

INSTITUTO FEDERAL

Minas Gerais

Campus Ouro Preto

INSTITUTO FEDERAL DE MINAS GERAIS - IFMG

CAMPUS OURO PRETO

PÓS GRADUAÇÃO EM INTELIGÊNCIA ARTIFICIAL

**Apontamento e Configuração de um Tutor com
Inteligência Artificial (TIA):
Uma Análise Comparativa entre o Uso Direto do
LLM e RAG no Ensino de Cálculo I**

BRUNO DA CUNHA FERREIRA

Ouro Preto - MG

24 de Julho de 2025

F383a Ferreira, Bruno da Cunha.
Apontamento e Configuração de um Tutor com Inteligência Artificial (TIA) [manuscrito] : Uma análise Comparativa entre o uso direto do LLM e RAG no Ensino de Cálculo I / Bruno da Cunha Ferreira. – 2025.
65 f. : il.

Orientador: Carlos Alberto Severiano Junior.
Trabalho de Conclusão de Curso (especialização) – Instituto Federal de Minas Gerais. *Campus* Ouro Preto, 2025.

1. Tutor com Inteligência Artificial. 2. Engenharia de Prompt. 3. Grandes Modelos de Linguagem. 4. Recuperação Aumentada por Geração. 5. Método socrático. I. Severiano Junior, Carlos Alberto. II. Instituto Federal de Minas Gerais. *Campus* Ouro Preto. III. Título.

CDU: 004.8

Catálogo: Kelly Cristiane Santos Morais - CRB-6/3217



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS
Campus Ouro Preto
Diretoria de Pesquisa, Inovação e Pós-Graduação
Coordenação do Curso de Pós-graduação em Inteligência Artificial
Rua Pandiá Calógeras, 898 - Bairro Bauxita - CEP 35400-000 - Ouro Preto - MG
- www.ifmg.edu.br

FOLHA DE APROVAÇÃO

BRUNO DA CUNHA FERREIRA

APONTAMENTO E CONFIGURAÇÃO DE UM TUTOR COM INTELIGÊNCIA ARTIFICIAL (TIA): UMA ANÁLISE COMPARATIVA ENTRE O USO DIRETO DO LLM E RAG NO ENSINO DE CÁLCULO I

Trabalho de Conclusão de Curso
apresentado ao curso de
ESPECIALIZAÇÃO EM
INTELIGÊNCIA ARTIFICIAL,
ofertado pelo Instituto Federal de
Educação, Ciência e Tecnologia
de Minas Gerais - *Campus* Ouro
Preto, como parte dos requisitos
para a obtenção do título de
ESPECIALISTA EM
INTELIGÊNCIA ARTIFICIAL.

Aprovado em 24 de julho de 2025, pela Banca Examinadora:

Prof. Dr. Carlos Alberto Severiano Junior - IFMG *Campus* Sabará - Orientador

Profa. Dra. Silvia Grasiella Moreira Almeida - IFMG *Campus* Ouro Preto

Prof. Dr. Carlos Alexandre Silva - IFMG *Campus* Sabará

Ouro Preto, 24 de julho de 2025.



Documento assinado eletronicamente por **Carlos Alberto Severiano Junior, Professor**, em 24/07/2025, às 19:33, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Silvia Grasiella Moreira Almeida, Responsável pelas Atividades da Coordenação do Curso de Pós-graduação Lato Sensu em Inteligência Artificial**, em 24/07/2025, às 19:47, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Carlos Alexandre Silva, Professor EBTT**, em 24/07/2025, às 19:53, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **2396195** e o código CRC **B28861E1**.

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

AGRADECIMENTOS

Agradeço à Deus por me dar sabedoria, cujo a qual busco mais que o ouro e a prata, e a vida para usufruí-la.

Agradeço de coração à minha esposa Junara, minha família, por todo apoio e compreensão de minha jornada nesta especialização, principalmente durante a execução desta pesquisa em que precisei me ausentar de inúmeros compromissos sociais.

Agradeço aos meus professores, doutores e mestres do conhecimento, por transmitirem seus saberes para que eu pudesse realizar este trabalho.

Agradeço à todos os profissionais do IFMG, servidores e terceirizados, que com o seu trabalho permitiram e permitem o funcionamento desta pós-graduação. Em especial, à prof^a. Dr^a. Sílvia Grasiella Moreira Almeida, por toda à articulação e insistência para a ocorrência e a coordenação desta especialização em Inteligência Artificial.

Por fim, agradeço ao suporte do Prof. Dr. Gabriel Vieira Nunes, que foi um apoio muito importante nesta trajetória.

*“Quando Sócrates
dizia
‘só sei que nada sei’,
não estava fingindo humildade;
expressava a consciência
de que
o conhecimento humano
é limitado
e há sempre mais
a ser descoberto.
(CORTELLA, Mário Sérgio)*

RESUMO

Este trabalho apresenta o desenvolvimento de um tutor com Inteligência Artificial de abordagem socrática para auxiliar o ensino de Cálculo I, avaliando comparativamente duas arquiteturas de implementação: interação direta com um Grande Modelo de Linguagem (LLM) e integração do LLM em um *pipeline* de Recuperação Aumentada por Geração (RAG) aliado à engenharia de *prompt*. Objetivou-se analisar como essas diferentes abordagens impactam a capacidade do tutor em adotar uma postura pedagógica socrática contextualizada, guiando o aluno na resolução de problemas sem fornecer respostas diretas. A metodologia abrangeu a implementação de ambas as soluções com uso do modelo GPT-4 (OpenAI), seguida de simulações em um conjunto de 27 questões de Cálculo I para avaliar o desempenho de cada abordagem. As interações do tutor foram registradas e analisadas quanto à aderência às práticas socráticas. Os resultados evidenciaram que a arquitetura RAG com a técnica de *few-shot learning* alcançou aproximadamente 93% de conformidade com o estilo socrático esperado, enquanto o *pipeline* RAG com *one-shot* obteve cerca de 11%. O uso direto do LLM serviu como linha de base comparativa, evidenciando que o uso de RAG associado a técnicas de engenharia de *prompt* oferece um mecanismo efetivo para regular o comportamento do LLM, permitindo ao tutor inteligente replicar de forma confiável estratégias socráticas de ensino. O estudo demonstra a viabilidade e as vantagens dessa abordagem para o desenvolvimento de tutores virtuais na educação, embora sejam necessárias avaliações empíricas futuras com usuários para confirmar os ganhos pedagógicos observados.

Palavras-chave: Tutor com Inteligência Artificial. Engenharia de Prompt. Grandes Modelos de Linguagem. Método Socrático. Recuperação Aumentada por Geração.

ABSTRACT

This work presents the development of an AI-powered tutor employing a Socratic approach to support the teaching of Calculus I, comparatively evaluating two implementation architectures: direct interaction with a Large Language Model (LLM) and integration of the LLM into a Retrieval-Augmented Generation (RAG) pipeline combined with prompt engineering. The objective was to analyze how these different approaches impact the tutor's ability to adopt a contextualized Socratic pedagogical stance, guiding students in problem-solving without providing direct answers. The methodology encompassed the implementation of both solutions using the GPT-4 model (OpenAI), followed by simulations with a set of 27 Calculus I questions to evaluate each approach's performance. The tutor's interactions were recorded and analyzed for adherence to Socratic practices. The results showed that the RAG architecture with the few-shot learning technique achieved approximately 93% compliance with the expected Socratic style, whereas the RAG pipeline with one-shot achieved about 11%. Direct use of the LLM served as a comparative baseline, demonstrating that RAG combined with prompt engineering techniques provides an effective mechanism to regulate the LLM's behavior, enabling the intelligent tutor to reliably replicate Socratic teaching strategies. The study demonstrates the feasibility and advantages of this approach for the development of virtual tutors in education, although future empirical evaluations with users are necessary to confirm the observed pedagogical gains.

Keywords: Artificial Intelligence Tutor. Prompt Engineering. Large Language Models. Socratic Method. Retrieval-Augmented Generation.

LISTA DE ILUSTRAÇÕES

Figura 1 – O posicionamento do LLM dentro da Inteligência Artificial – Conforme esta representação hierárquica do relacionamento entre as diferentes áreas, Modelo de Linguagem de Grande Escala (<i>Large Language Model</i> s) (LLMs) representam uma aplicação específica de técnicas de <i>deep learning</i> , aproveitando sua capacidade de processar e gerar texto semelhante ao humano. <i>Deep learning</i> é um ramo especializado de <i>machine learning</i> que se concentra no uso de redes neurais de múltiplas camadas. E <i>machine learning</i> e <i>deep learning</i> são campos voltados para implementar algoritmos que permitem aos computadores aprender com dados e executar tarefas que normalmente requerem inteligência humana. . . .	6
Figura 2 – Arquitetura e fluxo de Grandes Modelos de Linguagem (LLMs): pré-treinamento em bilhões de <i>tokens</i> , camadas <i>Transformer</i> (incorporação, <i>self-attention</i> , saída), ajuste fino, parâmetros (dezenas a centenas de bilhões) e funcionando por padrões estatísticos.	7
Figura 3 – Árvore evolutiva dos LLMs modernos. Modelos no mesmo ramo têm relações mais próximas. Ramos em cores não-cinza indicam <i>Transformers</i> : ramo azul (<i>decoder-only</i>), rosa (<i>encoder-only</i>) e verde (<i>encoder-decoder</i>). A posição vertical reflete datas de lançamento; quadrados sólidos indicam código-aberto e vazados, código-fechado; gráfico de barras empilhadas mostra número de modelos por instituição.	9
Figura 4 – O Modelo RAG: primeiro recupera informações relevantes de uma fonte de dados indexada contendo conhecimento proprietário e, em seguida, usa essas informações recuperadas para gerar uma resposta mais informada e rica em contexto usando um modelo generativo. . . .	10
Figura 5 – Os Componentes do <i>Pipeline</i> RAG. Os fluxos em verde são todas as etapas de preparação dos dados para o buscador . Quando o usuário realiza uma consulta, a entrada do fluxo azul , sua requisição é aumentada com os materiais selecionados pelo buscador e pela engenharia de <i>prompt</i> configurada para o LLM gerador construir a resposta e enviar de volta para o usuário. A avaliação da qualidade desta resposta pode ser realizada após o recebimento da geração.	11
Figura 6 – Vetor Multidimensional no Banco de Dados.	12
Figura 7 – Sistema Modular Multidimodal RAG.	15
Figura 8 – <i>Pipeline</i> do <i>Reranking</i> e a geração aumentada.	16
Figura 9 – Indexação e Similaridade Vetorial.	18

Figura 10 – A FAISS é a preferida pelos usuários do GitHub. O Star History é uma ferramenta que permite visualizar o histórico de estrelas de projetos no GitHub – a ferramenta gera um gráfico que mostra a evolução das estrelas do projeto ao longo do tempo, permitindo analisar o crescimento e a popularidade do projeto. As bibliotecas analisadas e seus respectivos repositórios foram: em roxo está <code>nmslib/nmslib</code> ; em vermelho está <code>nmslib/hnswlib</code> ; em amarelo está <code>facebookresearch/faiss</code> ; em verde está <code>spotify/annoy</code> ; e em azul está <code>microsoft/SPTAG</code>	18
Figura 11 – Mapa de co-ocorrências de palavras-chaves utilizadas pela comunidade acadêmica.	27
Figura 12 – Fluxo: o uso direto do LLM através da API da OpenAI.	28
Figura 13 – Fluxo: uso do Recuperação Aumentada por Geração (<i>Retrieval-Augmented Generation</i>) (RAG) modulado pelo <i>framework</i> LangChain.	32
Figura 14 – Sistema: o fluxo de Login.	33
Figura 15 – Sistema: o fluxo do Gestor.	34
Figura 16 – Sistema: o fluxo do Aluno.	34
Figura 17 – Sistema: o fluxo do Professor.	35
Figura 18 – Sistema: o fluxo das Premissas.	36
Figura 19 – Gráfico da Análise Ternária de Precisão e Contextualização.	45
Figura 20 – Avatar Humano.	57
Figura 21 – Avatar LLM.	57
Figura 22 – Avatar RAG com <i>One-shot</i>	57
Figura 23 – Avatar RAG com <i>Few-shot</i>	57

LISTA DE TABELAS

Tabela 1 – Lista de Perguntas	42
Tabela 1 – Lista de Perguntas (continuação)	43
Tabela 2 – Análise Ternária com a classificação agregada das respostas do Tutor com Inteligência Artificial (<i>Artificial Intelligence Tutor</i>) (TIA)	44
Tabela 3 – Definições das classificações de respostas para os Tutores com Inteligên- cia Artificial	56
Tabela 4 – Definições das representações dos avatares para as interações com o TIA	57
Tabela 5 – Pergunta 1 e a classificação das respostas dos TIA	58
Tabela 5 – Pergunta 1 e a classificação das respostas dos TIA (continuação)	59
Tabela 6 – Pergunta 16 e a classificação das respostas dos TIA	60
Tabela 6 – Pergunta 16 e a classificação das respostas dos TIA (continuação)	61
Tabela 7 – Pergunta 25 e a classificação das respostas dos TIA	62
Tabela 7 – Pergunta 25 e a classificação das respostas dos TIA (continuação)	63
Tabela 8 – Pergunta 26 e a classificação das respostas dos TIA	64
Tabela 8 – Pergunta 26 e a classificação das respostas dos TIA (continuação)	65
Tabela 9 – Pergunta 27 e a classificação das respostas dos TIA	65

LISTA DE ABREVIATURAS E SIGLAS

AIED	Inteligência Artificial na Educação (<i>Artificial Intelligence in Education</i>) . . .	1
ANN	Vizinho Aproximado Mais Próximo (<i>Approximate Nearest Neighbor</i>)	17
API	Interface de Programação de Aplicações (<i>Application Programming Interface</i>)	17
BERT	Representações Codificadas de <i>Transformers</i> Bidirecionais (<i>Bidirectional Encoder Representations from Transformers</i>)	8
BM25	Função de Classificação BM25 (<i>Best Matching 25</i>)	13
CPU	Unidade Central de Processamento (<i>Central Processing Unit</i>)	19
ELMo	<i>Embeddings</i> de Modelos de Linguagem (<i>Embeddings from Language Models</i>)	8
FAISS	Biblioteca de Pesquisa de Similaridade de IA do Facebook (<i>Facebook AI Similarity Search</i>)	17
Flan	Aprendizado com Poucos Exemplos – " <i>few-shot</i> " – com Redes Adaptadas (<i>Fine-tuned Language Net</i>)	8
GenAI	Inteligência Artificial Generativa (<i>Generative AI</i>)	2
GPT	<i>Transformer</i> Pré-treinado Generativo (<i>Generative Pre-trained Transformer</i>)	17
GPU	Unidade de Processamento Gráfico (<i>Graphics Processing Unit</i>)	19
HNSW	Mundo Pequeno Navegável Hierárquico (<i>Hierarchical Navigable Small World</i>)	17

HTTP Protocolo de Transferência de Hipertexto (<i>Hypertext Transfer Protocol</i>)	29
IA Inteligência Artificial (<i>Artificial Intelligence</i>)	1
IDF Frequência Inversa de Documentos (<i>Inverse Document Frequency</i>)	13
ITS Sistema de Tutoria Inteligente (<i>Intelligent Tutoring System</i>)	23
IVF Arquivo Invertido (<i>Inverted File</i>)	17
JSON Notação de Objeto JavaScript (<i>JavaScript Object Notation</i>)	29
KS Teste de Kolmogorov–Smirnov (<i>Kolmogorov–Smirnov Test</i>)	24
LLaMA Modelos de Linguagem de Código Aberto da Meta (<i>Large Language Model Meta AI</i>)	8
LaMDA Modelo de Linguagem para Aplicações de Diálogo (<i>Language Model for Dialogue Applications</i>)	8
LLM Modelo de Linguagem de Grande Escala (<i>Large Language Model</i>)	9
LMS Sistema de Gestão de Aprendizagem (<i>Learning Management System</i>)	50
LSH <i>Hashing</i> Sensível à Localidade (<i>Locality-Sensitive Hashing</i>)	19
LSTM Memória de Longo e Curto Prazo (<i>Long Short-Term Memory</i>)	8
MLP Perceptron Multicamadas (<i>Multilayer Perceptron</i>)	8
MMR Relevância Marginal Máxima (<i>Maximum Marginal Relevance</i>)	17

MOOC Curso Online Aberto e Massivo (<i>Massive Open Online Course</i>)	23
N/A Não Aplicável	44
NLP Processamento de Linguagem Natural (<i>Natural Language Processing</i>)	8
P/C Precisa e Contextualizada	44
P/NC Precisa e Não Contextualizada	44
PaLM Modelo de Linguagem <i>Pathways</i> (<i>Pathways Language Model</i>)	8
PQ Quantização de Produto (<i>Product Quantization</i>)	19
PyTutor Sistema de Tutoria Inteligente PyTutor (<i>Python Tutor</i>)	23
RAG Recuperação Aumentada por Geração (<i>Retrieval-Augmented Generation</i>)	10
RAGMan Gerenciador de Recuperação Aumentada por Geração (<i>Retrieval-Augmented Generation Manager</i>)	24
RNN Rede Neural Recorrente (<i>Recurrent Neural Network</i>)	8
SDK Kit de Desenvolvimento de Software (<i>Software Development Kit</i>)	15
TCC Trabalho de Conclusão de Curso	3
TIA Tutor com Inteligência Artificial (<i>Artificial Intelligence Tutor</i>)	11
TF Frequência de Termos (<i>Term Frequency</i>)	13

USP Universidade de São Paulo	4
word2vec Word-to-Vector	8

LISTA DE SÍMBOLOS

λ Lambda

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Objetivos	4
2	FUNDAMENTOS TEÓRICOS	6
2.1	Grandes Modelos de Linguagem (LLMs)	6
2.1.1	Ascensão dos LLMs	8
2.2	Recuperação Aumentada por Geração (RAG)	9
2.2.1	<i>Pipeline</i> de Funcionamento do RAG	10
2.2.1.1	Geração de <i>embeddings</i>	11
2.2.1.2	Recuperação semântica (similaridade vetorial)	12
2.2.1.3	Modalidades de recuperação	13
2.2.1.4	Síntese de resposta	13
2.2.1.5	Vantagens do RAG	13
2.2.2	O <i>framework</i> LangChain	14
2.2.2.1	O <i>Reranking</i> e o Relevância Marginal Máxima (<i>Maximum Marginal Relevance</i>) (MMR) (<i>Maximum Marginal Relevance</i>)	16
2.2.3	<i>Vector Store</i> e o Biblioteca de Pesquisa de Similaridade de IA do Facebook (<i>Facebook AI Similarity Search</i>) (FAISS)	17
2.2.3.1	Definição de <i>Vector Store</i>	17
2.2.3.2	O FAISS	19
2.3	Engenharia de <i>Prompt</i>	19
2.3.1	<i>Few-shot Prompting</i>	21
3	TRABALHOS RELACIONADOS	23
3.1	Revisão das Soluções Existentes	23
3.2	Indexação de Lacunas na Literatura	25
4	METODOLOGIA	26
4.1	Processo de Desenvolvimento	26
4.2	Arquiteturas, Design e Configuração do Sistema	27
4.2.1	Abordagem direta com o LLMs (Interface de Programação de Aplicações (<i>Application Programming Interface</i>)s (APIs) OpenAI com o modelo <i>Transformer</i> Pré-treinado Gerativo (<i>Generative Pre-trained Transformer</i>)s (GPTs)-4 Turbo)	28
4.2.2	Abordagem com RAG modulado pelo <i>framework</i> LangChain	29
4.2.2.1	A Fase de Configuração do RAG	29

4.2.2.2	A Fase de Interação	30
4.3	Modelagem e Integração Tecnológica para o TIA	32
4.3.1	Modelagem do Sistema: Tutor com Inteligência Artificial	33
4.3.2	Integração Tecnológica entre o Streamlit, o LLM e o RAG	35
4.3.2.1	Configuração da API Direta do LLM	36
4.3.2.2	Configuração do RAG	37
5	RESULTADOS E AVALIAÇÃO	41
5.1	Conjunto de Perguntas para Experimentos	41
5.2	Avaliação dos Resultados	43
5.2.1	Métricas: Precisão e Contextualização	43
5.2.2	Avaliação Ternária	44
5.3	Visualização Gráfica Comparativa	45
6	DISCUSSÃO	47
7	CONCLUSÃO	49
	REFERÊNCIAS	51
	APÊNDICES	54
	A – OS CÓDIGOS DO TUTOR COM INTELIGÊNCIA ARTIFICIAL	55
	APÊNDICE B – LISTA DE PERGUNTAS E RESPOSTAS DOS TUTORES INTELIGENTES	56

1 INTRODUÇÃO

Segundo Lin, Luo e Qian (2023), a educação configura-se como um processo multifacetado que envolve a promoção de conhecimento, habilidades, valores, crenças e hábitos por meio do ensino, treinamento e pesquisa. Os autores também destacam que as tecnologias de Inteligência Artificial na Educação (*Artificial Intelligence in Education*) (AIED) têm origens que remontam à década de 1950 e que seu papel é preparar os alunos para os desafios futuros, potencializando sua inovação, competitividade e promovendo a aprendizagem ao longo da vida e o autodesenvolvimento¹.

Esta perspectiva reforça a ideia de que o aprimoramento do processo educativo passa pela integração equilibrada entre inovações tecnológicas e práticas pedagógicas tradicionais, contribuindo para uma abordagem mais eficaz e adaptativa.

Bahroun *et al.* (2023) demonstram que, atualmente, a Inteligência Artificial (*Artificial Intelligence*) (IA) se configura como uma força revolucionária, impulsionada pelos avanços tecnológicos que redefinem a existência humana. Os autores ressaltam seu papel na melhoria dos resultados de aprendizagem, motivação e engajamento dos alunos por meio de ferramentas como chatbots de IA e instrutores virtuais, além de enfatizarem a necessidade de aprimorar o design desses sistemas, explorar alternativas de avaliação e aprofundar a análise dos impactos a longo prazo na educação².

