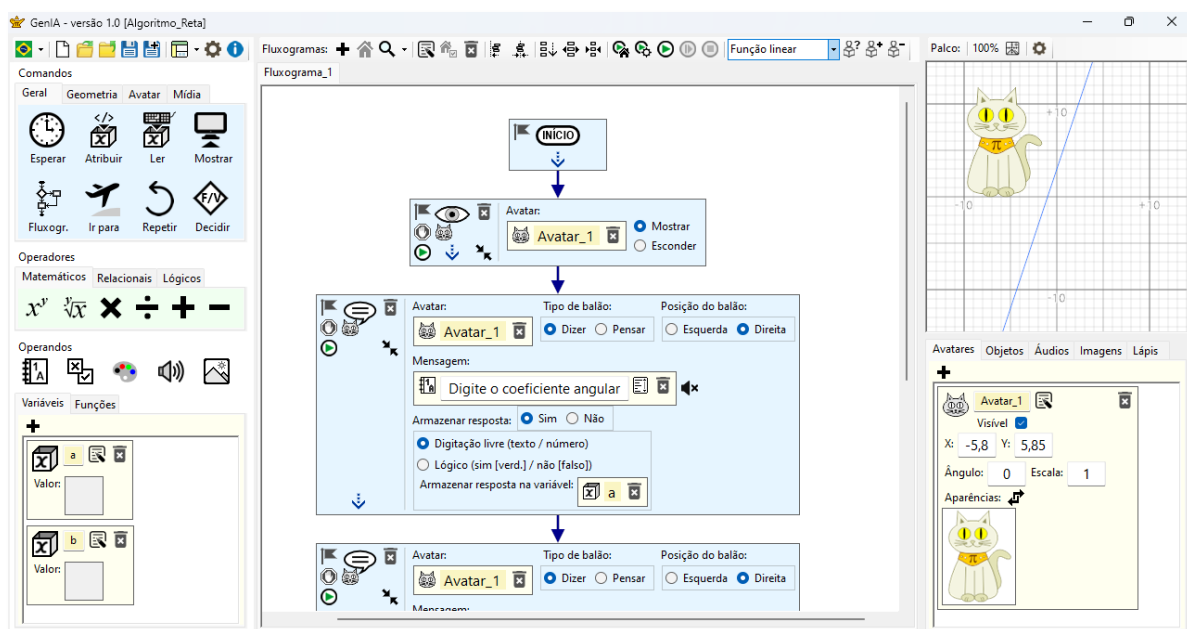
Percebe-se que sua interface pode ser considerada intuitiva, quando observada sob a óptica dos três pilares da intuição apontadas por Balbino *et al.* (2021), a serem lembrados: similaridade, visualização e acessibilidade.

O pilar da acessibilidade revela-se presente de imediato, pela apresentação de comandos representados por botões e ícones. Sob o pilar da similaridade, seus menus, botões e blocos foram traduzidos para diversos idiomas. Além disso, os

elementos apresentam dicas gráficas ao usuário. O pilar da visualização é evidenciado, dentre outras formas, pela separação dos blocos em cores de acordo com sua funcionalidade.

Sendo a programação intuitiva um dos focos desta pesquisa, era imperativo que a concepção da interface da GenIA (Figura 12) também considerasse os pilares da intuição. Cabe ressaltar que esta seção tem por objetivo apresentar as evidências de implementação da programação intuitiva na plataforma, e não ensinar o leitor a construir fluxogramas. Detalhes a respeito dos comandos e funcionalidades estão disponíveis no documento de apresentação do produto.

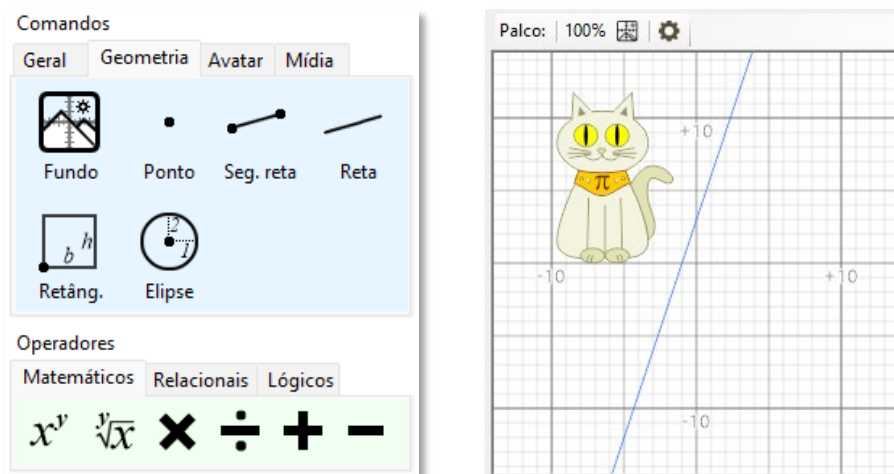
Figura 12 – Interface da GenIA



Fonte: Autoria própria (2023)

Começando pela similaridade, por se tratar de uma plataforma voltada a criação de conteúdo matemático, para atender tal pilar, os ícones e a organização dos comandos foram pensados considerando-se o conhecimento prévio de professores da área. Comandos específicos de geometria e a presença dos símbolos formais para operadores, são exemplos de tal preocupação durante a concepção da interface. Da mesma forma, o palco (área que se visualiza a execução do OA, recurso similar ao do Scratch) pode ser configurado para exibir um plano cartesiano. A Figura 13 apresenta recortes da interface, evidenciando tais elementos.

Figura 13 – Evidências do pilar da similaridade na GenIA



Fonte: Autoria própria (2023)

Ainda sob o pilar da similaridade, os textos e mensagens da GenIA podem ser visualizados tanto na Língua Portuguesa quanto na Língua Inglesa, estando estruturalmente preparada para receber prontamente traduções futuras para outros idiomas.

Com relação ao pilar da visualização, a GenIA o atende sob diferentes aspectos. O primeiro deles é que elementos gráficos, como avatares e objetos, podem ser posicionados diretamente no plano cartesiano com o habitual recurso de arrastar-e-soltar dos ambientes gráficos. Tal ação, por exemplo, reflete diretamente na alteração dos valores das coordenadas cartesianas do componente correspondente. Outras evidências da atenção dada a este pilar serão comentadas a seguir, pois dizem respeito à programação em si.

Por fim, o cuidado com o pilar da acessibilidade revela-se de imediato na medida que, de forma similar ao Scratch, os comandos são representados por botões e ícones, organizados por contexto e funcionalidade e apresentam dicas gráficas ao usuário. Além disso, a plataforma incorpora o mecanismo de conversão de texto em fala (voz), presente no Windows, possibilitando que as mensagens das caixas de diálogo (informações, alertas, perguntas) sejam narradas no idioma em operação. Tal iniciativa pretende incentivar a construção de OA que possam ser utilizados por estudantes cegos.

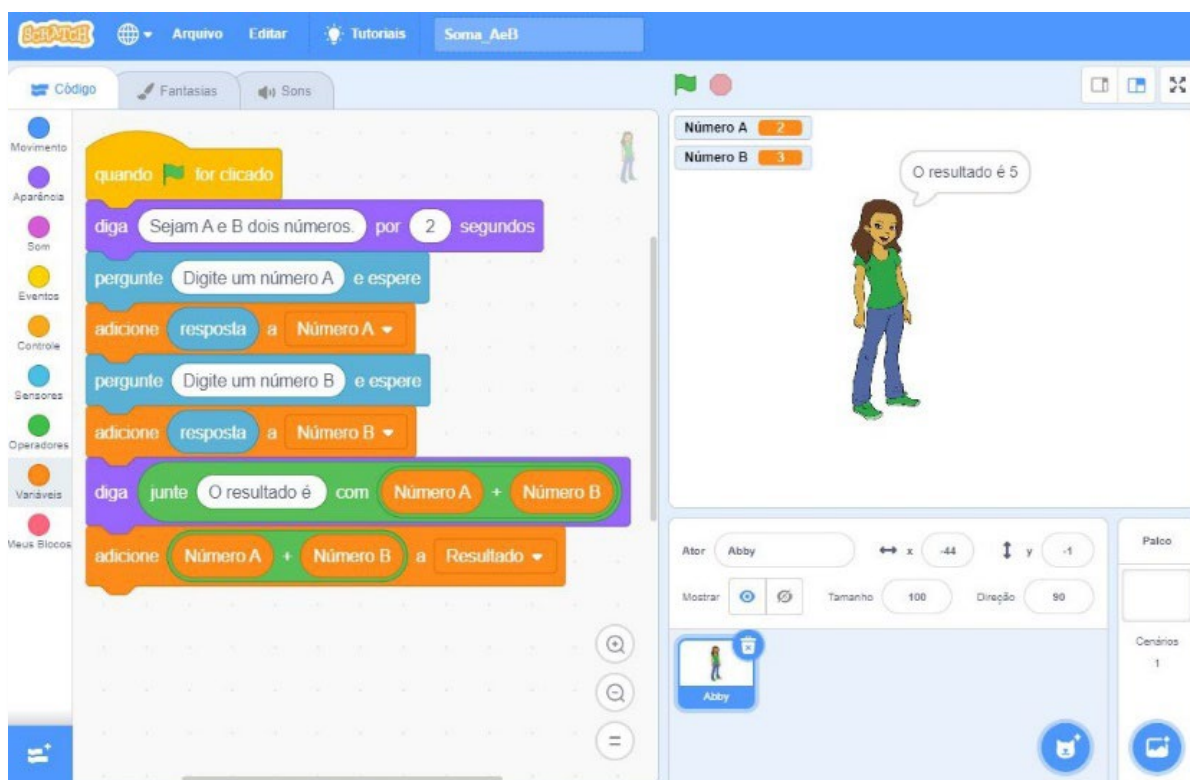
Concomitante à análise da interface, fez-se um estudo mais detalhado da programação. Ao explorar um projeto criado no Scratch, constatou-se que, em alguns aspectos, ele tem relação com a programação orientada a objetos. Tal associação

pode ser melhor compreendida observando-se as características essenciais inerentes a esse paradigma de programação:

O paradigma orientado a objetos pode ser considerado uma das formas mais assertivas de reduzir o golfo semântico sobre aquilo que se pretende representar. Isto porque a proposta é levar para o software os elementos mais próximos de como eles se encontram na natureza. Isto vem desde a observação, contextualização, análise, representação e, por fim, chegando à codificação. (ZATTI, 2017, p. 38)

Em um programa orientado a objetos, as ações normalmente são atreladas a elementos que são representações do mundo real. Tais representações são chamadas de classes. No caso do Scratch, normalmente a interatividade com o OA acontece por meio de um ator. Por conta disso, em sua programação, é comum que uma parte dos comandos (blocos) realize alguma tarefa atrelada a esse ator. Um exemplo está ilustrado na Figura 14.

Figura 14 – Projeto Scratch para somar dois números



Fonte: BALBINO *et al.* (2021, p. 11, adaptado)

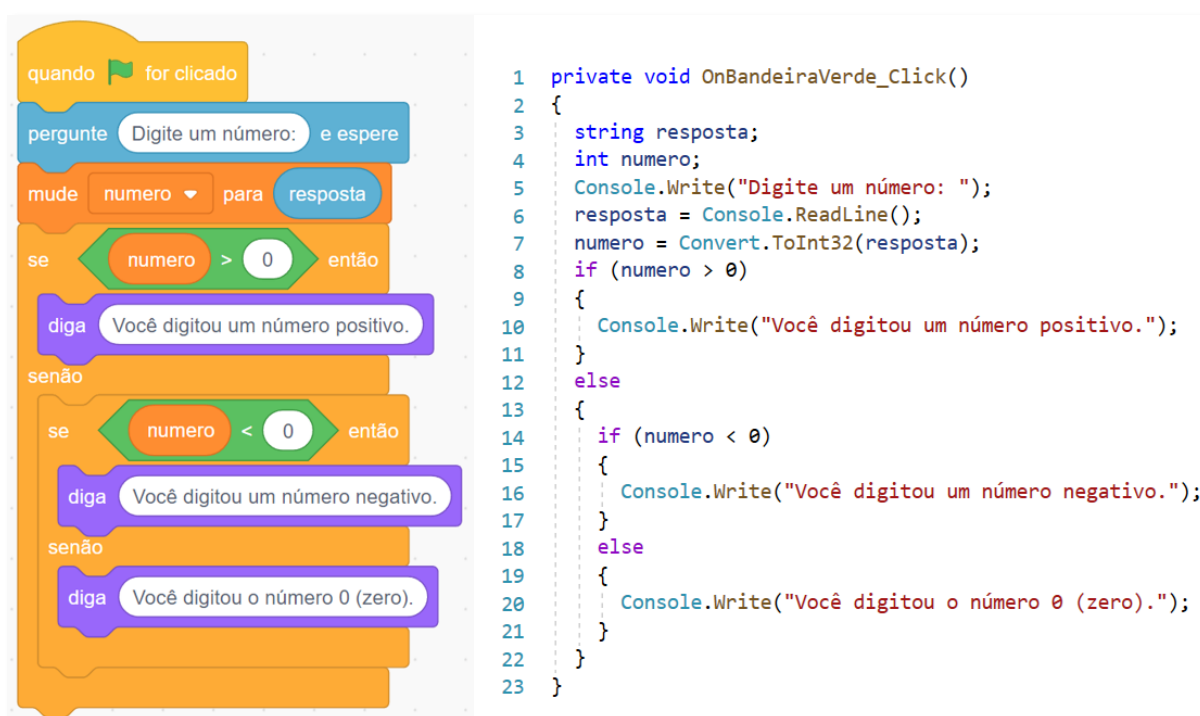
Neste caso, o conjunto de blocos apresentado está atrelado à Abby. Isso significa que, quando o programa for executado, ela é quem interage por meio dos

comandos “diga” e “pergunte”. Estabelecendo uma relação com o paradigma sugerido, entende-se, por exemplo, que Abby é um objeto da classe Ator.

Observa-se, também, que o conjunto de blocos segue uma estrutura visual que é análoga à programação estruturada. A programação estruturada é um paradigma de programação presente nos códigos das funcionalidades (métodos) de uma classe. Vale mencionar que a programação orientada a objetos é considerada, por alguns autores, como multiparadigma, e um dos fatores que sustenta essa interpretação é o fato de um método demandar elementos da programação estruturada no implementar de sua funcionalidade.

Com objetivo de elucidar essa associação, a Figura 15 estabelece um comparativo entre um código criado no Scratch e o trecho de código análogo na linguagem de programação C#.

