Um Framework RAG com Auditoria Híbrida de SKUs para Mitigar a Obsolescência Informacional em Suporte Técnico

Daniel Youssef de H. Lopes[1], **Gabriel M. Lunardi**[1], **Adriano Q. de Oliveira**[2]

[1] Centro de Tecnologia – Departamento de Computação Aplicada
Universidade Federal de Santa Maria (UFSM) – Santa Maria, RS – Brazil

[2]Coordenação Acadêmica – Campus Cachoeira do Sul
Universidade Federal de Santa Maria (UFSM) – Cachoeira do Sul, RS – Brazil

{daniel.lopes}@acad.ufsm.br, {gabriel.lunardi, adriano.q.oliveira}@ufsm.br

Resumo. *Modelos de Linguagem de Grande Escala (LLMs) frequentemente recuperam especificações ou drivers desatualizados em domínios de suporte técnico, comprometendo a confiabilidade do sistema. Este artigo propõe um framework de RAG Temporal que integra um algoritmo de decaimento linear (LinearV3) para relevância de documentos e um protocolo de auditoria híbrida (V8) baseado em raciocínio lógico de série. Resultados experimentais utilizando o framework DeepEval demonstram que o sistema proposto supera o limiar de segurança e atinge 89% de precisão contextual em cenários de hardware onde modelos RAG convencionais falham, garantindo fidelidade técnica através de ancoragem documental estrita.*

Abstract. *Large Language Models (LLMs) often retrieve outdated specifications or drivers in technical support domains, compromising system reliability. This paper proposes a Temporal Retrieval-Augmented Generation (RAG) framework that integrates a linear decay algorithm (LinearV3) for document relevance and a hybrid audit protocol (V8) based on series-logic reasoning. Experimental results using the DeepEval framework demonstrate that the proposed system exceeds safety thresholds, achieving 89% contextual precision in static hardware scenarios where vanilla RAG models fail, ensuring technical fidelity through strict documentary grounding.*

1. Introdução

A rápida evolução do hardware torna especialmente desafiadora a análise de informações em comunidades de suporte técnico, pois soluções antes válidas podem perder atualidade em pouco tempo [Chu et al. 2026]. Esse fenômeno ocorre tanto em plataformas abertas, como o Reddit, quanto em fóruns de fabricantes, como Dell, Acer, Lenovo e HP, nos quais problemas semelhantes podem reaparecer em gerações distintas de dispositivos, mas por causas técnicas diferentes [Park 2021]. Por exemplo, uma falha associada ao BIOS (Basic Input/Output System) em um notebook lançado inicialmente pode ressurgir anos depois em uma revisão posterior ou em um modelo sucessor, embora decorrente de outro fator, como alterações de firmware, controladores ou arquitetura interna. Ainda assim, discussões antigas tendem a concentrar mais interações e, por isso, podem ser classificadas como mais relevantes por sistemas de busca ou recuperação, mesmo quando já não representam a solução mais apropriada para o contexto atual [Hill 2026].

Essa volatilidade do conhecimento torna bases estáticas inadequadas para raciocínio atualizado [Piryani et al. 2025]. Em sistemas de Recuperação Aumentada por Geração (RAG) convencionais, esse cenário gera um conflito crítico. A literatura diagnostica que os Modelos de Linguagem de Grande Escala (LLMs) sofrem de "pontos cegos temporais" [Piryani et al. 2025]. Como a busca padrão prioriza a proximidade lexical, o sistema falha em distinguir discussões obsoletas de atuais, culminando em um "borrão semântico" (*semantic blur*) que reduz a acurácia técnica em mais de 20% [Ouyang et al. 2025].

Abordagens recentes, como o *framework* TG-RAG [Han et al. 2025] e a fusão de escores do TempRALM [Gade and Jetcheva 2024], mitigam essa vulnerabilidade com excelência. Contudo, sua validação depende rotineiramente de infraestruturas massivas (e.g., GPUs A100 com 40GB de VRAM) ou APIs fechadas, criando barreiras de adoção para ambientes com restrição de hardware ou requisitos de privacidade local. Para preencher essa lacuna, este trabalho introduz uma arquitetura de RAG Temporal otimizada para execução local sob condições limitadas de inferência. Nesse sentido propõe-se a substituição de LLMs massivos por *Small Language Models* (SLMs), especificamente os modelos Qwen com 3 e 4 Bilhões de parâmetros [Yang et al. 2025], Llama 3.2 de 3 Bilhões de parâmetros [Meta 2024] e, por fim, o Phi 3.5 de 3.8 Bilhões de parâmetros [Microsoft 2024]. Ainda, como parte do framework proposto, está o algoritmo determinístico de decaimento LinearV3, acoplado a um Protocolo de Auditoria Híbrida (V8).

