Benchmarking Local Large Language Models for an Intelligent Tutoring System in Digital Logic

Flaviano M. S. Junior [1] **, Derick S. S. Leite** [1] **, Luiz H. C. da Silva** [1] **,**
Marcos A. Luz [1] **, Túlio R. A. Tavares** [1] **, José N. S. Combé**[1] **,**
Maria V. P. Araújo [1] **, Marcus M. Bezerra** [1] **, Sérgio N. Silva** [1]

[1]Universidade Federal de Campina Grande (UFCG)
Campina Grande – PB – Brasil

{flaviano.junior, derick.leite, luiz.henrique.silva,
marcos.antonio.luz, tulio.tavares, jose.combe,
maria.vitoria.araujo}@ee.ufcg.edu.br
marcus.bezerra@virtus.ufcg.edu.br, sergionatan@dee.ufcg.edu.br

Abstract. *Intelligent Tutoring Systems (ITS) powered by Large Language Models (LLMs) face a critical trade-off between response quality and deployment constraints. While cloud-based models offer high performance, privacy concerns, recurring costs, and connectivity requirements motivate the adoption of locally hosted open-source alternatives. However, systematic evaluations comparing local 7–8B parameter models for Portuguese-language technical tutoring remain scarce. This paper presents a controlled benchmark of four local LLM models (Llama 3.1 8B Q8, Qwen 2.5 7B, Socratic Tutor Qwen 2.5, and Mathstral 7B Q4_K_M) deployed on consumer-grade hardware (NVIDIA RTX 4060 8 GB) within the TutorIA multi-collection RAG pipeline for Logic Circuits instruction. Using a standardized set of 45 questions evaluated through the RAGAS framework, a composite quality index, and one-way ANOVA, results reveal that Mathstral 7B achieves the highest composite score (0.618, statistically tied with Llama 3.1 within overlapping 95% confidence intervals) while exhibiting zero refusal rate and the best answer correctness (0.675). A four-group ANOVA with post-hoc tests confirms significant differences between models in faithfulness ($p_{FDR} = 0.004$), driven specifically by the misaligned Socratic Tutor rather than by parameter count. These findings provide practical guidance for selecting local models in resource-constrained educational deployments.*

Resumo. *Sistemas tutores inteligentes (Intelligent Tutoring Systems - ITS) baseados em modelos de linguagem de grande escala Large Language Models - LLMs) enfrentam um dilema entre qualidade e restrições de implantação. Modelos em nuvem oferecem alto desempenho, porém questões de privacidade, custos recorrentes e dependência de conectividade motivam o uso de alternativas locais de código aberto. Todavia, avaliações sistemáticas comparando modelos locais de 7–8B parâmetros para tutoria técnica em português são escassas. Este artigo apresenta um benchmark controlado de quatro modelos locais (Llama 3.1 8B Q8, Qwen 2.5 7B, Socratic Tutor Qwen 2.5 e Mathstral 7B Q4_K_M) implantados em hardware voltado ao consumidor (NVIDIA RTX 4060 8 GB) no pipeline RAG multi-collection do TutorIA para circuitos lógicos. Empregando 45 perguntas avaliadas pelo framework de avaliação de geração aumentada por recuperação (Retrieval-Augmented Generation Assessment - RAGAS), um índice composto de*

qualidade e ANOVA one-way, os resultados revelam que o Mathstral 7B alcança o maior escore composto (0,618, estatisticamente empatado com o Llama 3.1, com intervalos de confiança de 95% sobrepostos), com taxa de recusa zero e melhor acurácia (0,675). Uma ANOVA de quatro grupos com pós-teste confirma diferenças significativas em fidelidade ($p_{FDR} = 0{,}004$), atribuíveis especificamente ao desalinhamento do Socratic Tutor, e não à contagem de parâmetros. Os achados orientam a seleção de modelos locais em cenários educacionais com recursos limitados.

1. Introdução

A disciplina de circuitos lógicos é um componente fundamental em cursos superiores de engenharia de computação, ciência da computação e engenharia elétrica, englobando conceitos que variam desde a álgebra booleana elementar até o projeto complexo de máquinas de estados finitos. Historicamente, disciplinas de fundamentos lógicos e de programação apresentam altos índices de reprovação no contexto acadêmico brasileiro: Bosse e Gerosa (2015) documentaram taxa de reprovação de 30% em disciplinas de introdução à programação na USP, ultrapassando 50% em casos específicos, com os cursos da área de computação entre os de maior evasão no ensino superior, chegando a 36%, contra a média nacional de 22%. Essa realidade é agravada pela escassez de suporte individualizado fora da sala de aula, motivando a investigação de tutores inteligentes como estratégia de apoio.

Com o advento dos modelos de linguagem de grande escala (LLMs), tornou-se tecnicamente viável a concepção de sistemas tutores inteligentes (ITS) capazes de fornecer suporte personalizado sob demanda. No entanto, LLMs genéricos são propensos a alucinações (geração de respostas factualmente incorretas ou inconsistentes), uma limitação crítica em domínios técnicos que exigem precisão formal e raciocínio lógico-matemático [Ji et al. 2023, Chen et al. 2024]. A arquitetura de geração aumentada por recuperação (*Retrieval-Augmented Generation* - RAG) [Lewis et al. 2020] mitiga esse problema ao integrar um indexador documental que fornece trechos do material didático como contexto para a inferência do modelo [Gao et al. 2023].