Figura 15 – Comparativo entre blocos no Scratch e trecho de código em C#



Fonte: Autoria própria (2023)

Neste exemplo específico, optou-se por não utilizar o conjunto de blocos de Balbino *et al.* (2021) porque ele não implementa estruturas de programação aninhadas, que seriam correspondentes a um bloco dentro do outro (se/então/senão).

Comparando-se o conjunto de blocos com o trecho de código em C#, pode-se inferir que, sob a perspectiva de Lévy (2004), a programação por blocos poderia ser classificada como uma programação “vestida de imagens”:

Podemos dividir as linguagens de programação visuais em duas categorias:

1. As linguagens de programação “vestidas de imagens” não são intrinsecamente visuais, mas possuem representação visuais sobrepostas. São linguagens tradicionais com interface visual ou auxílio de tipo gráfico à engenharia informática.
2. As linguagens de programação intrinsecamente visuais procuram desenvolver novos paradigmas de programação cujas expressões são naturalmente visuais e nem sequer teriam equivalentes textuais.

(LÉVY, 2004, p. 209)

No sentido de implementar na GenIA uma forma intuitiva de programação que pudesse se aproximar do que Lévy chamou de “intrinsecamente visuais”, recorreu-se à representação gráfica de um algoritmo: o fluxograma. Neste tipo de diagrama, os comandos são interligados por setas que indicam a direção do fluxo, isto é, a sequência segundo a qual os comandos serão executados.

A adoção do fluxograma como base para a programação intuitiva da GenIA, além da perspectiva intrinsecamente visual de Lévy, foi impulsionada por uma situação particular vivenciada pelo pesquisador, relatada a seguir.

Neste momento, peço licença ao leitor para versar algumas palavras em primeira pessoa. Aprendi a programar, aos dez anos de idade, na Linguagem Logo. Estou falando do início dos anos 1980 e, portanto, daquela época em que não existiam os sistemas gráficos, como o Windows, e qualquer resultado gráfico era dependente da sequência de comandos enviados à tartaruga. A programação que aprendi era sequencial e, derivar a execução do programa para outras rotinas, acontecia por meio de um comando “mágico”: o vápara (vá para). Depois da Linguagem Logo veio a Linguagem BASIC, e os saltos entre sequências de comandos eram possíveis com o uso de um comando análogo: o GOTO (*go to*, em português: vá para). Durante muito tempo, tive toda a liberdade que me era conferida com a programação com saltos, até chegar na faculdade e me deparar com a programação estruturada das Linguagens Pascal e C. Perdi a liberdade, não pela inexistência do recurso de salto nessas linguagens (em ambas existe o comando *goto*), mas porque fazer uso dele não era uma prática aceitável, pois é algo que prejudica a legibilidade e manutenção de um

código de programação. Aqui, volto o texto para terceira pessoa, embasando a questão da legibilidade em Sebesta (2016):

“Sem restrições de uso [...] as instruções *goto* podem tornar os programas muito difíceis de ler e, como resultado, altamente não confiáveis e caros de manter.” (SEBESTA, 2016, p. 379, tradução nossa).

Sob uma perspectiva de programação no âmbito profissional, as linguagens baseadas no paradigma estruturado e, posteriormente, na orientação a objetos, tornaram o cotidiano de um programador mais fácil, principalmente ao dar manutenção em códigos criados por outros profissionais. Mas essa facilidade vem a um custo: para criar derivações e possibilidades durante a execução de um código escrito sob esses paradigmas, é necessário que o programador antecipe as intenções, e envolva blocos de comandos agrupados dentro de outros blocos.

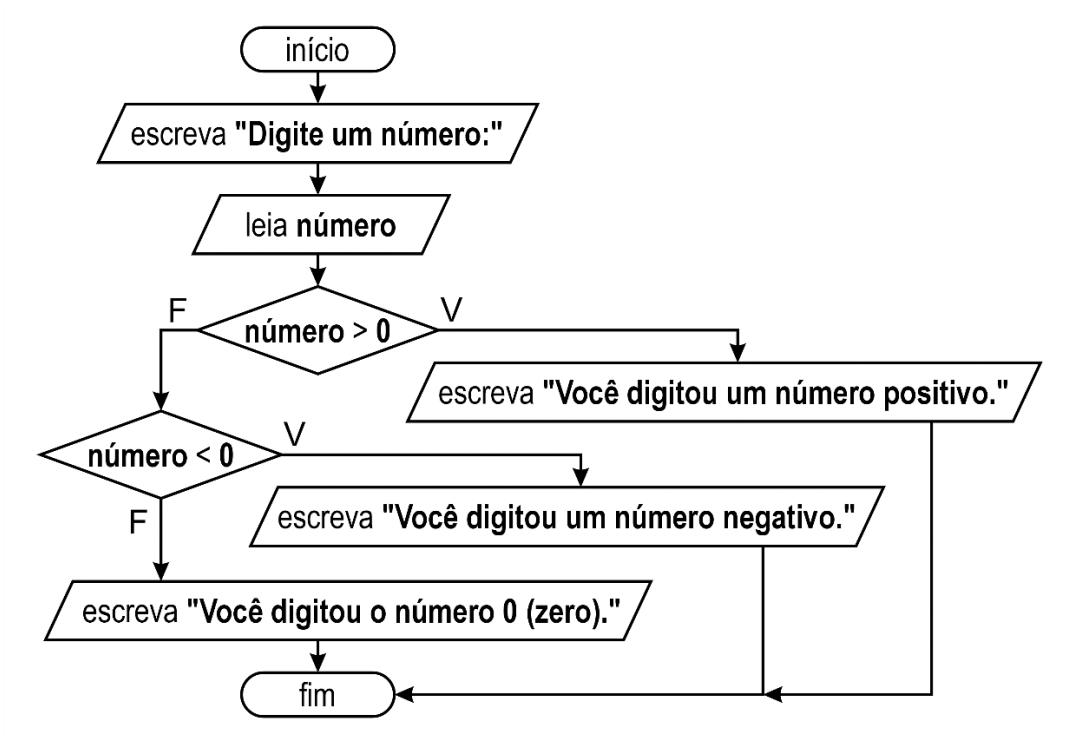
A estratégia da programação por blocos é, sem dúvida, uma prática voltada a organização e hierarquia, o que torna a leitura de um código muito mais clara. Porém, tal prática pode se revelar não tão intuitiva para alguém que não seja programador profissional e que venha se utilizar de recursos com os quais seja necessário expressar tarefas em uma organização lógica, que demande algumas derivações e/ou repetições. É o caso, por exemplo, de um professor de Matemática concebendo um OA.

Representações de conceitos em fluxo são comumente utilizados, em diferentes áreas de conhecimento, para organização de conteúdo e resolução de problemas. Por exemplo: o ciclo da água no ensino de Ciências e a classificação de quadriláteros na Educação Matemática fazem uso de setas para indicar o fluxo. Organogramas e mapas conceituais também fazem uso de figuras geométricas interligadas por linhas ou setas.

Sob tal perspectiva, espera-se que esta pesquisa possa auxiliar na compreensão se a disposição dos comandos em fluxo estaria mais próxima de uma programação intrinsecamente visual e, como consequência, se poderia promover uma redução no tempo de aprendizagem para utilização da plataforma.

A construção de um fluxograma apoia-se no uso de elementos gráficos que representam o desempenho de funcionalidades específicas. A Figura 16 apresenta um exemplo de fluxograma (análogo ao programa que foi apresentado na Figura 15) contendo alguns desses elementos para que, na sequência, seja apresentado como eles são representados na plataforma.

Figura 16 – Exemplo de fluxograma

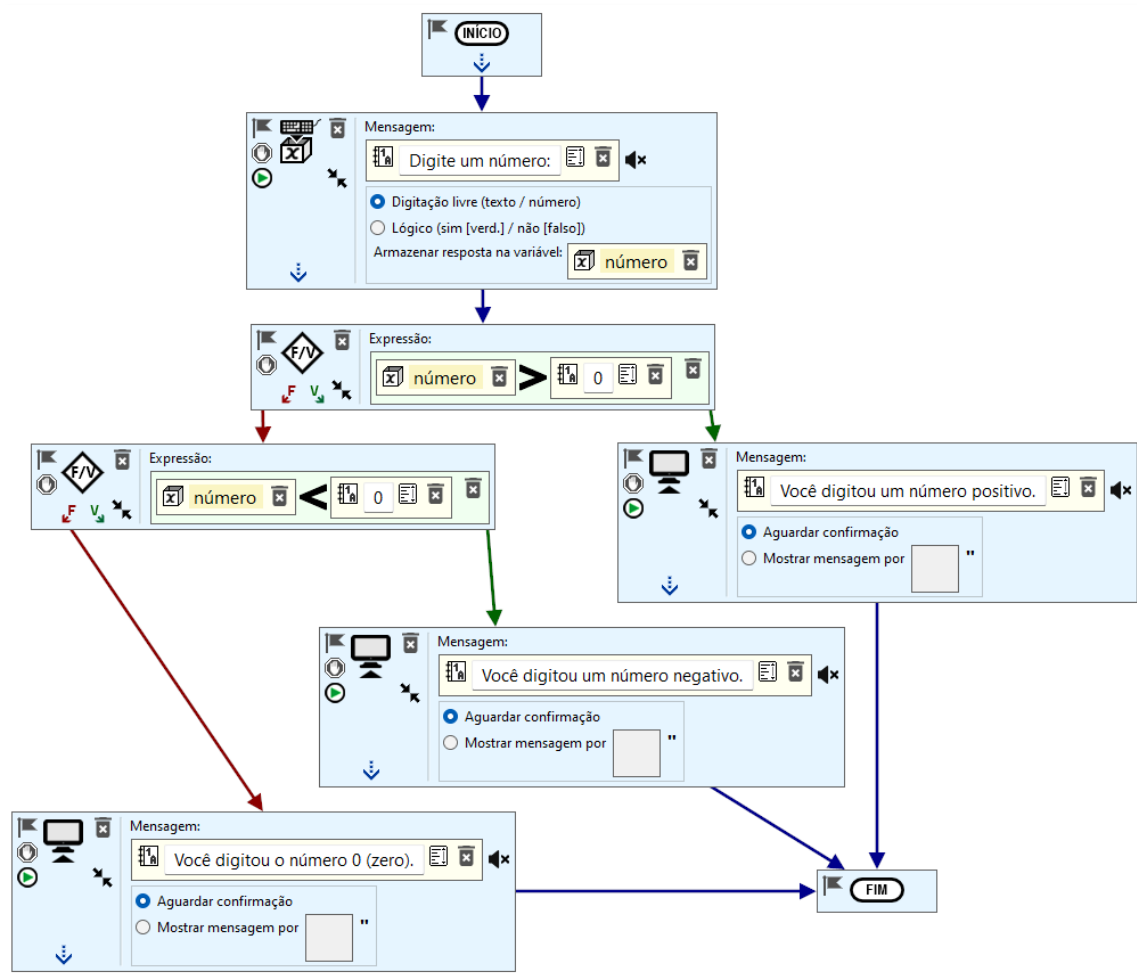


Fonte: Autoria própria (2023)

Os elementos gráficos utilizados para a criação de um fluxograma na GenIA são análogos aos elementos de um fluxograma comum, o que revela mais uma preocupação com o pilar da intuição condizente à similaridade. Porém, tais elementos não são idênticos. Isto porque foi necessária a inclusão de componentes visuais destinados à usabilidade, que incluem recursos como, por exemplo, a vinculação dos comandos ao fluxo, opções de disposição e visualização ou até mesmo a exclusão do comando.

A Figura 17 apresenta um fluxograma, criado na GenIA, também análogo ao fluxograma que foi apresentado na Figura 15.

Figura 17 – Fluxograma na GenIA



Fonte: Autoria própria (2023)

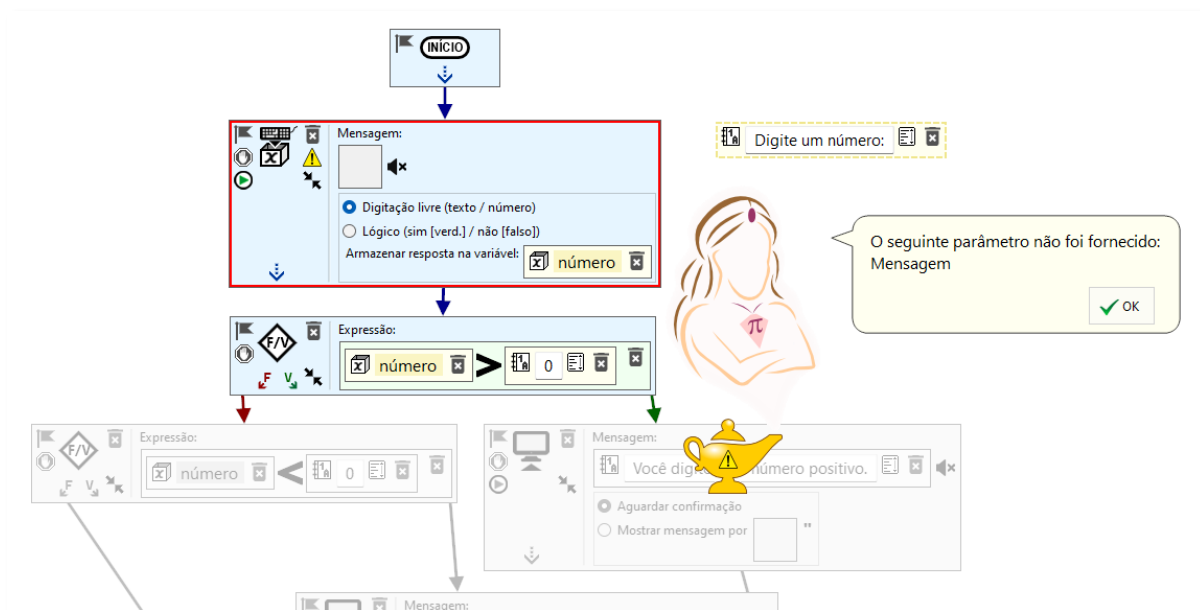
A montagem do fluxograma na GenIA é feita arrastando-se o ícone do componente (comando, operador ou operando) da região dos ícones para a região do código. No momento em que o ícone é solto na região de código, cria-se o componente correspondente, pronto para ser parametrizado.

Enquanto o componente não receber parametrização adequada, ele fica com sua borda destacada em vermelho e um ícone de alerta (⚠) permanece ativo que, quando acionado, emite uma mensagem com orientações sobre os parâmetros faltantes.

Se o componente, depois de arrastado, for solto sobre uma seta do fluxo, ele será automaticamente vinculado ao fluxo naquela posição. Quando o componente é arrastado para uma posição qualquer da área livre, sem vínculo com o fluxo, ele fica com a borda tracejada, indicando sua desconexão com o fluxograma.