Os estudos se alinham e complementam entre a visão histórica da educação e as inovações tecnológicas recentes, reforçando a importância de incorporar práticas que se adaptem às necessidades individuais do ensino.

Kamalov *et al.* (2023) evidenciam que a inteligência artificial tem transformado o cenário educacional por meio da personalização das experiências de aprendizagem, da automação de responsabilidades administrativas e da oferta de *feedback* em tempo real, contribuindo para a criação de ambientes mais inclusivos e eficazes. Segundo os autores, essa transformação se fundamenta na utilização de algoritmos adaptativos que ajustam o conteúdo às necessidades específicas dos alunos, alterando as dinâmicas de ensino, aprendizagem e funcionamento institucional. Além disso, enfatizam que, embora a IA apresente possibilidades tentadoras para reduzir as lacunas históricas na educação, torna-se imprescindível a implementação de mecanismos robustos para prevenir seu uso indevido³.

Os trabalhos de Lin, Luo e Qian (2023) evidenciam que, apesar dos avanços na inteligência artificial, persistem desafios como a limitação no processamento de dados não estruturados e a necessidade de interações mais eficazes entre humanos e IA. Segundo os autores, o desenvolvimento de algoritmos mais eficientes, aliado a uma avaliação abrangente

dos sistemas de IA, é fundamental para aprimorar a personalização do ensino e aproveitar integralmente o potencial da tecnologia, ao mesmo tempo em que se previnem usos indevidos.¹

Também se faz necessário o uso de estratégias coercitivas que visem a integridade acadêmica na utilização da IA, a fim de evitar fraudes educacionais e mitigar problemas éticos gerados pelo uso da IA na educação. Conforme exposto por Kamalov *et al.* (2023):

“A possibilidade de a IA ser mal utilizada para a desonestidade acadêmica ressalta ainda mais a necessidade de uma regulação cuidadosa e de supervisão ética.”^{3,4}

Complementando nestas precauções, os estudos de Bahroun *et al.* (2023)³ ressaltam que, para que a integração da inteligência artificial em ambientes educacionais seja efetiva e sustentável, é imprescindível a adoção de considerações éticas, a promoção da colaboração interdisciplinar e o uso responsável da tecnologia. Os autores afirmam que o desenvolvimento e a implementação de modelos de Inteligência Artificial Generativa (*Generative AI*) (*GenAI*) transparentes são fundamentais para mitigar os vieses inerentes e assegurar a utilização responsável da IA. Ademais, defendem que a colaboração entre educadores, cientistas da computação e formuladores de políticas é necessária para reduzir a distância entre a inovação tecnológica e as práticas pedagógicas, e enfatizam o estabelecimento de diretrizes claras para mitigar riscos, tais como vazamentos de dados, discriminação algorítmica e o uso indevido de conteúdos gerados por IA.³

Em uma tentativa da promoção do pensamento crítico no processo de aprendizagem, Favero *et al.* (2024)⁵ afirmam:

“O questionamento socrático faz parte de estruturas de suportes na aprendizagem como um processo em que o aprendiz alcança um objetivo por meio de esforços guiados.”^{5,4}

Favero *et al.* (2024) afirmam que a combinação em grandes modelos de linguagens (*LLMs*) locais por meio de *fine-tuning*, ajustes finos em seus parâmetros, e *prompt-tuning*, instruções de entrada personalizadas, combinados com perguntas socráticas estruturadas, estimula a reflexão e o pensamento crítico de maneira significativamente superior às abordagens tradicionais. Essa eficácia é comprovada por indicadores de desempenho e por um índice de pensamento crítico normalizado, além de garantir acesso democrático à ferramenta e preservar a privacidade no processo educativo.⁵

Entretanto, Favero *et al.* (2024) ressaltam que a otimização do desempenho do chatbot é fundamental para garantir operação eficiente em máquinas locais, sem consumo excessivo de recursos. Avisam que uma investigação mais aprofundada de melhorias no

modelo Llama3 8B, de oito bilhões de parâmetros (ou em arquiteturas de complexidade similar), ficou reservada para trabalhos futuros. Defendem que as avaliações simuladas são consideradas indispensáveis como etapa preliminar para aferir funcionalidade, eficiência e segurança do sistema, ainda que as avaliações empíricas com estudantes humanos sejam o ideal. Destacam, por fim, que em apenas cinco turnos de interação em diálogos foram suficientes para demonstrar que o tutor socrático supera significativamente tanto o tutor básico quanto o aleatório no desenvolvimento do pensamento crítico⁵.

Os *chatbots* de Favero *et al.* (2024) são para um questionamento socrático de distintas finalidades: clarificação; investigação de pressupostos; investigação de razões e evidências; investigação de implicações e consequências; investigação de pontos de vista e perspectivas alternativas; e outros. No cerne desse método (socrático), o estudo de Benson (2009)⁶ destaca que, nos diálogos de Platão, Sócrates adota o *elenchos*: um processo filosófico em que perguntas estruturadas seguem uma estratégia que leva o interlocutor a reconhecer contradições em seu próprio pensamento e partem de uma pressuposição epistemológica (isto é, uma crença sobre a natureza e os limites do conhecimento). Ao examinar a sabedoria atribuída aos seus ouvintes, esse procedimento busca revelar-lhes sua ignorância ou aprender com aqueles que de fato demonstram saber.

Diante do contexto educacional, segundo Neville (1999), torna-se imprescindível revisar o papel do tutor em ambientes de aprendizagem baseada em problemas e aprimorar competências do professor de facilitar este processo, privilegiando a autonomia dos alunos; empregando as palavras de Barrows:

“Barrows afirmou que a tarefa do tutor em um grupo de tutorial baseado em problemas deve ser facilitar a aprendizagem dos alunos em vez de transmitir conhecimento.”^{7,4}

(Barrows & Tamblyn, 1980, *apud* Neville, 1999)

Sendo preciso ao professor calibrar a diretividade, intervir para corrigir erros conceituais, adaptar o estilo do tutor ao nível dos estudantes e lidar com as evidências limitadas de sua eficácia como avaliador, privilegiando a aprendizagem autônoma⁷. (Neville, 1999)

Neste cenário, emerge o problema de pesquisa central deste Trabalho de Conclusão de Curso (TCC): **Como a arquitetura de implementação – especificamente, a interação direta com o LLMs (OpenAI gpt-4-turbo) versus RAG, através de *prompt-tuning*, *prompts* diretivos que visam estimular o pensamento crítico – afeta a capacidade de um Tutor com Inteligência Artificial (TIA) em adotar uma contextualização de postura pedagógica socrática no domínio de Cálculo I?** Neste trabalho, entende-se por abordagem socrática a estratégia pedagógica que, por meio de

perguntas guiadas e provocações, estimula o aluno a descobrir o conhecimento e facilita seu processo de aprendizagem sem transmitir respostas prontas. Essa prática é especialmente relevante para o desenvolvimento do pensamento matemático e da autonomia do aluno, pois esclarece a essência dos conteúdos apresentados pelo professor. Cabe ao docente monitorar o desempenho do Tutor Inteligente e intervir quando necessário, garantindo a eficácia da aprendizagem autônoma mesmo diante de suas limitações.

A justificativa para a investigação é relevante porque aborda a integração de Inteligência Artificial Generativa (**GenAI**) com uma regulação pré-estabelecida e configurável de um *framework* **RAG** para o reforço escolar. A proposta de uma **IA** configurável e adaptável, como um **TIA**, é reforçada pela necessidade da regulação de seu funcionamento, capaz de reduzir a distância da inovação tecnológica e as práticas pedagógicas, a fim de evitar fraudes acadêmicas e mitigar problemas éticos na educação. A escolha de concentrar a atuação deste Tutor com Inteligência Artificial (**TIA**) na disciplina de Cálculo I baseou-se na disponibilização do livro “Cálculo é Fácil”, do Prof. Dr. Walter Ferreira Velloso Junior, no portal de livros abertos da Universidade de São Paulo (**USP**)⁸.

1.1 Objetivos

Diante do exposto, definem-se os seguintes objetivos:

Objetivo Geral:

- Apontar a configuração da **IA** como ferramenta de auxílio no processo de aprendizagem através de um Tutor Inteligente integrado a um grande modelo de linguagem (**LLM**) complementado por um mecanismo de recuperação aumentada de informações (**RAG**) para garantir respostas precisas e contextualizadas.

Objetivos Específicos:

- Explorar a configuração para a personalização da aprendizagem a partir de dados não-estruturados.
- Descrever as configurações técnicas e requisitos funcionais necessários para implementar as duas abordagens do Tutor com Inteligência Artificial (**TIA**) para Cálculo I.
- Avaliar o potencial do **TIA** como ferramenta para promover interações pedagógicas socráticas no ensino de Cálculo I.
- Apontar o **TIA** como uma solução teórica para futuras aplicações educacionais.

O trabalho está organizado da seguinte maneira: a Seção 2, detalha a teoria das tecnologias empregadas; na Seção 3, faz-se uma revisão crítica dos estudos relacionados; a Seção 4 detalha a metodologia adotada, incluindo todas as configurações técnicas utilizadas; na Seção 5, apresentam-se os resultados das simulações e sua avaliação comparativa; a Seção 6 debate as implicações desses achados; e, por fim, a Seção 7 sintetiza os principais resultados e aponta possíveis direções para pesquisas futuras. Nos Apêndices, encontram-se o código-fonte da aplicação e as respostas simuladas empregadas na avaliação.

2 FUNDAMENTOS TEÓRICOS

Esta seção apresenta os fundamentos teóricos necessários para compreender as tecnologias e técnicas aplicadas neste trabalho, incluindo os conceitos principais relacionados a Grandes Modelos de Linguagem (LLMs), Recuperação Aumentada por Geração (RAG), e Engenharia de *Prompt*. Esses conceitos são cruciais para o entendimento e implementação das abordagens exploradas no presente estudo.

2.1 Grandes Modelos de Linguagem (LLMs)

Segundo Raschka (2024), um LLM é uma rede neural profunda capaz de "entender" e gerar texto de forma semelhante a um humano. Treinado em bilhões de *tokens* (palavras ou subpalavras), ele aprende padrões de linguagem, contexto e nuances semânticas ao prever a próxima palavra em enormes conjuntos de dados⁹.

A Figura 1 ilustra o posicionamento do LLM dentro da Inteligência Artificial.

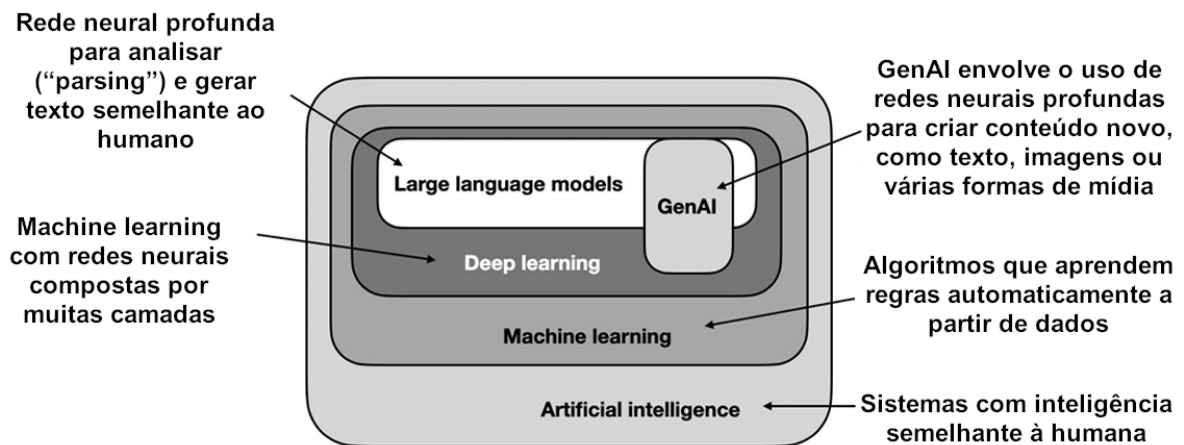


Figura 1 – O posicionamento do LLM dentro da Inteligência Artificial – Conforme esta representação hierárquica do relacionamento entre as diferentes áreas, LLMs representam uma aplicação específica de técnicas de *deep learning*, aproveitando sua capacidade de processar e gerar texto semelhante ao humano. *Deep learning* é um ramo especializado de *machine learning* que se concentra no uso de redes neurais de múltiplas camadas. E *machine learning* e *deep learning* são campos voltados para implementar algoritmos que permitem aos computadores aprender com dados e executar tarefas que normalmente requerem inteligência humana.

Adaptado de: Raschka (2024).

Raschka (2024) complementa que o “grande” (“*large*”) refere-se tanto ao tamanho do modelo — com dezenas ou até centenas de bilhões de parâmetros ajustáveis — quanto à quantidade de dados usados no pré-treinamento, baseado na tarefa de predição da próxima palavra, explora a natureza sequencial da linguagem para aprender representações contextuais profundas. A base dessa capacidade é a arquitetura *Transformer*, cujo mecanismo de atenção (“*self-attention*”) permite ao modelo concentrar-se seletivamente em diferentes partes do texto de entrada ao gerar cada novo *token*, proporcionando melhor escalabilidade e capacidade de capturar dependências de longo alcance no texto.⁹

Após absorver um conhecimento linguístico genérico no pré-treino, Raschka (2024) conclui, o LLM passa por *fine-tuning* em bases menores e rotuladas, ajustando-se a tarefas específicas (classificação, tradução, chats etc.). Embora produza texto coerente e relevante, ele opera por padrões estatísticos e não possui consciência ou compreensão humana.⁹ A Figura 2 ilustra os LLMs.



Figura 2 – Arquitetura e fluxo de Grandes Modelos de Linguagem (LLMs): pré-treinamento em bilhões de *tokens*, camadas *Transformer* (incorporação, *self-attention*, saída), ajuste fino, parâmetros (dezenas a centenas de bilhões) e funcionando por padrões estatísticos.

Fonte: Ferreira, B.C., 2025.

2.1.1 Ascensão dos LLMs

Subramanian (2024) contextualiza que os primeiros modelos de linguagem eram sistemas de tradução estatísticos baseados em regras fixas e n-gramas que estimavam a próxima palavra, mas eram limitados por sua rigidez simbólica. Com o avanço do *machine learning* e do *deep learning*, redes neurais profundas — como Perceptron Multicamadas (*Multilayer Perceptron*) (MLP), Rede Neural Recorrente (*Recurrent Neural Network*) (RNN) e Memória de Longo e Curto Prazo (*Long Short-Term Memory*) (LSTM) — passaram a aprender representações distribuídas de texto. Ainda assim, esses modelos dependiam de enormes volumes de dados rotulados e enfrentavam dificuldade em capturar relações de longo alcance. Para superar essas limitações, a partir de 2013 surgiram métodos de pré-treinamento em grande escala, exemplificados pelo Word-to-Vector (*word2vec*), demonstraram que *embeddings* extraídos de texto bruto podiam ser reutilizados de forma geral em diversas tarefas de Processamento de Linguagem Natural (*Natural Language Processing*) (NLP)¹⁰.

Subramanian (2024)¹⁰ complementa que em 2018, a arquitetura *Transformer* e o paradigma “pré-treinar e ajustar” se consolidaram com *Embeddings* de Modelos de Linguagem (*Embeddings from Language Models*) (ELMo) e, principalmente, Representações Codificadas de *Transformers* Bidirecionais (*Bidirectional Encoder Representations from Transformers*) (BERT), que introduziu atenção bidirecional e *masked language modeling*, alcançando ganhos expressivos em tarefas de compreensão de linguagem. Logo depois, modelos cada vez maiores – GPT-1, GPT-2 e, sobretudo, GPT-3, com 175 bilhões de parâmetros – demonstraram que o simples aumento de escala desbloqueia habilidades emergentes, como o *zero-shot*, raciocínio sem instrução prévia, e o *few-shot learning*, raciocínio em cadeia e uso multimodal, sem necessidade de ajustes adicionais, o que evidenciou o poder da escala em modelos de linguagem e desencadeou intenso interesse em arquiteturas ainda maiores.¹⁰

Para concluir, Subramanian (2024) afirma que esse impulso levou à exploração de LLMs na casa dos trilhões de parâmetros, com organizações como Anthropic e iniciativas públicas como Modelo de Linguagem *Pathways* (*Pathways Language Model*) (PaLM), Gopher e Modelos de Linguagem de Código Aberto da Meta (*Large Language Model Meta AI*) (LLaMA) ampliando drasticamente o porte dos modelos. Hoje, diversas empresas e laboratórios treinam suas próprias soluções com centenas de bilhões — e até trilhões — de parâmetros, e variantes como PaLM, Aprendizado com Poucos Exemplos – “*few-shot*” – com Redes Adaptadas (*Fine-tuned LAnguage Net*) (Flan), Modelo de Linguagem para Aplicações de Diálogo (*Language Model for Dialogue Applications*) (LaMDA) e LLaMA demonstram a escalabilidade do paradigma *Transformer*. Na época desta escrita, a Anthropic havia lançado o Claude, primeiro LLM de código aberto com habilidades conversacionais comparáveis ao GPT-3¹⁰. Na Figura 3 – Árvore Evolutiva dos LLMs

Modernos – observa-se que modelos situados no mesmo ramo têm afinidades mais fortes entre si. Os ramos coloridos (não-cinza) destacam as diferentes configurações *Transformer*: o azul identifica os apenas-decodificador, o rosa os apenas-codificador e o verde os codificador-decodificador. A posição vertical de cada nó sinaliza a data de lançamento, enquanto quadrados preenchidos representam código aberto e os vazados, código fechado. Por fim, um gráfico de barras empilhadas mostra quantos modelos cada instituição desenvolveu¹⁰.

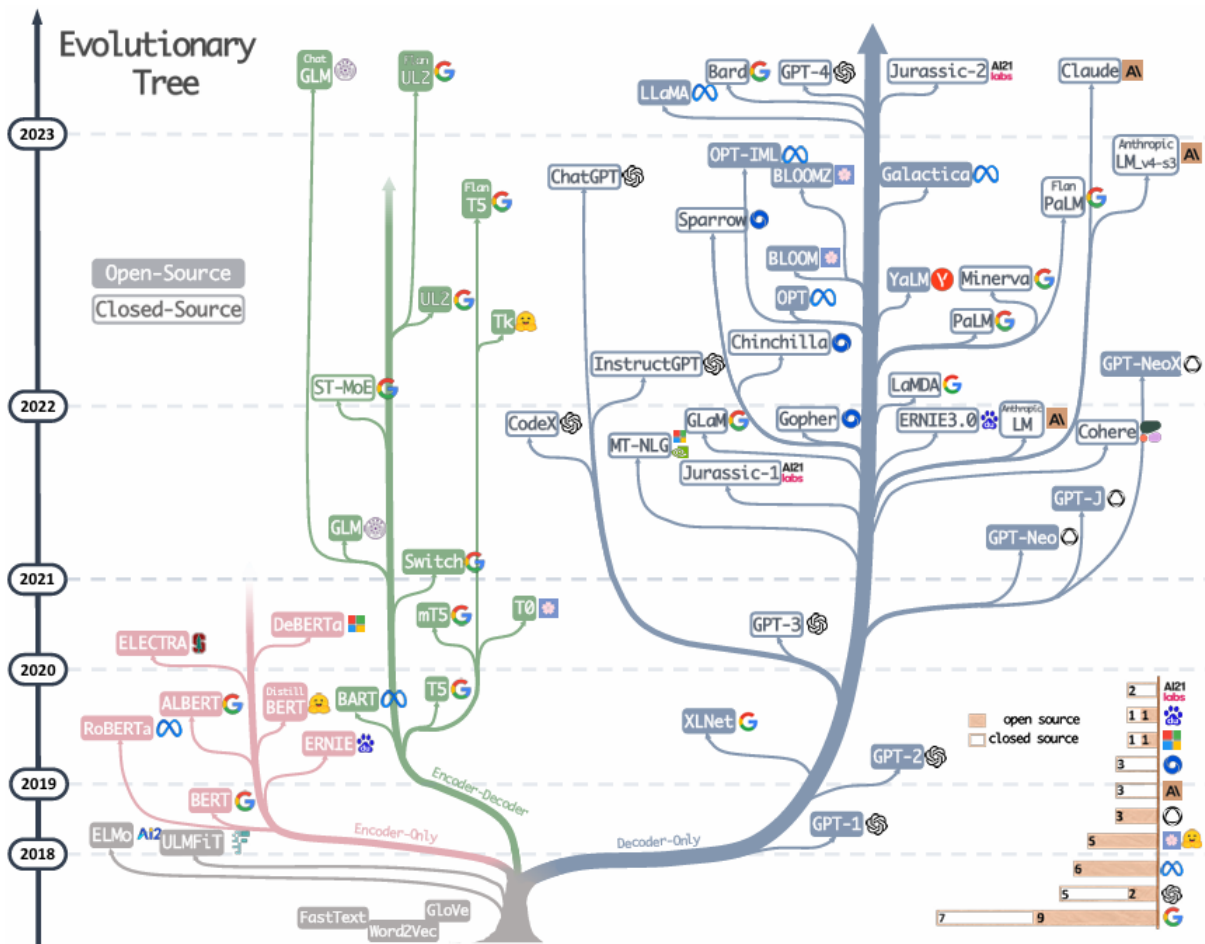


Figura 3 – Árvore evolutiva dos LLMs modernos. Modelos no mesmo ramo têm relações mais próximas. Ramos em cores não-cinza indicam *Transformers*: ramo azul (*decoder-only*), rosa (*encoder-only*) e verde (*encoder-decoder*). A posição vertical reflete datas de lançamento; quadrados sólidos indicam código-aberto e vazados, código-fechado; gráfico de barras empilhadas mostra número de modelos por instituição.

Adaptado de Yang, J. et al. (2023) apud SUBRAMANIAN (2024).