Apesar da eficácia do RAG, a maioria dos sistemas comerciais baseia-se em APIs proprietárias na nuvem (como GPT-4 e Claude), o que acarreta custos recorrentes proibitivos para implantação em larga escala universitária, além de levantar preocupações severas quanto à privacidade e à custódia de dados educacionais dos estudantes. A execução local de modelos de código aberto de pequeno e médio porte (como os *Small Language Models* - SLMs e os *Large Language Models* - LLMs de 7B a 8B parâmetros) apresenta-se como uma alternativa viável e segura. No entanto, a literatura científica nacional carece de avaliações sistemáticas comparando o desempenho qualitativo e de infraestrutura desses modelos locais aplicados a tutores inteligentes em português brasileiro.

A lacuna científica que este trabalho visa preencher reside justamente na falta de um benchmark controlado que avalie o desempenho de diferentes modelos de linguagem de código aberto, executados localmente em hardware voltado ao consumidor, dentro de uma arquitetura RAG estruturada por estágios curriculares. A justificativa do estudo apoia-se na necessidade de viabilizar a adoção sustentável de assistentes virtuais pedagógicos em universidades públicas e privadas. O objetivo deste artigo é avaliar comparativamente quatro modelos locais (Llama 3.1 8B Q8, Qwen 2.5 7B, Socratic Tutor Qwen 2.5 e Mathstral 7B Q4_K_M) integrados ao ecossistema do TutorIA, analisando suas métricas

de qualidade pedagógica (RAGAS) e de viabilidade computacional.

2. Trabalhos Relacionados

2.1. Sistemas Tutores Inteligentes e Mitigação de Alucinações via RAG

A literatura de sistemas tutores inteligentes destaca que a eficácia pedagógica depende de explicações factualmente corretas e metodologicamente alinhadas ao plano de ensino [Abdelghani et al. 2024]. O uso puro de LLMs falha em garantir esse alinhamento, gerando alucinações conceituais que podem induzir o aluno ao erro [Ji et al. 2023]. A adoção de RAG resolve essa limitação ao ancorar as respostas em materiais didáticos oficiais do curso, trabalhos como [Modran et al. 2025, Liu et al. 2025] defendem e demonstram esse ponto.

2.2. Modelos de Linguagem Locais e Avaliação de Sistemas RAG

Modelos de linguagem executados localmente (via ferramentas como o Ollama [Ollama Team 2024] têm se tornado populares devido à privacidade e independência de rede. Contudo, avaliar o desempenho desses modelos em português brasileiro e em domínios específicos é complexo. O framework RAGAS [Es et al. 2024] consolidou-se como o padrão-ouro para avaliação automática de RAG, quantificando dimensões cruciais como a fidelidade ao contexto (*faithfulness*), a correção da resposta (*answer correctness*) e a relevância semântica (*answer relevancy*). Além disso, estudos recentes de avaliação em português brasileiro, como o *benchmark* PoETa v2 conduzido por [Almeida et al. 2025], mostram que o desempenho varia significativamente de acordo com a escala computacional e a adaptação específica do modelo ao idioma.

3. Descrição da Proposta

Na Figura 1 está apresentado o diagrama geral da proposta do TutorIA, um sistema tutor inteligente baseado em pipeline de geração aumentada por recuperação (RAG) multi-coleção para a disciplina de circuitos lógicos.

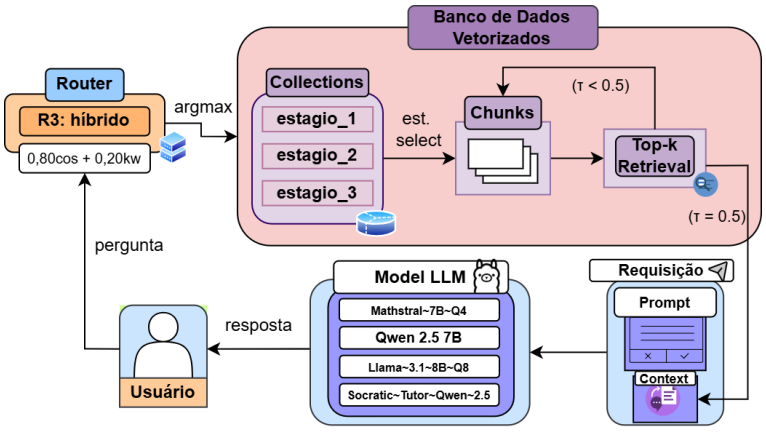


Figura 1. Pipeline do TutorIA. O estudo varia o modelo gerador (bloco roxo) mantendo fixos o roteador, as coleções e o *prompt* de instrução.