A Figura 18 destaca ambas as situações: o operando contendo o texto “Digite um número:” foi removido do comando que emitia a mensagem. Com esta alteração, o comando de mensagem ficou sem parâmetro (borda vermelha) e o operando ficou solto na área livre (borda tracejada). Ao acionar o ícone de alerta do comando de mensagem, foi emitida a mensagem orientando a parametrização.

Figura 18 – Evidências do pilar da visualização na GenIA



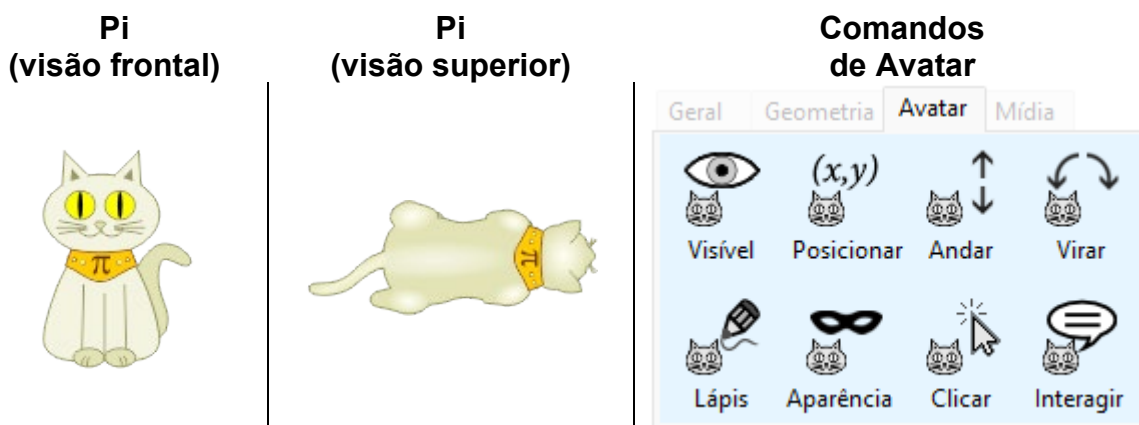
Fonte: Autoria própria (2023)

Esse comportamento da plataforma, que realça condições de uso ou parametrização dos comandos, é mais uma evidência da preocupação com o pilar intuitivo da visualização.

Como já foi mencionado, com o intuito de aprimorar a plataforma voltando-se para o pilar da similaridade, houve um direcionamento no sentido de implementar comandos específicos para criação de conteúdo matemático. Para tal feito, buscou-se inspiração no GeoGebra. Tais comandos foram apresentados na Figura 13.

Ainda no sentido de resgatar recursos já conhecidos pela comunidade de professores que fazem uso de TD relacionadas à programação na Educação, buscou-se referência no Logo. Porém, ao invés do personagem-robô ser uma tartaruga (conhecida como Tat), que recebe comandos de movimentação para andar e girar, combinados com o uso de lápis, a GenIA traz como mascote um gato, chamado de Pi. A Figura 19 apresenta o Pi e os comandos específicos para seu aproveitamento.

Figura 19 – Apresentação do Pi e seus comandos



Fonte: Autoria própria (2023)

Atendendo ao pilar da visualização, a GenIA ainda possui dois recursos que são diferenciais em relação a outras plataformas: por meio de comandos específicos, é possível executar o fluxograma em modo de depuração³⁵, facilitando bastante a visualização do seguimento do fluxo e do resultado de cada comando. Ainda sob este viés, o outro recurso é a existência de um botão em cada comando do fluxograma que permite que ele seja executado individualmente, independentemente de sua posição no fluxo ou do fluxograma estar em execução.

Ao mesmo tempo em que a programação intuitiva está fundamentada nos três pilares apresentados, ela também é um dos três pilares tecnológicos da GenIA. Até este ponto da leitura foram explorados dois deles, que, de forma conjunta, conferem material considerável para o desenvolvimento da plataforma. Porém, com base nas palavras de Doerr (2014), o trabalho de um cientista é determinado por duas coisas: seus interesses e os interesses de seu tempo.

Não há dúvidas de que a IA seja assunto de grande interesse para o pesquisador deste trabalho. E, em sintonia com tal afirmação, entende-se que a IA talvez seja o assunto global de maior relevância ao passo em que esta pesquisa apresenta seus primeiros frutos. É momento de inserir a IA no contexto desta pesquisa.

³⁵ Na programação de computadores, depurar um programa consiste em executá-lo passo a passo, com vistas a identificar e corrigir erros.

6 INTELIGÊNCIA ARTIFICIAL

Com o passar dos anos, aumenta a gama de TD que fazem uso de IA ou são, por ela, suportadas. Tikhomirov (1981) considerava, ainda no início dos anos 1970, que era “impossível discutir o problema da influência dos computadores no desenvolvimento dos processos mentais humanos sem levar em conta as pesquisas em inteligência artificial.” (TIKHOMIROV, 1981, p. 260, tradução nossa).

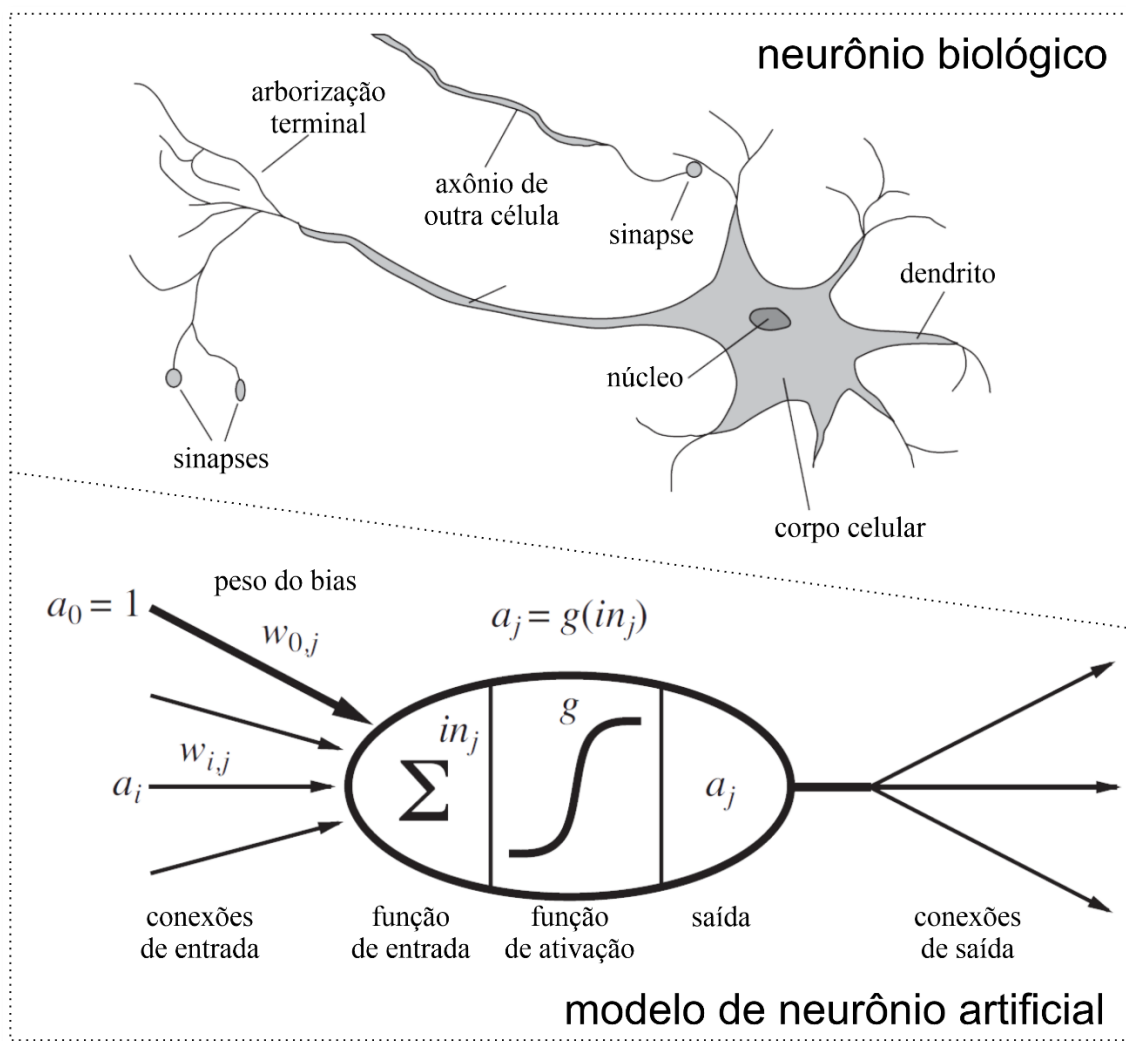
A IA é um campo de estudo multidisciplinar, que tem sido foco contínuo de pesquisadores de diversas áreas. Não há uma definição consensual para o termo, ainda que diversos autores concordem que se trata de uma forma de fazer o computador realizar tarefas que normalmente seriam desempenhadas por um ser humano.

A origem do termo “inteligência artificial” data de meados dos anos 1950, sendo atribuída ao cientista da computação e cientista cognitivo John McCarthy, quando cunhou o termo ao referir-se a uma proposta de estudo para uma conferência:

Propomos que um estudo de inteligência artificial de 2 meses e 10 homens seja realizado durante o verão de 1956 no Dartmouth College em Hanover, New Hampshire. O estudo deve prosseguir com base na conjectura de que cada aspecto do aprendizado ou qualquer outra característica da inteligência pode, em princípio, ser descrito com tanta precisão que uma máquina pode ser feita para simulá-lo. Será feita uma tentativa de descobrir como fazer as máquinas usarem a linguagem, formar abstrações e conceitos, resolver tipos de problemas agora reservados aos humanos e se aprimorarem. (MCCARTHY *et al.*, 1955, p. 1, tradução nossa).

Os experimentos envolvendo IA começaram ainda antes da origem do termo. Durante a Segunda Guerra Mundial, McCulloch e Pitts (1943) apresentaram a ideia de simular o funcionamento dos neurônios por meio de modelos matemáticos. O modelo sugeria o uso de resistores e amplificadores para representar as sinapses (conexões entre neurônios biológicos). A Figura 20 ilustra uma analogia entre um neurônio biológico e um modelo de neurônio artificial.

Figura 20 – Analogia entre neurônio biológico e modelo de neurônio artificial



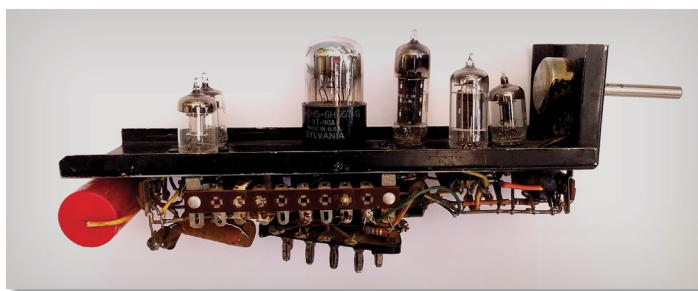
Fonte: RUSSELL; NORVIG (2021, pp. 30, 683-690, adaptado)

Observa-se que o neurônio artificial traz em seu “núcleo” a função de ativação. Ela é a principal responsável pelo comportamento do neurônio e irá definir os valores para as conexões de saída, que servirão de entrada para o próximo neurônio a ele conectado.

6.1 IA Conexionista

A partir do modelo de neurônio artificial proposto por McCulloch e Pitts (1943), em 1950 Marvin Minsky e Dean Edmonds construíram a primeira rede neural artificial (RNA), chamada SNARC (*Stochastic Neural Analog Reinforcement Calculator*), composta por 40 neurônios artificiais interconectados (Figura 21).

Figura 21 – Um dos neurônios da SNARC

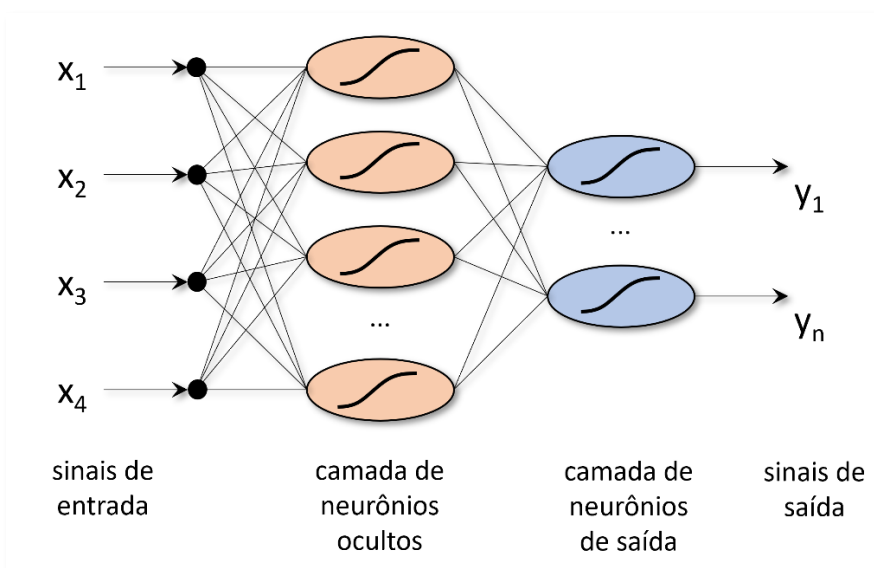


Fonte: AKST (2019, n.p.)

Na mesma década, Rosenblatt (1958) também propôs a criação de um dispositivo eletrônico análogo a um neurônio biológico, ao qual nomeou “Perceptron”. No final da década de 1950, Minsky e Papert (1988) construíram e realizaram experimentos com uma rede de dispositivos que batizaram de perceptron em reconhecimento ao trabalho pioneiro de Rosenblatt.

O ramo da IA que trabalha com RNA é, atualmente, denominado conexionista e os neurônios artificiais são chamados simplesmente de células de uma RNA. As células são distribuídas em camadas (Figura 22) e podem ser classificadas de acordo com sua função de ativação.

Figura 22 – Exemplo de RNA com duas camadas interconectadas



Fonte: KUBAT (2017, p. 92, tradução nossa, adaptado)

Valendo-se do uso de RNA, Samuel (1959) foi um dos primeiros a obter resultados de programas de autoaprendizagem, ao propor um jogo de damas, sendo considerado um dos pioneiros em experimentos envolvendo aprendizado de máquina.