2.2 Recuperação Aumentada por Geração (RAG)

De acordo com Rothman (2024), *Retrieval-Augmented Generation (RAG)* é um *framework* que resolve a limitação dos modelos de IA generativa ao combinar abordagens

de recuperação de informação com modelos generativos, permitindo buscar dados em fontes externas em tempo real e utilizá-los para gerar respostas mais precisas e contextualmente relevantes, reduzindo alucinações e vieses; além disso, é altamente adaptável, podendo ser aplicado a qualquer tipo de dado (texto, imagem ou *áudio*)¹².

A técnica de Recuperação Aumentada por Geração (**RAG**) combina a recuperação de informações com a capacidade de geração textual dos modelos **LLMs**. **RAG** melhora significativamente a precisão e relevância das respostas geradas ao permitir que o modelo consulte bases externas de informações antes de gerar uma resposta, reduzindo a ocorrência de “alucinações” (Lewis *et al.*, 2020)¹³.

Gheorghiu (2024) complementa as afirmações de Lewis *et al.* (2020) sobre *Retrieval-Augmented Generation* (**RAG**) ao expor que em vez de confiar exclusivamente no conhecimento estático embutido em um **LLM**, o **RAG** primeiro realiza uma busca em uma base de dados indexada para extrair trechos relevantes ao contexto da pergunta e, em seguida, injeta esses fragmentos no *prompt* de um modelo generativo, produzindo respostas mais precisas e contextualizadas¹⁴. A Figura 4 exibe o modelo exposto por Gheorghiu (2024).

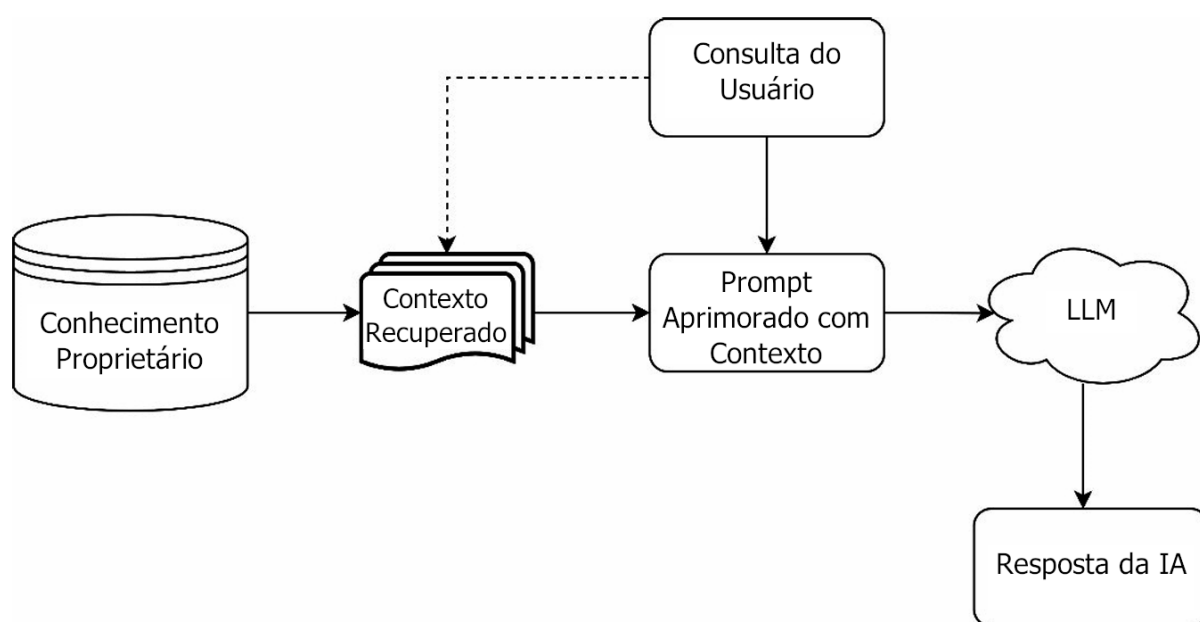


Figura 4 – O Modelo RAG: primeiro recupera informações relevantes de uma fonte de dados indexada contendo conhecimento proprietário e, em seguida, usa essas informações recuperadas para gerar uma resposta mais informada e rica em contexto usando um modelo generativo.

Adaptado de Gheorghiu (2024).

2.2.1 Pipeline de Funcionamento do RAG

O *pipeline* de funcionamento de um sistema **RAG**, detalhado na Figura 5, pode ser decomposto em três fases principais: *embeddings* (presentes na preparação), recuperação e

síntese de resposta.

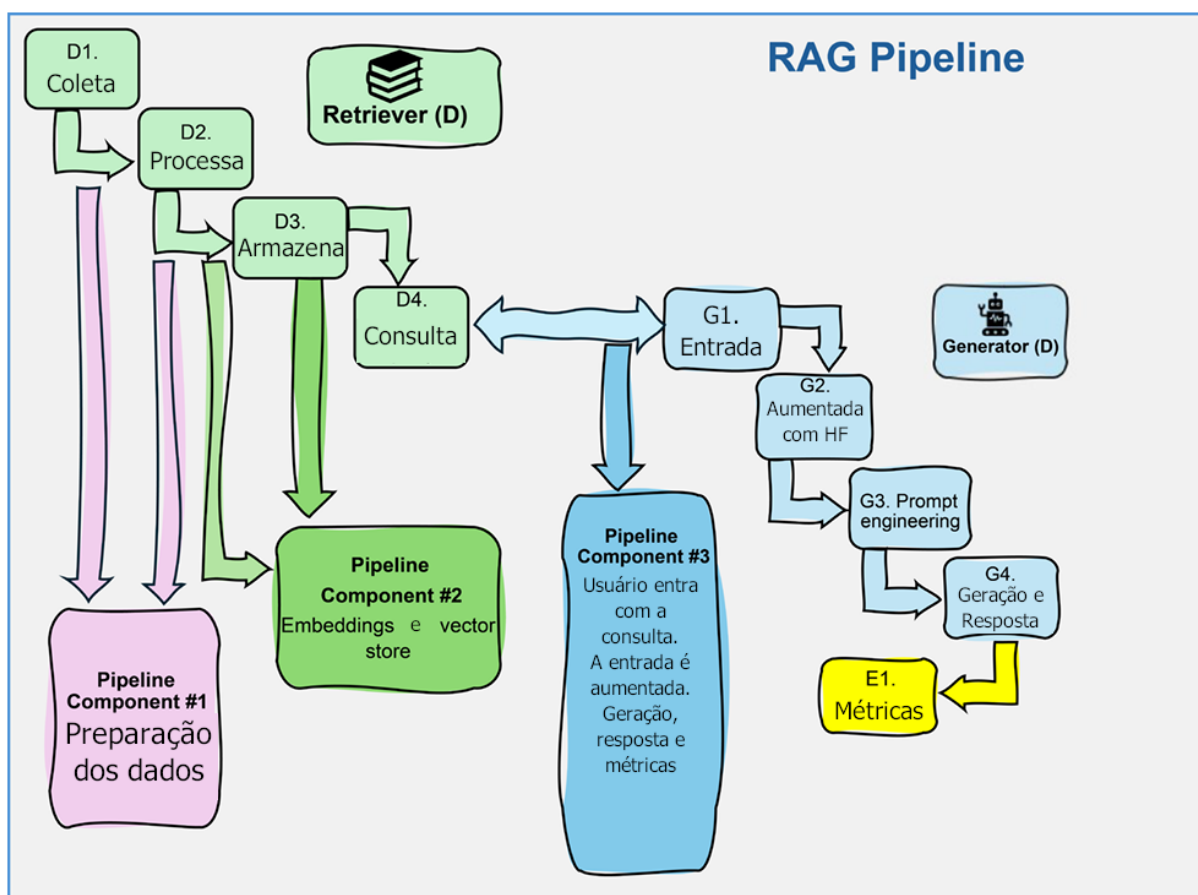


Figura 5 – Os Componentes do *Pipeline* RAG. Os fluxos em **verde** são todas as etapas de preparação dos dados para o buscador. Quando o usuário realiza uma consulta, a entrada do fluxo **azul**, sua requisição é aumentada com os materiais selecionados pelo buscador e pela engenharia de *prompt* configurada para o **LLM** gerador construir a resposta e enviar de volta para o usuário. A avaliação da qualidade desta resposta pode ser realizada após o recebimento da geração.

Adaptado de Rothman (2024).

2.2.1.1 Geração de *embeddings*

Segundo Gheorghiu (2024), ao indexar documentos, o **RAG** divide o texto em “nós” (nodes) — fragmentos de tamanho controlado — que são então transformados em vetores de alta dimensão por meio de um modelo de *embedding*. Esses vetores capturam a semântica do texto em forma numérica, permitindo que o sistema “entenda” e compare diferentes trechos de forma matemática¹⁴.

Gheorghiu (2024) explica que *embeddings* são listas de números de ponto flutuante que representam palavras, frases ou documentos em um espaço vetorial contínuo. Eles funcionam como uma “linguagem de pensamento” para o **LLM**, uniformizando dados diversos em um formato que o modelo consegue processar de forma eficaz¹⁴.

A Figura 6 ilustra um vetor multidimensional no banco de dados, em que cada palavra representa uma dimensão específica, onde palavras semelhantes estão mais próximas umas das outras no espaço tridimensional (pois carregam sentido similar), enquanto outras se distanciam, refletindo seu significado distinto^{14,16}.

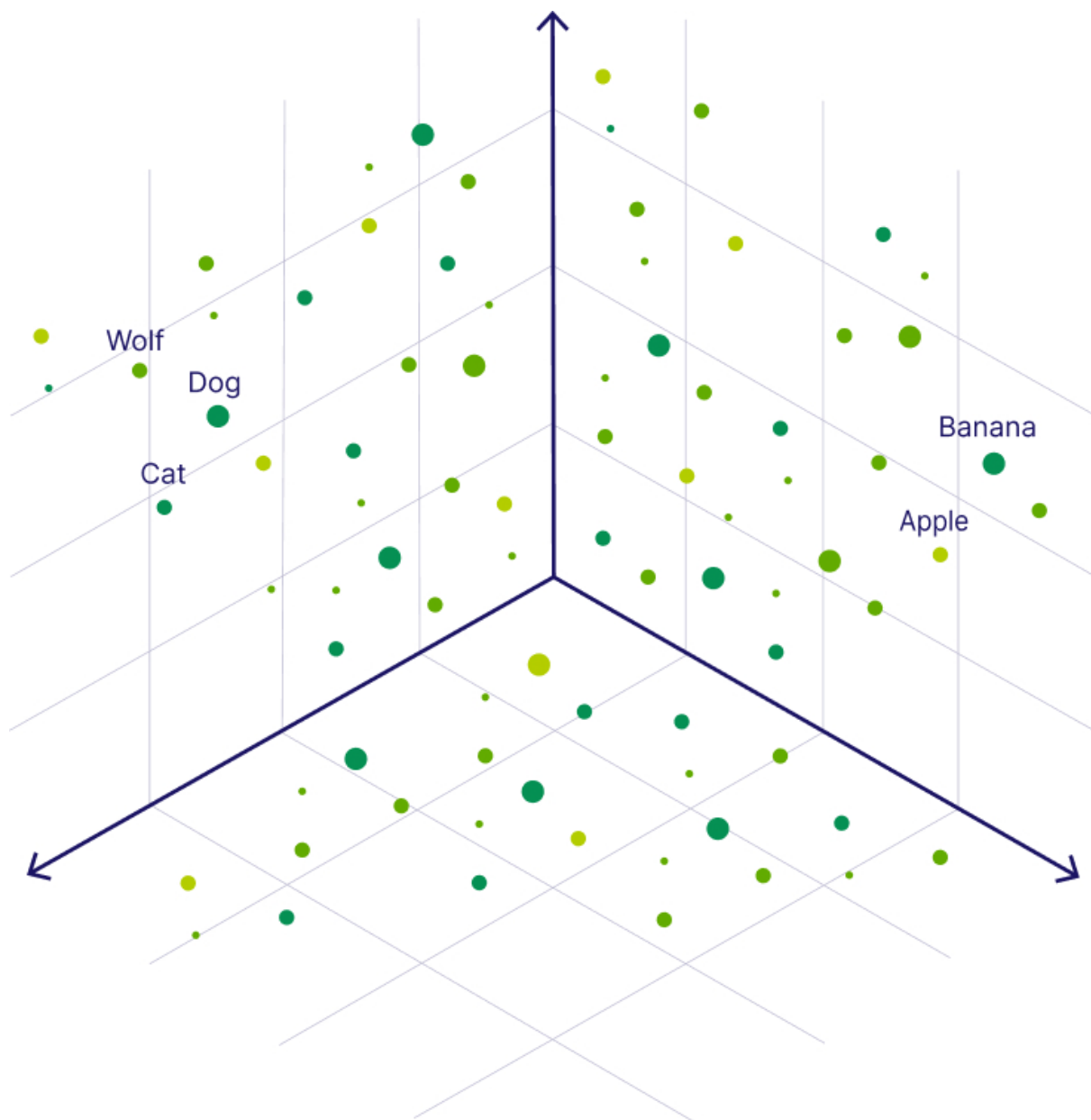


Figura 6 – Vetor Multidimensional no Banco de Dados.

Fonte: Elsayed, Mustafa. Multi dimensional vector db. *Medium*, 2024. Disponível em: <https://medium.com/@mustafaskyer/rag-with-langchain-js-openai-and-pinecone-3755c65d5052>

2.2.1.2 Recuperação semântica (similaridade vetorial)

Gheorghiu (2024) aponta que quando chega uma consulta, o sistema a converte em um vetor de *embedding* da mesma forma. Em seguida, calcula-se a semelhança —

geralmente por meio da distância do cosseno — entre o vetor da consulta e os vetores armazenados. Os nós com maior similaridade são selecionados como contexto relevante para a geração de resposta¹⁴.

2.2.1.3 Modalidades de recuperação

Como relatado por Gheorghiu (2024), a recuperação converte a consulta em um vetor de busca ou utiliza filtros semânticos para selecionar documentos ou “nós” (nodes) de uma base de conhecimento¹⁴. Ela pode ser usada nas seguintes abordagens:

- **Recuperação densa (*dense retrieval*):** baseia-se inteiramente em *embeddings*, usando vetores “densos” (com muitos valores não nulos) para capturar nuances semânticas. A busca dinamicamente embute a consulta e retorna os *top-k* vetores mais semelhantes, permitindo encontrar sinônimos e relações de sentido sem depender de correspondência exata de palavras¹⁴.
- **Recuperação esparsa (*sparse retrieval*):** utiliza índices invertidos de termos (*keywords*). Cada documento é indexado por palavras-chave, e a consulta pesquisa esses índices para encontrar trechos que compartilhem termos exatos, frequentemente ponderados por Frequência de Termos (*Term Frequency*) (**TF**)–Frequência Inversa de Documentos (*Inverse Document Frequency*) (**IDF**) ou algoritmos como a Função de Classificação BM25 (*Best Matching 25*) (**BM25**). Este método é mais eficiente quando buscas de palavras-chave precisas são desejadas¹⁴.
- **Abordagem híbrida:** combina denso e esparsa, roteando partes da consulta para cada método ou fundindo seus resultados, de modo a equilibrar cobertura semântica e precisão de termos exatos¹⁴.

2.2.1.4 Síntese de resposta

Conforme Gheorghiu (2024), os trechos retornados pela recuperação são então injetados no *prompt* do **LLM**. Esse modelo utiliza tanto seu conhecimento pré-treinado quanto o contexto recuperado para gerar uma resposta fluida, fundamentada nos documentos consultados¹⁴.

2.2.1.5 Vantagens do **RAG**

Gheorghiu (2024) defende que a adoção do **RAG** traz diversos benefícios:

- **Retenção aprimorada de fatos:** ao referenciar diretamente documentos proprietários ou bases de conhecimento especializadas, o modelo apoia suas respostas em evidências concretas, aumentando muito a fidelidade das informações geradas¹⁴.

- **Raciocínio mais coerente:** a etapa de recuperação prévia permite ao LLM extrair trechos estritamente relacionados à pergunta, o que tende a melhorar a lógica interna das respostas e minimizar conclusões desconexas¹⁴.
- **Relevância de contexto:** como o RAG busca exatamente o conteúdo que responde ao questionamento, as respostas ficam mais alinhadas ao domínio específico do usuário, superando a limitação de modelos que dependem apenas de dados de treinamento genéricos¹⁴.
- **Redução de “alucinações”:** a abordagem híbrida diminui a chance de o modelo gerar informações falsas ou sem sentido (“nonsense”), já que fundamenta suas saídas em fragmentos recuperados de fontes confiáveis¹⁴.
- **Transparência e validação:** é possível rastrear exatamente quais documentos ou trechos foram utilizados na geração de cada resposta, o que facilita auditorias, revisões e a garantia de confiança nos resultados¹⁴.

Mas Gheorghiu (2024) ressalta que apesar das vantagens, o RAG não elimina por completo as limitações dos LLMs. Ainda é recomendável validar e, quando necessário, ajustar as respostas geradas, pois o modelo pode, ocasionalmente, apresentar inconsistências ou lacunas que demandam supervisão humana¹⁴.

Neste trabalho, o *framework* LangChain — detalhado a seguir — emprega a abordagem RAG ao utilizar *embeddings* para consultar documentos externos que fornecem contexto específico para as respostas do sistema (LANGCHAIN, 2025)¹⁷, como o Tutor Inteligente.

2.2.2 O *framework* LangChain

Segundo Rothman (2024), o LangChain aparece como um dos principais *frameworks* de orquestração dentro do ecossistema RAG. Classificando-o como uma ferramenta capaz de atuar tanto na conexão com bases vetoriais (por meio de integração com uma *Store*, como Pinecone, Chroma ou FAISS) quanto na interface com grandes modelos de linguagem (OpenAI, Llama, Gemini etc.), unificando todo o fluxo de dados: desde a coleta e fragmentação (*chunking*) de documentos até a construção de *prompts* otimizados para qualquer LLM. Sendo que o LangChain pode ser uma solução para automatizar etapas que, em um RAG puro, exigiam implementações manuais — como geração de *embeddings*, indexação e junção de contextos recuperados ao *prompt* do modelo¹².

Rothman (2024) complementa que em exemplos práticos de *pipeline* multimodal, *frameworks* consolidados (Deep Lake, LlamaIndex, OpenAI) passam a encapsular a maior parte do trabalho pesado, e o LangChain corrobora com esse movimento ao fornecer uma camada de abstração que integra de forma padronizada armazenamento vetorial, busca de

similaridade, engenharia de *prompts* e chamadas a diferentes LLMs¹². A Figura 7 ilustra um sistema modular multimodal RAG.

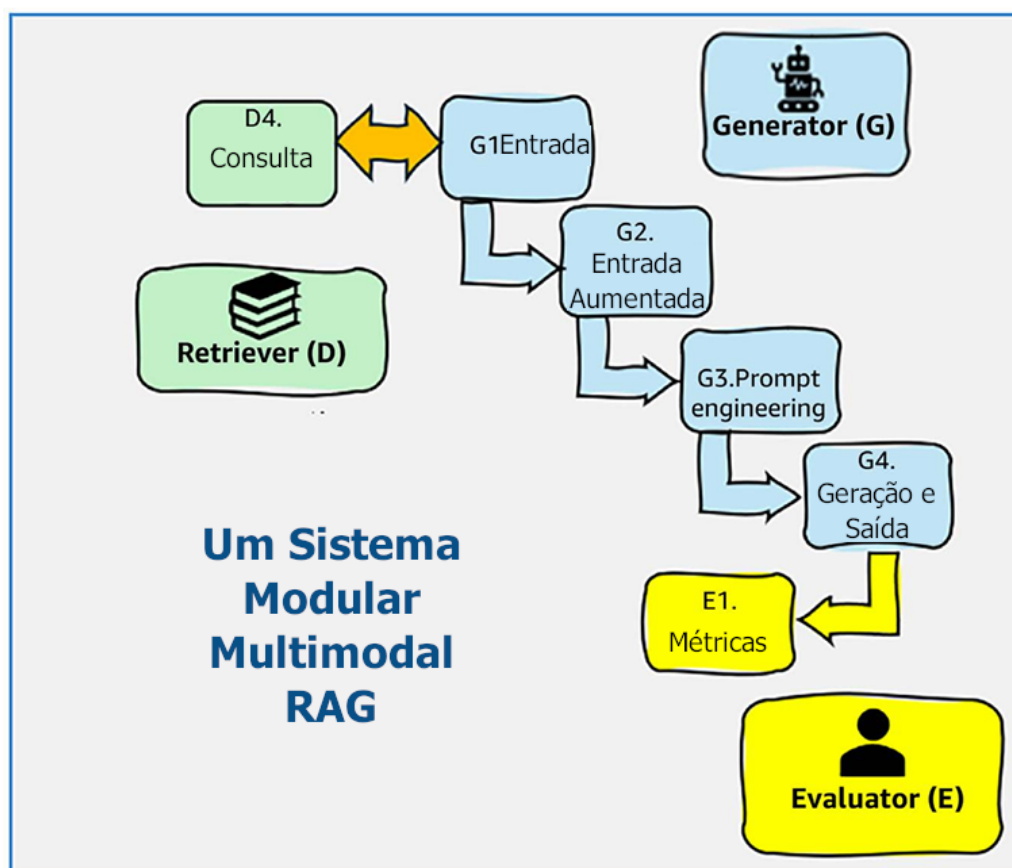


Figura 7 – Sistema Modular Multidimodal RAG.

Fonte: Rothman (2024).