O TutorIA organiza o material didático em três coleções separadas no banco de dados vetorial ChromaDB [Chroma AI 2024], refletindo os estágios curriculares do curso:

- **estagio_1:** Fundamentos de circuitos, sistemas de numeração, níveis lógicos, portas lógicas e álgebra booleana;

- **estagio_2:** Circuitos combinacionais, simplificação com mapas de karnaugh, multiplexadores, decodificadores e somadores aritméticos;
- **estagio_3:** Circuitos sequenciais, flip-flops, registradores, contadores e máquinas de estados finitos (FSM).

O fluxo do sistema consiste em:

1. **Roteamento Curricular:** A consulta do aluno em português brasileiro é submetida a um roteador híbrido que calcula uma pontuação ponderada. O escore combina a similaridade cosseno de *embeddings* (usando o modelo denso M3-Embedding [Chen et al. 2024] com peso 0,80) e a correspondência léxica de palavras-chave (peso 0,20). O modelo foi escolhido por sua robustez em tarefas de recuperação multilíngue, suporte adequado ao português e capacidade de processar contextos longos de até 8.192 *tokens*. A consulta é direcionada à coleção correspondente ao estágio curricular sugerido.
2. **Recuperação de Contexto:** São recuperados os $k = 3$ trechos mais relevantes do estágio selecionado, aplicando-se um limiar mínimo de similaridade $\tau = 0{,}5$.
3. **Fallback:** Se nenhum trecho na coleção primária atingir o limiar de similaridade, o sistema consulta coleções adjacentes antes de retornar uma recusa estruturada.
4. **Geração Pedagógica:** O contexto recuperado e a pergunta são formatados em um *prompt* de instrução estruturada que exige fidelidade estrita aos fatos do material didático, enviando a requisição para inferência do modelo local.

A escolha dos parâmetros de roteamento e recuperação foi feita a partir de testes empíricos preliminares, com o objetivo de melhorar o tempo de resposta e a qualidade do contexto. Os pesos do roteador híbrido foram ajustados para priorizar o sinal semântico (0,80), que lida melhor com a forma como os alunos escrevem, enquanto a busca por contribuição léxica (0,20) ajuda a não perder os termos técnicos importantes da disciplina. Na etapa de recuperação, o valor $k = 3$ foi estabelecido como uma heurística para fornecer informações suficientes sem sobrecarregar o modelo ou deixar o sistema mais lento. Já o limite de $\tau = 0{,}5$ funcionou como um filtro de segurança básico, usado apenas para descartar textos que não tenham ligação com a pergunta.

4. Metodologia

4.1. Configurações Gerais da Pipeline

O experimento foi desenhado para isolar o comportamento do modelo gerador no TutorIA. Mantendo fixos o roteamento híbrido (peso 0,80/0,20), a base de conhecimento vetorial, a estratégia de recuperação ($k = 3$, $\tau = 0{,}5$) e o *prompt* de instrução, isolou-se o efeito do modelo gerador sobre o desempenho. Esses valores foram selecionados com base em testes exploratórios preliminares de tentativa e erro, sem otimização sistemática. Embora os valores adotados tenham produzido resultados satisfatórios no contexto avaliado, ressalta-se que essa configuração pode não ser ótima, sendo potencialmente sensível a variações no domínio, na base documental ou no perfil das perguntas submetidas ao sistema.

4.2. Modelos Avaliados

Com a configuração descrita na seção anterior, avaliou-se comparativamente quatro LLMs locais de pequeno e médio porte:

- **Llama 3.1 8B Q8:** Versão quantizada a 8-bits do modelo de 8 bilhões de parâmetros da Meta [Dubey et al. 2024];

- **Qwen 2.5 7B:** Modelo geral de 7 bilhões de parâmetros da Alibaba [Qwen Team 2025];
- **Socratic Tutor Qwen 2.5:** Ajuste fino especializado em estilo socrático [Wang et al. 2026];
- **Mathstral 7B Q4_K_M:** Variante do Mistral focada em raciocínio matemático e exatas [Jiang et al. 2023].

4.3. Estrutura de Benchmark

Consistiu em 45 questões de circuitos lógicos elaboradas pela equipe de desenvolvimento e auditadas pelos professores da disciplina, distribuídas igualmente entre os três estágios avaliados (15 questões por estágio), em conformidade com a ementa vigente da Universidade Federal de Campina Grande (UFCG - Campus Sede). A qualidade foi avaliada com o framework RAGAS [Es et al. 2024], que baseia-se na técnica de *LLM-as-a-judge* utilizando um modelo juiz forte (`qwen-2.5-72b-instruct` via OpenRouter) para as métricas de *faithfulness*, *answer correctness* e *answer relevancy*. De forma geral, o funcionamento do LLM juiz pode ser interpretado como composto por três etapas: recebimento das entradas necessárias para a métrica avaliada, realização do julgamento por meio de *prompts*, estruturados pelo próprio RAGAS, e agregação dos resultados em uma pontuação final que depende de qual métrica está sendo avaliada.

4.4. Estrutura Física

As inferências e medições foram executadas localmente em uma *workstation* (Intel Core i9, 64 GB RAM DDR5, GPU NVIDIA GeForce RTX 4060 8 GB VRAM, Windows 11). Os modelos foram servidos via Ollama [Ollama Team 2024] v0.1.48 sem chamadas a APIs externas durante a inferência. O juiz RAGAS operou via OpenRouter exclusivamente na avaliação.