Segundo Haykin (2001), uma RNA se assemelha ao cérebro humano basicamente em dois aspectos: 1) o conhecimento é adquirido pela rede por intermédio do processo de aprendizagem; 2) são utilizadas forças de conexão entre neurônios (pesos sinápticos) para armazenar o conhecimento adquirido.

O aprendizado de máquina em uma RNA funciona essencialmente da seguinte maneira: são fornecidos valores (sinais) de entrada para cada um dos neurônios da primeira camada (camada de entrada); tais valores são multiplicados por pesos previamente estabelecidos (peso do bias); obedecendo às funções de entrada e de ativação, cada neurônio da camada de entrada irá gerar uma saída, que servirá de sinal de entrada para a próxima camada de neurônios (camada oculta), e assim sucessivamente, até chegar na última camada da rede (camada de saída).

A cada execução, a saída da rede é analisada e, caso não gere o resultado esperado, são ajustados os pesos do bias dos neurônios e todo o processo é repetido. Cada iteração no processo é conhecida como “geração”. O aprendizado acontece na camada oculta da RNA e, a quanto mais gerações a rede é submetida, isto é, quanto maior o volume de dados (experiências), mais refinado é o resultado gerado.

Porém, Pinhanez (2022) pontua que existe um limite para que o número de gerações seja proveitoso no treinamento de uma RNA. Alimentar a rede com um número elevado de experiências pode levar a um efeito conhecido como *overfitting* (sobreajuste), sob o qual a rede, a partir de determinado ponto, começa a apresentar resultados menos satisfatórios ou inconclusivos.

De acordo com Russell e Norvig (2021), os algoritmos são treinados sob três modos distintos de aprendizado: modo supervisionado, modo não supervisionado, e aprendizagem por reforço.

No aprendizado supervisionado, o algoritmo observa pares de entrada-saída e aprende uma função que mapeia o aprendizado supervisionado da entrada para a saída. Por exemplo:

As entradas podem ser imagens de câmeras, cada uma acompanhada por uma saída dizendo “ônibus” ou “pedestre” etc. Uma saída como essa é chamada de etiqueta. O agente aprende uma função que, ao receber uma nova imagem, prediz o rótulo apropriado. (RUSSELL; NORVIG, 2021, p. 671, tradução nossa).

No aprendizado não supervisionado, o algoritmo aprende padrões na entrada sem qualquer *feedback* explícito. “A tarefa de aprendizado não supervisionado mais comum é o *clustering*: detectar conjuntos potencialmente úteis de exemplos de entrada.” (RUSSELL; NORVIG, 2021, p. 671, tradução nossa). Por exemplo, a partir de um número expressivo de imagens extraídas da internet, o algoritmo poderia identificar automaticamente um conjunto delas que fosse de um mesmo objeto.

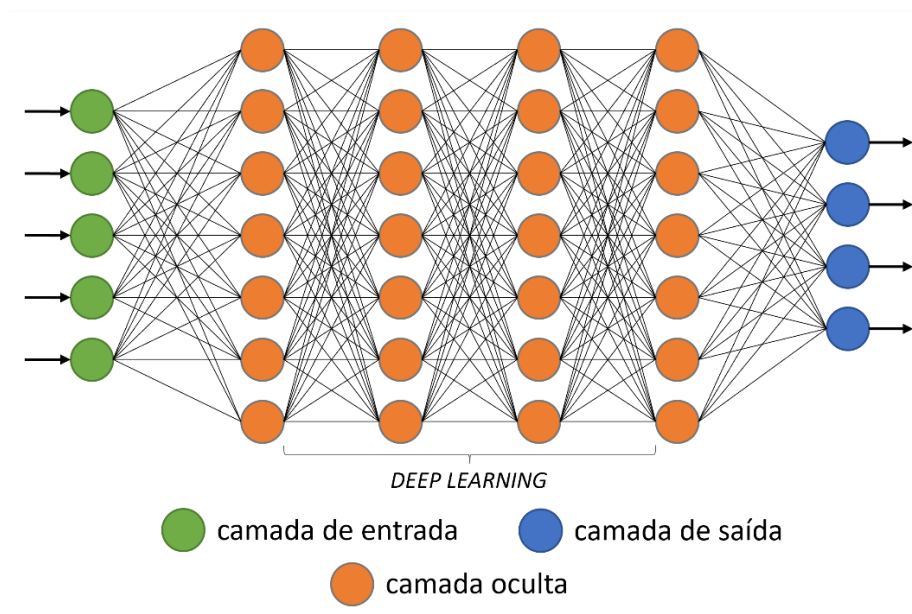
Por fim, no aprendizado por reforço, o algoritmo aprende por meio de recompensas (reforço positivo) e punições (reforço negativo). Por exemplo:

Ao final de uma partida de xadrez, o agente é informado de que ganhou (uma recompensa) ou perdeu (uma punição). Cabe ao agente decidir quais das ações anteriores ao reforço foram as mais responsáveis por ele, e alterar suas ações para almejar mais recompensas no futuro. (RUSSELL; NORVIG, 2021, p. 671, tradução nossa).

A GenIA trabalha com algoritmos no modo supervisionado e detalhes sobre a estratégia adotada encontram-se na seção 6.5.

Quando a RNA é organizada contendo mais de uma camada oculta, ocorre o que se chama de *deep learning* (aprendizagem profunda). Neste caso, o poder de processamento é elevado de forma exponencial, mas os resultados também tendem a ser mais refinados. A Figura 23 ilustra uma RNA de aprendizagem profunda.

Figura 23 – Modelo de RNA de aprendizagem profunda



Fonte: Autoria própria (2022)

Justamente por conta dessa demanda de alto poder computacional para treinamento das RNA é que as pesquisas em IA entraram em uma onda recessiva em meados dos anos 1960. Naquela época, os computadores ainda eram equipamentos rudimentares e não conseguiram atender às necessidades, o que levou a uma perda de interesse e investimentos na área. “Por aproximadamente uma década (entre 1970 e 1980), com pouca ou quase nenhuma contribuição, a IA passou por um período que ficou conhecido como *AI Winter* (inverno da IA)” (ZATTI; KALINKE, 2021, p. 79).

Para Minsky e Papert (1988), é possível que os resultados de suas pesquisas tenham contribuído para a desmotivação em pesquisas sobre o paradigma conexionista. Atualmente este cenário é bem diferente, pois além dos computadores terem alto poder de processamento, os algoritmos são executados paralelamente em diversos equipamentos interconectados em nuvem.

6.2 IA Simbólica

No começo da década de 1980, Buchanan, Sutherland e Feigenbaum (1969) apresentaram uma nova vertente para a IA: os sistemas especialistas. Os estudos iniciaram com a criação de um sistema para resolver o problema de inferir a estrutura molecular da informação fornecida por um espectrômetro de massa. O sistema, denominado Dendral, serviu como ponto de partida para que surgissem propostas de sistemas aplicados a outras áreas do conhecimento (RUSSELL; NORVIG, 2021).

Esse novo direcionamento nas pesquisas revela outro paradigma da IA, denominado simbólico. “Um sistema simbólico de IA funciona realizando uma série de etapas de raciocínio lógico sobre representações semelhantes à linguagem.” (GARNELO, M.; SHANAHAN, M., 2019, [s.p.], tradução nossa). Os trabalhos consideram a criação de sistemas voltados a resolver problemas específicos, pois a IA conexionista buscava solucionar problemas genéricos por meio de autoaprendizado.

Dentre as diversas aplicações que se abrigam sob as asas da IA simbólica, uma começou a despertar atenção especial no final dos anos 1990: a mineração de dados. Naquela época, as empresas começaram a acumular dados históricos de suas operações nos chamados armazém de dados (do inglês: *data warehouse*), definidos como “um conjunto de dados orientado por assunto, integrado, variável com o tempo

e não-volátil, que fornece suporte ao processo de tomada de decisão do negócio.” (INMON, 2002, p. 31).

De acordo com Fayyad et al. (1996), a mineração de dados é uma das etapas do KDD (do inglês: *Knowledge Discovery in Databases* – descoberta de conhecimento em bases de dados), cujo objetivo é identificar padrões potencialmente úteis nos dados, o que não é possível por meio de algoritmos triviais.

Se, em 1996, a mineração de dados já era considerada não-trivial, transpor o processo para 2023 (quase trinta anos depois) não deixa dúvidas a respeito do poder computacional necessário.

Apesar do inverno da IA, os estudos com a IA conexionista nunca pararam. Do *big data*³⁶ à ciência de dados, são tantos os recursos reutilizados com e pela IA que atualmente torna-se difícil classificar um sistema sob um paradigma ou outro. Pelo contrário, ambos os paradigmas são comumente utilizados em conjunto na solução de problemas diversos. É o caso, por exemplo, das redes de aprendizagem profunda sendo utilizadas para identificar padrões em textos, imagens, vídeos, todos espalhados pela grande rede (internet), em uma constante de mineração.

6.3 Outras áreas de estudo envolvendo IA

Outro exemplo de uso concomitante dos paradigmas são os algoritmos de busca que, com o advento da Internet, passaram a receber atenção especial, sendo constantemente aprimorados, envolvendo outras técnicas de IA, como o reconhecimento de padrões e o processamento de linguagem natural (PNL).

Há uma série de ramificações de estudos que partem de uma prerrogativa comum: imitar o comportamento humano na solução de problemas. O Quadro 15 apresenta algumas dessas ramificações e uma breve descrição de como elas podem contribuir para a solução de tais problemas.

³⁶ Termo cunhado em 1997, por Michael Cox e David Ellsworth, engenheiros da NASA, para “fazerem referência ao problema computacional de se manipular uma quantidade muito grande de dados, o que requer técnicas específicas de processamento.” (ZATTI; KALINKE, 2021, p. 82)

Quadro 15 – Algumas áreas de estudo envolvendo IA

Área de estudo	Descrição
Chatbots	Também chamado de <i>chatterbot</i> (<i>chatter</i> = pessoa que conversa + <i>bot</i> = robô) é um programa de computador que tenta simular um ser humano na conversação com as pessoas.
Jogos digitais	Os jogos digitais são programas que tem por objetivo entreter. Técnicas de IA aplicada a jogos incluem o <i>pathfinding</i> (encontrar caminhos), máquinas de estado finitas e árvores de comportamento.
Lógica difusa (fuzzy logic)	É uma forma de lógica na qual os valores verdade não ficam limitados simplesmente a verdadeiro e falso. As variáveis podem assumir qualquer valor real entre 0 e 1.
Reconhecimento de padrões	Tem por objetivo classificar objetos em classes (ou categorias). Incluem o reconhecimento de fala, reconhecimento (processamento) de imagens, classificação de documentos etc.
Robótica	É “o estudo das máquinas que podem substituir o ser humano na execução de uma tarefa, tanto no que diz respeito à atividade física quanto à tomada de decisões.” (SICILIANO <i>et al.</i> , 2010, p. 1, tradução nossa)
Sistemas multiagentes	Alvares e Sichman (1997) consideram que um agente é uma entidade computacional com um comportamento autônomo que lhe permite decidir suas próprias ações. Os sistemas multiagentes (SMA) concentram-se no estudo de agentes autônomos em um universo multiagente. Nestes sistemas, o termo autônomo designa o fato de que os agentes têm uma existência própria, independentemente da existência de outros agentes.
Sistemas tutoriais inteligentes	“Os STI são programas de software que dão suporte às atividades da aprendizagem.” (GAMBOA; FRED, 2001, p. 452)
Visão computacional	Segundo Roberts (1965), a visão computacional é uma área que busca construir sistemas autônomos que possam realizar algumas das tarefas do sistema visual humano ou até mesmo superá-lo em alguns aspectos.

Fonte: Autoria própria (2023)

Seguindo na linha da imitação comportamental, considerando as ramificações apresentadas sob uma perspectiva funcional, percebe-se que algumas estão mais próximas da execução, enquanto outras aparentam estar em outro patamar, mais complexo, em uma esfera decisória.

Filósofos como Searle (1980) usam o termo “IA fraca” ao considerarem que a IA é e sempre será uma simulação das funções cognitivas humanas, e que os computadores parecem apenas pensar, mas não são realmente conscientes, em nenhum sentido.

Por outro lado, a “IA forte” pretende emular as funções do cérebro humano e, por consequência, as ações de um ser humano, incluindo seu poder de compreensão e até sua consciência.

Para Russell (2021), é improvável que num futuro próximo as máquinas venham a ter acesso a um modelo completo da cognição humana, seja genérica, seja sob medida para indivíduos específicos. Em vez disso, faz sentido, de um ponto de vista prático, examinar como os humanos costumam se afastar da racionalidade e tentar descobrir preferências a partir de comportamentos que apresentem esses desvios.

Esta pesquisa não tem por objetivo adentrar as nuances e discussões éticas a respeito dos sistemas de IA que pretendem imitar o comportamento humano. Oportuno lembrar que uma das pesquisas que integram o macroprojeto, apresentada na seção 3.1, aborda algumas dessas questões. Ainda assim, uma revisita à questão norteadora remete a uma área que visa o desenvolvimento do indivíduo à medida que abraça suas potencialidades: a Educação.

6.4 IA na Educação

Zatti e Kalinke (2021) apontam que existem iniciativas internacionais voltadas especificamente à pesquisa e desenvolvimento da AIED (do inglês: *Artificial Intelligence in Education* – Inteligência Artificial na Educação). A IAIED³⁷, por exemplo, é uma comunidade interdisciplinar que promove pesquisa e desenvolvimento voltados a ambientes de aprendizagem interativos e adaptativos, envolvendo pesquisadores de diferentes países e diferentes áreas de conhecimento.

Nesse sentido, Holmes, Bialik e Fadel (2019), buscam investigar de que forma a IA pode transformar a Educação. Para isso, adotam um *framework* que se baseia em quatro dimensões que devem ser consideradas em relação às tecnologias educacionais: substituição, aumento, modificação e redefinição. O nome do *framework*, SAMR, é um acrônimo para as iniciais das dimensões.

O Quadro 16 apresenta como a IA poderia contribuir para a transformação educacional sob cada uma das dimensões.

³⁷ *International Artificial Intelligence in Education Society* (www.iaied.org).