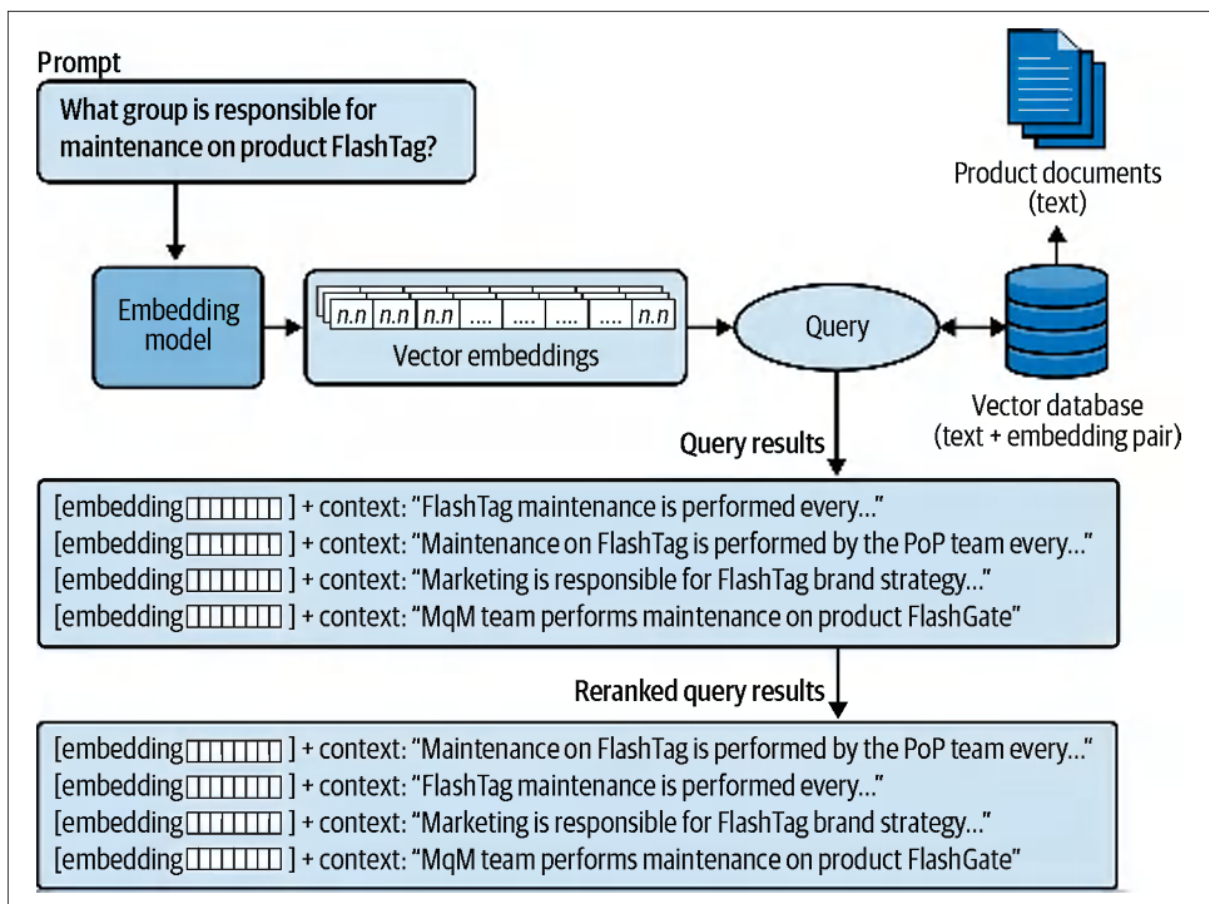
Subramanian (2024) afirma que o LangChain representa tanto um paradigma quanto uma biblioteca prática para construir aplicações baseadas em LLMs: ele organiza componentes reutilizáveis — como modelos de *prompt*, índices de recuperação, memória de conversa e conectores a APIs externas — em “cadeias” modulares que simplificam o fluxo de dados do usuário até o modelo e de volta à interface. Com o seu Kit de Desenvolvimento de Software (*Software Development Kit*) (SDK), é possível criar desde *pipelines* simples (onde um template formata a entrada, o LLM gera a resposta e um *parser* finaliza o texto) até agentes autônomos capazes de decidir, em tempo de execução, quais ferramentas invocar (busca na web, cálculo, acesso a bancos de dados etc.) para atingir objetivos complexos. Assim, LangChain abstrai a infraestrutura e oferece padrões de “saída fora da caixa” para acelerar o desenvolvimento de soluções com GenAI robustas e escaláveis¹⁰.

Por fim, Rothman (2024) insere o LangChain entre as ferramentas essenciais para escalar aplicações RAG em ambientes de produção, enfatizando seu papel modular multiplataforma que permite conectar repositórios de vetores, índices de busca e mo-

delos generativos sem reescrever código, tornando-se peça-chave para operacionalizar e industrializar soluções avançadas de IA¹², como o Tutor Inteligente.

2.2.2.1 O Reranking e o MMR (Maximum Marginal Relevance)

Segundo Fregley (2023), enquanto o buscador realiza a busca semântica o *reranking* (reclassificação) revisa esse conjunto, aplicando critérios adicionais — como diversidade ou relevância refinada — para reordenar os resultados, garantindo um contexto com mais informações e menos redundante para o modelo¹⁸, A Figura 8 ilustra o fluxo para um geração aumentada com *reranking*.



Augmented prompt:

What group is responsible for maintenance on product FlashTag?
 FlashTag maintenance is performed every Saturday with no downtime by the PoP Team. PoP team is responsible for sending automated notifications.

Completion prompt:

The PoP Team is responsible for Product Maintenance on the FlashTag product.

Figura 8 – Pipeline do Reranking e a geração aumentada.

Fonte: Fregley (2023).

Fregley (2023) complementa que, para introduzir variedade de documentos sem perder relevância, o LangChain oferece a **MMR** como opção de *reranking*. O **MMR** equilibra dois fatores: a afinidade de cada documento com o *prompt* e a diferença entre os próprios documentos, controlado por um parâmetro λ que varia de 0 (prioriza máxima diversidade) a 1 (prioriza máxima similaridade). Ao aplicar **MMR**, obtém-se um conjunto de contexto que explora múltiplas perspectivas, reduzindo redundâncias e ampliando a cobertura temática¹⁸.

2.2.3 *Vector Store* e o **FAISS**

2.2.3.1 Definição de *Vector Store*

Para Auffarth (2023) uma *Vector Store* é, essencialmente, um banco de dados otimizado para guardar e recuperar vetores de alta dimensão – normalmente *embeddings* gerados por grandes modelos de linguagem e multimodais. Ela desempenha um papel central em sistemas de **RAG** na busca semântica, recomendação e memórias de assistentes baseados em **LLMs**¹⁶.

Auffarth (2023) complementa que no fluxo típico de uma aplicação **RAG** com **LLMs**, o processo começa transformando trechos de documentos em vetores numéricos por meio de um modelo de *embedding*. Esses vetores são então indexados em estrutura de Vizinho Aproximado Mais Próximo (*Approximate Nearest Neighbor*) (**ANN**), como Mundo Pequeno Navegável Hierárquico (*Hierarchical Navigable Small World*) (**HNSW**) ou Arquivo Invertido (*Inverted File*) (**IVF**), para permitir buscas ultrarrápidas. Quando chega uma consulta, ela também é convertida em *embedding* e comparada com os vetores armazenados, retornando os trechos mais semanticamente próximos. Esses resultados servem de “pontos de apoio” para o **LLM**, que os incorpora como contexto adicional na geração de respostas, reduzindo alucinações e aumentando a precisão¹⁶.

Auffarth (2023) enfatiza quatro componentes principais: o motor de *embedding*, responsável por criar as representações vetoriais; o indexar, que organiza esses vetores em estruturas **ANN**; o armazenamento de metadados, que associa cada vetor a identificadores e informações auxiliares; e a camada de consulta, que orquestra todo o processo de busca e filtragem. Muitas *Vector Stores* modernas também oferecem filtros híbridos – além da similaridade vetorial, é possível restringir resultados por atributos como categoria ou data de criação¹⁶. A Figura 9 representa um compilado dos processos de indexação e similaridade vetorial.

Auffarth (2023) classifica as soluções de armazenamento vetorial em três grupos: bibliotecas *on-premise*, como Faiss e Annoy, que oferecem instalação local e otimização para baixa latência; plataformas *open-source* escaláveis, como Milvus, voltadas para grandes volumes de dados; e serviços gerenciados em nuvem, como Pinecone, Weaviate

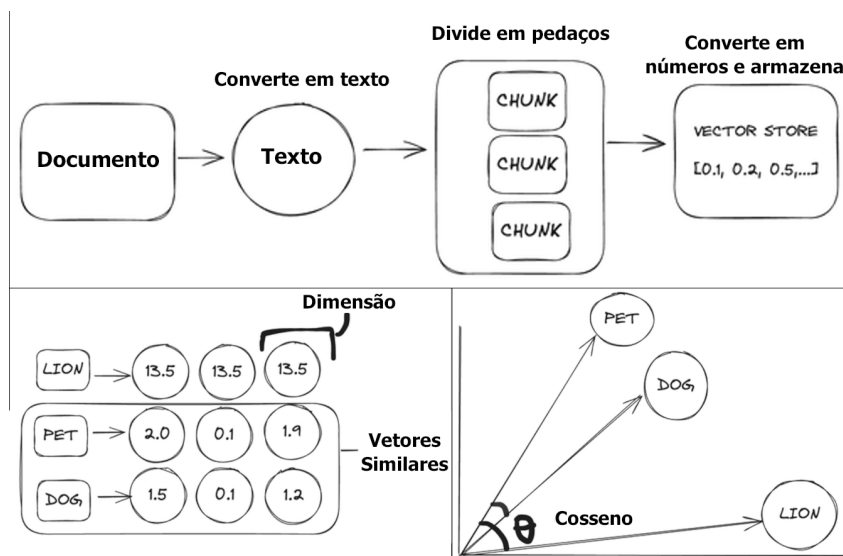


Figura 9 – Indexação e Similaridade Vetorial.

Adaptado de Oshin e Campos (2025).

e Qdrant, que cuidam automaticamente de replicação, escalonamento e monitoramento. Para prototipagem rápida em Python, o Chroma é apontado como a opção mais ágil, enquanto a Faiss se destaca por se manter como a escolha preferida, Figura 10, entre as soluções *open-source* pelos usuários do GitHub, segundo o site *Star History*¹⁶.

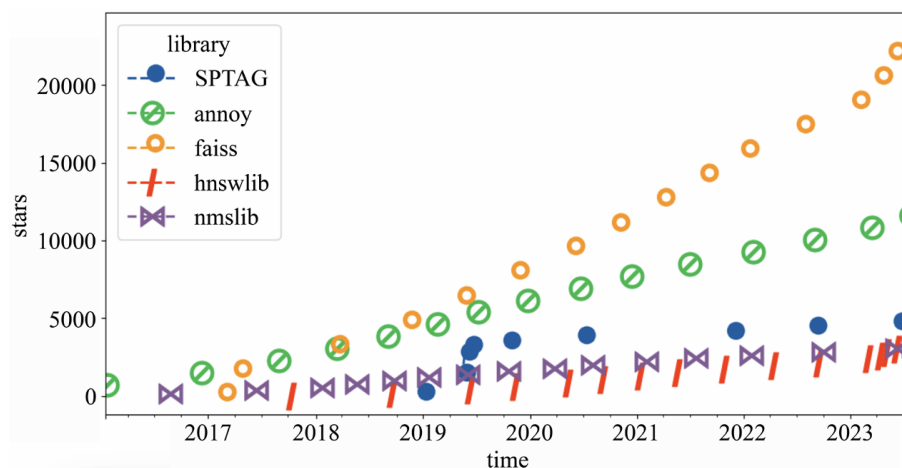


Figura 10 – A FAISS é a preferida pelos usuários do GitHub. O Star History é uma ferramenta que permite visualizar o histórico de estrelas de projetos no GitHub – a ferramenta gera um gráfico que mostra a evolução das estrelas do projeto ao longo do tempo, permitindo analisar o crescimento e a popularidade do projeto. As bibliotecas analisadas e seus respectivos repositórios foram: em **roxo** está `nmslib/nmslib`; em **vermelho** está `nmslib/hnswlib`; em **amarelo** está `facebookresearch/faiss`; em **verde** está `spotify/annoy`; e em **azul** está `microsoft/SPTAG`.

Adaptado de Auffarth (2023).

Auffarth (2023) ressalta que a *Vector Store* atua no cerne semântico de RAG com aplicações de GenAI, habilitando buscas contextuais e respostas mais confiáveis. Sua escolha, parametrização e manutenção adequadas fazem toda a diferença na qualidade de fluxos de trabalho de RAG, recomendação inteligente e chatbots com “memória”¹⁶.

2.2.3.2 O FAISS

Segundo Auffarth (2023), o FAISS é uma biblioteca desenvolvida pela Meta (anteriormente Facebook) que fornece busca de similaridade e agrupamento eficientes de vetores densos que suporta aceleração em Unidade Central de Processamento (*Central Processing Unit*) (CPU) e Unidade de Processamento Gráfico (*Graphics Processing Unit*) (GPU). Com diversos algoritmos de indexação — como Quantização de Produto (*Product Quantization*) (PQ), *Hashing* Sensível à Localidade (*Locality-Sensitive Hashing*) (LSH) e HNSW —, ele se tornou referência em buscas vetoriais em larga escala¹⁶. Fregley (2023) complementa afirmando que em arquiteturas de geração aumentada por recuperação (RAG), ele atua como núcleo de consulta semântica: documentos ou trechos de texto são convertidos em *embeddings* e, quando um *prompt* é enriquecido, o FAISS identifica rapidamente os vetores mais relevantes para guiar a geração de respostas pelos LLMs¹⁸.

O fluxo de uso do FAISS, conforme Fregley (2023), envolve três fases bem definidas: primeiro, gera-se *embeddings* a partir dos dados brutos — texto, imagem ou outro tipo de entrada — utilizando modelos especializados, como os *endpoints* (pontos de acesso de API) do *SageMaker JumpStart*; em seguida, esses vetores são organizados em índices (por exemplo, IVF, HNSW ou *Product Quantization*) que permitem buscas aproximadas de vizinhos mais próximos (ANN). Por fim, uma consulta é transformada em *embedding* e confrontada com o índice, retornando os elementos cujo grau de similaridade semântica é mais alto¹⁸.

2.3 Engenharia de *Prompt*

A Engenharia de *Prompt* é uma técnica fundamental para otimizar a interação com LLMs. Ela envolve projetar e refinar *prompts* ou instruções para direcionar o comportamento do modelo, aumentando sua precisão, relevância e aderência a comportamentos desejados (Brown *et al.*, 2020)²⁰, como a metodologia socrática utilizada neste estudo. A eficácia da Engenharia de *Prompt* é ampliada com técnicas como *Few-shot prompting*, onde exemplos específicos e limitados são fornecidos para ilustrar claramente o comportamento esperado (Auffarth, 2023)¹⁶.

Fregley (2023) defende que a Engenharia de *Prompt* surge da necessidade de direcionar grandes modelos de linguagem (LLMs) — que operam estatisticamente sobre sequências de *tokens* — a produzirem respostas precisas, relevantes e livres de “alucinações”

(informações sem respaldo nos dados de treinamento). Uma vez que não é possível alterar os pesos internos dessas redes, o principal desafio consiste em formular instruções claras e bem estruturadas, capazes de guiar o modelo para o comportamento desejado em cada contexto de uso¹⁸.

Fregley (2023) afirma que um *prompt* consistente reúne quatro elementos principais, todos interligados:

- **Instrução:** o comando que define a tarefa, descrito de forma direta (“Resuma em uma frase:”). Deve indicar explicitamente o formato ou restrições, quando necessário¹⁸.
- **Contexto:** informações complementares que ajudam o modelo a compreender escopo e estilo, como exemplos de entrada e saída — prática conhecida por “*few-shot learning*” ou aprendizado em contexto¹⁸.
- **Dados de Entrada:** o conteúdo bruto a ser processado (texto, imagem ou outro artefato), que estabelece o material-fonte para a geração¹⁸.
- **Indicador de Saída:** sinal claro de onde o modelo deve iniciar sua resposta (por exemplo, “Resposta:”), evitando ambiguidade sobre o ponto de partida da geração¹⁸.

Fregley (2023) endossa que esses componentes, quando bem combinados, aumentam a coesão e a fidelidade da saída, alinhando-a às expectativas do usuário¹⁸.

A engenharia de *prompt* — também chamada de aprendizado *in-contexto* — consiste em guiar o comportamento de modelos de linguagem apenas por meio de *prompts* cuidadosamente elaborados, sem alterar seus pesos internos (Auffarth, 2023). Por meio dessa técnica, é possível estender o uso dos LLMs a tarefas para as quais eles não foram originalmente treinados, oferecendo flexibilidade e economia de recursos ao dispensar retrainamento¹⁶.

Auffarth (2023) corrobora ao afirmar que um *prompt* eficaz combina três elementos fundamentais:

- **Instruções**, que descrevem de forma clara e inequívoca os requisitos, objetivos e o formato da tarefa;
- **Exemplos**, que demonstram pares de entrada e saída desejados, fornecendo ao modelo casos concretos de como mapear diferentes inputs em outputs apropriados;
- **Entrada**, que é o dado real sobre o qual o modelo deve atuar para gerar a resposta.

Segundo Fregley (2023), há técnicas de aprendizagem por contexto, cujo o qual explana sobre três técnicas de aprendizagem por contexto:

- **Zero-shot:** o modelo recebe apenas a instrução, sem exemplos prévios; ideal para tarefas que já são familiares ao LLM, mas que podem resultar em saídas mais criativas ou imprecisas.
- **One-shot:** inclui-se um exemplo ilustrativo; serve como demonstração pontual do formato esperado, elevando a consistência da resposta.
- **Few-shot:** insere-se de dois a cinco exemplos, definindo padrões de estilo, tom e estrutura. Essa técnica costuma aprimorar significativamente a qualidade e a padronização das saídas em tarefas complexas.

Mas Fregley (2023) alerta que é fundamental revisar os exemplos para evitar ensinar comportamentos indesejados¹⁸.

2.3.1 *Few-shot Prompting*

Fregley (2023) aborda que o *Few-shot Prompting* aproveita a habilidade dos LLMs de “aprender na hora”, um fenômeno de *in-context learning* — em que o LLM ajusta sua geração ao padrão apresentado no contexto, ao inserir, no próprio *prompt*, poucos exemplos de entrada e saída, sem qualquer ajuste nos pesos do modelo. Ao fornecer de dois a cinco pares de exemplos, o usuário estabelece um guia implícito que orienta o formato, o estilo e o nível de detalhe desejados, reduzindo respostas excessivamente longas ou que fujam do escopo esperado¹⁸.

Fregley (2023) destaca a maior uniformidade na formatação das saídas, o foco nas informações essenciais e a rapidez na prototipação de soluções — tudo isso sem a necessidade de *fine-tuning*. Mas, alerta que é preciso atentar para o limite de *tokens* da janela de contexto, pois muitos exemplos podem ultrapassá-lo e fragmentar o *prompt*, e também para o risco de enviesamento: exemplos mal calibrados podem induzir respostas incorretas ou distorcidas¹⁸.

Auffarth (2023) reforça que o *Few-Shot Learning* aproveita o amplo conhecimento prévio de um LLM fornecendo-lhe apenas alguns exemplos anotados — tipicamente entre um e dez pares “entrada → saída” — para que ele ajuste seu comportamento a uma nova tarefa sem a necessidade de *fine-tuning*. Uma vez que os LLMs já “sabem” muito sobre linguagem, esses poucos exemplos servem de guia suficiente para direcionar a geração¹⁶.

Auffarth (2023) aponta como principais vantagens a rapidez de adaptação — pois não há fase de treinamento adicional — e a flexibilidade — já que basta trocar ou reorganizar os exemplos para mudar o estilo ou o formato da resposta; e complementa as limitações relatadas por Fregley (2023) afirmando que a ordem e a formulação dos exemplos influenciam diretamente a qualidade do resultado, exigindo bom domínio de engenharia de *prompt*¹⁶.

Em suma, o *Few-Shot Learning* encurta distâncias entre o pré-treinamento massivo e aplicações pontuais, oferecendo uma forma ágil de customizar LLMs para novas demandas com pouquíssimos exemplares. Esta técnica permite que o modelo capture rapidamente a tarefa desejada a partir de poucos exemplos, melhorando a robustez e precisão das respostas ao demonstrar claramente o comportamento esperado. Neste trabalho, a técnica será aplicada no RAG para garantir respostas pedagógicas eficazes e alinhadas ao método socrático.

Esses fundamentos teóricos estabelecem uma base sólida para compreender as tecnologias utilizadas, permitindo uma análise clara e estruturada das abordagens arquitetônicas comparadas neste trabalho.

3 TRABALHOS RELACIONADOS

A área de [AIED](#) tem experimentado um crescimento exponencial, impulsionado em grande parte pelos avanços em [LLMs](#) e pela busca por soluções de aprendizagem mais personalizadas e eficazes. Esta seção revisa a literatura recente sobre soluções de [TIAs](#) em suas abordagens de implementação e avaliação, identificando lacunas que justificam a presente investigação.

3.1 Revisão das Soluções Existentes

Yang Albert C. M. *et al.* (2024) descrevem que sistemas de tutoria inteligente baseados no ChatGPT, como o Sistema de Tutoria Inteligente PyTutor (*Python Tutor*) ([PyTutor](#)), representam um marco na personalização do ensino, ao oferecerem sugestões de resolução de problemas que se adaptam ao perfil de cada estudante. Segundo os autores, essa tecnologia não apenas amplia o acesso dos aprendizes a conteúdos contextualmente relevantes, mas também proporciona aos usuários avançados a oportunidade de comparar diferentes abordagens de codificação, o que eleva seu engajamento e persistência ao enfrentar exercícios desafiadores²¹.

Yang Albert C. M. *et al.* (2024) alertam para os potenciais riscos de dependência excessiva de sistemas de tutoria inteligente, observando que alunos com baixo conhecimento prévio podem deixar de adquirir competências básicas de resolução de problemas ao se apoiarem demais nas dicas automatizadas. Os autores acrescentam que esse fenômeno adiciona uma dimensão crítica à teoria de design instrucional e ressalta a necessidade de gerar dicas que sejam simultaneamente precisas, compreensíveis e desafiadoras para todos os níveis de experiência. Tais constatações encontram respaldo em Candy (1991) *apud* Yang Albert C. M. *et al.* (2024), que defende a autonomia estudantil como pré-requisito para a transferência efetiva do aprendizado e o desenvolvimento do pensamento crítico²¹.