Operando sob a filosofia Zero-Trust, o sistema combina o isolamento determinístico de componentes (*SKU-Lock*) com limiares rígidos de validação generativa via *framework* DeepEval. Essa arquitetura aborda a incerteza temporal [Piryani et al. 2025], exigindo que o modelo priorize a omissão segura (recusar-se a responder) à entrega de "alucinações superconfiantes" baseadas em dados obsoletos [Ouyang et al. 2025].

Neste contexto, este trabalho propõe um framework de RAG temporal voltado a suporte técnico de hardware em ambiente local com forte restrição computacional. As contribuições principais são: (*i*) um mecanismo de re-ranking temporal linear para penalizar evidências obsoletas; (*ii*) um protocolo híbrido de auditoria com trava de SKU e validação por LLM-as-a-Judge; (*iii*) uma implementação executável sob 4GB de VRAM; e (*iv*) uma avaliação comparativa entre uma variante baseline (V1) e a variante proposta (V8) em cenários de suporte técnico de hardware.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 detalha o método proposto, abordando a curadoria do corpus, o motor LinearV3 e a auditoria híbrida; a Seção 4 descreve a configuração experimental, o avaliador e as métricas; a Seção 5 discute a avaliação quantitativa e qualitativa dos resultados; e, por fim, a Seção 6 conclui o estudo e aponta direções para trabalhos futuros.

2. Trabalhos Relacionados

Para mitigar a obsolescência temporal, abordagens recentes alteram a mecânica de recuperação. O framework TempRALM, por exemplo, penaliza informações antigas ao incorporar um escore temporal baseado no recíproco (inverso) da diferença de tempo entre a consulta e o documento [Gade and Jetcheva 2024]. O sucesso dessa estratégia valida o uso de funções de decaimento acopladas ao motor de busca, princípio que adaptamos em nosso algo-

ritmo LinearV3 ao substituir a curva inversa por um decaimento linear baseado na idade da informação.

Sob a ótica de avaliação de risco, o estudo de Ouyang et al. [Ouyang et al. 2025], ao introduzir o benchmark dinâmico HOH, revelou que modelos vetoriais (*embeddings*) de estado da arte sofrem de severa "cegueira temporal", recuperando informações obsoletas em mais de 50% das vezes nos cinco primeiros resultados (*top-5*). Essa incapacidade intrínseca dos *embeddings* em distinguir a validade temporal justifica a nossa decisão arquitetural de não delegar o isolamento do contexto apenas ao motor semântico (BGE-M3), submetendo-o a um re-ranqueamento determinístico e a uma rigorosa auditoria generativa posterior.

Além da dimensão temporal, a eficiência em domínios técnicos restritos depende da eliminação de ruídos estruturais. Estudos focados no desenvolvimento de aplicações LLM seguras alertam que delegar correspondência estrita de padrões (como extração de SKUs) exclusivamente a modelos generativos eleva o tempo de inferência e a propagação de erros [Wang et al. 2024]. A literatura sugere a integração de métodos determinísticos baseados em expressões regulares, justificando a adoção de um motor de extração e canonização estrutural operando como barreira primária em nosso pipeline.

3. Método Proposto

Este trabalho introduz um sistema de RAG temporal com auditoria híbrida, estruturada para transformar dados brutos de comunidades não estruturadas em conhecimento técnico auditável. O fluxo diferencia-se de abordagens convencionais ao não confiar exclusivamente na similaridade vetorial, submetendo os resultados a filtros cronológicos e lógicos determinísticos.