Quadro 16 – AIED e o *framework* SAMR

	Tecnologia Educacional	AIED
Substituição	Tecnologia atua como uma substituta direta de ferramenta, sem mudança funcional	Não se aplica (até o presente momento)
Aumento	Tecnologia atua como uma substituta direta de ferramenta, com melhoria funcional.	<ul style="list-style-type: none"> • Sistemas tutores inteligentes • Sistemas tutores baseados em diálogo • Ambientes de aprendizagem exploratória • Avaliação de escrita automática • Aprendizado de línguas • <i>Chatbots</i> • Suporte a aprendizagem colaborativa • Monitoria de fórum de estudante
Modificação	Tecnologia permite reformulação de tarefas de forma significativa.	<ul style="list-style-type: none"> • Experiências de aprendizagem com realidade virtual e realidade aumentada • Colegas de aprendizagem com IA • Assistentes de ensino com IA • IA como ferramenta de pesquisa em ciências de aprendizagem
Redefinição	Tecnologia permite a criação de novas tarefas, anteriormente inconcebíveis.	IA tirando a necessidade de exames pontuais, proporcionando testes contínuos altamente adaptativos

Fonte: HOLMES; BIALINK; FADEL (2019, p. 182, tradução nossa, adaptado)

De acordo com o *framework* SAMR, entende-se que a versão inicial da GenIA estaria inserida na dimensão de Modificação, pois sua IA pode proporcionar uma modificação na maneira como o professor prepara o conteúdo (OA). Porém, quando se considera o que já está sendo produzido a partir das pesquisas do macroprojeto, a plataforma poderá, futuramente, fazer parte da dimensão de Aumento, estendendo suas possibilidades para uso com *chatbots* e sistemas tutores inteligentes.

Em contraponto às perspectivas da AIED apontadas por Holmes, Bialik e Fadel (2019) como agente transformadora, Ouyang e Jiao (2021) consideram que:

Embora trabalhos relevantes tenham apresentado categorizações, desafios e visões de futuro da AIED, poucos estudos examinam explicitamente quais são os diferentes papéis da IA na Educação, como a IA está conectada às teorias educacionais e de aprendizagem existentes e em que medida o uso de tecnologias de IA influenciam a aprendizagem e a instrução. (OUYANG; JIAO, 2021, p. 1, tradução nossa)

Em seu artigo, a partir de uma lista de trabalhos publicados relacionados a AIED, os autores propõem três paradigmas sob os quais a AIED poderia ser observada. Após breve descrição, o Quadro 17 sumariza os paradigmas com algumas características e exemplos de utilização.

- 1) Direcionado por IA (aprendiz-como-destinatário): a IA representa o conhecimento do domínio e direciona os processos de aprendizagem, enquanto o aprendiz atua como destinatário do serviço de IA para seguir os caminhos de aprendizagem específicos.
- 2) Suportado por IA (aprendiz-como-colaborador): o sistema de IA renuncia a seu poder de controle para servir como uma ferramenta de apoio, enquanto o aprendiz trabalha como colaborador do sistema para se concentrar no processo de aprendizagem do aprendiz individual.
- 3) Empoderado pela IA (aprendiz-como-líder): mantém a ação do aprendiz como o núcleo da AIEd, e vê a IA como uma ferramenta para aumentar a inteligência humana.

Quadro 17 – Três paradigmas da AIEd

Paradigma	Fundamentação teórica	Implementações	Técnicas de IA	Exemplos
1: Direcionado por IA, aprendiz-como-destinatário	Behaviorismo	Trabalhos anteriores de Sistemas Tutores Inteligentes (STIs)	IA baseada em técnicas relacionais estatísticas	Tutor de Programação ACT ³⁸ Stat Lady ³⁹
2: Suportado por IA, aprendiz-como-colaborador	Construtivismo cognitivo e social	Sistemas de Tutoria Baseados em Diálogo (SETs); Ambientes Exploratórios de Aprendizagem (ELEs)	Rede bayesiana, processamento de linguagem natural, árvores de decisão de Markov	Um ambiente exploratório QUE ⁴⁰
3: Empoderado pela IA, aprendiz-como-líder	Conectivismo, Sistema adaptativo complexo	A cooperação humano-computador; Aprendizagem personalizada/adaptativa	Interface cérebro-computador, aprendizado de máquina, aprendizado profundo	Uma modelagem preditiva MOOC em tempo real ⁴¹

Fonte: OUYANG; JIAO (2021, p. 3, tradução nossa, adaptado)

A partir de seu estudo, os autores consideram que a AIEd não se limita à implementação de tecnologia de IA, mas sim, “[...] é uma integração das dimensões pedagógica, social, cultural e econômica durante os processos de aplicação da tecnologia.” (OUYANG; JIAO, 2021, p. 5, tradução nossa).

Considerando que a GenIA contém algoritmos de aprendizado de máquina, seria natural inferir que ela estivesse sob o paradigma 3. Porém, entende-se que, nesta versão inicial, ela esteja mais concordante com o paradigma 2 (suportado por

³⁸ [https://doi.org/10.1016/0004-3702\(90\)90093-F](https://doi.org/10.1016/0004-3702(90)90093-F)

³⁹ <https://www.sciencedirect.com/science/article/abs/pii/000437029090093F>

⁴⁰ https://link.springer.com/chapter/10.1007/3-540-68716-5_19

⁴¹ https://link.springer.com/chapter/10.1007/978-3-319-93843-1_18

IA), no qual o aprendiz trabalha como colaborador do sistema, podendo, futuramente, atingir o próximo patamar quando incorporada, por exemplo, a MOOCs inteligentes.

Em âmbito nacional, Vicari (2018) realizou uma pesquisa contendo resultados de um mapeamento sistemático sobre documentos publicados de 2014 a 2017, com objetivo de identificar as tendências mundiais em tecnologias baseadas em IA na Educação, para o período de 2017 até 2030.

A IA aplicada à Educação é uma área de pesquisa multi e interdisciplinar, pois contempla o uso de tecnologias da IA em sistemas cujo objetivo é o ensino e a aprendizagem. Dessa forma, sistemas educacionais são um campo de aplicação e testes para as tecnologias da IA. (VICARI, 2018, p. 12).

De acordo com Vicari (2018), a revisão sistemática realizada considerou uma base internacional de teses e dissertações, quatro bases de patentes, duas bases internacionais de artigos científicos, anais e apresentações de pesquisadores em congressos internacionais de IA.

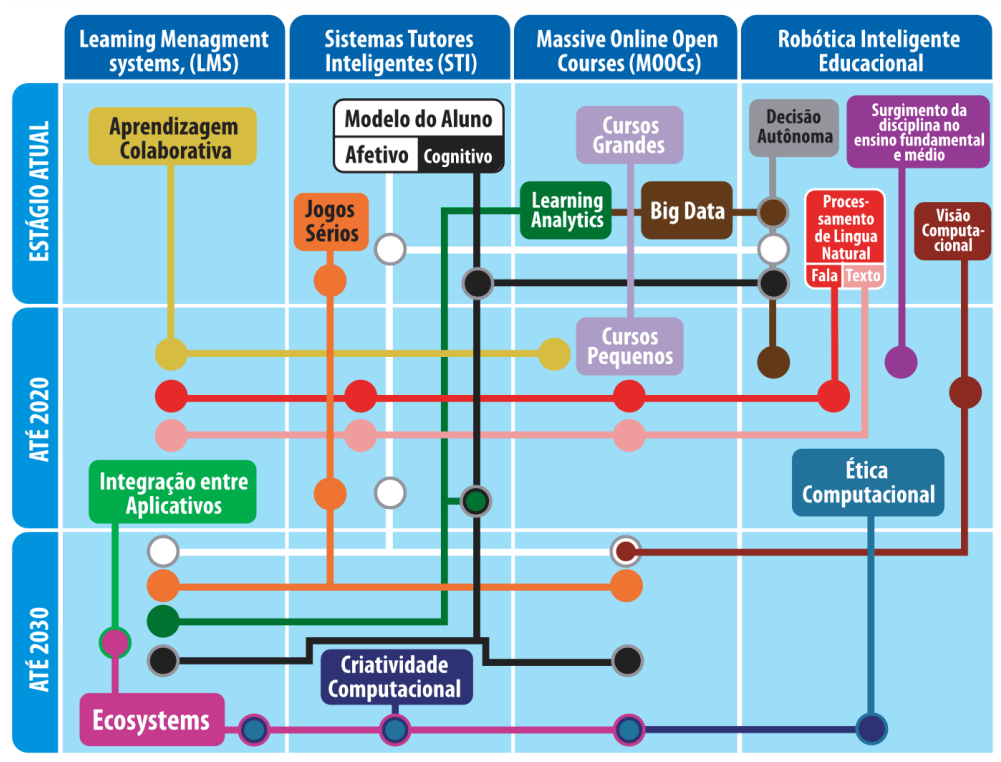
Durante o mapeamento, foram utilizados os seguintes termos para filtrar sua busca: *education* (educação) *AND artificial intelligence* (IA); *intelligent tutoring systems* (ITS – sistemas inteligentes de tutoria); *affective intelligent tutor systems* (sistemas inteligentes de tutoria afetivos); *education* (educação) *AND natural language processing* (processamento de linguagem natural); *collaborative learning* (aprendizagem colaborativa); *game-based* (baseado em jogos) *AND learning* (aprendizagem); *serious games* (jogos sérios); *massive open online courses* (MOOC – cursos online abertos massivos); *robotics intelligent* (robótica inteligente) *AND education* (educação).

A partir do mapeamento, Vicari (2018) utilizou a técnica de *roadmapping* tecnológico para inferir uma prospecção de uso da IA na Educação até o ano de 2030.

O *roadmapping* tecnológico é uma técnica específica para planejamento de tecnologia, que se encaixa em um conjunto mais geral de atividades de planejamento. [...] Ela identifica (para um conjunto de necessidades de produto) os requisitos críticos do sistema, as metas de desempenho do produto e do processo e as alternativas de tecnologia e marcos para atingir essas metas. (GARCIA; BRAY, 1997, pp. 9-12, tradução nossa).

A Figura 24 apresenta o *roadmap* proposto.

Figura 24 – Prospecção das tendências em IA na Educação até 2030



Fonte: VICARI (2018, p. 22)

Com base no *roadmap*, o Quadro 18 apresenta as tendências tecnológicas apontadas por Vicari (2018) e as possíveis contribuições para a Educação.

Quadro 18 – Tendências tecnológicas da IA na Educação

processamento de linguagem natural (PLN)	tradução simultânea de voz, em tempo real, para textos
afetividade/emoções	tanto os STI quanto os LMS e MOOCs poderiam detectar emoções expressas pela face do estudante
robótica educacional	poderá estar mais presente nos currículos pela diminuição dos custos
ensino personalizado	favorecido pelos STI e <i>smartbooks</i> , por meio do modelo afetivo/cognitivo dos estudantes
learning analytics	interpretação dos dados produzidos pelos estudantes, a fim de avaliar o progresso acadêmico
MOOCs	menores e modularizados
redes sociais	presentes nas plataformas LMS
visão computacional	tanto para reconhecimento de emoções quanto certificação de identidade dos estudantes
criatividade computacional	produção artística vinculada à programação
ética computacional	-
ecossistemas educacionais	integração de vários aplicativos ou componentes nos quais a utilização de tecnologias complementa a educação formal ou informal

Fonte: VICARI (2018, adaptado)

Diante de um número cada vez mais expressivo de possibilidades, dialogar a respeito das potencialidades das TD no âmbito da Educação pode ser cativante. Porém, é uma rota que, além das facilidades, também integra muitos desafios. Castañeda e Selwyn (2018, p. 1) pontuam que “não é mais estranho se deparar com questões desafiadoras das conotações sociais, culturais, políticas e econômicas do uso da tecnologia digital no ensino superior”. Dentre tais questões, para os autores, as que demandam atenção especial incluem:

- A tecnologia educacional ainda não lidou adequadamente com as questões de aprendizagem; sistemas de gestão de aprendizagem tendem a apoiar a gestão ao invés da aprendizagem;
- A pedagogia representa uma lacuna nas discussões em torno das TD; aspectos intelectuais, estruturais e instrumentais ainda precisam ser mais explorados e balanceados;
- É importante que as discussões sobre a tecnologia educativa ultrapassem os aspectos racionais (informacionais, cognitivos ou neurológicos) do processo educativo e considerem também a educação como um processo emocional e humano;
- Deve-se tomar cuidado com a (hiper)individualização da educação; as construções individualistas da educação digital estão fazendo pouco para refletir questões de justiça social, desigualdade ou a noção de educação como um bem público coletivo;
- A digitalização em larga escala do ensino e da aprendizagem suscitam aspectos comerciais acentuados; sistemas de gerenciamento de conteúdo, sistemas de gerenciamento de carga de trabalho, métricas de desempenho são exemplos de elementos que estariam levando à corporativização do ensino superior.

A partir dos estudos de Vicari, Holmes, Bialik e Fadel (2018, 2019), é possível visualizar algumas contribuições da IA especificamente para a Educação Matemática. Zatti e Kalinke (2021) sugerem que tais contribuições podem levar em conta algumas tecnologias, dentre elas a internet e os OA.

A internet “já é natural, ela já modificou o humano, os coletivos seres-humanos-com-internet protagonizam cenários educacionais e moldam os modos de

pensamento e produção do conhecimento” (BORBA; MALHEIROS; ZULATTO, 2007, p. 129).

A relação de benefícios que ela pode trazer aos processos pedagógicos contempla uma gama extensa de tópicos, que vai da facilidade para a pesquisa, passando pela participação em cursos virtuais, visita a sites interativos, comunicação dinâmica, publicação de materiais e prática da leitura em línguas estrangeiras. (KALINKE, 2014, p.41).

Zatti e Kalinke (2021) sugerem que alguns desses benefícios podem ser suportados por IA: a facilidade para a pesquisa, potencializada pelos algoritmos de busca e classificação; a participação em cursos virtuais, considerando os STI estão inseridos nos MOOCs; e a prática de leitura em línguas estrangeiras, podendo acontecer por meio de tradutores simultâneos, tanto de texto quanto de voz, que fazem uso do PLN. Os OA também podem receber contribuições da IA, desde a sua construção até sua utilização.

A IA pode auxiliar a construção de tais objetos de diferentes maneiras, como por exemplo na identificação dos tipos de objetos mediante treinamento de algoritmos de aprendizado de máquina ou aprendizagem profunda; no auxílio à criação dos objetos, identificando padrões de características ou de atendimento aos descritores de aprendizagem; na percepção do pensamento computacional e das reações emocionais sobre o uso dos objetos; e na interligação de repositório por serviços de nuvem, contribuindo para uma aprendizagem colaborativa. (ZATTI; KALINKE, 2021, p. 92).

Nesse viés, esta pesquisa busca implementar algoritmos de aprendizado de máquina para identificar: 1) se o que está sendo construído na GenIA pode ser considerado um OA; 2) se trata-se de um OA de Matemática; 3) se é um OA que trata de um conteúdo específico. Além disso, a plataforma deverá também identificar aspectos da MPEDUC.