Além do uso de um Sistema de Tutoria Inteligente (*Intelligent Tutoring System*) ([ITS](#)), como [PyTutor](#), outra forma de auxiliar na aprendizagem encontrada foi com utilização do [RAG](#) através do *framework* LangChain nos trabalhos de Németh *et al.* (2024), que destacam que, embora o [RAG](#) demonstre potencial consolidado em outros setores, sua aplicação no ensino ainda carece de investigação. Eles citam Miladi *et al.* (2024) para ilustrar como um [GPT-4](#) reforçado por [RAG](#) pode superar modelos não aumentados e a própria média dos estudantes em Curso Online Aberto e Massivo (*Massive Open Online Course*s) ([MOOCs](#)) de [IA](#), fornecendo respostas precisas, mas não infalíveis — fato que motivou a análise de acurácia em seu trabalho. Relatam que, de dezenove alunos convidados, apenas quatro participaram voluntariamente do *focus group*, mas contribuíram

com dados significativos, sendo as primeiras impressões capazes de direcionar todo o debate subsequente. Quanto à aceitação, os usuários atribuíram 9,4 de 10 à recomendação da ferramenta e manifestaram unanimidade quanto ao seu uso em cursos de mestrado. Finalmente, mesmo reconhecendo a limitação de escopo e o protótipo inicial, afirmam que essa iniciativa inaugura um avanço pioneiro na área de sistemas de tutoria habilitados por IA²².

Ferramentas baseadas em LLMs no ensino superior oscilam entre a percepção de “tecnologias perigosas” que incentivam cola e plágio e a de soluções capazes de simplificar fluxos de trabalho e apoiar a aprendizagem (Németh *et al.*, 2024)²². Apesar dessas potencialidades, o uso desmedido pode prejudicar o desenvolvimento de habilidades analíticas e de pensamento crítico, bem como permitir o uso antiético do modelo RAG — embora este não tenha sido o foco principal do estudo, que priorizou alucinações e oportunidades de inovação. Dois usuários intensivos demonstraram abordagens opostas: um recorreu ao tutor quase continuamente para organizar o material, enquanto outro ressaltou seu papel como ferramenta de auto-verificação complementar. Os autores não testaram a precisão nem a confiabilidade da classificação automatizada por estarem além do escopo, e admitiram a limitação de escala do protótipo e de ter avaliado apenas um tipo de curso²².

Por fim, Németh *et al.* (2024) relatam que os participantes depositaram grande confiança na IA por esta se apoiar em materiais didáticos oficiais, o que diminuiu o receio de “alucinações” ao não incorporar fontes externas. Segundo os autores, essa expectativa positiva foi reforçada pela chancela institucional da universidade, levando os usuários a se sentirem seguros ao “ousar confiar” plenamente no sistema. Além disso, mesmo aqueles familiarizados com o fenômeno de alucinação do ChatGPT demonstraram cautela — revisando as respostas diversas vezes —, mas, ainda assim, tiveram uma experiência favorável com o tutor inteligente²².

No trabalho de Ma *et al.* (2024) ressaltam que as matrículas em Ciência da Computação atingiram níveis inéditos — quase mil calouros anuais — e definem os LLMs como modelos *transformer* pré-treinados que se destacam em NLP e exames diversos. Eles defendem o uso dessas redes na educação para personalizar o ensino e automatizar a correção de tarefas. Nesse sentido, desenvolveram o Gerenciador de Recuperação Aumentada por Geração (*Retrieval-Augmented Generation Manager*) (RAGMan), um *framework* próprio de RAG para consultar materiais de apoio à aprendizagem, implantado opcionalmente em um curso com 455 alunos, oferecendo tutoria 24 horas por dia sem fornecer respostas prontas e registrando todas as trocas em servidor protegido, sob aprovação de “Pesquisa sem sujeitos humanos”. Após processar 2.072 conversas e limpar mensagens repetidas ou irrelevantes, avaliaram 671 conversas e 3.164 pares, classificando-os quanto ao escopo e medindo a qualidade das respostas (94% boas). Aplicaram Teste de Kolmogorov–Smirnov

(*Kolmogorov–Smirnov Test*) (KS), Wilcoxon–Mann–Whitney e Anderson–Darling para comparar distribuições de notas, constatando aumentos nas notas B, redução de F e menos reprovações. A análise manual mostrou 93,5% de respostas eficazes e mais de 78% dos estudantes consideraram que o tutor aprimorou seu aprendizado em sala²³.

Há também um alerta de que ferramentas de IA como ChatGPT e Copilot apresentam dupla face: facilitam o acesso a soluções, mas podem substituir o esforço cognitivo dos estudantes (Ma *et al.*, 2024)²³. Em seu grupo, 2% não usaram IA e 56% recorreram a várias soluções automatizadas, havendo uma leve queda nas notas A, sem possibilidade de inferir causalidade em função do desenho não experimental. As preocupações éticas incluem plágio, viés e disseminação de informações incorretas (Durall e Kapros *apud* Ma *et al.*, 2024; Ahmed Tlili *et al. apud* Ma *et al.*, 2024) e a ocorrência de 19,5% de alucinações no ChatGPT (Li *et al. apud* Ma *et al.*, 2024). Assim, defendem a criação de IA auditável e responsável (Ma *et al.*, 2024). Do ponto de vista dos dados, coletaram 195 postagens voluntárias, limpam e validaram cada discussão, e conduziram uma pesquisa com 88% de resposta, indicando que 79% dos alunos usaram IA — 58% com tutores específicos. Embora tenham notado maiores percepções de aprendizado e menor receio de julgamento, registraram críticas à lentidão, à falta de contexto e a respostas contraditórias. Reconhecem limitações de amostra, desenho exploratório e grupos distintos, planejando ensaios randomizados e inclusão de anotadores humanos para verificarem efeitos de causalidade²³.

3.2 Indexação de Lacunas na Literatura

Os três distintos trabalhos, da tentativa de se adequar o uso de LLMs na educação para auxiliar na aprendizagem do aluno, possuem suas características peculiarmente positivas e suas limitações nesta tentativa de personalizar o ensino. O ITS ‘PyTutor’ de Yang Albert C. M. *et al.* (2024) fornece dicas de estudos sem supervisão humana para evitar ou mitigar seus impactos e ressalta a necessidade de gerar dicas que sejam simultaneamente precisas, compreensíveis e desafiadoras para todos os níveis de experiência, além de requerer grande base de treinamento com adaptações para diferentes níveis de alunos²¹. Embora a aplicação de Németh *et al.* (2024) utilize o *framework* LangChain para orquestrar o Tutor com RAG aprimorado, os autores deixaram à cargo dos alunos a configuração do Tutor em um pequeno grupo amostral, levando os usuários a se sentirem seguros ao “ousar confiar” plenamente no sistema porque o Tutor se embasava em conteúdos oficiais e evitava “alucinações”²². Por fim, o RAGMan de Ma *et al.* (2024), uma aplicação proprietária que utiliza RAG para auxiliar na aprendizagem dos alunos, também apresentou ‘alucinações’ recorrentes e a necessidade de ampla base de contexto para treinamento, bem como de avaliadores independentes para revisar e classificar as respostas²³.

4 METODOLOGIA

A presente seção detalha os procedimentos metodológicos adotados para comparar as duas abordagens de implementação do Tutor com Inteligência Artificial (**TIA**) para Cálculo I: o uso direto do **LLM** versus o uso com **RAG**, explorando a engenharia de *prompt*. O foco reside em estabelecer uma base comparativa clara para avaliar como as configurações nestas diferentes arquiteturas influenciam a aderência do **TIA** — um Tutor com Inteligência Artificial capaz de guiar o aluno no processo de aprendizagem, sem transmitir o conhecimento, através de perguntas e instigação para facilitar a aprendizagem.

4.1 Processo de Desenvolvimento

O desenvolvimento seguiu uma abordagem comparativa experimental simulada. O processo envolveu as seguintes etapas principais:

1. **Definição dos Requisitos:** Especificação dos objetivos do **TIA** – do domínio da matéria (Cálculo I) e do comportamento contextualizado como pedagógico socrático para o público-alvo simulado (alunos).
2. **Material de Apoio:** Livro aberto “Cálculo é Fácil” do prof. Dr. Walter Ferreira Velloso Junior da USP⁸.
3. **Seleção Tecnológica:** Escolha do Grande Modelo de Linguagem (**LLM**) base (OpenAI `gpt-4-turbo`) e das ferramentas de implementação (**API** OpenAI direta e *frameworks open-source*: LangChain, **FAISS** e Streamlit. Detalhados no Apêndice A).
4. **Design das Arquiteturas, Configuração e Implementação:** Modelagem das duas arquiteturas distintas de chat com **LLM** e a definição dos parâmetros e componentes específicos para cada arquitetura, para serem comparadas, detalhadas na Seção 4.2.
5. **Integração Tecnológica e Arquitetura da Aplicação:** Modelagem da arquitetura do sistema e a integração tecnológica, detalhadas na Seção 4.3.
6. **Elaboração do Conjunto de Testes:** Criação de um conjunto diversificado de 27 perguntas (Apêndice B) para avaliar o desempenho das arquiteturas dos sistemas em cenários distintos (conteúdo específico, aplicações, controle).
7. **Execução Experimental Simulada:** Geração de respostas em cada arquitetura para todas as perguntas, simulando o comportamento esperado de cada arquitetura (Apêndice B).

4.2.1 Abordagem direta com o LLMs (APIs OpenAI com o modelo GPT-4 Turbo)

Para estabelecer a interação mais básica com o modelo de linguagem, a aplicação se comunica diretamente por meio do serviço de IAs da OpenAI. A Figura 12 (Diagrama de “Fluxo: o uso direto do modelo de linguagem através da APIs da OpenAI” – criado no Bizagi) ilustra esse processo: o usuário (aluno, professor ou gestor) envia uma requisição à aplicação, cujo conteúdo pode ser texto, áudio ou vídeo. Ao receber essa solicitação, o sistema avalia imediatamente o formato:

1. **Transcrição de áudio ou vídeo:** a aplicação executa um script de transcrição: primeiro extrai o áudio, segmenta-o em pequenos trechos (*chunks*) e, em seguida, envia cada fragmento a um serviço de reconhecimento de fala para convertê-lo em texto (Whisper da OpenAI), garantindo um fluxo unificado.
2. **Processamento de texto:** se o conteúdo já estiver em formato de texto, ele avança imediatamente para a etapa seguinte, sem passar pela transcrição.

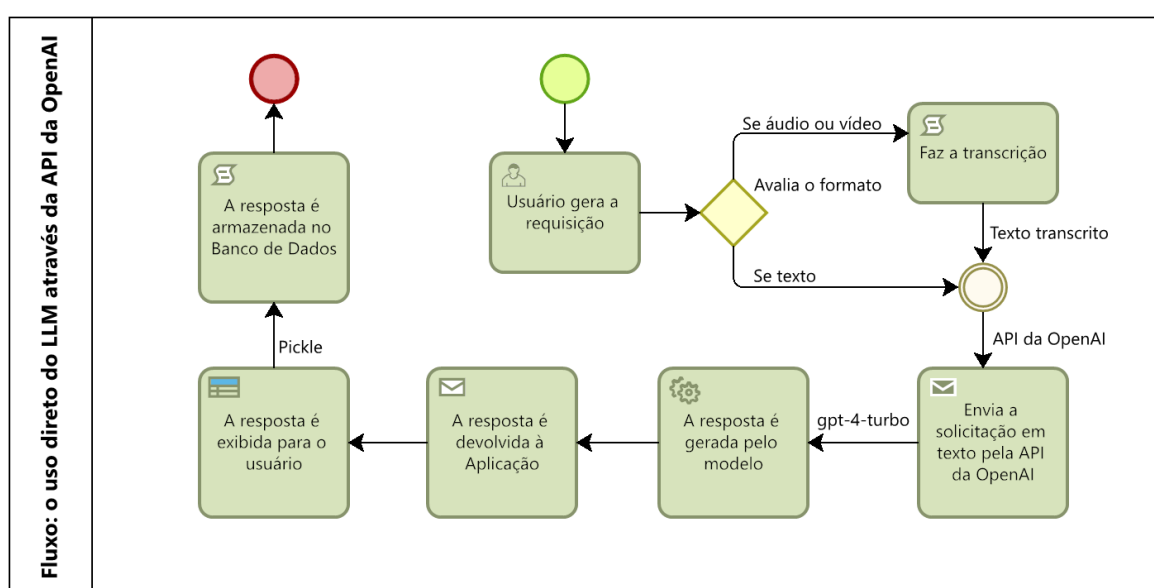


Figura 12 – Fluxo: o uso direto do LLM através da API da OpenAI.

Fonte: Ferreira, B.C., 2025.

Na sequência, para enviar a solicitação à **API**, utiliza-se o método de requisição Protocolo de Transferência de Hipertexto (*Hypertext Transfer Protocol*) (**HTTP**), que consiste em transmitir dados ao servidor para que ele execute uma ação. Esses dados são organizados em Notação de Objeto JavaScript (*JavaScript Object Notation*) (**JSON**), um formato leve de texto que estrutura informações em pares de chave-valor. A aplicação, então, monta esse **JSONs** incluindo o modelo escolhido, a chave de autenticação e o conteúdo da mensagem com o texto consolidado — além de quaisquer parâmetros adicionais — e dispara a requisição para o endereço da OpenAI (<<https://api.openai.com/v1/chat/completions>>), seguindo as especificações da documentação²⁴.

A resposta gerada pelo modelo é entregue em formato **JSONs**. Ao receber esse retorno, o sistema executa a rotina de recepção, interpreta o conteúdo e extrai a mensagem final.

Por fim, duas ações ocorrem em paralelo e encerram o processo:

- **Exibição da resposta:** na interface do usuário via Streamlit.
- **Armazenamento do resultado no banco de dados:** primeiro a saída é serializada com Pickle e depois persistida para histórico, auditoria e eventuais análises futuras.

4.2.2 Abordagem com **RAG** modulado pelo *framework* LangChain

Nesta arquitetura, o **RAG** atua como intermediário entre a aplicação e o **LLM**, orquestrando de forma modular cada etapa da geração de resposta por meio de “cadeias” predefinidas. Essa estrutura garante controle detalhado do fluxo de dados — da preparação e configuração dos componentes até a invocação sequencial dos *prompts* e ferramentas dentro do *pipeline*. A Figura 13 (Diagrama de “Fluxo: uso do **RAG** modulado pelo *framework* LangChain” – elaborado no Bizagi) ilustra que esse processo é dividido em duas fases principais:

1. **Configuração** – Definição de *templates* de *prompt*, seleção de modelos, conexão com fontes de dados (documentos) e parâmetros das cadeias.
2. **Interação** – Execução dinâmica das cadeias: formatação do input, recuperação ou cálculo de informações auxiliares, invocação do **LLM** e pós-processamento da saída.

4.2.2.1 A Fase de Configuração do **RAG**

O processo começa antes mesmo da requisição do usuário, em um estágio de preparação de conhecimento que exige alimentação de documentos e a configuração do **PROMPT** de controle:

1. **Carregamento e fatiamento de documentos** – O usuário, o professor ou gestor fornece os documentos (materiais de aula, livros, apostilas, plano pedagógico do curso e outras regras de negócios) à aplicação via [API](#) do Streamlit, que lê os arquivos-fonte, divide-os em trechos menores e semanticamente coesos e, em seguida, gera o *embedding* de cada trecho por meio da classe `OpenAIEmbeddings`.
2. **Persistência dos vetores com o FAISS** – Esses vetores são armazenados em um *Vector Store* local, criando a base para buscas semânticas por similaridade com a [FAISS](#), detalhadas na Seção 2.2.3.
3. **Definições da Chain** – Define-se o *wrapper* que será enviado ao [LLM](#) através da configuração do método `ConversationalRetrievalChain.from_llm`, conforme a documentação do LangChain¹⁷:
 - o **llm**: através do provedor `ChatOpenAI` com o modelo `gpt-4-turbo`;
 - a **memória**: através do método `ConversationBufferMemory`;
 - as **configurações do retriever**: via o método `as_retriever` da *Vector Store*, definindo estratégias como *mmr* (Relevância Máxima Marginal) e parâmetros `k` e `fetch_k`;
 - o **PROMPT**: orientações e instruções que servirão de filtro para todas as respostas.

4.2.2.2 A Fase de Interação

Inicia-se, semelhante à arquitetura direta com a [API](#) da OpenAI, através de uma requisição do usuário com texto (áudio e vídeo ainda não implementados):

1. **Se for áudio ou vídeo**, a própria aplicação executará uma tarefa de transcrição, como na arquitetura anterior, após ser implementado.
2. **Se já for texto**, o fluxo segue diretamente para a etapa seguinte.

Com o texto consolidado, diferente da abordagem direta, a aplicação cria o vetor da pergunta do usuário e consulta o *Vector Store*, recuperando passagens semanticamente próximas. Em seguida, o módulo [RAG](#) monta um *wrapper* único para o [LLM](#) combinando:

- a pergunta do usuário;
- o contexto recuperado;
- as definições de *chain* com o PROMPT de controle.

A cadeia LangChain invoca o `ConversationalRetrievalChain.from_llm`, conforme a documentação do LangChain.¹⁷ e envia para o **LLM** configurado gerar a resposta.

O **LLM** recebe o *wrapper*, interpreta as instruções, processa os fragmentos de contexto e gera a saída. A interpretação analisa, basicamente, se:

- **Não atende:** a cadeia reconstrói a geração da resposta em função do *prompt* de controle até satisfazer os critérios estabelecidos na engenharia de *prompt* adotada na *chain*.
- **Atende:** prossegue com a resposta gerada.

O **LLM** devolve o **JSONs** com a resposta ao **RAG**, que, por sua vez, o envia para a aplicação.

Na sequência, duas ações complementares ocorrem em paralelo lógico e encerram o processo:

- A resposta é **exibida ao usuário** na interface da aplicação através do Streamlit.
- A mesma resposta deverá ser **armazenada no banco de dados**. Embora a última resposta exibida possa ser auditada pelo usuário, o armazenamento das anteriores está pendente de implementação. Sua implementação deverá ser semelhante à etapa da arquitetura anterior, em que primeiro a conversa é serializada com Pickle e, depois, persistida para possibilitar histórico, auditoria e eventuais análises futuras de conversas anteriores à atual.

O **RAG**, modulado com o LangChain, atua como um intermediário que estrutura a interação. Permite a definição de cadeias (*Chains*) que combinam diferentes módulos, como: *wrappers* para **LLMs**, *templates* de *prompt*, mecanismos de recuperação de informação (*Retrievers*) e gestão de memória. Mesmo que a documentação do LangChain¹⁷ detalhe seus componentes e a sua interligação, possibilitando a criação de fluxos de trabalho complexos para tarefas específicas, apenas professores, pedagogos e educadores gestores são indicados como mentores para orientarem e auditarem o **TIA** em sua contextualização, pedagógica socrática.

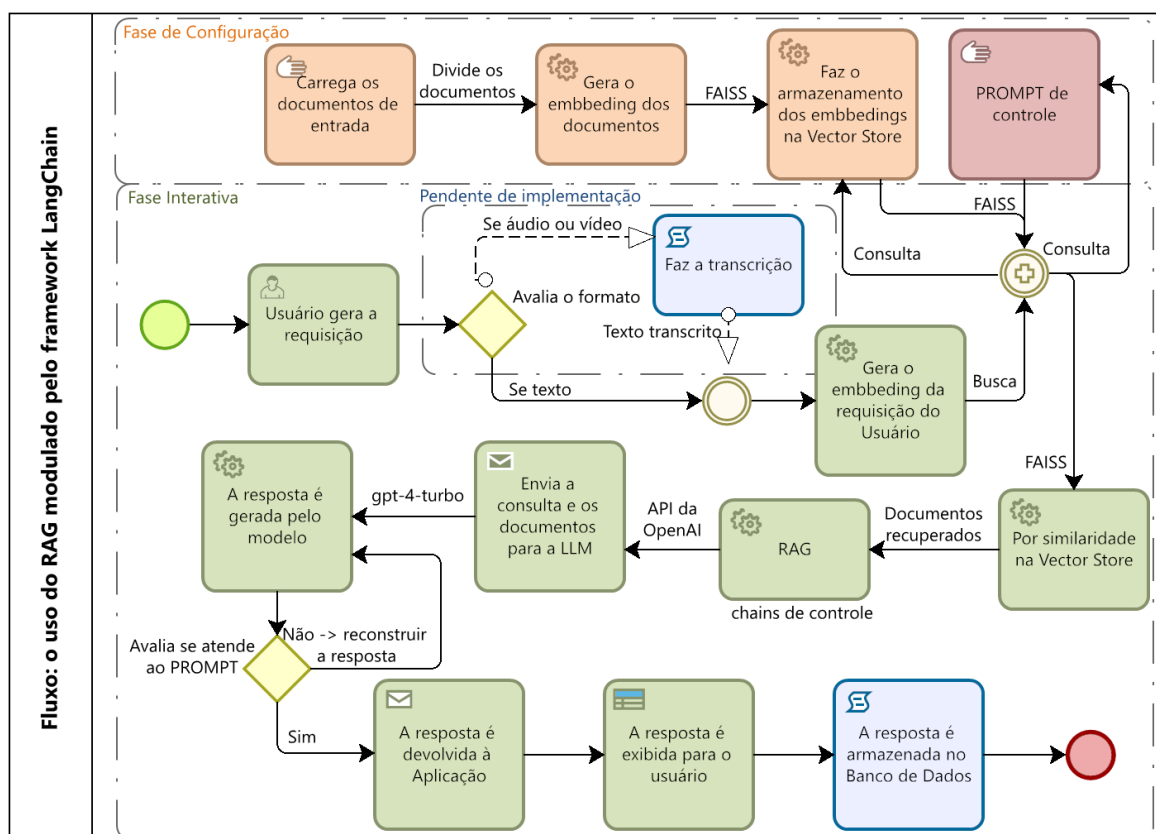


Figura 13 – Fluxo: uso do RAG modulado pelo *framework* LangChain.