3.1. Visão Geral do Pipeline

O pipeline operacional é decomposto em três etapas fundamentais, como ilustrado na Figura 1. Inicialmente, o *corpus* bruto é processado por um motor de Reconhecimento de Entidades Nomeadas (NER) e Sumarização Técnica baseado em modelos locais, que identifica componentes críticos e gera relatórios densos com atributos como *Detected Fault*, *Suggested Solution* e metadados temporais. Na segunda etapa, ocorre a vetorização e a recuperação (*Retrieval*), onde o motor de decaimento LinearV3 penaliza documentos obsoletos no momento da busca. Por fim, a terceira etapa consiste em um *Reasoning Pipeline* interno governado por regras de *SKU-Lock*, no qual o modelo de linguagem valida a pertinência do contexto antes da geração final.

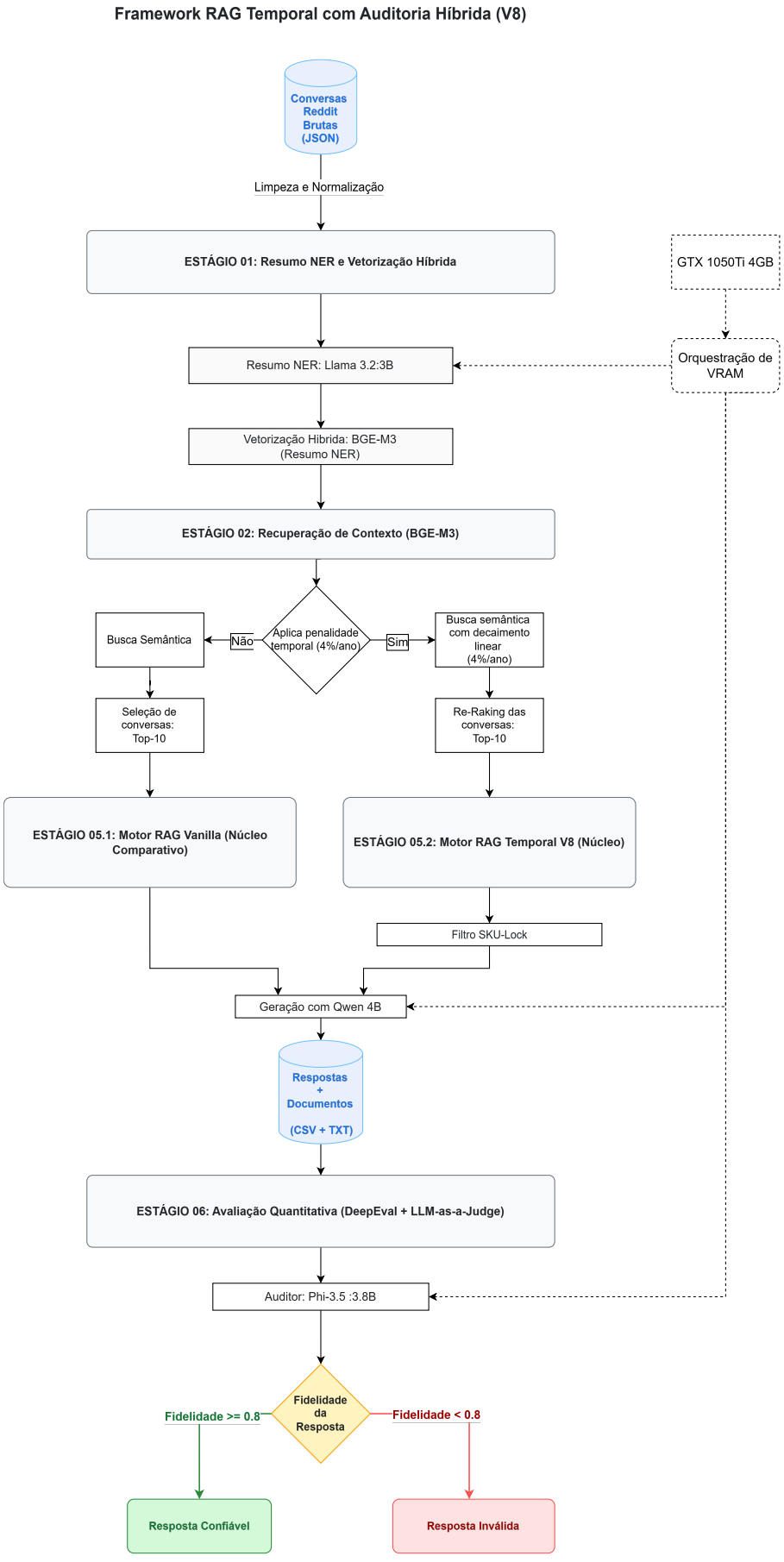

Figura 1. Fluxo de processamento do framework RAG Temporal.