Para garantir a reprodutibilidade, os hiperparâmetros de geração foram mantidos idênticos para todos os modelos, conforme a Tabela 1. A decodificação com temperatura 0 minimiza a aleatoriedade da saída, dispensando o controle de *seed* na geração, embora não garanta determinismo absoluto. Para viabilizar a execução local, empregou-se a quantização (redução da precisão dos tensores para economia de VRAM [Gholami et al. 2021]). As versões exatas via Ollama foram: `llama3.1:8b-instruct-q8_0` e `socratic-tutor-qwen2.5:Q8_0` ambos em 8-*bits*); `mathstral-7B-v0.1-GGUF:Q4_K_M` (quantização mista de 4-*bits* para reter qualidade); e `qwen2.5:7b-instruct` (precisão padrão).

Tabela 1. Hiperparâmetros de inferência, idênticos para os quatro modelos.

Parâmetro	Valor
Temperatura	0,0 (decodificação gulosa)
top-p	1,0 (sem efeito sob temperatura 0)
Máx. de *tokens* de saída	350
Seed	não aplicável (geração determinística)
Timeout por consulta	120 s
Trechos recuperados (k)	3
Limiar de similaridade (τ)	0,5

O texto integral do *prompt* de instrução utilizado na geração, mantido fixo em todos os experimentos, é reproduzido no bloco de citação a seguir:

> Você é um tutor universitário de Circuitos Lógicos. Sua função é explicar conceitos utilizando exclusivamente o CONTEXTO fornecido.
>
> **CONTEXTO:** {base_de_conhecimento}
> **PERGUNTA:** {pergunta}
>
> **REGRAS:** (i) responda em português brasileiro, com tom didático de tutor universitário; (ii) use SOMENTE o CONTEXTO, sem conhecimento externo; (iii) não invente informações; (iv) não use LaTeX, empregue a notação como aparece no CONTEXTO; (v) se a pergunta pedir N itens, cubra os N itens com base no CONTEXTO; (vi) priorize definição e comportamento; (vii) não diga que falta informação se ela aparecer em qualquer trecho do CONTEXTO; (viii) se a resposta não estiver no CONTEXTO, responda exatamente: "Não encontrei essa informação no material."

4.5. Métricas

Adotaram-se as seguintes métricas do *framework* RAGAS [Es et al. 2024]: ***faithfulness*** (grau em que a resposta é sustentada pelo contexto), ***answer correctness*** (similaridade com a resposta de referência) e ***answer relevancy*** (relevância semântica frente à pergunta). Registrou-se ainda a **taxa de recusa** e métricas operacionais (tok/s, VRAM e RAM).

Para consolidar as dimensões de qualidade em um único indicador, definiu-se o **índice composto** (IC):

$$IC = 0{,}35 \times \text{faithfulness} + 0{,}35 \times \text{answer_correctness} + 0{,}10 \times (1 - \text{refusal_rate}) \quad (1)$$

A soma dos pesos é $0{,}80$: os $0{,}20$ restantes são reservados a uma avaliação humana de qualidade linguística (escala 1–5), planejada como etapa futura. Optou-se por não imputar esse componente para preservar a transparência do índice, de modo que o IC reportado é um limite inferior conservador. Como proxy de juízes humanos nesta fase, adotou-se um avaliador automático de alta capacidade (`qwen-2.5-72b-instruct`), dado que o RAGAS foi validado quanto à correlação com julgamentos humanos [Es et al. 2024]. As diferenças entre os quatro modelos foram avaliadas por ANOVA *one-way* ($\alpha = 0{,}05$), com pós-testes de Tukey HSD e Games-Howell (este robusto a variâncias desiguais, indicadas pelo teste de Levene) e correção de múltiplas comparações por Holm e Benjamini-Hochberg (FDR) sobre as seis métricas. Como a geração é determinística (temperatura 0), repetições com sementes distintas seriam idênticas; a variabilidade amostral provém apenas das 45 questões, quantificada por intervalos de confiança de 95% via *bootstrap* (10.000 reamostragens).

5. Resultados e Discussão

5.1. Comparação Geral dos Modelos

Os resultados consolidados dos quatro modelos avaliados integrados ao pipeline do TutorIA revelam o desempenho obtido no suporte ao ensino de circuitos lógicos. A análise articula as métricas de qualidade pedagógica do framework RAGAS com indicadores de viabilidade operacional em hardware voltado ao consumidor, fornecendo uma visão objetiva sobre a

eficácia de cada solução. Os dados apresentados evidenciam os *trade-offs* críticos entre a fidelidade ao material didático, a acurácia das respostas e a eficiência computacional, abrangendo aspectos como velocidade de geração e ocupação de memória. Para sintetizar o desempenho global e facilitar o ranqueamento das alternativas, utiliza-se o Índice Composto (IC) como métrica central para determinar a adequação de cada modelo às exigências do sistema.