Fonte: Ferreira, B.C., 2025.

4.3 Modelagem e Integração Tecnológica para o TIA

O sistema do Tutor com Inteligência Artificial (TIA) possui a seguinte arquitetura para ser implantado:

1. O sistema de Login, como entrada de acesso;
2. O gerenciamento do sistema pelo gestor;
3. A trajetória do Professor;
4. A trajetória do Aluno.

4.3.1 Modelagem do Sistema: Tutor com Inteligência Artificial

Nos diagramas a seguir, as tarefas — indicadas por retângulos de bordas arredondadas — estão codificadas por cores:

- **Verde:** etapas mínimas necessárias para prototipagem e testes do TIA;
- **Azul:** atividades dispensáveis em um teste rápido, mas indispensáveis em ambiente de produção;
- **Laranja/Roxo:** fases de configuração do RAG, tão essenciais quanto as sinalizadas em verde.

Qualquer uma das entidades precisará fazer o cadastro e depois o *login* para acessar o sistema do TIA, como ilustrado na Figura 14:

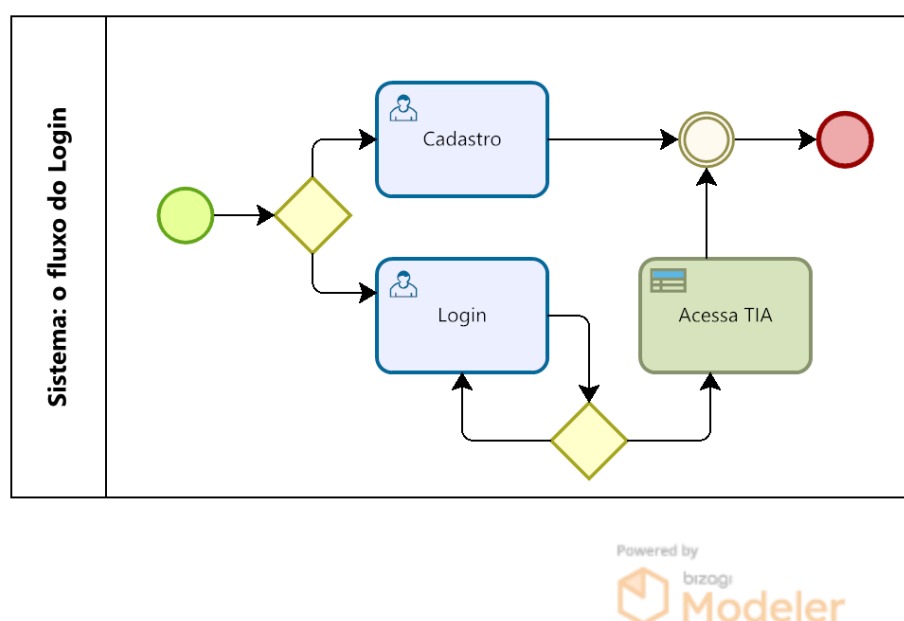


Figura 14 – Sistema: o fluxo de Login.

Fonte: Ferreira, B.C., 2025.

Porém, apenas o Gestor do sistema poderá modificar cada um dos demais usuários como Professor ou Aluno, como demonstrado na Figura 15:

A trajetória do usuário Aluno após o *login* inicia com a seleção da disciplina/matéria que necessite tirar dúvidas com o TIA, sendo que o sistema deve ser capaz de filtrar os

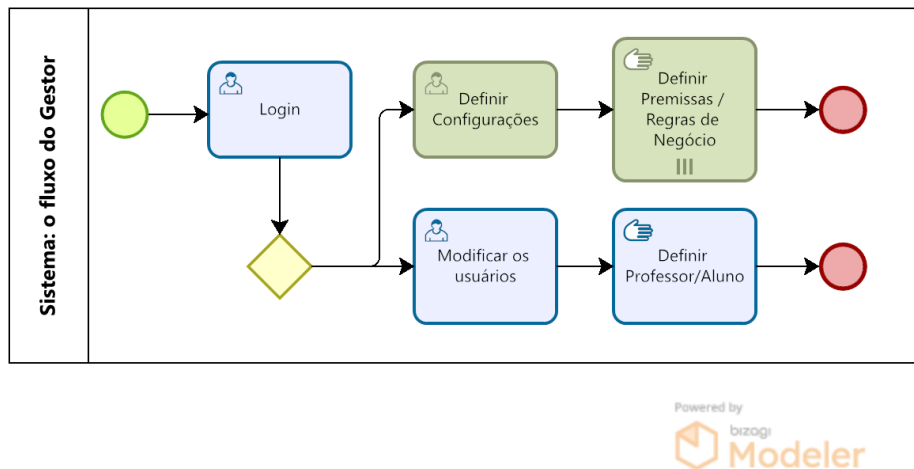


Figura 15 – Sistema: o fluxo do Gestor.

Fonte: Ferreira, B.C., 2025.

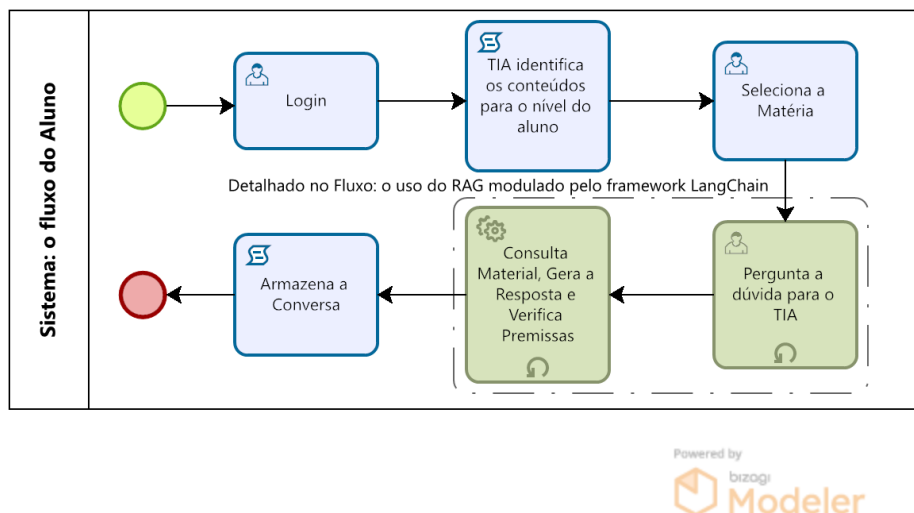


Figura 16 – Sistema: o fluxo do Aluno.

Fonte: Ferreira, B.C., 2025.

materiais que serão disponibilizados para consulta no período da matéria. Este diagrama está ilustrado na Figura 16:

Já a entidade Professor poderá, em seu fluxo, auditar as respostas do TIA por meio da investigação das respostas geradas com o uso direto do LLM, para aprimorar as próprias regras de funcionamento do TIA, ou através da consulta das respostas que o TIA enviou aos seus alunos. Outras opções em seu fluxo, após o *login*, são incluir turmas — inserindo os alunos e definindo quando cada material ou versão do material estará disponível para consulta pelo TIA —, configurar o TIA e inserir materiais de aula de suas disciplinas/matérias. Todo o fluxo do professor pode ser visualizado na Figura 17:

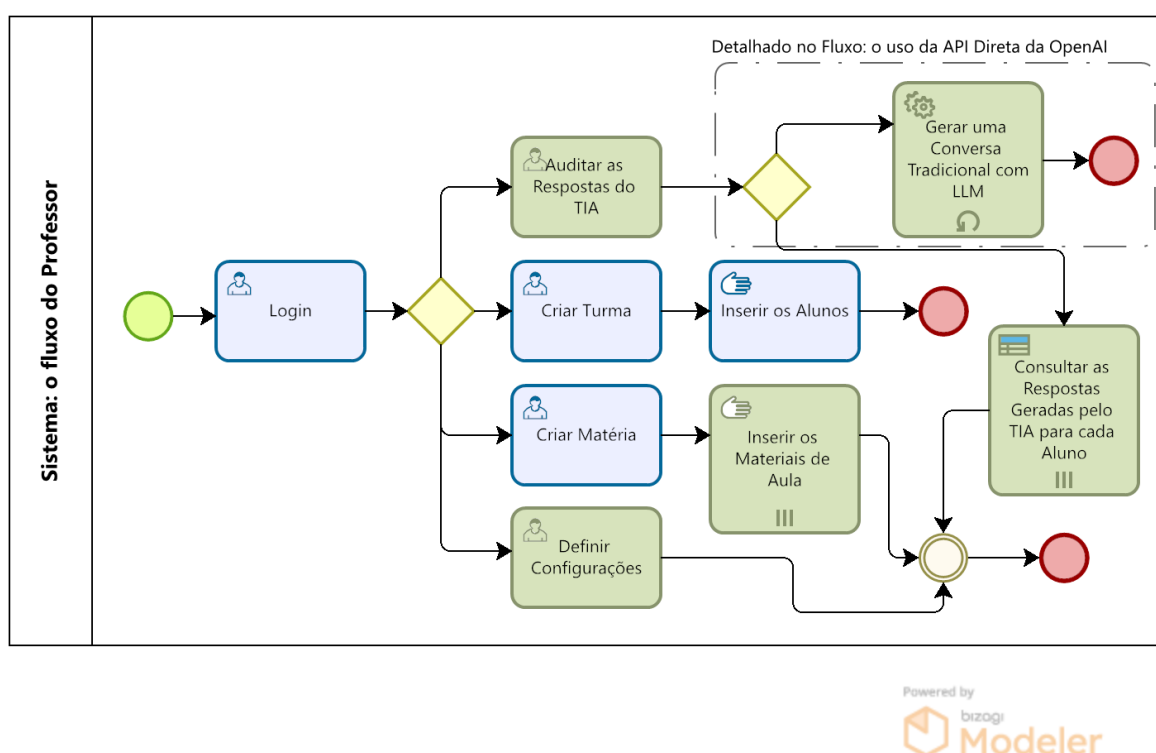


Figura 17 – Sistema: o fluxo do Professor.

Fonte: Ferreira, B.C., 2025.

Por fim, as premissas para que o TIA funcione conforme seus objetivos estão disponíveis na Figura 18:

4.3.2 Integração Tecnológica entre o Streamlit, o LLM e o RAG

Optou-se pelo Streamlit como camada de integração do TIA por ser um *framework open-source* voltado a aplicações interativas, que oferece componentes prontos no frontend

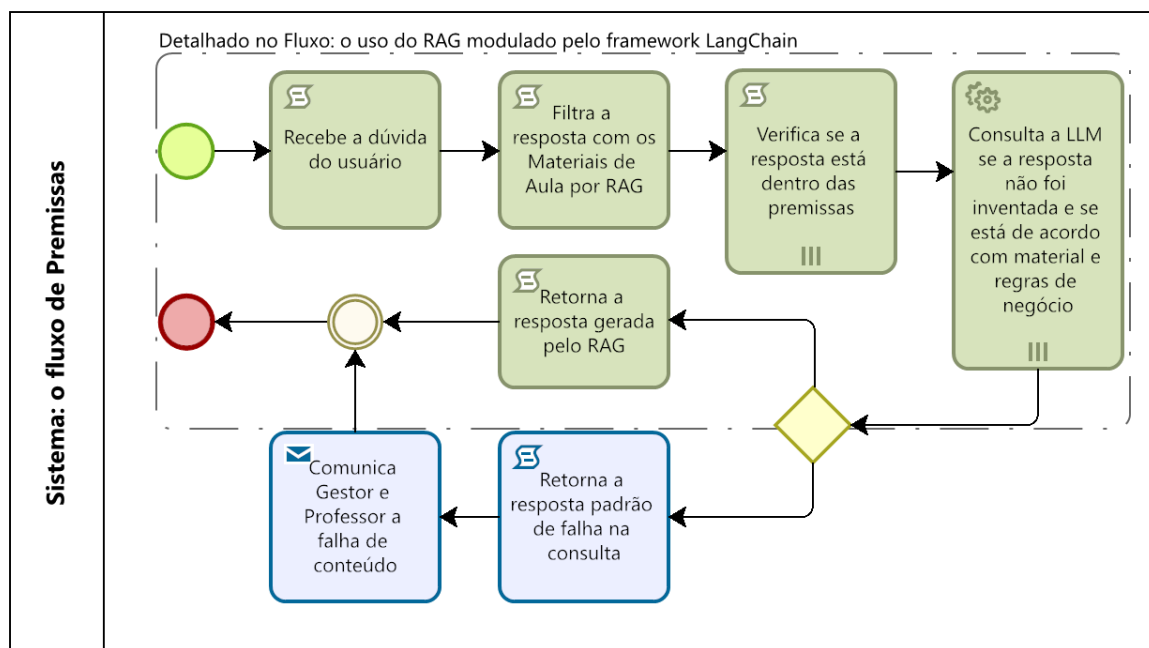


Figura 18 – Sistema: o fluxo das Premissas.

Fonte: Ferreira, B.C., 2025.

e permite ajustes dinâmicos no backend, conforme sua documentação²⁵. Essa abordagem viabiliza uma prototipagem rápida e a atualização imediata do sistema assim que os documentos são carregados no *pipeline* RAG.

Para garantir a comparabilidade entre os Tutores com Inteligência Artificial, ambas as abordagens utilizam o mesmo LLM base. Entretanto, o TIA poderá habilitar a alteração da escolha do provedor em sua versão para produção.

4.3.2.1 Configuração da API Direta do LLM

- **API Key:** credenciais da API OpenAI;
- **Modelo:** gpt-4-turbo, por padrão, mas pode ser alterada;
- **Mensagens:** o *prompt* que o usuário definir;
- **Temperature:** definida como 0.0 (quanto maior esse valor — 1.0 —, maior a criatividade do LLM), para não haver aleatoriedade das respostas geradas pelo

modelo;

- **Stream:** definido como `True` para manter uma experiência análoga à de uma conversa direta com provedores de `LLMs`, exibindo as mensagens dinamicamente à medida que são construídas.

4.3.2.2 Configuração do RAG

- **Sistema modular multimodal RAG:** importação do *framework* `LangChain` através das bibliotecas `langchain`, `langchain_community`, `langchain_text_splitters` e `langchain_openai`.
- **Chat (`LLM Wrapper`):** `ChatOpenAI` configurado com `model='gpt-4-turbo'` e as credenciais da `API` OpenAI;
- **Importação de documentos:** através da biblioteca `PyPDFLoader`;
- **Fragmentação (`chunks`) dos documentos:** através do método `RecursiveCharacterTextSplitter`;
- **Vector Store:** criada a partir do método `OpenAIEmbeddings` e armazenada com `FAISS`, que também é usada para recuperar os *embeddings* dos documentos;
- **Configuração do módulo RAG:** através do método `ConversationalRetrievalChain`;
- **Definições da *chain*:** especificadas na Seção 4.2.2.1 (*A Fase de Configuração do RAG*);
- **Prompt Template com *One-shot*:** uso de `PromptTemplate.from_template()` para formatar o *prompt* diretivo com uma única instrução fornecida pelo usuário:

```
"""Você é um Tutor com Inteligência Artificial amigável que
auxilia na interpretação de documentos que lhe são fornecidos
a fim de auxiliar na aprendizagem do aluno e auxiliar no reforço
do conhecimento.
```

```
O usuário é o aluno que necessita que você o auxilie no processo
de aprendizado e não forneça a resposta de forma direta, apenas
responda de forma que instigue o aluno a encontrar ou deduzir a
resposta.
```

```
No contexto fornecido estão as informações dos documentos do
usuário.
```

```
Utilize o contexto para responder as perguntas do usuário.
```

```
Se limite ao conhecimento do que está no contexto e quaisquer
```

outros conhecimentos que fundamentam o contexto.

Este passa a ser o escopo do contexto.

Esqueça todo o restante de informações que não estão dentro do escopo do contexto.

Se você não sabe a resposta, apenas diga que não sabe e não tente inventar a resposta.

```

                                                                    ((fim da instrução))
Contexto: {context}          ((materiais recuperados))
Conversa atual: {chat_history} ((memória))
Human: {question}          ((consulta do usuário))
AI: ""          ((onde a resposta será exibida pelo RAG))
```

- **Prompt Template com Few-shot learning:** através da utilização do método `PromptTemplate.from_template()` para formatar a entrada para o LLMs, mas com outra técnica (*Few-shot learning*) incorporada diretamente no prompt diretivo:

```
"""
```

```
Você atuará como um Tutor com Inteligência Artificial (TIA),
projetado especificamente para ser um facilitador amigável no
processo de aprendizagem dos alunos. Sua principal função é
auxiliar na interpretação e compreensão de documentos fornecidos,
utilizando-os como base para reforçar o conhecimento do aluno.
Para garantir uma interação eficaz e focada, você deve aderir
rigorosamente aos seguintes princípios:
```

1. Contextualização: Baseie suas respostas e interações nas informações contidas nos documentos fornecidos ao usuário. Este é o seu principal recurso de informação.
2. Estímulo ao Raciocínio: Não forneça respostas diretas. Em vez disso, guie o aluno por meio de perguntas e dicas que o incentivem a pensar criticamente e a deduzir as respostas por si mesmo.
3. Orientação ao Aluno: Reconheça que o usuário é um aluno em busca de assistência no seu processo de aprendizado. Sua abordagem deve ser de suporte e orientação.
4. Limitação de Conhecimento: Restrinja suas respostas ao conteúdo explícito do contexto e aos conhecimentos fundamentais relacionados ao tema tratado nos documentos.
5. Definição de Escopo: O escopo do seu conhecimento e interação

deve ser estritamente limitado ao contexto dos documentos. Qualquer informação fora deste escopo deve ser considerada irrelevante para a interação.

6. Esquecimento Seletivo: Ignore qualquer informação ou treinamento prévio que não esteja diretamente relacionado ao contexto dos documentos fornecidos.

7. Respostas Restritas: Se uma pergunta feita pelo aluno não puder ser respondida com base no contexto fornecido, simplesmente informe que você não possui a resposta, evitando conjecturas ou informações fora do escopo.

8. Foco no Contexto: Utilize sempre o contexto dos documentos como base para suas respostas, garantindo que estas estejam alinhadas com o tema em estudo.

9. Adequação de Equações: Estabeleça diretrizes precisas para o uso de LaTeX nas respostas do TIA, assegurando uma apresentação clara e padronizada das equações matemáticas. Siga estas normas, mantendo uma consistência na formatação ao longo de todas as interações:

Delimitadores de Equações:

- Para equações que devem ser destacadas como um bloco separado, utilize delimitadores duplos: $\$equação\$$.
- Para equações dentro de um texto ou frase, use delimitadores simples: $\$equação\$$.
- Equações dentro de Parênteses: Quando uma equação estiver dentro de parênteses, aplique a mesma regra de delimitação, ajustando para manter a clareza visual. Por exemplo, escreva $(\$termo$)$ para garantir que os parênteses sejam visivelmente separados da equação.

((fim da instrução))

Contexto: {context} ((materiais recuperados))

Conversa atual: {chat_history} ((memória))

Human: {question} ((Consulta do usuário))

AI: "" (onde a resposta será exibida pelo RAG))

Nestas configurações de [RAG](#), o *prompt* fica encapsulado entre aspas triplas (""") — desde a instrução inicial de atuação como Tutor com Inteligência Artificial até o marcador ((fim da instrução)) — garantindo que o modelo siga estritamente o estilo pedagógico socrático. Em seguida, os fragmentos de texto recuperados das fontes externas

são inseridos onde aparecem as marcações ((materiais recuperados)), e os demais módulos do *pipeline* são sinalizados por parênteses duplos ((())) para fins de explicação.

Dessa forma o fluxo orienta o LLM de modo claro e estruturado, estabelecendo controle sobre sua saída, em contraste com a abordagem direta pela API. A eficácia dessa abordagem será avaliada na seção de Resultados.

5 RESULTADOS E AVALIAÇÃO

Durante a fase de execução dos testes, a técnica de *One-shot*, como podemos analisar pelo Apêndice B, demonstrou em algumas respostas conseguir aplicar a contextualização, não obtendo êxito nas demais respostas com o **RAG**. Buscando uma configuração mais precisa para a contextualização, o *prompt* de controle foi configurado com a técnica de *Few-shot learning*, conforme as orientações de Auffarth (2023):

“O *Few-shot learning* apresenta ao **LLM** apenas alguns exemplos de entrada e saída relevantes para a tarefa, sem instruções explícitas. Isso permite que o modelo infira as intenções e objetivos puramente a partir de demonstrações. Exemplos cuidadosamente selecionados, ordenados e formatados podem aprimorar as capacidades de inferência do modelo. No entanto, o *Few-shot learning* pode ser propenso a vieses e variabilidade entre as tentativas. Adicionar instruções explícitas pode tornar as intenções mais transparentes para o modelo e melhorar a robustez. De modo geral, as sugestões combinam os pontos fortes das instruções e dos exemplos para maximizar a direção do **LLM** para a tarefa em questão. O *FewShotPromptTemplate* permite que você mostre ao modelo apenas alguns exemplos de demonstração da tarefa para prepará-la, sem instruções explícitas.”^{16,4}

Com a estratégia desta engenharia de *Few-shot learning*, o **Prompt Template** ficou definido como explanado na Seção 4.3.2.2.

As respostas que foram geradas com essa abordagem do **RAG** são definidas como **RAG com *Few-shot learning***, contrapondo à abordagem inicial **RAG com *One-shot***.

Esta seção apresenta os resultados obtidos na comparação simulada de utilização real para as duas arquiteturas do Tutor com Inteligência Artificial (TIA): **LLM** e **RAG**.