Com as etapas definidas, apresentamos um prompt para a V8 Híbrida, que irá abranger suas instruções, variáveis de entrada e restrições:

Template do Prompt de Geração e Síntese (Variante V8 Híbrida)

System instruction:
You are a Technical Support Specialist. Use the reasoning process to evaluate information internally, but provide a clean, professional technical response without any internal tags or meta-talk.

User prompt:
TECHNICAL CONTEXT: {`context`}
USER QUERY: {`query`}
REFERENCE DATE: {`ref_date`}
INSTRUCTION: Apply strict SKU validation. If the exact model specifications are not in the context, do not use data from other models in the series. Deliver only accurate, SKU-specific data.

Variables:
- {`context`}: Documentos recuperados do FAISS após decaimento temporal.
- {`query`}: Pergunta técnica submetida pelo usuário.
- {`ref_date`}: Data de referência explícita para a simulação temporal.

Constraints:
- **Strict SKU Lock:** Base hardware specifications EXCLUSIVELY on the provided context for the EXACT SKU requested.
- **No Series Inference:** Do not infer data from similar models (e.g., using `AN515-58` data for an `AN515-56` query).
- **Anachronism Guard:** Do not use internal knowledge for hardware released after 2024 unless guided by the context.
- **Temporal Provenance:** Explicitly identify the knowledge "Era" if internal concepts are used.

Expected output:
A precise technical response adhering to Zero-Trust policies, followed by a provenance audit log.

3.2. Corpus, Canonização de SKUs e Curadoria

O banco de dados compõe-se de um *dataset* de especificações técnicas e de uma base dinâmica de discussões oriundas da plataforma Reddit. As discussões concentram-se em cinco temas principais que descrevem problemas frequentes em notebooks, abrangendo desde falhas de *hardware* físico até questões de *software* e desempenho.

Para as especificações técnicas, consolidou-se um corpus unificado a partir de conjuntos de dados públicos, resultando em 6.599 registros únicos mapeados por SKU após um processo rigoroso de canonização de nomenclaturas. A canonização é crítica para que o modelo de *embedding* não disperse sua atenção em *tokens* irrelevantes (como variações de grafia para a mesma marca ou componente), otimizando a precisão vetorial [Wang et al. 2024]. A Tabela 1 sumariza a distribuição do *dataset* e as médias de especificações por categoria, evidenciando a diversidade do corpus utilizado.

Para garantir a representatividade técnica e evitar o ruído de fóruns abertos, o

Tabela 1. Profiling do dataset de especificações técnicas por categoria.

Categoria	Qtd.	RAM Méd. (GB)	RAM Máx. (GB)	Armaz. Méd. (GB)	Tela Méd. (“)
Business	761	11,31	32	482,83	14,37
Creator	81	16,62	32	598,88	14,41
Gamer	2122	16,66	64	754,35	15,74
Geral	3635	11,28	48	508,42	14,85
Total	**6599**	**12,48**	**64**	**569,21**	**15,02**

corpus do Reddit passou por um rigoroso “afunilamento de fidelidade”. Aplicou-se um método de amostragem estratificada focado em cinco categorias: Performance, Térmica, Energia, Display e Estabilidade. Importante ressaltar que para aumentar a chance de encontrar discussões dos temas foi introduzido as variantes semânticas deles, de 4 a 5, para enriquecer as buscas dos temas originais que podem não surgir pelo termo exato. Modelos cujas discussões estavam concentradas em apenas uma categoria foram descartados para mitigar o “borrão semântico”. Estabeleceu-se um teto de 50 postagens por modelo para evitar redundâncias cíclicas e sobrecarga na janela de contexto. Este processo refinou a base de conhecimento para 106 notebooks de alta densidade técnica.

Os metadados temporais (Unix Timestamp) das interações foram extraídos para compor duas variáveis fundamentais: a âncora Temporal, que estabelece uma referência de tempo para mitigar a perda de contexto cronológico [Piryani et al. 2025], e a timeline da discussão, que mapeia a evolução temporal da falha relatada.