Conforme detalhado na Tabela 2, o Mathstral 7B e o Llama 3.1 8B Q8 lideram o índice composto em condição de equivalência estatística (0,618; intervalos de confiança sobrepostos), embora apresentem perfis de desempenho distintos. O Llama 3.1 8B Q8 destaca-se pela maior *faithfulness* (0,907), indicando uma adesão rigorosa ao contexto fornecido, porém com a menor velocidade de processamento entre os avaliados (9,88 tok/s). Em contrapartida, o Mathstral 7B oferece o melhor equilíbrio operacional para o cenário de tutoria, alcançando taxa de recusa zero, a maior correção de respostas (0,675) e uma fluidez de interação superior com 43,73 tok/s.

O modelo Qwen 2.5 7B, apesar de registrar a maior eficiência computacional (44,96 tok/s), apresenta uma taxa de recusa elevada de 20%, o que reduz sua utilidade prática em domínios técnicos. Já o Socratic Tutor Qwen 2.5 obteve o desempenho mais baixo em quase todas as métricas de qualidade, apresentando a maior taxa de recusa (33,3%), possivelmente devido à incompatibilidade entre seu ajuste fino e as instruções de fidelidade do sistema. A relevância estatística dessas variações foi confirmada por ANOVA de quatro grupos, que apontou diferenças significativas em fidelidade ($p_{\text{FDR}} = 0{,}004$) e relevância ($p_{\text{FDR}} < 0{,}001$), validando que a escolha do modelo gerador é o fator determinante para a qualidade da tutoria, uma vez que as métricas de recuperação permaneceram estáveis entre os testes ($p_{\text{FDR}} > 0{,}75$).

Tabela 2. Comparação dos modelos locais no experimento . Métricas RAGAS, consumo de recursos e índice composto (IC). Valores em negrito indicam o melhor resultado por métrica.

Modelo	tok/s	VRAM (MB)	RAM (GB)	Recusa (%)	Faith.	Correct.	Relev.	Ctx.P	Ctx.R	IC
Llama 3.1 8B Q8	9,88	7038	42,77	2,2	**0,907**	0,579	**0,785**	0,693	0,750	**0,618**
Qwen 2.5 7B	**44,96**	**5709**	**32,62**	20,0	0,779	0,588	0,678	**0,704**	0,728	0,558
Socratic Tutor Qwen 2.5	16,23	7563	41,34	33,3	0,637	0,509	0,486	0,682	**0,808**	0,468
Mathstral 7B Q4_K_M	43,73	5842	41,17	**0,0**	0,806	**0,675**	0,800	**0,704**	0,739	**0,618**

5.2. Análise das Métricas de Qualidade

A avaliação detalhada da qualidade da geração evidencia perfis de desempenho complementares entre os modelos, revelando como cada arquitetura privilegia diferentes dimensões da interação pedagógica. Para subsidiar essa análise, a Figura 2 apresenta uma comparação visual dos escores de *faithfulness* (fidelidade), *answer correctness* (correção) e *answer relevancy* (relevância). Os resultados indicam que o desempenho não é uniforme, expondo *trade-offs* importantes entre a aderência estrita ao material didático e a capacidade de produzir respostas semanticamente relevantes e tecnicamente consistentes no domínio de circuitos lógicos.

O Socratic Tutor Qwen 2.5, embora corresponda a uma variante ajustada (*fine-tuned*) para aplicações de tutoria, apresentou o pior desempenho em praticamente todas as métricas

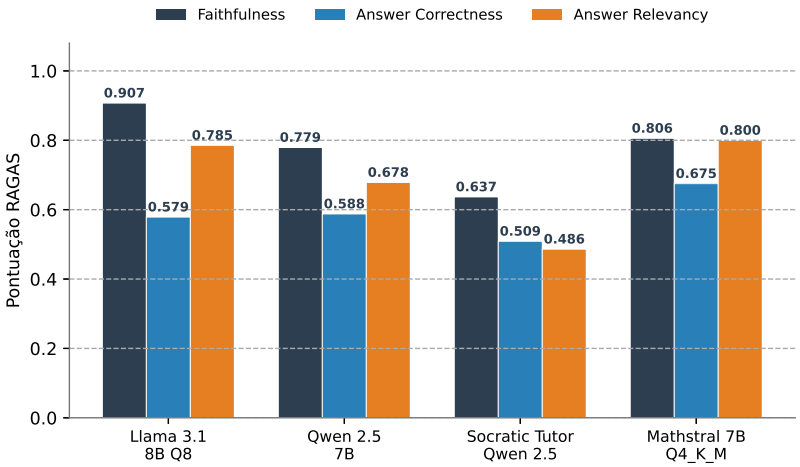


Figura 2. Comparação de fidelidade, correção e relevância entre os quatro modelos. Llama 3.1 lidera em fidelidade, enquanto o Mathstral apresenta os maiores escores de correção e relevância.

avaliadas (fidelidade: 0,637; correção: 0,509; relevância: 0,486), além da maior taxa de recusa (33,3%). Esse resultado aparentemente contraditório pode ser explicado pelo conflito entre a estratégia socrática de *prompting*, que privilegia perguntas orientadoras em detrimento de respostas diretas, e o *prompt* de instrução do TutorIA, que orienta o modelo a fornecer explicações fundamentadas no contexto recuperado. A incompatibilidade entre essas estratégias resulta em respostas excessivamente evasivas, classificadas pelo RAGAS como de baixa qualidade.