De uma maneira geral, a inserção da IA na Educação Matemática ainda é, de certa forma, embrionária. Por conta disso, este trabalho pretende contribuir para a área no que tange a temática apresentada. Na próxima seção, são apresentados detalhes de como a IA foi implementada na plataforma.

6.5 IA na GenIA

Desde o início da pesquisa, diferentes abordagens foram consideradas para implementar a IA na GenIA. No início do projeto, cogitou-se o uso de uma vertente específica da IA, a saber: o raciocínio baseado em casos (RBC). O principal enfoque do RBC é resolver um novo problema lembrando um problema anterior similar, e assim reutilizar a informação e o conhecimento daquela situação (SCHANK e RIESBECK, 1989).

Souza *et al.* (2016) pontuam sobre a importância do RBC no que tange ao tempo de resposta exigido para o sistema:

A capacidade de reutilizar experiências passadas para resolver novas situações é um comportamento comum dos seres humanos. Porém, nem sempre é possível tirar o melhor proveito dessas experiências, principalmente quando o processo de tomada de decisão precisa ser rápido. (SOUZA *et al.*, 2016, p. 4579, tradução nossa)

Utilizando essa técnica, é necessário transformar em casos o conhecimento de um especialista do domínio em questão. Neste ponto entraria o professor de Matemática e as possíveis soluções para um determinado problema, no caso os diversos OA de um conteúdo específico. “Esta atividade é denominada aquisição de conhecimento, e pode ser feita de maneira manual, semi-automática ou automática.” (REZENDE, 2003).

A representação dos casos e aquisição do conhecimento devem compor uma base de dados que será utilizada para a recuperação dos casos.

Esta é considerada uma etapa crítica em um sistema de raciocínio baseado em casos, pois dada a descrição de um problema, é necessário que um algoritmo de recuperação encontre casos similares da base de casos. Para esta tarefa de recuperação de casos se faz necessário algoritmos de busca eficientes, uma vez que a busca envolve validar o quão semelhante um problema é de um ou mais casos. Esta comparação pode não ser simples, em particular quando envolve o julgamento de valor e não apenas se um valor é igual a outro. (SOUZA, 2018, p. 32)

Ao adotar o RBC, parte-se do pressuposto que, durante a criação de um OA, a recuperação de OA semelhantes ao que está sendo criado pode servir como base para inferência das ações do professor no processo de criação. Diante das

particularidades observadas sobre o RBC, entende-se que ele seria uma opção viável para a GenIA.

Porém, seguindo a ideia inicialmente proposta pelo pesquisador, de adotar os recursos disponíveis no *framework* .NET, buscou-se uma aproximação com o ML.NET, que é um conjunto de bibliotecas de aprendizado de máquina disponíveis para serem utilizadas diretamente em programas escritos na Linguagem C#.

Pelo disposto na pesquisa a respeito das expectativas de IA para a GenIA, entendeu-se que o aprendizado de máquina seria do tipo aprendizagem supervisionada. Isto porque, desde o princípio, a proposta era de que, quando o professor criasse um fluxograma, a plataforma identificasse se aquela sequência de comandos corresponderia a um OA de conteúdo específico. Neste caso, os pares entrada-saída seriam: o fluxograma do OA como entrada; e um rótulo “sim” ou “não” como saída, indicando se tratar ou não do conteúdo treinado.

Para garantir um grau aceitável de acuidade, seria necessário produzir uma massa de dados suficiente para gerar os pares entrada-saída, tanto com amostras positivas, quanto negativas, para o treinamento dos algoritmos. Tendo-se em mente a ideia de que os integrantes do GPTM pudessem produzir uma quantidade suficiente de OA para um treinamento inicial, buscou-se por um assunto matemático que pudesse contribuir de alguma forma com a comunidade acadêmica.

Neste sentido, encontrou-se uma possibilidade nos relatórios de uma avaliação diagnóstica de desempenho realizada por estudantes do Ensino Médio, no estado do Paraná, em 2019. Os resultados dessa avaliação revelaram uma defasagem no conteúdo de função polinomial do 1º grau, o que sugere uma demanda para ser explorada.

Cabe aqui pontuar que este foi um dos fatores motivadores da já mencionada pesquisa de mestrado, vinculada ao macroprojeto e ao PPGFCET, que buscou investigar concepções a respeito da construção de OA para ensino de função polinomial.

Considerando a aprendizagem supervisionada, seria produzido um número expressivo de arquivos (fluxogramas) que considerassem as seguintes variações:

- Não ser um OA;
- Ser um OA de qualquer conteúdo que não fosse matemático;
- Ser um OA que não fosse de Matemática, mas que contemplasse função polinomial de 1º grau;

- Ser um OA de Matemática que não contemplasse função polinomial do 1º grau;
- Ser um OA de Matemática que contemplasse função polinomial do 1º grau.

Durante a implementação da IA na GenIA, com a realização de testes iniciais feitos a partir dos OA criados pelo grupo de pesquisa, identificou-se que: 1) é possível o treinamento dos algoritmos de IA para atender os anseios da pesquisa; porém 2) a quantidade de amostras produzida não seria suficiente para que o treinamento dos algoritmos de IA pudessem atingir a acurácia necessária dentro do prazo previsto para a conclusão do doutorado.

Neste viés, vale destacar que a GenIA é um protótipo vinculado a uma pesquisa e está em fase inicial. A título de comparação, sistemas disponibilizados para domínio público e já consagrados, como o Chat GPT⁴², ainda apresentam problemas de acurácia, fornecendo respostas incorretas a respeito de diversos assuntos. Neste caso, isto acontece porque os algoritmos, baseados em modelos de linguagem, buscam informações em linguagem natural, a partir de bases de dados a eles fornecida. De maneira similar, outras técnicas ou algoritmos de IA também podem apresentar problema de acurácia decorrente da quantidade de amostras inerentes ao contexto sob o qual estão sendo utilizados.

Considerando a questão da insuficiência de amostras, ao invés de treinar a GenIA com OA sobre função polinomial, optou-se por prepará-la para ser treinada para reconhecer qualquer conteúdo. Dessa maneira, os algoritmos foram implementados usando o ML.NET e permanecerão disponíveis para treinamento constante a partir de OA criados pelo grupo de pesquisa, podendo servir de base para futuras pesquisas.

Segundo documentação técnica da Microsoft⁴³, a implementação do aprendizado de máquina no ML.NET deve seguir algumas etapas, a saber:

- 1) Criação de um contexto ML;
- 2) Carga dos dados;
- 3) Transformação dos dados;
- 4) Escolha do algoritmo;

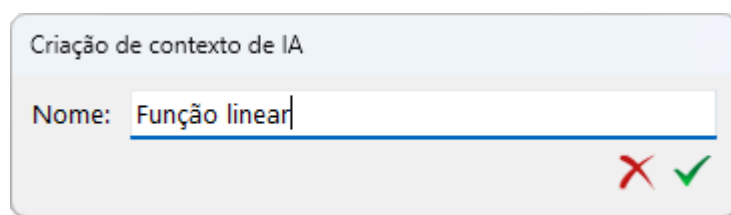
⁴² O ChatGPT (<https://openai.com/chatgpt>) é um *chatbot online*, desenvolvido pela OpenAI, lançado em novembro de 2022.

⁴³ Disponível em: <https://dotnet.microsoft.com/en-us/learn/ml-dotnet/what-is-mldotnet>. Acesso em: 17 jun. 2023.

- 5) Treinamento do modelo;
- 6) Avaliação do modelo;
- 7) Uso do modelo.

A criação do contexto diz respeito ao conteúdo que está sendo treinado. Para atender a esta etapa, foi implementada na GenIA uma opção em que o professor pode criar rótulos para identificar os diferentes conteúdos e posteriormente selecioná-los para o treinamento. A Figura 25 apresenta a tela de criação de contexto:

Figura 25 – Criação de contexto para treinamento da IA



Criação de contexto de IA

Nome:

X ✓

Fonte: Autoria própria (2023)

A carga dos dados é feita a partir da leitura do arquivo próprio da GenIA, no qual estão inseridos os fluxogramas no formato Json, sendo eles a variável de entrada, e o indicativo de se tratar ou não do conteúdo para o qual o algoritmo está sendo treinado.

A transformação dos dados é uma etapa adicional, para caso se queira direcionar quais atributos do conteúdo devem ser tratados pelos algoritmos. Esta possibilidade de escolha não foi implementada neste momento, o que significa que a GenIA irá utilizar comandos e parâmetros internamente definidos pelo programador. Para fins de esclarecimento, caso fosse aberto ao professor parametrizar a entrada de valores, seria como o professor poder escolher quais comandos do fluxograma permaneceriam a partir da carga de dados e quais comandos seriam excluídos do processo; regra similar seria aplicada aos parâmetros dos comandos.

A escolha do algoritmo é uma tarefa atribuída exclusivamente pelo programador. Os algoritmos disponibilizados para uso pelo ML.NET são apresentados no Quadro 19.

Quadro 19 – Algoritmos para treinamento do ML.NET

Tarefa do ML	Algoritmo
Classificação binária	AveragedPerceptronTrainer, SdcaLogisticRegressionBinaryTrainer
Classificação multi-classe	LightGbmMulticlassTrainer, OneVersusAllTrainer
Regressão	LbfgsPoissonRegressionTrainer, FastTreeRegressionTrainer
Clusterização	KMeansTrainer
Detecção de anomalia	RandomizedPcaTrainer
Recomendação	MatrixFactorizationTrainer
Ranking	LightGbmRankingTrainer, FastTreeRankingTrainer

Fonte: ML.NET (2023, tradução nossa)

Para classificação e identificação do conteúdo da GenIA, foi usado o modelo baseado na tarefa de Classificação binária, utilizando uma classificação linear simples, por meio do algoritmo AveragedPerceptronTrainer, baseado em RNA Perceptrons.

Por conta do direcionamento da pesquisa e dos motivos expostos, as etapas 5, 6 e 7, que correspondem, respectivamente, ao treinamento, avaliação e uso do modelo, ainda se encontram em fase inicial de aproveitamento, com perspectivas de serem abordadas por pesquisas futuras, algumas já em andamento no macroprojeto. Contudo, é possível executar o treinamento, ainda que com uma quantidade de amostras reduzidas, conforme apresentado na Figura 26, escolhendo-se o contexto e acionando a função de alimentação do algoritmo com uma amostra positiva ou negativa. Também é possível: avaliar a acurácia dos algoritmos em relação a um contexto; excluir um contexto; consultar o quanto o fluxograma que está sendo construído se aproxima de um determinado contexto.

Figura 26 – Treinamento e uso da IA

Fonte: Autoria própria (2023)

É importante destacar que a versão inicial da GenIA, por se tratar de um protótipo, armazena os dados de treinamento dos algoritmos em uma base de dados local. Isto significa que um contexto treinado em um computador, neste momento, só poderá ser utilizado naquele mesmo equipamento. Porém, sob a solução de

desenvolvimento adotada, baseada no *framework* .NET, será possível, em futuras versões, compartilhar os dados dos contextos de treinamento dos algoritmos em serviços de nuvem como, por exemplo, o Microsoft Azure⁴⁴ ou o Amazon AWS⁴⁵.

Conhecendo-se o trabalho resultante da tese, o capítulo a seguir apresenta as considerações finais, com sugestões de trabalhos futuros a partir de um olhar cuidadoso sobre o caminho percorrido.

Oportuno mencionar, mais uma vez, que no documento de apresentação do produto, o leitor interessado em explorar a GenIA irá encontrar a exploração completa da GenIA, com instruções de acesso e uso.

⁴⁴ Serviço de nuvem da Microsoft (<https://azure.microsoft.com>)

⁴⁵ Serviço de nuvem da Amazon (<https://aws.amazon.com>)

7 CONSIDERAÇÕES

Com esta pesquisa, buscou-se legitimar a tese de que seria possível utilizar recursos de IA, aliados à programação intuitiva, para a construção de OA de Matemática. Com objetivo de concretizar as premissas estabelecidas, desenvolveu-se uma plataforma, denominada GenIA, como produto educacional ao qual a pesquisa está vinculada.

Em busca dos aportes teóricos para a tese apresentada, a pesquisa assumiu uma abordagem qualitativa, calcando-se em duas bases metodológicas: a PDE, sugerida para orientar pesquisas sob o contexto educacional, e o processo de software por prototipação, como norteador dos aspectos técnicos do desenvolvimento de software.

Na fase preliminar da PDE, foram realizados mapeamentos e revisões bibliográficas com intuito de compor a fundamentação teórica para as principais temáticas identificadas: OA, programação intuitiva e IA.

Sobre OA, foram identificadas características, classificações e metodologias para sua produção. A programação de computadores e o design de interação deram suporte à compreensão dos pilares da programação intuitiva. Como modalidade de programação intuitiva, adotou-se o fluxograma como mecanismo de implementação da lógica de funcionamento dos OA, diferentemente da programação por blocos adotada por outras plataformas.

Não menos importante, buscou-se relatar os desafios que podem emergir da conversa entre a Educação Matemática e a Ciência da Computação no que tange as temáticas envolvidas nesta pesquisa. Dentre tais desafios, que ainda estão esboçados em artigo para publicação futura, podem ser adiantadas algumas provocações: os algoritmos de IA são treinados ou ensinados? Isto é, pode-se dizer que uma IA aprende? Quando se fala de usabilidade, entre o professor e a GenIA, existe interação ou interatividade? Questionamentos como os ilustrados aqui fizeram parte da trajetória desta pesquisa, provocando discussões muitas vezes acaloradas, entre os pesquisadores das diferentes áreas do macroprojeto, revelando o quão complexo, importante, e até fascinante um trabalho interdisciplinar pode ser.

Os estudos teóricos que circundaram o desenvolvimento da GenIA serviram de base para a produção e publicação de quatro artigos em periódicos Qualis A na

área de Ensino (sendo um A1, dois A2 e um A3)⁴⁶, considerando-se a classificação do quadriênio 2017-2020 sob a Plataforma Sucupira. Além dos artigos, a pesquisa foi apresentada como trabalho em congressos da área (VI SIPEQ, XXIV EBRAPEM, II EPTM e XIV ENEM), além de alguns workshops dentro e fora da UTFPR.

Uma visita ao histórico e fundamentos da IA proporcionou vislumbrar possibilidades de implementação de algoritmos inteligentes, pretendendo-se, naquele momento, fazer uso de algoritmos de aprendizado de máquina.

O percurso metodológico revelou que o desenvolvimento da GenIA iria acontecer em dois momentos bastante distintos: 1) o desenvolvimento do software contemplando a programação intuitiva; 2) a implementação dos recursos de IA.