3.3. Re-ranking Temporal com LinearV3

O motor LinearV3 atua no re-ranqueamento pós-recuperação. Para compensar a “cegueira temporal”inerente aos modelos de *embedding* [Ouyang et al. 2025], o sistema realiza uma busca inicial ampliada ($k = 50$) no índice FAISS utilizando a Distância L2 a partir da qual valores menores indicam maior similaridade. Cada documento recebe uma penalidade linear que aumenta matematicamente essa distância, baseada no tempo decorrido (Δt, em anos) entre a data da publicação e a data da consulta:

$$S_{final} = S_{FAISS} \times (1 + (\Delta t \times \lambda)) \tag{1}$$

A taxa de penalização $\lambda = 0,04$ (4% ao ano) foi determinada heuristicamente para equilibrar dois objetivos conflitantes: (i) rebaixar informações obsoletas que induzem ao “borrão semântico”; e (ii) não silenciar discussões relevantes sobre hardware mais antigo, cuja última interação significativa pode ter ocorrido há mais de quatro anos. Uma taxa muito alta (e.g., $> 8\%$) penalizaria excessivamente esses casos, excluindo conhecimento ainda válido, enquanto uma taxa muito baixa (e.g., $< 2\%$) seria inócua para diferenciar documentos de 2020 e 2024. Esse equilíbrio é crucial para o contexto de suporte técnico, onde a longevidade de um modelo não implica, necessariamente, a obsolescência de suas resoluções.

3.4. Auditoria Híbrida com SKU-Lock e LLM-as-a-Judge

Para mitigar a alucinação em modelos compactos, a arquitetura impõe um pipeline de raciocínio estruturado em quatro estágios baseados na filosofia *Zero-Trust*: (i) SKU-Lock, que exige a verificação estrita do modelo do hardware no contexto recuperado;

(ii) priorização de dados recentes; (iii) negação de lógica de série (*anti-series logic*, impedindo a suposição de que falhas de uma geração de hardware sejam herdadas automaticamente pela seguinte); e (iv) log de proveniência.

Esse método atua como uma auditoria em tempo de inferência, forçando o modelo a atestar a existência de evidência documental. O sistema é programado para preferir a omissão segura (recusar-se a responder e alertar a falta de dados) à entrega de dados paramétricos não verificados que poderiam resultar em danos ao hardware.

4. Configuração Experimental

Para garantir a reprodutibilidade e o rigor científico, o protocolo de avaliação foi projetado para mensurar três aspectos críticos do sistema: 1) a eficácia em recuperar a documentação exata do hardware alvo sem misturar modelos similares; 2) a capacidade de impedir que o modelo gerador invente especificações com base em seu treinamento prévio (alucinação paramétrica); e 3) a habilidade de descartar resoluções técnicas antigas em favor de dados atualizados (obsolescência informacional). Todos os artefatos de código, *datasets* e análises estão disponíveis no repositório anônimo do projeto[1].

4.1. Ambiente Computacional e Modelos Utilizados

A arquitetura foi estritamente definida pelo limite de *4GB de VRAM* de uma GPU local (NVIDIA GTX 1050 Ti Mobile). A execução simultânea de modelos de linguagem e vetoriais exigiu um particionamento rigoroso e orquestração de *offloading*, operando sob quantização de 4-bit (GGUF/AWQ). O pipeline sequencial utilizou as seguintes famílias de modelos: o Llama 3.2 3B para a etapa de extração inicial (NER); o BGE-M3 (dimensionalidade 1024) para a vetorização do espaço semântico, escolhido por sua versatilidade e alta capacidade em janelas longas; e o Qwen 3 4B-Instruct para a etapa principal de geração (RAG), devido à sua ampla janela de contexto. Por fim, o Phi 3.5 Mini foi isolado como o avaliador (*LLM-as-a-Judge*). A literatura recente sugere que *Small Language Models* (SLMs), quando configurados para gerar rastros de raciocínio explícito (*thinking mode*), atingem precisão e robustez superiores contra vieses de avaliação, operando a uma fração do custo computacional de LLMs massivos, mitigando assim o viés de endogenia na auditoria do pipeline.