Finalmente, o Mathstral 7B Q4_K_M destacou-se pelo equilíbrio entre as diferentes dimensões avaliadas, apresentando ausência de recusas, o maior índice de correção (0,675), elevada relevância (0,800) e fidelidade competitiva (0,806). Sua velocidade de geração (43,73 tok/s) é comparável à do Qwen 2.5 7B, favorecendo interações mais fluidas. Adicionalmente, o índice composto obtido pelo Mathstral é estatisticamente indistinguível do alcançado pelo Llama 3.1 8B Q8: os intervalos de confiança de 95% (*bootstrap*) se sobrepõem amplamente (Mathstral: 0,616, IC95% [0,577; 0,653]; Llama: 0,619, IC95% [0,584; 0,649]), de modo que a diferença na terceira casa decimal está inteiramente contida na margem de erro amostral e não deve ser interpretada como vantagem de um modelo sobre o outro. A escolha entre ambos, portanto, recai sobre as vantagens operacionais do Mathstral em latência, disponibilidade e ausência de recusas, características particularmente desejáveis em sistemas de tutoria baseados em RAG.

5.3. Índice Composto

A síntese do desempenho global dos modelos é realizada por meio do Índice Composto (IC), uma métrica agregadora desenhada para facilitar o ranqueamento das alternativas conforme sua adequação ao ensino de circuitos lógicos. Diferente da análise isolada de métricas, o IC permite uma visão holística ao ponderar a fidelidade ao material didático (35%), a acurácia das respostas (35%) e a disponibilidade do sistema frente a possíveis recusas (10%). Na Figura 3 está ilustrada essa consolidação, permitindo identificar quais modelos melhor atendem aos requisitos de um Sistema Tutor Inteligente (ITS) operando integralmente em hardware de consumo.

Os dados sintetizados revelam uma equivalência estatística no topo do ranking entre o Mathstral 7B e o Llama 3.1 8B Q8, ambos com escore próximo de 0,618 e intervalos de

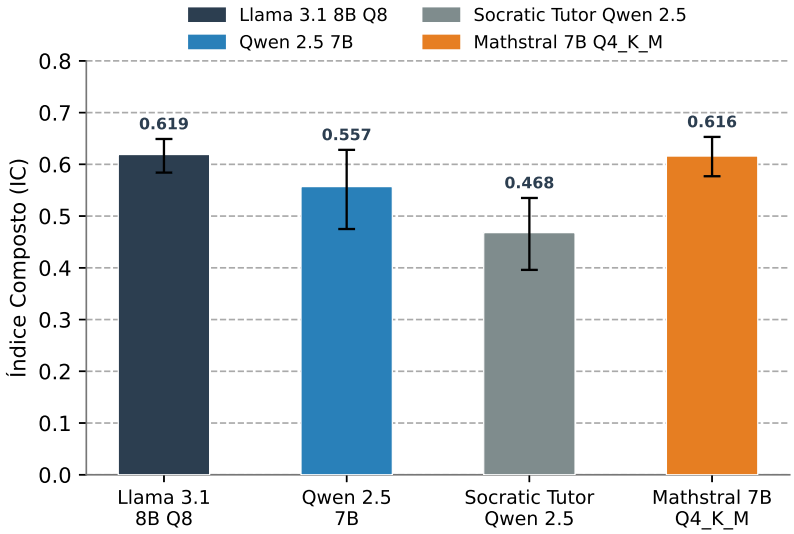


Figura 3. Índice composto (IC) dos modelos avaliados, com barras de erro de IC95% (*bootstrap*, 10.000 reamostragens). A ampla sobreposição entre Mathstral 7B e Llama 3.1 8B indica equivalência estatística no topo do ranking.

confiança sobrepostos (Figura 3). Como a diferença entre eles não é distinguível do ruído amostral, o desempate dá-se pelas vantagens práticas do Mathstral para a implementação: ausência total de recusas e velocidade de inferência cerca de quatro vezes superior à do Llama 3.1 8B Q8. Em contrapartida, o Qwen 2.5 7B (0,558) e o Socratic Tutor (0,468) apresentam escores menores, penalizados por altas taxas de recusa que comprometem a utilidade prática do tutor em ambientes educacionais.

5.4. Análise Estatística (ANOVA de Quatro Grupos e Pós-teste)

A validação estatística foi conduzida por ANOVA *one-way* de quatro grupos (um por modelo), evitando o agrupamento de modelos heterogêneos em uma única "classe 7B". Os resultados ômnibus, com correção de múltiplas comparações pelo procedimento de Benjamini-Hochberg (FDR), estão na Tabela 3. As diferenças entre modelos são significativas em *faithfulness* ($F = 5{,}21$; $p_{\text{FDR}} = 0{,}004$), *answer correctness* ($F = 3{,}72$; $p_{\text{FDR}} = 0{,}019$), *answer relevancy* ($F = 11{,}60$; $p_{\text{FDR}} < 0{,}001$) e *answer similarity* ($F = 6{,}40$; $p_{\text{FDR}} = 0{,}001$). As métricas de recuperação (*context precision* e *context recall*) não diferem ($p_{\text{FDR}} > 0{,}75$), resultado esperado pois o componente de recuperação é idêntico para todos os modelos, o que valida o desenho experimental.