Simultaneamente ao desenvolvimento da primeira fase, atendendo aos requisitos técnicos, foram feitas verificações e validações em diferentes níveis, de acordo com o processo de software adotado. Sob a perspectiva educacional, considerando aspectos de usabilidade e da programação intuitiva, os professores integrantes do GPTM fizeram sucessivas avaliações, de acordo com as iterações orientadas pela PDE.

Durante a primeira fase, anterior à qualificação do doutorado, foram geradas aproximadamente 40 versões de compilação da GenIA, para inserção de novas funcionalidades e correções. Como já foi comentado, estas versões não integraram o versionamento semântico, sendo considerada sempre em sua primeira versão até a defesa da tese. Para que se possa ter uma ideia aproximada a respeito do esforço

⁴⁶ Artigos publicados a partir da pesquisa:

1. MATTOS, S. G.; ZATTI, E. A.; BALBINO, R. O.; MOCROSKY, L. F.; KANLINKE, M. A. Uma Análise de Propostas de Uso de Tecnologias Digitais na Rede Estadual do Paraná nos Anos de 2015 a 208. **RPEM** – Revista Paranaense de Educação Matemática, v. 9, n. 20, 153-173. nov-dez 2020. Disponível em: <https://doi.org/10.33871/22385800.2020.9.20.153-173>. Acesso em: 26 ago. 2023.
2. STAVNY, F. M.; MATTOS, S. G. DE; BALBINO, R. O.; ZATTI, E. A.; KALINKE, M. A. Em Busca de Compreensões sobre Utilização de Recursos Digitais na Criação de Objetos de Aprendizagem de Matemática. **Revemat**: Revista Eletrônica de Matemática, v. 16, 1-22. 25 out. 2021. Disponível em: <https://doi.org/10.5007/1981-1322.2021.e80312>. Acesso em: 26 ago. 2023.
3. BALBINO, R. O.; KALINKE, M. A.; ZATTI, E. A.; MATTOS, S. G. DE; LOSS, T.; MOTTA, M. S. Programação Intuitiva: em Busca de Compreensões. **Perspectivas da Educação Matemática**, v. 14, n. 36, 1-22. 17 dez 2021. Disponível em: <https://doi.org/10.46312/pem.v14i36.12121>. Acesso em: 26 ago. 2023.
4. ZATTI, E. A.; BALBINO, R. O.; MATTOS, S. G. DE; KALINKE, M. A. Uma Proposta Para A Criação De Uma Plataforma Assistida Pela Inteligência Artificial Para Construção De Objetos De Aprendizagem De Matemática. **Revista Paradigma** – Edição Temática, v. XLIII, n 43, 259-281, 02 mai. 2022. Disponível em: <https://doi.org/10.37618/PARADIGMA.1011-2251.2022.p259-281.id1226>. Acesso em: 26 ago. 2023.

empreendido em sua programação, a versão apresentada na qualificação contava com aproximadamente 20 mil linhas de código-fonte.

Depois de qualificada a pesquisa, o desenvolvimento rumou à etapa seguinte: a implementação da IA. A partir do momento em que o desenvolvimento da GenIA adentrou a exploração das possibilidades da IA, optou-se por interromper o desenvolvimento de novas funcionalidades pertinentes a comandos e parametrização. Além da validação por parte dos pesquisadores do GPTEM, a plataforma foi aplicada em ambiente real acadêmico em duas circunstâncias: primeiramente com estudantes de Bacharelado em Engenharia de Software, que teceram considerações técnicas a respeito da construção dos fluxogramas, e posteriormente como recurso para elaboração do projeto prático de uma disciplina da Licenciatura em Matemática. Naquele momento, versionamento interno do software já excedia a marca de cem compilações, com um número aproximado de 45 mil linhas de código-fonte. Ainda que não se possa comparar diretamente programação com produção textual, sob uma métrica simples, seria o equivalente a 1500 laudas. Cabe ressaltar que o registro no INPI foi feito somente para as classes principais do código, suficientes para caracterizar sua autoria, e gerou um arquivo com 465 páginas em tamanho A3.

Um olhar sobre o trajeto percorrido e, não menos importante, sobre a GenIA enquanto produto educacional concretizado, testado sob o ponto de vista técnico, avaliado por pesquisadores da Educação Matemática, e usado como recurso em ambiente educacional real, revela que a tese levantada é viável, sendo possível, sim, a criação de OA de Matemática com uso de programação intuitiva e assistida por IA.

A respeito dos movimentos que circundam esta pesquisa, percebeu-se que a inserção da IA entre as temáticas de estudo e pesquisa do GPTEM tem revelado um caminho bastante insólito. À medida em que surgem novas propostas como a desta pesquisa, manifestando vertentes a serem exploradas no âmbito educacional, há também um certo receio por parte dos pesquisadores. E isto não é uma percepção que recaia exclusivamente sobre os professores do grupo. Durante o caminhar da pesquisa, em diversos momentos em que a GenIA era apresentada a pesquisadores de outras instituições ou até mesmo de outras áreas, não era incomum a surpresa com a proposta de se tentar aproximar tanto a IA do ofício do professor.

Tal comportamento evidencia a importância de que se tenham mais pesquisas interdisciplinares envolvendo IA e Educação, para que alguns preceitos sejam desmistificados. Talvez seja necessário que os professores desenvolvam simpatia por

recursos de IA, para que atuem como mediadores, e que esta área tão promissora possa ser uma forte aliada nos processos de ensino e de aprendizagem.

Evidentemente, um trabalho dessa natureza, envolvendo diferentes áreas de conhecimento e com temáticas tão complexas, deixa lacunas a serem exploradas. A seguir, são apontadas algumas possibilidades de pesquisas que podem emergir a partir desta.

7.1 Recomendações para trabalhos futuros

As provocações para trabalhos futuros começam por aspectos técnicos que ainda podem contribuir diretamente com a GenIA. Por questões de direcionamentos já apresentados, a versão inicial da plataforma foi desenvolvida para ambiente Windows. Porém, se trata de um sistema operacional proprietário, com licença paga, o que incorre em algumas limitações de uso. Portanto, a primeira proposta é que se porte a plataforma para ambiente web, para que ela possa ser utilizada em qualquer equipamento que possua os requisitos mínimos computacionais para sua execução. Concomitante a esta mudança, sugere-se que os OA criados a partir da GenIA também possam ser executados de maneira independente em diversas plataformas, incluindo dispositivos móveis.

Sob óptica da usabilidade, sabe-se que a plataforma carece de diversos recursos comumente encontrados em ambientes visuais, como, por exemplo, a cópia e colagem de objetos, recursos de arrastar-e-soltar, opções de configuração de cores, disposição de leiaute. Ainda há que se falar de recursos como salvamento automático e compartilhamento dos OA por meio de nuvem.

Redirecionando o olhar dos aspectos técnicos para questões mais abrangentes, sugere-se que seja possível integrar a plataforma com componentes de hardware para criação de OA que possam ser utilizados em aulas de Robótica Educacional, espelhando-se em soluções como as que integram Scratch com Arduíno, por exemplo.

A GenIA carrega desde suas fundações uma preocupação especial com a acessibilidade. Sua versão inicial proporciona a escolha entre dois idiomas (português e inglês) para sua utilização, tendo sido preparada com estrutura técnica que possibilita, de forma relativamente fácil, sua tradução para outros idiomas, como italiano ou francês, já numa próxima versão.

A IA foi utilizada na GenIA para uma tarefa específica, relacionada à classificação dos OA sob um determinado conteúdo indicado pelo professor, por meio de aprendizado supervisionado. Sob este viés, considera-se importante seguir explorando possibilidades de inserção de novos algoritmos de IA e seus respectivos treinamentos.

Entende-se, por exemplo, que seja possível utilizar algoritmos de aprendizado não-supervisionado para classificar automaticamente OA de diversos conteúdos. E, com estudos mais avançados, visando o uso de aprendizagem profunda, que haja possibilidade de se importar OA de outras plataformas ou exportar os OA construídos com a GenIA para outros ambientes. Nesta mesma linha, a IA poderia ser utilizada para combinar e recombinar OA a partir de um problema específico.

Outra contribuição que a IA poderia trazer para a plataforma seria durante a construção dos fluxogramas. Atualmente, ferramentas de desenvolvimento voltada a programadores profissionais incorporam recursos que sugerem automaticamente a complementação de comandos, inferindo a digitação para sua parametrização, ou até mesmo comandos subsequentes. Em analogia, durante a composição de um fluxograma na GenIA, ela poderia inferir sobre uma sequência de objetos que fariam sentido na resolução de um problema comum. Por exemplo: ao montar um fluxograma com intenção de traçar uma reta por meio de função linear, ao se arrastar o comando que solicita ao usuário o coeficiente angular, a plataforma poderia sugerir automaticamente o comando análogo para que fosse informado o coeficiente linear, bem como criar automaticamente as variáveis para seu armazenamento.

Atualmente, a GenIA interage com o usuário por meio de caixas de diálogo comuns a outros aplicativos visuais: textos informativos, botões de confirmação etc. Ao invés desse mecanismo padrão, seria incorporado a ela um *chatbot*, por meio do qual a interação, além de ser mais natural, poderia dialogar com o professor a respeito de suas intenções e ações na criação de um OA para um problema ou conteúdo conhecido. Ao mesmo tempo, a IA estaria sendo treinada a respeito do padrão de uso do professor para que tal diálogo acontecesse de maneira personalizada.

Sobre o uso da IA, as sugestões para futuras pesquisas poderiam compor uma lista extensa, à medida que se coloca a IA como um elemento quase-humano. Para esta vertente, findam-se aqui as provocações, para que não se corra o risco de pensar que, com IA, qualquer coisa pode ser feita com e a partir da plataforma, da mesma maneira que uma pessoa estaria limitada apenas à sua criatividade.

Ampliando a visão sob uma perspectiva acadêmico-científica, sugere-se o uso da GenIA por diversos públicos (professores, acadêmicos de licenciatura, estudantes de ensino médio etc.), acompanhado de pesquisadores que busquem compreender como ela pode ser explorada em atividades educacionais.

REFERÊNCIAS

- AKST, J. Machine, Learning, 1951. **The Scientist**. 2019. Disponível em: <https://www.the-scientist.com/foundations/machine--learning--1951-65792>. Acesso em: 26 ago. 2023.
- ALVARES, L.O.; SICHMAN, J. Introdução aos Sistemas Multiagentes. *In*: MEDEIROS, C. M. B (ed.). **Jornada de Atualização em Informática**, 16.; Congresso da SBC, 17. Anais. Brasília: SBC, 1997. p.1-38.
- ARAÚJO, J. D. L.; BORBA, M. D. C. **Pesquisa qualitativa em educação matemática**. Belo Horizonte: Grupo Autêntica, 2019. 9788551305942. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788551305942/>. Acesso em: 26 ago. 2023.
- BICUDO, M. A.; BORBA, M.C. (org.). **Educação Matemática: pesquisa em movimento**. 2. ed. revisada. São Paulo: Cortez, 2005.
- BICUDO, M. A.; COSTA, A. P. (org.). **Leituras em Pesquisa Qualitativa**. São Paulo: Editora Livraria da Física, 2019.
- BALBINO, R. O. *et al.* Programação Intuitiva: em Busca de Compreensões. **Perspectivas da Educação Matemática**, v. 14, n. 36, p. 1-22, 17 dez. 2021. Disponível em: <https://doi.org/10.46312/pem.v14i36.12121>. Acesso em: 26 ago. 2023.
- BALBINO, R. O.; MATTOS, S. G. Uma proposta de utilização da MPEDUC para a construção de objetos de aprendizagem. *In*: MOTTA, M. S.; KALINKE, M. A. (org.). **Inovações e Tecnologias Digitais na Educação: uma busca por definições e compreensões**. Campo Grande, MS: Life Editora, 2021, p. 57-74.
- BORBA, M. C.; MALHEIROS, A. P. S.; ZULATTO, R. B. A. **Educação a Distância on-line**. Belo Horizonte: Autêntica, 2007.
- BORBA, M. C.; SCUCUGLIA, R. R. S.; GADANIDIS, G. **Fases das tecnologias digitais em Educação Matemática: sala de aula e internet em movimento**. 2. ed.; 2. reimp. Belo Horizonte: Autêntica, 2018.
- BRAGA, J. C. *et al.* Desafios para o Desenvolvimento de Objetos de Aprendizagem Reutilizáveis e de Qualidade. **DESAFIE – Workshop de Desafios da Computação Aplicada à Educação**, 2012, p. 90-99. Disponível em: <https://www.br-ie.org/pub/index.php/desafie/article/view/2779> Acesso em: 26 ago. 2023.
- BRAGA, J. C.; PIMENTEL, E.; DOTTA, S. Processos e Metodologias para o Desenvolvimento de Objetos de Aprendizagem. *In*: BRAGA, J. C. (org.). **Objetos de aprendizagem, volume 2: metodologia de desenvolvimento**. Santo André: Editora da UFABC, 2015, p. 23–56.
- BRASIL, CAPES. **Documento de Área – Área 46 – Ensino**. 2019. Disponível em: <https://www.gov.br/capes/pt-br/centrais-de-conteudo/ENSINO.pdf>. Acesso em: 26 ago. 2023.

BRASIL, CAPES. **Relatório de Avaliação – Ensino – Avaliação Quadrienal 2017 (2013-2016)**. 2017. Disponível em: https://www.gov.br/capes/pt-br/centrais-de-conteudo/documentos/avaliacao/relatorio_quadrienal_ensino.pdf. Acesso em: 17 jun. 2023.

BUCHANAN, B. G.; SUTHERLAND, G. L., FEIGENBAUM, E. A. Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry. In: MELTZER et al (Orgs.), **Machine Intelligence**, vol. 4, pp. 209-254. Edinburgh University Press, 1969.

CASTAÑEDA, L.; SELWYN, N. More than tools? Making sense of the ongoing digitizations of higher education. **International Journal of Educational Technology in Higher Education**, [s. l.], v. 15, n. 22, p. 1-10, mai. 2018. Disponível em: <https://doi.org/10.1186/s41239-018-0109-y>. Acesso em: 26 ago. 2023.

CHALMERS, A. F. **What is this thing called Science?** 3rd ed. Indianapolis: Hackett Publishing, 1999.

D'AMBROSIO, U. In: ARAÚJO, J. L., BORBA, M. C. **Pesquisa Qualitativa em Educação Matemática**. Belo Horizonte: Editora Autêntica, 2004, 120p, p. 9-21 (Prefácio). Entrevista: <https://www.youtube.com/watch?v=o8OUA7JE2UQ>

DOERR, A. **All The Light We Cannot See**. New York: Scribner, 2014.