4.2. Baseline V1 e Variante V8

O experimento compara o desempenho de duas variantes arquiteturais. A variante V1 (Baseline) representa a abordagem convencional, que utiliza busca no índice FAISS baseada apenas na similaridade vetorial, sem a aplicação de decaimento temporal. Nesta configuração, o prompt de geração instrui o modelo a responder com base no contexto ou, na sua ausência, utilizar livremente seu conhecimento paramétrico. Em contrapartida, a variante V8 (Híbrida) consiste na abordagem proposta por este trabalho, que integra o motor LinearV3 de decaimento temporal durante o ranqueamento e aplica um *reasoning pipeline* de quatro estágios com instruções estritas de Auditoria Híbrida, impondo a trava determinística de *SKU-Lock* e a política *Zero-Trust* durante o processo de síntese das respostas.

[1] `https://github.com/dyoussef11/TimeAware-RAG-Auditor`

4.3. Métricas, Thresholds e Avaliador

A validação técnica foi orquestrada integrando o modelo local *Phi-3.5 Mini* ao framework *DeepEval* [Ip and Vongthongsri 2026], operando offline sob temperatura zero ($T = 0$) para garantir um julgamento determinístico.

O modelo juiz operou sob um conjunto de cinco regras estritas de avaliação, sumarizadas a seguir: (1) *TAG NEUTRALITY*: ignorar marcadores como [RAG] ou [INTERNAL] para a métrica de Relevância; (2) *TEMPORAL PROVENANCE*: verificar se o modelo identifica corretamente a "era" do seu conhecimento interno; (3) *VERACITY LOCK*: penalizar com escore zero de Fidelidade qualquer invenção de especificações de SKU não presentes no contexto; (4) *RECENCY BIAS*: exigir que a resposta priorize os timestamps mais recentes do contexto; e (5) *OUTPUT*: produzir exclusivamente um objeto JSON com os campos `score` e `reason`.

A definição empírica dos limiares (*thresholds*) de cada métrica baseou-se na tolerância ao risco inerente a cenários críticos de suporte de hardware [Sarmah et al. 2024], onde recomendações incorretas podem resultar em danos físicos ou falhas de sistema. Para sistematizar a avaliação, as métricas e suas justificativas foram estruturadas:

A precisão contextual (limiar 0,7) avalia a capacidade do motor de busca em posicionar os documentos mais relevantes no topo do ranqueamento. O limiar rigoroso de 0,7 foi estipulado para garantir que o ruído semântico seja minimizado. Um exemplo prático: Se um usuário consulta a capacidade de RAM do modelo específico "Nitro 5 AN515-56" e o sistema ranqueia os manuais do modelo paralelo "AN515-58" nas primeiras posições, o modelo é penalizado matematicamente por falhar na trava de SKU, reprovando no teste.

A revocação contextual (limiar 0,6) avalia se o contexto devolvido pelo FAISS englobou as informações essenciais esperadas. Por lidarmos com dados estocásticos e ruidosos extraídos de fóruns, exigiu-se um limiar mais brando (0,6). Isso reflete a premissa de que capturar a resolução técnica central de um defeito é suficiente, sem a necessidade de recuperar as redundâncias periféricas da discussão. Exemplo prático, Ao buscar uma solução para tela preta, recuperar a instrução exata de atualização de BIOS é suficiente para atingir o limiar, mesmo omitindo comentários secundários de outros usuários.

A fidelidade (limiar 0,8) impõe a menor tolerância ao erro de todo o pipeline para assegurar a política *Zero-Trust*. A alta exigência previne que a LLM utilize o seu conhecimento interno para inventar especificações não presentes na evidência documental. Exemplo prático: Caso a LLM afirme que o notebook suporta um *upgrade* para uma hipotética placa de vídeo "NVIDIA RTX 6080" não mencionada no contexto recuperado, o auditor detecta a alucinação paramétrica e aplica uma penalização máxima, zerando o escore e reprovando a variante.

A relevância da resposta (limiar 0,7) assegura a objetividade técnica da LLM, deduzindo pontos de sentenças evasivas ou redundantes perante a pergunta direta do usuário. Exemplo prático: Se o sistema não encontrar dados precisos no contexto e responder com *"Não possuo essas especificações, mas os notebooks da Acer costumam ser bons para jogos"*, a métrica penaliza a verbosidade desnecessária e a evasão de foco.