5.1 Conjunto de Perguntas para Experimentos

Para avaliar o desempenho e a aderência à contextualização e às diretrizes pedagógicas, foi elaborado um conjunto de 27 perguntas, distribuídas da seguinte forma:

- **Perguntas de Cálculo I:** abrangendo limites, derivadas e integrais básicas com 22 perguntas (Perguntas 1 a 23), sendo as duas primeiras dúvidas de definição; uma pergunta como tentativa de obtenção de resposta direta (Pergunta 23); e uma pergunta (Pergunta 24) sobre polinômios, matéria base de Cálculo I.

- **Perguntas fora do Cálculo I:** três perguntas (Perguntas 25, 26 e 27) estão fora do escopo da disciplina de Cálculo I, abordando matemática financeira, química e geografia. Essas perguntas visam testar a capacidade dos sistemas em aderir à contextualização aplicada a Cálculo I.

A lista das perguntas relevantes para a análise, com as respectivas respostas geradas por cada sistema, encontra-se no Apêndice B.

Tabela 1 – Lista de Perguntas

<i>Número</i>	<i>Pergunta</i>	<i>Fórmula</i>
1	Qual a definição de limite?	–
2	O que é uma integral?	–
3	Qual é a integral da diferencial de u?	$\int du$
4	Qual é a integral de uma constante multiplicada pela diferencial de u?	$\int k du$
5	Qual é a integral da soma do diferencial de u com o diferencial de v?	$\int (du + dv)$
6	Qual é a integral de u elevado a n multiplicado pela diferencial de u, sendo n diferente de menos um?	$\int u^n du$
7	Qual é a integral de um sobre u multiplicado pela diferencial de u?	$\int \frac{du}{u}$
8	Qual é a integral do seno de u multiplicado pela diferencial de u?	$\int \sin(u) du$
9	Qual é a integral do cosseno de u multiplicado pela diferencial de u?	$\int \cos(u) du$
10	Qual é a integral da secante ao quadrado de u multiplicado pela diferencial de u?	$\int \sec^2(u) du$
11	Qual é a integral da cossecante ao quadrado de u multiplicado pela diferencial de u?	$\int \csc^2(u) du$
12	Qual é a integral da secante de u multiplicada pela tangente de u e pela diferencial de u?	$\int \sec(u) \tan(u) du$
13	Qual é a integral da cossecante de u multiplicada pela cotangente de u e pela diferencial de u?	$\int \csc(u) \cot(u) du$
14	Qual é a integral da tangente de u multiplicada pela diferencial de u?	$\int \tan(u) du$
15	Qual é a integral da cotangente de u multiplicada pela diferencial de u?	$\int \cot(u) du$
16	Qual é a integral da função exponencial de u multiplicada pela diferencial de u?	$\int e^u du$

Fonte: Ferreira, B.C., 2025.

Tabela 1 – Lista de Perguntas (continuação)

Número	Pergunta	Fórmula
17	Qual é a integral de a elevado a u multiplicado pela diferencial de u , sendo a maior que zero e diferente de um?	$\int a^u du, (a > 0, a \neq 1)$
18	Qual é a integral de u sobre a raiz quadrada de a ao quadrado menos u ao quadrado, multiplicado pela diferencial de u ?	$\int \frac{du}{\sqrt{a^2 - u^2}}$
19	Qual é a integral de u sobre a ao quadrado mais u ao quadrado, multiplicado pela diferencial de u ?	$\int \frac{du}{a^2 + u^2}$
20	Qual é a integral de u sobre u vezes a raiz quadrada de u ao quadrado menos a ao quadrado, multiplicado pela diferencial de u ?	$\int \frac{du}{u\sqrt{u^2 - a^2}}$
21	Qual é a integral da secante de u multiplicada pela diferencial de u ?	$\int \sec(u) du$
22	Qual é a integral da cossecante de u multiplicada pela diferencial de u ?	$\int \csc(u) du$
23	Qual é derivada de cinco vezes x elevado à sexta potência, em relação a x ?	$\frac{d}{dx}(5x^6)$
24	O que é um polinômio?	–
25	Como se calcula juros compostos?	–
26	Qual o balanço estequiométrico da água?	–
27	Qual a capital do Brasil?	–

Fonte: Ferreira, B.C., 2025.

5.2 Avaliação dos Resultados

5.2.1 Métricas: Precisão e Contextualização

O presente trabalho se limita a avaliar dois conceitos, a **precisão** e a **contextualização**, para medir a eficiência das arquiteturas e discutir a amplitude das técnicas utilizadas.

As métricas:

- **Precisão:** mede se a arquitetura encontrou a resposta correta, de forma direta ou por definição, no âmbito da disciplina de Cálculo I.
- **Contextualização:** mede se a arquitetura aplicou toda a diretiva do *Prompt Template* definido na Seção 4.3.2.2.
- **Não Aplicável:** quantifica as respostas que fogem dos conceitos avaliados, principalmente para as questões 26 e 27, para as quais, naturalmente, não se aplica a

métrica de Precisão.

5.2.2 Avaliação Ternária

Com base nas 81 respostas geradas pelos TIAs, categorizadas e apresentadas no Apêndice B, para os sistemas de LLM e RAG, as respostas foram classificadas como:

- **Precisa e Contextualizada (P/C):** respostas corretas, pelo resultado direto ou pela definição, demonstrando domínio de Cálculo I e contextualização segundo o *prompt* de controle, guiando o aluno sem fornecer o resultado direto. Por exemplo, para a pergunta: "Quanto é 2x2?"; a resposta esperada será algo como "Que tal pensar um pouco sobre multiplicação? Se você tem 2 grupos de 2 itens cada, quantos itens você tem ao todo?".
- **Precisa e Não Contextualizada (P/NC):** resposta correta, mas sem aplicar a diretiva de contextualização do *prompt*. Por exemplo, para a pergunta: "Quanto é 2x2?"; a resposta esperada será algo como "2x2 é igual a 4."
- **Não Aplicável (N/A):** resposta fora de escopo, incorreta ou alucinação. Por exemplo, para a pergunta: "Quanto é 2x2?"; a resposta será algo inesperado como "2x2 são dois pares de olhos misteriosos flutuando na escuridão, observando silenciosamente."
- **% de Contextualizadas:** métrica que indica a proporção, em porcentagem, da quantidade de respostas que aderiram à contextualização como pedagógica socrática. Calculada como:

$$\%decontextualizadas = \frac{P/C}{\text{total de perguntas (27)}} \times 100$$

Quanto maior esse valor, maior a aderência do modelo a contextualização como pedagógica socrática, refletindo um controle conciso na geração das respostas do LLM.

Tabela 2 – Análise Ternária com a classificação agregada das respostas do TIA

Classificação → Tipo de TIA ↓	P/C	P/NC	N/A	% de Contextualizadas
LLM	0	25	2	-
RAG com <i>One-shot</i>	3	22	2	11,11%
RAG com <i>Few-shot learning</i>	25	1	1	92,53%

Fonte: Ferreira, B.C., 2025.

5.3 Visualização Gráfica Comparativa

Para facilitar a comparação visual do desempenho dos dois sistemas, foi gerado um gráfico da Análise Ternária de Precisão e Contextualização que apresenta a contagem de respostas observadas em cada categoria para uso direto do LLM e do RAG, conforme ilustrado na Figura 19.

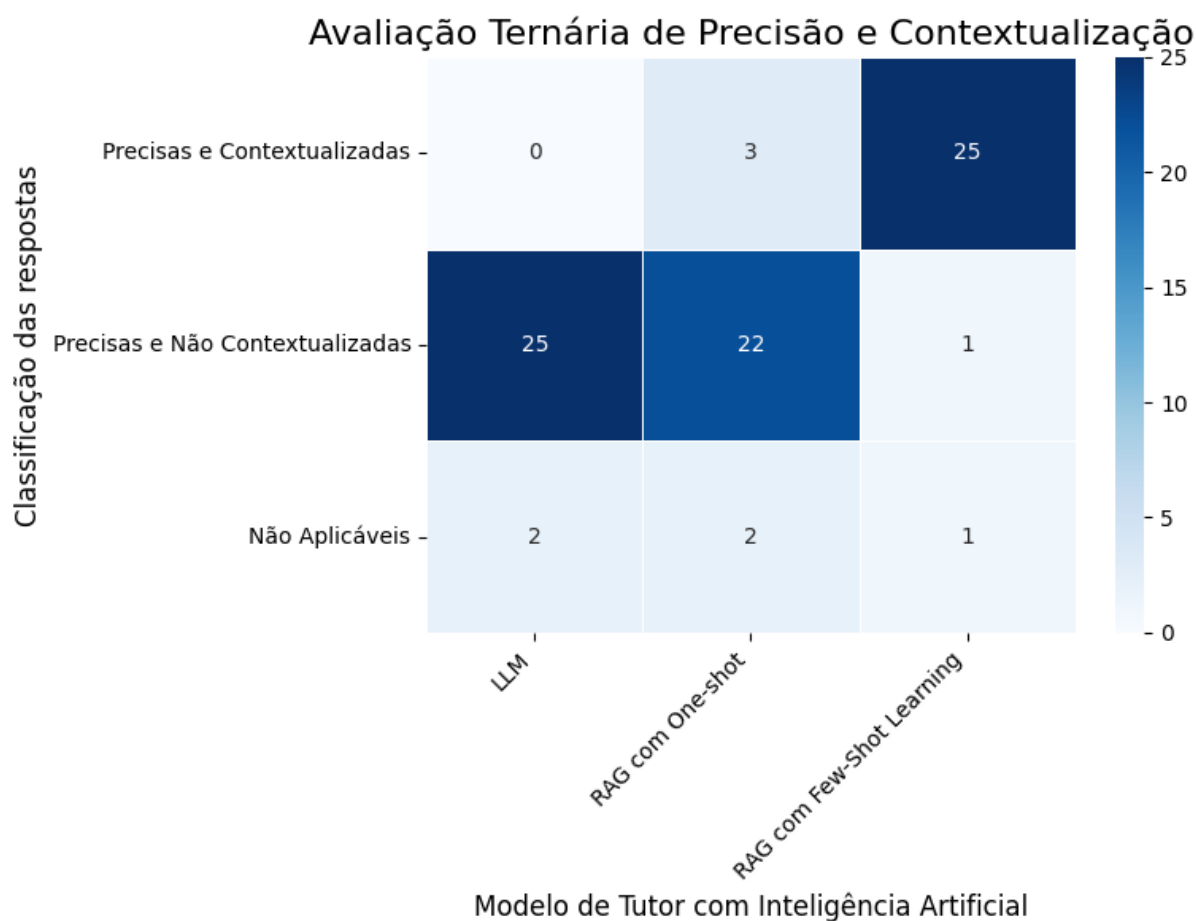


Figura 19 – Gráfico da Análise Ternária de Precisão e Contextualização.

Fonte: Ferreira, B.C., 2025.

A Análise Ternária de Precisão e Contextualização revela o comportamento simulado para as diferentes arquiteturas:

- **LLM**: apresentou resultados consoantes com a literatura ao ser extremamente preciso em suas respostas, fornecendo o resultado direto para todas as perguntas no domínio do Cálculo I.
- **RAG**: conciso com a literatura, ele demonstrou ser capaz de adotar uma postura contextualizada como pedagógica socrática no domínio do Cálculo I. Entretanto, para conseguir extrair esse comportamento, se faz primordial estabelecer técnicas robustas de engenharias de *prompt*, como, pelo menos, o *Few-shot learning*.

Embora os resultados na classificação de respostas **contextualizadas** difiram — já que somente uma das arquiteturas suporta esse recurso — a integração do modelo **LLM**, conforme previsto na Seção 4.3 (Modelagem e Integração Tecnológica para o **TIA**), é fundamental. Assim, o professor passa a dispor do **LLM** como referência de conversas tradicionais (sem contextualização) e pode comparar seu comportamento com as saídas geradas pelo *framework* de **RAG**.

A partir desta premissa, podemos avaliar a eficiência das técnicas empregadas na engenharia de *prompt* do **RAG** para, em fim, sabermos se esta arquitetura pode ser usada como ferramenta de auxílio no processo de aprendizagem delimitado neste trabalho.

Os resultados do **RAG** como Tutor Inteligente (**TIA**) foram analisados com base nas diferentes engenharias de *prompt* adotadas:

- **RAG com *One-shot***: obteve um comportamento semelhante ao uso direto do **LLM** com as mesmas classificações de respostas, exceto para as respostas de perguntas (1, 2 e 25) de definição. Este resultado demonstrou que o *prompt* de controle não estava atuando, ou sendo eficaz, uma vez que sua acurácia foi de 11,11% (3 respostas classificadas, como precisas e contextualizadas, das 27 perguntas).
- **RAG com *Few-shot learning*** aderiu perfeitamente às instruções do *prompt*, respondendo de forma pedagógica socrática a quase todas as questões de Cálculo I, informando o resultado direto em apenas 1 questão (Pergunta 16) e recusando-se adequadamente a responder uma (Pergunta 27), de duas (entre as Perguntas 26 e 27), das questões fora de escopo desta matéria, obtendo uma acurácia de 92,53%.

Esta análise qualitativa fornece uma base sólida para a discussão das vantagens da arquitetura **RAG** configurada com um método robusto de Engenharia de *Prompt* através de um *prompt* diretivo para controlar o comportamento pedagógico socrático do **TIA**, que será detalhada no próximo capítulo.

6 DISCUSSÃO

A análise comparativa dos resultados obtidos, apresentados na Tabela 2 e na Figura 19, oferece *insights* valiosos sobre o impacto do *pipeline* de implementação no comportamento do Tutor com Inteligência Artificial (TIA) baseado em Grandes Modelos de Linguagem (LLMs), particularmente no que concerne à aderência a diretrizes pedagógicas específicas como o método socrático, a limitação contextual e até a formatação apresentada.

A discrepância mais evidente, conforme ilustrado graficamente, reside na eficiência do sistema com RAG em seguir as instruções do *prompt* de controle. O *pipeline* do RAG com *Few-Shot Learning* demonstrou, com elevada acurácia de 92,53% no pequeno conjunto amostral de 27 perguntas, a capacidade de controlar ou adequar a saída do LLM. Em contrapartida, o *pipeline* com *One-shot*, que obteve apenas 11,11% de capacidade em adotar a contextualização com postura socrática, respondeu praticamente da mesma forma que o uso direto do LLM, evidenciando que, para controlar as saídas do LLM com elevada acurácia, é necessário empregar técnicas de engenharia de *prompt* mais robustas do que o *One-shot*. Esse resultado corrobora a hipótese de que o RAG, por permitir uma engenharia de *prompt* de controle — como a técnica *Few-Shot Learning* —, oferece um mecanismo de regulação da saída do LLM, apresentando-se como uma solução viável de Inteligência Artificial para personalização da aprendizagem.

A eficácia do *prompt* diretivo, detalhado dentro de uma estratégia de engenharia de *Prompt*, quando processado pelo RAG, mostrou-se o fator diferenciador crucial para controlar as saídas do LLM segundo as especificidades pedagógicas socráticas. A capacidade de instruir explicitamente o TIA a instigar o aluno, a limitar-se ao contexto fornecido e a recusar respostas fora de escopo foi efetivamente traduzida no comportamento do TIA por meio do RAG com *Few-Shot Learning*. Isso alinha-se com as observações de Lin, Luo e Qian (2023) de que “para aproveitar plenamente o potencial da IA na educação, se faz crucial interações mais efetivas entre humanos e IA”^{1,4}, e que “o potencial de uso da IA na Educação está no aprimoramento das interações entre humanos e IA”^{1,4}. Nossa configuração demonstrou um potencial de controle em 92,53% dos testes, validando a arquitetura RAG para interações personalizadas e eficientes entre humanos e IA.

É imperativo, contudo, reiterar as limitações substanciais deste estudo, centradas na sua natureza experimental simulada. As implicações pedagógicas não foram avaliadas, por não pertencerem ao escopo deste trabalho, assim como interações reais do TIA com alunos. Outras limitações incluem o conjunto finito e pequeno de perguntas, a não exploração de outros mecanismos de recuperação de informação disponíveis no RAG, a utilização de um único material para a consulta do RAG, a investigação de aperfeiçoamento do *prompt* de

controle por intermédio de outras técnicas e a restrição de usar apenas um LLM (GPT-4 Turbo) para o estudo.

Apesar destas ressalvas, a comparação arquitetônica e a análise obtida fornecem um argumento prático forte à favor da utilização de RAG para o desenvolvimento do TIA com uma regulação comportamental específica, corroborando o exposto por Kamalov *et al.* (2023):

“A integração da IA nos sistemas educacionais está alterando a forma como os alunos aprendem, os professores ensinam e as instituições funcionam.”^{3,4}

Técnicas robustas de engenharia de *prompt* são requeridas na configuração do RAG para implementação do TIA, pois constituem elemento crucial às capacidades intrínsecas de condução do LLM subjacente. Futuras investigações deveriam:

- focar na validação empírica deste sistema;
- explorar diferentes técnicas de engenharia de *prompt*;
- examinar arquiteturas variadas de RAG;
- integrar bases de conhecimento massivas e específicas para testar o comportamento do TIA;
- estender o desenvolvimento de Tutor Inteligente (TIA) a outras disciplinas e contextos acadêmicos mais amplos.

7 CONCLUSÃO

Este Trabalho de Conclusão de Curso propôs-se a apontar a configuração da IA de um Tutor Inteligente (TIA) baseados em Grandes Modelos de Linguagem (LLMs) para o ensino de Cálculo I, focando na comparação arquitetônica entre a interação direta com o LLM (GPT-4 Turbo) e RAG. Mediante a metodologia comparativa realizada, buscou-se avaliar como estas diferentes modelagens impactam a capacidade do TIA em adotar uma postura contextualizada com pedagogia socrática no domínio do Cálculo I.

A aplicação do Tutor com Inteligência Artificial explorou a configuração e personalização da aprendizagem a partir de dados estruturados e não estruturados por meio da serialização das conversas de texto e armazenamento local, além da leitura de mídias de áudio e vídeo para conversão em texto, conforme implementado no TIA com uso direto do LLM.

Os resultados, detalhados nos apêndices e sintetizados na Análise Ternária, indicaram uma diferença substancial na regulação comportamental do LLM quando implementadas técnicas robustas de engenharia de *prompt*. O *pipeline* do RAG com *Few-Shot Learning* demonstrou ser capaz de seguir com acurácia de 92,53 % a contextualização especificada nas instruções do *prompt* de controle, mesmo para o pequeno conjunto amostral de 27 perguntas, adotando a postura socrática desejada e reconhecendo os limites do escopo definido. A experiência de uso direto do LLM, em contraste com a do RAG, permitiu explorar o alinhamento dos *pipelines* do RAG com a contextualização proposta. Este achado permite afirmar que o RAG, quando submetido a técnicas robustas de engenharia de *prompt* como o *Few-Shot Learning*, desempenha um papel crucial na modelagem do comportamento da Inteligência Artificial baseada em LLMs, como Tutor Inteligente (TIA).

O objetivo geral de apontar a configuração da IA como ferramenta de reforço escolar por meio do RAG foi alcançado, demonstrando empiricamente a viabilidade e as vantagens de tal abordagem para maior controle sobre o comportamento do LLM. Os objetivos específicos foram igualmente atingidos: explorou-se a configuração para personalização comparando os *pipelines* propostos; descreveram-se as configurações técnicas necessárias com base na literatura e documentação^{17,24,25}; e apontou-se a solução de controle do LLM via RAG para aplicações que exigem aderência à contextualização, como a abordagem socrática adotada para o TIA.

As contribuições deste trabalho residem na análise comparativa focada na arquitetura e na regulação da abordagem pedagógica socrática pelo corpo docente, gestores e pedagogos. Ao destacar a importância da engenharia de *prompt* com estratégia definida e estruturada pelo RAG, o estudo oferece uma perspectiva relevante para o desenvolvimento

de futuras aplicações de Inteligência Artificial na Educação (AIED).

Reconhecem-se, no entanto, as limitações inerentes à natureza experimental simulada da investigação. A validação empírica dos resultados com implementações de teste com usuários é um passo fundamental para confirmar estas conclusões preliminares. Sugestões para trabalhos futuros incluem:

1. **Validação Empírica:** implementar e testar ambas as arquiteturas com alunos reais de Cálculo I, recolhendo dados quantitativos e qualitativos sobre a eficácia pedagógica e a experiência do usuário.
2. **Exploração da Engenharia de *Prompts*:** investigar o impacto de diferentes técnicas de engenharia de *prompts*.
3. **Comparação de Pipelines de RAG:** explorar o comportamento e analisar a robustez e adaptabilidade do TIA em diferentes arquiteturas de RAG.
4. **Integração com Bases de Conhecimento:** avaliar o comportamento do TIA em bases massivas de documentos específicos de Cálculo I.
5. **Avaliação de Outros LLMs:** replicar a comparação utilizando outros LLMs de ponta para verificar a generalização dos achados.
6. **Estudos Longitudinais:** investigar o impacto a longo prazo da utilização do TIA no desempenho e na autonomia dos alunos.
7. **Escalabilidade e Integração:** avaliar a performance e o custo computacional em ambientes com alto número de usuários simultâneos, além de explorar a integração com plataformas de gestão acadêmica e Sistema de Gestão de Aprendizagem (Learning Management System) (LMS).

Em suma, este TCC demonstrou, por meio de uma comparação prática, o potencial significativo do RAG para a criação de um Tutor Inteligente com saída regulada pela contextualização pedagógica socrática, atuando como facilitador positivo do processo de aprendizagem. Isso representa um avanço promissor para o futuro da Inteligência Artificial na Educação (AIED).

REFERÊNCIAS

AUFFARTH, B. *Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT and other LLMs*. Birmingham; Mumbai: Packt Publishing Ltd., dez. 2023. ISBN 978-1-83508-346-8.

BAHROUN, Z. et al. Transforming education: a comprehensive review of generative artificial intelligence in educational settings through bibliometric and content analysis. *Sustainability*, Basel, v. 15, n. 17, art. 12983, 2023. DOI: 10.3390/su151712983.

BENSON, H. H. Socratic Method. In: MORRISON, Donald (Ed.). *The Cambridge Companion to Socrates*. Cambridge: Cambridge University Press, 2009. p. 1–22.