FAYYAD, U. *et al*. From Data Mining to Knowledge Discovery in Databases. **AI Magazine**, vol. 17, n. 3, pp. 37-54, AAAI Press, MIT Press, 1996.

FLECK, L. **Gênese e desenvolvimento de um fato científico**: introdução à doutrina do estilo de pensamento e do coletivo de pensamento. Tradução: Georg Otte, Mariana Camilo de Oliveira. Belo Horizonte: Fabrefactum Editora, 2010.

GAMBOA, H.; FRED, A. Designing Intelligent Tutoring Systems: a Bayesian Approach. **3rd International Conference on Enterprise Information Systems, ICEIS'2001**.

GARNELO, M.; SHANAHAN, M. Reconciling deep learning with symbolic artificial intelligence: representing objects and relations. **Current Opinion in Behavioral Sciences**, v. 29, p. 17-23, 2019. Disponível em: <https://doi.org/10.1016/j.cobeha.2018.12.010>. Acesso em: 26 ago. 2023.

GUTIERRES, S. S. Distribuição de conteúdos e aprendizagem on-line. **RENOTE – Revista Novas Tecnologias na Educação**, v. 2, 2004, p. 1-14.

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

HOLMES, W.; BIALIK, M.; FADEL, C. **Artificial Intelligence in Education: Promises and Implications for Teaching & Learning**. Boston: Center for Curriculum Redesign, 2019.

IEEE. Standard for System and Software Verification and Validation. In: **IEEE Std 1012-2012**. 25 mai 2012. Disponível em: <https://standards.ieee.org/ieee/1012/4021/>. Acesso em: 26 ago. 2023.

IEEE. Standard for Learning Object Metadata. *In: IEEE Std 1484.12.1-2020*, vol., no., pp.1-50, 16 nov. 2020, doi: 10.1109/IEEESTD.2020.9262118. Disponível em: <https://standards.ieee.org/ieee/1484.12.1/7699>. Acesso em: 26 ago. 2023.

INMON, W. H. **Building the Data Warehouse**. 3rd ed. New York: John Wiley & Sons, 2002.

KALINKE, M. A. **Tecnologias no Ensino: a linguagem matemática na web**. Curitiba: CRV, 2014.

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

KALINKE, M. A.; MOCROSKY, L. F.; MOTTA, M. S. Mapeamento das dissertações que versam sobre o uso de tecnologias educacionais no ensino de física. **Actio**, Curitiba, v. 3, n. 3, set./dez. 2017 (2018). Disponível em: <https://doi.org/10.3895/actio.v3n3.7591>. Acesso em: 26 ago. 2023.

KEMCZINSKI, A.; COSTA, I. A.; WEHRMEISTER, M. A.; HOUNSELL, M. S.; VAHLDICK, A. Metodologia para Construção de Objetos de Aprendizagem Interativos. **Anais do 23º Simpósio Brasileiro de Informática na Educação (SBIE 2012)**, Rio de Janeiro, 2012. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/1739>. Acesso em 17 jun. 2023.

KENSKI, V. M. Aprendizagem mediada pela tecnologia. **Revista Diálogo Educacional**, Curitiba, v. 4, n.10. 2003. Disponível em: <https://periodicos.pucpr.br/index.php/dialogoeducacional/article/viewFile/6419/6323>. Acesso em: 26 ago. 2023.

KENSKI, V. M. **Educação e tecnologias: O novo ritmo da informação**. 8. ed. São Paulo: Papirus, 2011.

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

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

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

_____. **As tecnologias da inteligência: o futuro do pensamento na era da informática**. 2. ed. São Paulo: Editora 34, 2010.

_____. **O que é o virtual?** Tradução de Paulo Neves. 2. ed. São Paulo: Editora 34, 2011.

_____. **O big data e a próxima revolução científica**. 2016. Disponível em: <https://www.youtube.com/watch?v=W5hlcxKPVRw>. Acesso em: 26 ago. 2023.

_____. **Cibercultura**. Tradução de Carlos Irineu Costa. 3. ed. São Paulo: Editora 34, 2018.

- LIMA, J. M.; SIPLE, I. Z. GeoGebra Classroom: uma plataforma virtual com ferramentas matemáticas interativas. **Revista Paranaense de Educação Matemática**, Campo Mourão, v. 10, p. 493-515, mai.-ago. 2021. Disponível em: <https://doi.org/10.33871/22385800.2021.10.22.493-515>. Acesso em: 26 ago. 2023.
- LIMA, J. M.; SIPLE, I. Z. GeoGebra grupos e objetos de aprendizagem: um recurso para exploração do raciocínio covariacional em tempos de aulas não presenciais. **Revista BOEM**, Florianópolis, v. 9, n. 18, p. 253-273, out. 2021. Disponível em: <http://doi.org/10.5965/2357724X09182021253>. Acesso em: 26 ago. 2023.
- LOLLINI, Massimo; FARLEY, Arthur; LEVY, Pierre. Collective Intelligence, the Future of Internet and the IEML. **Humanist Studies & the Digital Age**, [S.l.], v. 6, n. 1, p. 5-31, dec. 2019. ISSN 2158-3846. Disponível em: <http://dx.doi.org/10.5399/uo/hsda.6.1.2>. Acesso em: 26 ago. 2023.
- MCCARTHY, J. *et al.* **A Proposal for The Dartmouth Summer Research Project on Artificial Intelligence**. 1955. Disponível em: <http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>. Acesso em: 26 ago. 2023.
- MCCULLOCH, W. S.; PITTS, W. H. A Logical Calculus of The Ideas Immanent. *In: Nervous Activity. Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943.
- MCKENNEY, S.; REEVES, T. C. **Conducting Educational Design Research**. Abingdon, UK: Routledge, 2012.
- MEIRELES, T. F. **Desenvolvimento de um Objeto de Aprendizagem de Matemática usando o Scratch**: da elaboração à construção. Curitiba, 2017. Dissertação (Mestrado em Educação) – Universidade Federal do Paraná.
- MENDES, R. M.; SOUZA, V. I.; CAREGNATO, S. I. **A Propriedade Intelectual na Elaboração de Objetos de Aprendizagem**. 2007. Disponível em <https://lume.ufrgs.br/handle/10183/548>. Acesso em: 26 ago. 2023.
- MINSKY, M. L.; PAPERT, S. A. **Perceptrons**: An Introduction to Computational Geometry. Expanded Edition. Third printing. Boston: MIT Press, 1988.
- ML.NET. **What is ML.NET?** Disponível em: <https://dotnet.microsoft.com/en-us/learn/ml-dotnet/what-is-mldotnet>. Acesso em: 26 ago. 2023.
- MOTTA, M. S. **Contribuições do SuperLogo ao ensino de geometria do sétimo ano da educação básica**. 2008. 226p. Dissertação (Mestrado em Ensino de Matemática) – Pontifícia Universidade Católica de Minas Gerais, Programa de Pós-Graduação em Ensino de Ciências e Matemática.
- MOTTA, M. S.; BASSO, S. J. L.; KALINKE, M. A. Mapeamento sistemático das pesquisas realizadas nos programas de mestrado profissional que versam sobre a aprendizagem matemática na educação infantil. **Actio: Docência em Ciências**, Curitiba, v. 4, n. 3, p. 204-225, 2019.
- MOTTA, M. S.; KALINKE, M. A. Uma proposta metodológica para a produção de objetos de aprendizagem na perspectiva da dimensão educacional. *In: _____* (org.). **Objetos de aprendizagem**: pesquisas e possibilidades na Educação Matemática. Campo Grande, MS: Life Editora, 2019, p. 203-218.

- MOTTA, M.S.; KALINKE, M. A. Em busca de compreensões sobre os objetos de aprendizagem na educação matemática por meio de uma revisão sistemática de literatura. **Educação Matemática Pesquisa**: Revista do Programa de Estudos Pós-Graduados em Educação Matemática, v. 23, n. 1, 2021, p. 140-170.
- MOTTA, M. S.; KALINKE, M. A.; MOCROSKY, L. F. Mapeamento das dissertações que versam sobre o uso de tecnologias educacionais no ensino de Física. **Actio: Docência em Ciências**, Curitiba, v. 3, n. 3, p. 65-85, 2018.
- MULHOLLAND, P. Incorporating Software Visualization into Prolog teaching: a challenge, a restriction, and an opportunity. *In: Proceedings of ICLP'97 Postconference workshop on logic programming environment*, 14, 1997, Leuven. Proceedings of ICLP'97 Postconference Workshop on Logic Programming Environment. MIT Press, p. 33-42, 1997.
- MUNHOZ, A. S. **Objetos de Aprendizagem**. Curitiba: InterSaber, 2013, 220p.
- MYERS, G. J.; BADGETT, T.; SANDLER, C. **The Art of Software Testing**. 3rd ed. New Jersey: John Wiley & Sons, 2011.
- OUYANG, F.; JIAO, P. Artificial intelligence in education: The three paradigms. **Computers and Education: Artificial Intelligence**, v. 2, 2021, p. 1-6. Disponível em <https://doi.org/10.1016/j.caeai.2021.100020>. Acesso em: 26 ago. 2023.
- PAPERT, S. **Mindstorms**: children, computers and powerfull ideas. New York, Basic Books. 1980.
- PINHANEZ, C. S. Inteligência Artificial: conceitos fundamentais. **Aula Inaugural do do Curso de Computação IME/USP em 2022**. 10 mar 2022. Disponível em: https://www.youtube.com/watch?v=MAiqP__1Yeg. Acesso em: 26 ago. 2023.
- POLSANI., P. R. Use and Abuse of Reusable Learning Objects. **Journal Of Digital Information**. Arizona, p. 9 20. 19 fev. 2003. Disponível em: <https://journals.tdl.org/jodi/index.php/jodi/article/view/jodi-105/88>. Acesso em: 26 ago. 2023.
- POWELL, A. B.; ALI, K. V. Design Research in Mathematics Education: Investigating A Measuring Approach To Fraction Sense. *In: CUSTODIO, J. F. et al. (org.). Programa De Pós-Graduação Em Educação Científica E Tecnológica (PPGECT): contribuições para pesquisa e ensino – São Paulo: Editora Livraria da Física*, 2018, p. 221-242.
- PLOMP, T. Educational design research: an introduction. *In: PLOMP, T., NIEVEEN, N. Educational design research – Part A: an introduction*. Enschede, November 2013, p. 10-51.
- PRESSMAN, R. W; MAXIM B. R. **Software Engineering – A Practitioner's Approach**. 8th ed. New York: McGraw-Hill, 2015.
- PRESTON-WERNER, T. **Versionamento Semântico 2.0.0**. [s. d.]. Disponível em: <https://semver.org/lang/pt-BR/>. Acesso em: 26 ago. 2023.
- REZENDE, S. O. **Sistemas Inteligentes**. São Paulo, Ed. Manole, 2003.

RESNICK, M. *et al.* (2009). Scratch: Programming for All. **Communications of the ACM**, **52(11)**, p. 60-67. Disponível em: <https://doi.org/10.1145/1592761.1592779>. Acesso em: 26 ago. 2023.

ROBERTS, L. G. Machine Perception of Three-Dimensional Solids. *In*: TIPPETT *et al* (eds.). **Optical and Electro-Optical Information Processing**, May 1965.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v.65, n.6, p. 386-408, 1958.

RUSSELL, S. **Inteligência artificial a nosso favor**: Como manter o controle sobre a tecnologia. Tradução de Berilo Vargas. 1. ed. São Paulo: Companhia das Letras, 2021.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence**. 4th ed. Harlow: Pearson Education, 2021.

SAMUEL, A. L. Some Studies in Machine Learning Using the Game of Checkers. **IBM Journal of Research and Development**, vol. 3, n. 3, p. 1-20, 1959. Disponível em: <https://www.cs.virginia.edu/~evans/greatworks/samuel.pdf>. Acesso em: 26 ago. 2023.

SEARLE, J. Minds, brains, and programs. **The Behavioral and Brain Sciences**, 3, pp. 417-457, 1980. Disponível em: <http://cogprints.org/7150/1/10.1.1.83.5248.pdf>. Acesso em: 26 ago. 2023.

SEBESTA, R. W. **Concepts of Programming Languages**. 11th global ed. Harlow: Pearson Education, 2016.

SCHANK, C. R.; RIESBECK, C. K. **Inside Case-based Reasoning**. [S.l.]: Lawrence Erlbaum, 1989.

SICILIANO, B. *et al.* **Robotics: Modelling, Planning and Control**. London: Springer-Verlag, 2010.

SOFTWARE. *In*: **Grande Dicionário Houaiss**. São Paulo: UOL, 2021. Disponível em: <http://houaiss.uol.com.br>. Acesso em: 26 ago. 2023.

SOMMERVILLE, I. **Software Engineering**. Harlow: Pearson Education, 2016.

SOUZA, V. D. M. **Otimização Multi-Objetivo Aplicada em Raciocínio Baseado em Casos**. Curitiba, 2018. Tese (Doutorado em Informática) – Pontifícia Universidade Católica do Paraná.

SOUZA, V. D. M. *et al.* Automatic Knowledge Learning Using Case-Based Reasoning. *In*: **International Joint Conference on Neural Networks (IJCNN)**, 2016, Vancouver. Neural Networks (IJCNN), 2016 International Joint Conference on, 2016. p. 4579-4585. Disponível em: <https://ieeexplore.ieee.org/document/7727800>. Acesso em: 26 ago. 2023.

TANENBAUM, A. S. **Structured Computer Organization**. London: Pearson Education Inc, 2013.

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

TIOBE. **TIOBE Programming Community Index**. Disponível em: <https://www.tiobe.com/tiobe-index/> Acesso em: 26 ago. 2023.

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

VICARI, R. M. **Tendências em inteligência artificial na educação no período de 2017 a 2030: sumário executivo**. Brasília: SENAI, 2018. Disponível em: <http://tracegp.sesi.org.br/handle/uniepro/259>. Acesso em: 26 ago. 2023.

WILEY, D. Connecting Learning Objects to Instructional Design Theory: A Definition, a Metaphor, and a Taxonomy. *In*: WILEY, D. (ed.). **The Instructional Use of Learning Objects**. Bloomington, Indiana: Tichenor Printing, 2002, p. 3-23.

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

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

ZATTI, E. A. *et al.* Una Propuesta para la Creación de una Plataforma Asistida por la Inteligencia Artificial para la Construcción de Objetos de Aprendizaje de Matemática. **Paradigma**, [s. l.], v. 43, n. 2, pp. 259-281, 2022. DOI: 10.37618/PARADIGMA.1011-22512022. pp. 259-281. id1226. Disponível em: <http://revistaparadigma.online/ojs/index.php/paradigma/article/view/1226>. Acesso em: 26 ago. 2023.