O pós-teste de Tukey HSD, corroborado pelo de Games-Howell, revela que a significância em *faithfulness* não reflete uma superioridade homogênea de uma "classe" de parâmetros, mas decorre de um único par: Llama 3.1 *vs.* Socratic Tutor ($\Delta = 0{,}270$; $p = 0{,}001$). Em particular, a diferença de fidelidade entre Llama 3.1 e Mathstral 7B **não** é estatisticamente significativa ($\Delta = 0{,}102$; $p = 0{,}465$), assim como Llama 3.1 *vs.* Qwen 2.5 ($p = 0{,}280$). Padrão análogo ocorre em *answer relevancy*, em que os pares significativos envolvem sempre o Socratic Tutor (*vs.* Llama, $\Delta = 0{,}299$, $p < 0{,}001$; *vs.* Mathstral, $\Delta = 0{,}314$, $p < 0{,}001$), enquanto Llama e Mathstral são indistinguíveis ($\Delta = -0{,}014$; $p = 0{,}995$). Conclui-se que o fator determinante é o *modelo específico* e não a contagem de parâmetros, refinando a interpretação de que "a escolha do modelo gerador importa".

Tabela 3. ANOVA *one-way* de quatro grupos por métrica RAGAS. p_{FDR}: p-valor corrigido por Benjamini-Hochberg; ANOVA de Welch reportada por robustez a variâncias desiguais. Significância a $\alpha = 0{,}05$.

Métrica	F	p_{FDR}	Welch p	Resultado
Faithfulness	5,21	0,004	0,001	Significativo
Answer Correctness	3,72	0,019	0,005	Significativo
Answer Relevancy	11,60	<0,001	<0,001	Significativo
Answer Similarity	6,40	0,001	<0,001	Significativo
Context Precision	0,04	0,991	0,992	Não significativo
Context Recall	0,57	0,759	0,564	Não significativo

5.5. Correlação entre Fidelidade e Recuperação de Contexto

A análise de correlação de Pearson entre *faithfulness* e *context_recall* por modelo revelou valores fracos para todos os modelos: Llama 3.1 ($r = -0{,}090$; $p = 0{,}569$), Qwen 2.5 ($r = -0{,}248$; $p = 0{,}123$), Socratic Tutor ($r = 0{,}192$; $p = 0{,}207$) e Mathstral ($r = 0{,}070$; $p = 0{,}652$). Nenhuma das correlações é estatisticamente significativa, ou seja, não há evidência de associação linear entre a quantidade de contexto relevante recuperado e a fidelidade da resposta. Esse resultado é compatível com a hipótese de que a fidelidade depende primariamente da capacidade intrínseca do modelo de aderir ao material fornecido durante a geração, embora a confirmação dessa hipótese exija amostras maiores.

5.6. Limitações do Estudo

Apesar dos achados relevantes, o presente estudo apresenta algumas limitações. O tamanho da amostra foi restrito a 45 questões; embora suficiente para detectar os efeitos principais entre modelos (inclusive após correção de múltiplas comparações), limita o poder para discriminar diferenças finas entre pares de modelos de desempenho próximo, como evidenciado pela ampla sobreposição dos intervalos de confiança no índice composto. Apenas quatro modelos locais foram testados, dada a natureza exaustiva das avaliações de *hardware*.

Cabe observar que a métrica de *faithfulness* mensura a proporção de afirmações da resposta gerada que podem ser inferidas a partir do contexto recuperado. Contudo, essa definição não contempla situações em que o próprio contexto recuperado é impreciso ou incompleto. Nesses casos, uma resposta altamente fiel a um contexto factualmente incorreto receberá pontuação elevada de *faithfulness*, ainda que o conteúdo gerado seja equivocado caracterizando uma limitação inerente à métrica. Por fim, o uso de um juiz LLM automatizado (RAGAS) sem uma avaliação humana correspondente pode introduzir vieses metodológicos, particularmente no que diz respeito à qualidade subjetiva e fluência em português brasileiro. Esse risco é parcialmente atenuado por três fatores: (i) o emprego de um modelo juiz de grande porte (`qwen-2.5-72b-instruct`), substancialmente mais capaz do que os modelos avaliados. Ainda assim, reconhece-se a literatura recente sobre o viés de auto-preferência (*self-preference bias*) [Wataoka et al. 2024], fenômeno onde o juiz favorece modelos de sua própria família arquitetural. Portanto, admite-se que a pontuação do Socratic Tutor Qwen 2.5 exige ressalvas metodológicas específicas; contudo, o fato de este modelo ter obtido o pior desempenho geral sugere que eventuais favorecimentos não foram suficientes para mascarar sua incompatibilidade com o *pipeline* RAG; (ii) o uso das métricas RAGAS de forma comparativa entre modelos sob protocolo idêntico, de modo que