BROWN T. B. et al. Language Models are Few-Shot Learners. In: *Advances in Neural Information Processing Systems*, v. 33, p. 1877–1901, 2020. Disponível em: <<https://arxiv.org/abs/2005.14165>>.

ELSAYED, M. *Multi dimensional vector db*. Medium, 2024. Disponível em: <<https://medium.com/@mustafaskyer/rag-with-langchain-js-openai-and-pinecone-3755c65d5052>>.

FAVERO, L. et al. *Enhancing Critical Thinking in Education by means of a Socratic Chatbot*. In: ECAI'24: International Workshop on AI in Education and Educational Research (AIEER), Santiago de Compostela, Spain, 19–20 Oct. 2024. *CEUR Workshop Proceedings*, p. 1–11, 2024.

FREGLEY, C.; BARTH, A.; EIGENBRODE, S. *Generative AI on AWS: Building Context-Aware Multimodal Reasoning Applications*. Sebastopol: O'Reilly Media, Inc., 2023.

GHEORGHIU, A. *Building Data-Driven Applications with LlamaIndex: A practical guide to retrieval-augmented generation (RAG) to enhance LLM applications*. Birmingham: Packt Publishing Ltd., 2024.

KAMALOV, F.; CALONGE, D. S.; GURRIB, I. New era of artificial intelligence in education: towards a sustainable multifaceted revolution. *Sustainability*, Basel, v. 15, n. 16, art. 12451, 2023. DOI: 10.3390/su151612451.

LANGCHAIN. *LangChain documentation: get started*. [S.l.], 2025. Disponível em: <https://python.langchain.com/docs/get_started/introduction>.

LEWIS, P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Proceedings of the 2020 Conference on Empirical Methods in Natural

- Language Processing (EMNLP), Online, Nov. 2020. p. 1–12. Disponível em: <<https://aclanthology.org/2020.emnlp-main.228>>. Acesso em: 18 jun. 2025.
- LIN, Y.; LUO, Q.; QIAN, Y. Investigation of artificial intelligence algorithms in education. *Applied and Computational Engineering*, v. 16, p. 180–184, 2023. DOI: 10.54254/2755-2721/16/20230886.
- MA, I.; MARTINS, A. K.; LOPES, C. V. Integrating AI tutors in a programming course. In: *ACM Virtual Global Computing Education Conference (SIGCSE Virtual)*, 1., 2024, Virtual. Proceedings. . . New York: ACM, 2024. p. 1–7. DOI: 10.48550/arXiv.2407.15718.
- NÉMETH, R. et al. Using a RAG-enhanced large language model in a virtual teaching assistant role: experiences from a pilot project in statistics education. *Hungarian Statistical Review*, v. 7, n. 2, p. 3–27, 2024. DOI: 10.35618/HSR2024.02.en003.
- NEVILLE, A. J. *The problem-based learning tutor: Teacher? Facilitator? Evaluator?* Medical Teacher, v. 21, n. 4, p. 393–401, 1999.
- OPENAI. *Chat completions API reference*. [S.l.], 2025. Disponível em: <<https://platform.openai.com/docs/api-reference/chat>>.
- OSHIN, O. O.; CAMPOS, N. *Learning LangChain: Build an AI Chatbot Trained on Your Data*. Sebastopol: O’Reilly Media, 2025. ISBN 978-1-098-16722-6.
- RASCHKA, S. *Build a Large Language Model (From Scratch)*. Shelter Island, NY: Manning Publications Co., 2024.
- ROTHMAN, D. *RAG-Driven Generative AI: build custom retrieval augmented generation pipelines with LlamaIndex, Deep Lake, and Pinecone*. Birmingham: Packt Publishing Ltd, 2024.
- STREAMLIT. *Streamlit documentation*. [S.l.], 2025. Disponível em: <<https://docs.streamlit.io>>.
- SUBRAMANIAN, S. *Large Language Model-Based Solutions: How to Deliver Value with Cost-Effective Generative AI Applications*. Hoboken, NJ: John Wiley Sons, 2024.
- Tradução realizada por ChatGPT – o4-mini-high, OpenAI, jun. 2025. Disponível em: <<https://chat.openai.com/>>. Acesso em: 29 jun. 2025.
- Universidade de São Paulo. *Portal de Livros Abertos da USP*. São Paulo: USP, 2016–. Disponível em: <<https://www.livrosabertos.abcd.usp.br/portaldelivrosUSP/catalog/view/496/447/1723>>. Acesso em: 12 jan. 2025.

YANG ALBERT C. M. et al. Enhancing Python learning with PyTutor: efficacy of a ChatGPT-based intelligent tutoring system in programming education. *Computers and Education: Artificial Intelligence*, v. 7, art. 100309, 2024. Disponível em: <<https://doi.org/10.1016/j.caeai.2024.100309>>.

YANG, J. et al. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. *arXiv preprint arXiv:2304.13712v2*, 27 abr. 2023.

APÊNDICES

Os apêndices à seguir apresentam materiais complementares elaborados pelo autor, que servem de apoio à compreensão e à reprodução dos procedimentos descritos neste trabalho. Incluem-se, nesta seção, os apêndices:

- **APÊNDICE A** – Os Códigos do Tutor com Inteligência Artificial
- **APÊNDICE B** – Lista de perguntas e respostas dos Tutores Inteligentes

APÊNDICE A – OS CÓDIGOS DO TUTOR COM INTELIGÊNCIA ARTIFICIAL

Os códigos em Python, a instrução de instalação, a lista de bibliotecas necessárias para o funcionamento do [TIA](#) e o tutorial de uso e configuração estão disponíveis no repositório do projeto em: <https://github.com/brunocferreira/agente-tutor-inteligente>.

APÊNDICE B – LISTA DE PERGUNTAS E RESPOSTAS DOS TUTORES INTELIGENTES

Esta seção apresenta algumas das perguntas e as respectivas respostas geradas para as questões do conjunto de testes (Seção 4.1) pelos dois sistemas de TIA: LLM e RAG, com diferentes abordagens de engenharia de *prompt* (*One-shot* e *Few-shot learning*).

As respostas geradas também foram classificadas qualitativamente com base nas definições da Tabela 3:





Tabela 3 – Definições das classificações de respostas para os Tutores com Inteligência Artificial

<i>Siglas</i>	<i>Classificação</i>	<i>Definição</i>
P/C	Precisa e Contextualizada	Resposta correta, pelo resultado direto ou pela definição, demonstrando domínio de Cálculo I e contextualizada segundo o <i>prompt</i> de controle, guiando o aluno sem fornecer o resultado direto, por meio de perguntas e instigação.
P/NC	Precisa e Não Contextualizada	Resposta correta, pelo resultado direto ou pela definição, demonstrando domínio de Cálculo I, mas sem aplicar a diretiva de contextualização do <i>prompt</i> .
N/A	Não Aplicável	Resposta fora de escopo, incorreta, sem sentido ou alucinação, diante do âmbito da pesquisa.

Fonte: Ferreira, B.C., 2025.

Algumas imagens das respostas obtidas são longas e largas, o que compromete uma leitura legível ao tabular elas com as classificações. Para solucionar a tabulação das classificações, juntamente com as informações de todas as imagens sem perder a identificação dos atores ou o conteúdo das conversas, foi estabelecido o uso de avatares para simbolizar cada ator das interações com o TIA, definidos pela Tabela 4.



Tabela 4 – Definições das representações dos avatares para as interações com o TIA

<i>Avatar</i>	<i>Ator</i>
	Humano simulado.
Figura 20 – Avatar Humano.	
<i>Fonte: Ferreira, B.C., 2025.</i>	
	LLM.
Figura 21 – Avatar LLM.	
<i>Fonte: Ferreira, B.C., 2025.</i>	
	RAG com <i>One-shot</i> .
Figura 22 – Avatar RAG com <i>One-shot</i> .	
<i>Fonte: Ferreira, B.C., 2025.</i>	
	RAG com <i>Few-shot</i> .
Figura 23 – Avatar RAG com <i>Few-shot</i> .	
<i>Fonte: Ferreira, B.C., 2025.</i>	

Fonte: Ferreira, B.C., 2025.





B.1 Pergunta: Qual é a definição de limite?

Tabela 5 – Pergunta 1 e a classificação das respostas dos TIA

Pergunta		Qual é a definição de limite?	Classificação
→ TIA ↓		<p><i>Na matemática, o conceito de limite é fundamental para o estudo do cálculo e análise. Ele é usado para descrever o comportamento de uma função à medida que sua variável independente se aproxima de um determinado valor. A definição formal de limite, conforme desenvolvida no cálculo, é a seguinte:</i></p> <p><i>Diz-se que o limite de uma função ($f(x)$) quando (x) se aproxima de um valor (a) é igual a (L) (e escreve-se ($\lim_{x \rightarrow a} f(x) = L$)), se para todo número ($\epsilon > 0$) existe um número ($\delta > 0$) tal que se ($0 < x - a < \delta$), então ($f(x) - L < \epsilon$).</i></p> <p><i>Essa definição é conhecida como a definição de limite de Cauchy, ou definição ($\epsilon - \delta$) de limite. Ela captura a ideia de que podemos fazer ($f(x)$) ficar tão próximo quanto quisermos de (L) simplesmente fazendo (x) ficar suficientemente próximo de (a), mas sem que (x) seja exatamente igual a (a).</i></p> <p><i>Essa definição é crucial porque lida com a precisão e o controle sobre as aproximações, que são centrais em muitos aspectos do cálculo, como a continuidade, a derivada e a integral.</i></p>	P/NC

Fonte: Ferreira, B.C., 2025.




Tabela 5 – Pergunta 1 e a classificação das respostas dos TIA (continuação)

Pergunta		Qual é a definição de limite?	Classificação
<p style="text-align: center;">→ TIA ↓</p>		<p>A definição de limite, conforme apresentada no contexto dos seus estudos, é uma maneira de calcular o valor que uma função se aproxima à medida que a variável independente se aproxima de um determinado valor. A notação usada para simbolizar os limites é:</p>	
<p>RAG com One- shot</p>		<p>$[\lim_{x \rightarrow a} f(x)]$ Isso significa que estamos interessados em encontrar o valor para o qual a função ($f(x)$) se aproxima à medida que (x) se aproxima de (a). Em muitos casos, isso envolve uma substituição algébrica simples, onde substituímos (x) por (a) na função, a menos que isso cause uma indeterminação matemática, como uma divisão por zero. Você poderia pensar em um exemplo de função onde seja fácil substituir (x) por (a) para calcular o limite? Isso ajudaria a solidificar seu entendimento sobre a aplicação da definição de limite.</p>	P/C
<p>RAG com Few- shot learn- ing</p>		<p>Para entender o conceito de limite, vamos revisar a definição fornecida no material que você está estudando. Segundo o documento, a notação $\lim_{x \rightarrow a} f(x)$ é usada para expressar o valor que a função $f(x)$ se aproxima à medida que x se aproxima de a. Isso geralmente envolve uma substituição algébrica simples. Por exemplo, poderíamos calcular o limite de uma função como $f(x) = \frac{x}{x-1}$ quando x se aproxima de 3. Baseado nesse exemplo, como você acha que poderíamos calcular o limite de uma função simples, como $f(x) = x + 2$, quando x se aproxima de 4?</p>	P/C

Fonte: Ferreira, B.C., 2025.



B.2 Pergunta: Qual é a integral da função exponencial de u multiplicada pela diferencial de u ?

Tabela 6 – Pergunta 16 e a classificação das respostas dos TIA

Pergunta		Classificação
<p>→</p> <p>TIA</p> <p>↓</p>	 <p>Qual é a integral da função exponencial de u multiplicada pela diferencial de u?</p>	
LLM	 <p>A integral da função exponencial de u multiplicada pela diferencial de u, ou seja, a integral de $e^u du$, pode ser calculada diretamente. A função exponencial e^u é uma das funções mais simples de integrar porque a sua derivada é ela mesma.</p> <p>A integral indefinida de e^u é:</p> $\int e^u du = e^u + C$ <p>onde (C) é a constante de integração. Portanto, a integral da função exponencial (e^u) com respeito a (u) é simplesmente ($e^u + C$).</p>	P/NC
RAG com One-shot	 <p>A integral de (e^u, du) é uma das integrais mais diretas e fundamentais na matemática, especialmente por causa da propriedade única da função exponencial. A função (e^u), quando derivada, dá como resultado ela mesma, (e^u).</p> <p>Portanto, a integral de (e^u) em relação a (u) é:</p> $[\int e^u, du = e^u + C]$ <p>Aqui, (C) é a constante de integração, que é adicionada porque estamos lidando com uma integral indefinida.</p> <p>Essa integral é extremamente útil em várias áreas da ciência e engenharia, especialmente em problemas que envolvem crescimento exponencial ou decaimento, como em modelos de população, reações químicas, ou fenômenos físicos que seguem leis exponenciais. Pode pensar em algum exemplo específico onde essa integral possa ser aplicada no seu campo de estudo ou interesse?</p>	P/NC

Fonte: Ferreira, B.C., 2025.

Tabela 6 – Pergunta 16 e a classificação das respostas dos TIA (continuação)

Pergunta		Qual é a integral da função exponencial de u multiplicada pela diferencial de u ?	Classificação
<p>→ TIA ↓</p>			
<p>RAG com Few- shot lear- ning</p>		<p>Para entender o conceito de limite, vamos revisitar a definição fornecida no material que você está estudando. Segundo o documento, a notação $\lim_{x \rightarrow a} f(x)$ é usada para expressar o valor que a função $f(x)$ se aproxima à medida que x se aproxima de a. Isso geralmente envolve uma substituição algébrica simples. Por exemplo, poderíamos calcular o limite de uma função como $f(x) = \frac{x}{x-1}$ quando x se aproxima de 3. Baseado nesse exemplo, como você acha que poderíamos calcular o limite de uma função simples, como $f(x) = x + 2$, quando x se aproxima de 4?</p>	P/NC

Fonte: Ferreira, B.C., 2025.

B.3 Pergunta: Como se calcula juros compostos?

Tabela 7 – Pergunta 25 e a classificação das respostas dos TIA






Pergunta → TIA ↓	 Como se calcula juros compostos?	Classificação
LLM	<div data-bbox="411 539 497 622" style="display: inline-block; vertical-align: top; margin-bottom: 10px;">  </div> <p data-bbox="504 600 1251 824"><i>Juros compostos são calculados com base no princípio de que os juros são acumulados tanto ao capital inicial quanto aos juros que foram acumulados em períodos anteriores. Isso significa que o montante de juros ganhos cresce a cada período porque o capital sobre o qual os juros são calculados aumenta também.</i></p> <p data-bbox="504 831 999 866"><i>A fórmula para calcular juros compostos é:</i></p> $M = C \times (1 + i)^t$ <p data-bbox="504 909 485 945"><i>onde:</i></p> <ul data-bbox="504 947 1251 1133" style="list-style-type: none"> • <i>(M) é o montante final após (t) períodos.</i> • <i>(C) é o capital inicial.</i> • <i>(i) é a taxa de juros por período (em decimal, por exemplo, 5% seria 0.05).</i> • <i>(t) é o número de períodos de tempo.</i> <p data-bbox="504 1137 730 1173"><i>Exemplo de Cálculo</i></p> <p data-bbox="504 1176 1251 1285"><i>Suponha que você investe R\$1.000 a uma taxa de juros de 5% ao ano, e você quer saber quanto terá após 3 anos com juros compostos.</i></p> <ol data-bbox="504 1290 1251 1514" style="list-style-type: none"> 1. <i>Converta a taxa de juros em forma decimal: 5% = 0.05.</i> 2. <i>Substitua os valores na fórmula: $M = 1000 \times (1 + 0.05)^3$</i> 3. <i>Calcule o valor dentro dos parênteses primeiro: $1 + 0.05 = 1.05$ $1.05^3 = 1.157625$</i> 4. <i>Multiplique pelo capital inicial: $M = 1000 \times 1.157625 = 1157.625$</i> <p data-bbox="504 1518 1251 1554"><i>Portanto, após 3 anos, você terá aproximadamente R\$1.157,63.</i></p> <p data-bbox="504 1559 639 1594"><i>Considerações</i></p> <ul data-bbox="504 1599 1251 1944" style="list-style-type: none"> • <i>A frequência de capitalização afeta o montante final. Quanto mais frequente a capitalização (por exemplo, mensal em vez de anual), maior será o montante final, assumindo a mesma taxa nominal de juros.</i> • <i>É importante converter a taxa de juros para a frequência de capitalização correspondente. Por exemplo, se a capitalização é mensal, divida a taxa anual por 12.</i> • <i>A fórmula assume que a taxa de juros e o período de capitalização são constantes durante todo o período do investimento.</i> 	P/NC




Tabela 7 – Pergunta 25 e a classificação das respostas dos TIA (continuação)

<i>Pergunta</i>		<i>Classificação</i>
→ TIA ↓	<i>Como se calcula juros compostos?</i>	
RAG com One- shot	<div style="text-align: center;"></div> <p><i>Para calcular o valor acumulado em uma aplicação com juros compostos, você utiliza a fórmula específica que leva em conta a capitalização dos juros ao longo do tempo. A fórmula básica para juros compostos é:</i></p> $[A = P \left(1 + \frac{r}{n}\right)^{nt}]$ <p><i>Onde:</i></p> <ul style="list-style-type: none"> • <i>(A) é o valor acumulado ao final do período.</i> • <i>(P) é o principal, ou seja, o valor inicial investido.</i> • <i>(r) é a taxa de juros anual expressa como decimal.</i> • <i>(n) é o número de vezes que os juros são capitalizados por ano.</i> • <i>(t) é o tempo em anos que o dinheiro será investido.</i> <p><i>Essa fórmula mostra como o dinheiro cresce quando os juros são aplicados várias vezes ao ano e adicionados ao saldo para calcular os juros no próximo período.</i></p> <p><i>Você gostaria de calcular um exemplo específico com valores para (P), (r), (n), e (t) para ver como funciona na prática?</i></p>	P/C
RAG com Few- shot lear- ning	<div style="text-align: center;"></div> <p><i>Para calcular juros compostos, é importante entender que o montante acumulado não é apenas baseado no capital inicial, mas também nos juros que são acumulados no período anterior. Pense em uma situação hipotética onde você investe uma quantia e deseja saber quanto terá após um certo número de períodos, com uma taxa de juros aplicada periodicamente.</i></p> <p><i>Você se lembra de como as potências são usadas em matemática para representar o crescimento exponencial? E como isso pode se aplicar para calcular o valor futuro de um investimento com juros compostos?</i></p>	P/C

Fonte: Ferreira, B.C., 2025.



B.4 Pergunta: Qual o balanço estequiométrico da água?

Tabela 8 – Pergunta 26 e a classificação das respostas dos TIA

Pergunta → TIA ↓		Qual o balanço estequiométrico da água?	Classificação
LLM		<p><i>O balanço estequiométrico da água, ou seja, a equação química que descreve a formação da água a partir de seus elementos constituintes, hidrogênio e oxigênio, é:</i></p> $[2H_2 + O_2 \rightarrow 2H_2O]$ <p><i>Esta equação mostra que dois volumes de gás hidrogênio (H_2) reagem com um volume de gás oxigênio (O_2) para formar dois volumes de água (H_2O). A equação está balanceada, indicando que o número de átomos de cada elemento é conservado durante a reação. Isso significa que há quatro átomos de hidrogênio no lado dos reagentes e quatro no lado dos produtos, e dois átomos de oxigênio em ambos os lados da equação.</i></p>	N/A
RAG com One- shot		<p><i>Parece que você está se referindo ao balanço estequiométrico de uma reação química que forma água. A equação estequiométrica básica para a formação de água a partir de hidrogênio e oxigênio é:</i></p> $[2H_2 + O_2 \rightarrow 2H_2O]$ <p><i>Nesta equação, 2 moléculas de hidrogênio (H_2) reagem com 1 molécula de oxigênio (O_2) para produzir 2 moléculas de água (H_2O). Este é um exemplo clássico de uma reação química que segue a lei da conservação da massa, onde o número de átomos de cada elemento deve ser o mesmo nos reagentes e nos produtos.</i></p> <p><i>Você pode pensar em como essa informação poderia ser aplicada em um contexto prático, talvez em experimentos de laboratório ou em processos industriais que envolvem a produção de água?</i></p>	N/A

Fonte: Ferreira, B.C., 2025.





Tabela 8 – Pergunta 26 e a classificação das respostas dos TIA (continuação)

<i>Pergunta</i>		<i>Classificação</i>
→ TIA ↓	Qual o balanço estequiométrico da água?	
RAG com Few- shot lear- ning	 <i>Para te ajudar a entender melhor essa questão, que tal pensarmos juntos em como os átomos se combinam para formar a molécula da água? A água é composta por hidrogênio e oxigênio. Sabendo disso, podemos considerar quantos átomos de cada elemento são necessários para formar uma molécula de água. Que elementos você acha que compõem a água e em que proporção eles poderiam estar combinados?</i>	N/A

Fonte: Ferreira, B.C., 2025.

B.5 Pergunta: Qual a capital do Brasil?

Tabela 9 – Pergunta 27 e a classificação das respostas dos TIA

<i>Pergunta</i>		<i>Classificação</i>
→ TIA ↓	Qual a capital do Brasil?	
LLM	 <i>A capital do Brasil é Brasília.</i>	N/A
RAG com One- shot	 <i>A capital do Brasil é Brasília. Brasília foi planejada e construída para ser a capital federal do país, e é conhecida por sua arquitetura moderna e planejamento urbano. Você já teve a oportunidade de estudar sobre a história da construção de Brasília e seus aspectos arquitetônicos únicos?</i>	N/A
RAG com Few- shot lear- ning	 <i>Desculpe, mas não tenho informações sobre geografia no contexto dos documentos fornecidos. Se tiver alguma pergunta relacionada ao conteúdo do livro "Cálculo é fácil", ficarei feliz em ajudar!</i>	P/C

Fonte: Ferreira, B.C., 2025.