4.4. Suíte de Testes e Protocolo de Avaliação

Para garantir total transparência metodológica e mitigar a subjetividade do avaliador automatizado, a suíte de testes foi composta por 13 casos de uso reais extraídos do *corpus*, adotando-se o ano de 2026 como data de referência explícita para as consultas. Cada caso foi curado manualmente por dois autores com experiência em suporte técnico, recebendo um contexto de referência e uma resposta esperada. Discrepâncias entre os anotadores foram resolvidas por consenso, fornecendo assim um gabarito estruturado para o cálculo objetivo das métricas pelo modelo juiz.

Os eixos de avaliação foram distribuídos em três categorias. A primeira categoria, de conhecimento Estático (2 Casos), foca na extração de especificações físicas imutáveis, como a capacidade máxima de memória RAM ou os *slots* SSD suportados por um modelo exato, testando a eficácia da trava de *SKU-Lock*. A segunda categoria, de conflito temporal e anomalias (7 Casos), submete o sistema a cenários de contradição cronológica ou consultas sobre hardware inexistente (como uma placa de vídeo hipotética RTX 6080); este eixo exige que o modelo ative sua trava de anacronismo e admita a ausência da informação em vez de extrapolar dados. Por fim, a categoria de conhecimento híbrido (4 Casos) testa a correlação de conceitos físico-temporais complexos, como os benefícios contínuos de um *design* de ventoinha dupla ou os impactos de atualizações de BIOS, exigindo que o RAG temporal sintetize múltiplas evidências vigentes em uma única resposta coerente.

Embora os 13 casos de teste tenham sido curados manualmente para cobrir diferentes categorias de risco, reconhecemos que esse número é pequeno para generalizações estatísticas robustas. Os resultados devem ser interpretados como prova de conceito em um domínio restrito.

5. Resultados e Discussão

A avaliação do sistema baseou-se na comparação de desempenho entre uma abordagem de RAG convencional (*Vanilla* - V1) e a arquitetura proposta com Auditoria Híbrida e Decaimento Temporal (*V8 Hybrid*). Para corroborar a viabilidade da proposta, todo o processo de inferência e avaliação foi executado estritamente sob a restrição de 4GB de VRAM. Os testes foram conduzidos utilizando o framework DeepEval, com o modelo `Phi-3.5 Mini` atuando como juiz (*LLM-as-a-Judge*) sob temperatura zero, garantindo o determinismo da validação e que é sumarizado numericamente na Tabela 2.

Tabela 2. Métricas Médias de Auditoria: V1 (Vanilla) vs V8 (Híbrido)

Métrica de Avaliação	V1 (Baseline)	V8 Hybrid	Threshold	Status V8
Contextual Precision	0,00	**0,89**	0,70	*Aprovado*
Contextual Recall	0,00	**0,60**	0,60	*Aprovado*
Faithfulness (Fidelidade)	0,62	**0,91**	0,80	*Aprovado*
Answer Relevancy	0,95	**0,97**	0,70	*Aprovado*

Para garantir transparência metodológica, a suíte de testes foi composta por 13 casos de uso reais extraídos do *corpus*, tendo o ano de 2026 como data de referência explícita para as consultas temporais. Cada caso foi curado manualmente com um contexto de referência e uma resposta esperada, fornecendo um "gabarito" de como o modelo juiz deve calcular as métricas de forma objetiva, mitigando o viés de subjetividade na avaliação.

5.1. Análise Quantitativa e Eficácia de Recuperação

A análise das métricas detalhadas revela um contraste severo na mecânica de recuperação. A linha de base (*Vanilla* - V1) falha sistematicamente em mapear as consultas aos documentos técnicos, registrando sucessivos escores nulos $(0,00)$ em Precisão e Revocação Contextual. A análise dos *logs* de auditoria indica que, na variante V1, o contexto retornado frequentemente é vazio ou povoado por nós irrelevantes.

Em contrapartida, a variante V8 demonstrou alta capacidade de ancoragem técnica. Em consultas sobre conhecimento estático, a precisão contextual média saltou para $0,89$, com os *logs* do auditor confirmando que a variante V8 isolou com sucesso as especificações corretas no topo do ranqueamento, descartando menções a modelos da mesma série com sufixos diferentes (validação da trava de SKU). O *Contextual Recall* $(0,60)$ situou-se no limiar de aceitação estrito, refletindo a eficácia do isolamento documental: o sistema abdica de recuperar grandes volumes de texto ruidoso, garantindo que apenas evidências focadas no modelo exato alimentem o contexto da resposta.