eventual viés sistemático do juiz afeta todos os modelos na mesma direção, preservando a validade do ranqueamento relativo; e (iii) a reserva explícita de 20% do índice composto para validação humana futura, o que evita superestimar a confiabilidade das pontuações automáticas. Ainda assim, a confirmação definitiva dos achados em termos de qualidade percebida exige a avaliação por especialistas humanos, conduzida como trabalho futuro.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou uma avaliação comparativa controlada de quatro modelos de linguagem de código aberto (Llama 3.1 8B Q8, Qwen 2.5 7B, Socratic Tutor Qwen 2.5 e Mathstral 7B Q4_K_M) para utilização em um sistema tutor inteligente baseado em RAG para a disciplina de circuitos lógicos, operando integralmente em hardware de consumo.

Os resultados indicam que o Mathstral 7B Q4_K_M constitui a opção mais equilibrada para implantação prática, combinando taxa de recusa zero, a maior correção de respostas (0,675), alta velocidade de inferência (43,73 tok/s) e consumo moderado de recursos. O Llama 3.1 8B Q8, embora empate no índice composto, é recomendado para cenários em que a fidelidade ao material didático é a prioridade absoluta (0,907), ainda que com latência mais elevada. O Socratic Tutor, apesar de concebido para tutoria, revelou-se inadequado para pipelines RAG estruturados, evidenciando que a especialização via *fine-tuning* pode conflitar com estratégias de *prompting* externas.

A análise estatística por ANOVA de quatro grupos, com pós-teste de Tukey HSD e correção de múltiplas comparações, confirmou diferenças significativas em fidelidade ($p_{\text{FDR}} = 0{,}004$) e relevância ($p_{\text{FDR}} < 0{,}001$) entre modelos específicos, sobretudo o desalinhamento do Socratic Tutor, ao passo que as métricas de recuperação permaneceram estáveis, validando o desenho experimental.

Como trabalhos futuros, pretende-se: (i) avaliar modelos de nova geração, incluindo Phi-4 e DeepSeek-R1 32B (com quantização para adequação à VRAM disponível); (ii) conduzir avaliação manual com especialistas em circuitos lógicos para validar as métricas automatizadas em língua portuguesa; e (iii) investigar estratégias de *prompt engineering* otimizadas para modelos com perfis distintos de especialização.

Agradecimentos

Este trabalho foi apoiado em parte pela Universidade Federal de Campina Grande (UFCG) e em parte pelo Projeto Base iSOP apoiado pelo Centro de Competência Embrapii Virtus em Hardware Inteligente Para Indústria (VIRTUS-CC), por meio do Programa Prioritário (PPI) HardwareBR do Ministério da Ciência, Tecnologia e Inovação (MCTI), firmado com a Empresa Brasileira de Pesquisa e Inovação Industrial (EMBRAPII), sob concessão 055/2023.

Referências

Abdelghani, R. et al. (2024). GPT-3-driven pedagogical agents to train children's curious question-asking skills. *International Journal of Artificial Intelligence in Education*, 34:483–518.

Almeida, T. R. S. et al. (2025). PoETa v2: Toward more robust evaluation of large language models in Portuguese. *IEEE Access*, 13:214180–214200.

Bosse, Y. and Gerosa, M. A. (2015). Reprovações e trancamentos nas disciplinas de introdução à programação da universidade de são paulo: Um estudo preliminar. In *Anais*

do Workshop sobre Educação em Computação (WEI 2015), WEI, page 426–435. Sociedade Brasileira de Computação - SBC.

Chen, J. et al. (2024). BGE M3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Chroma AI (2024). ChromaDB: The AI-native open-source embedding database. https://www.trychroma.com. Acesso em: 1 jun. 2026.

Dubey, A. et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Es, S. et al. (2024). RAGAS: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL) - System Demonstrations*, pages 150–158.

Gao, Y. et al. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Gholami, A. et al. (2021). A survey of quantization methods for efficient neural network inference. *Low-Power Computer Vision*, pages 291–326.

Ji, Z. et al. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Jiang, A. Q. et al. (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Lewis, P. et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9459–9474.

Liu, Z. et al. (2025). LPITutor: An LLM-based personalized intelligent tutoring system using RAG and prompt engineering. *PeerJ Computer Science*, 11:e2991.

Modran, H. et al. (2025). *LLM Intelligent Agent Tutoring in Higher Education Courses Using a RAG Approach*, page 589–599. Springer Nature Switzerland.

Ollama Team (2024). Ollama: Get up and running with large language models. https://ollama.ai. Acesso em: 1 jun. 2026.

Qwen Team (2025). Qwen2.5 technical report. Technical report, Alibaba Group.

Wang, X. et al. (2026). Beyond direct answering: Aligning educational llms as socratic guides via heuristic reinforcement learning. *arXiv preprint arXiv:2607.22996*.

Wataoka, K. et al. (2024). Self-preference bias in llm-as-a-judge. *arXiv preprint arXiv:2410.21819*.