É notável, contudo, que mesmo com recuperação nula, o *baseline* V1 atingiu pontuação elevada na métrica de Relevância da Resposta (*Answer Relevancy*), com média de 0,95. O *baseline* V1 instrui o modelo a "fornecer uma resposta direta e útil" e, na ausência de contexto, a "usar seu conhecimento geral". Essa diretriz vaga potencializa a ocorrência alucinação, o modelo V1, mesmo com contexto vazio, é compelido a gerar respostas tematicamente relevantes usando seu conhecimento paramétrico. O acerto ocasional dessas respostas, longe de atestar competência, configura uma falso positivo. Por outro lado, o *V8 Hybrid* impõe um *reasoning pipeline*, técnica de engenharia de prompt que define uma cadeia de raciocínio, que reforça o modelo verificar SKU da pergunta com a resposta, e, que na ausência de evidências, pratica a omissão segura (recusar-se a responder). Essa restrição intencional explica a ligeira superioridade da V8 (0,97), ao responder "Não tenho dados", a saída é tematicamente direta e semanticamente completa, enquanto a V1 frequentemente se perde em explicações prolixas e factualmente frágeis.

A divergência nos escores de relevância, portanto, não é uma falha, mas a demonstração empírica de que métricas de relevância isoladas são incapazes de detectar alucinações fluentes em modelos de linha de base.

5.2. Análise Qualitativa e Comportamento do Auditor

Os registros das justificativas (*reasons*) do juiz automatizado fornecem evidência concreta do como os mecanismos de segurança do protocolo estão sendo analisados, que estão expressos inicialmente de forma numérica, e chegando em uma justificativa verbal.

No aspecto da mitigação de alucinação paramétrica (conhecimento estático - Caso `TC_STATIC_02`), o cenário sobre a quantidade de *slots* M.2 SSD evidenciou duas camadas de falha do modelo *Vanilla*. Por um lado, o prompt permitiu o uso de conhecimento geral, a V1 utilizou seu treinamento paramétrico para afirmar a existência de uma baia de 2,5 polegadas não documentada no contexto recuperado. Por outro, e mais grave, a resposta evadiu-se da pergunta central: em vez de informar objetivamente o número de slots M.2, o modelo produziu um texto genérico sobre compatibilidade de armazenamento e recomendou a consulta ao manual do usuário. O juiz automatizado penalizou essa dupla falha com uma nota de Fidelidade de $0,25$, capturando tanto a invenção de especificações (*unsupported claim*) quanto a ausência de uma resposta direta ao questionamento técnico.

A variante V8, por sua vez, operou sob as restrições de *SKU-Lock* e não inferência por série. Sem evidência documental para o modelo exato, o sistema praticou a omissão segura, atingindo Fidelidade de $1,00$ ao afirmar que a informação não estava disponível no contexto. Esse contraste ilustra o princípio central da arquitetura: em domínios de hardware, é preferível recusar uma resposta a fornecer uma informação não verificada ou evasiva.

O rigor do juiz automatizado também se manifesta em cenários de sensibilidade temporal. No caso `TC_TIME_SENS_01_EXPIRED`, que consultava a validade de uma garantia de 12 meses para um notebook adquirido em janeiro de 2024 com data de referência em março de 2026, a variante V8 respondeu incorretamente que a garantia ainda estaria ativa. O auditor, ancorado na regra de *RECENCY BIAS*, penalizou a Fidelidade em $0,60$, demonstrando que o sistema de avaliação pune ativamente respostas cronologicamente inconsistentes, mesmo quando bem formuladas.

Em relação à resolução de conflitos temporais, o motor LinearV3 atuou como o mitigador do borrão semântico. Em problemas cujos sintomas e soluções são textualmente similares com a diferença em anos (e.g., falhas de *driver* de vídeo), a penalização heurística de 4% ao ano permitiu ao sistema desempatar a distância semântica para priorizar a resolução vigente mais próxima de 2026. Enquanto o *baseline* foi atraído por discussões obsoletas de 2021, pela alta sobreposição lexical, a variante V8 isolou a solução atualizada, elevando a precisão contextual do caso para $0,80$.

Por fim, a incerteza explícita e o rigor científico da arquitetura são demonstrados na média de $0,91$ da variante V8. Apesar de uma média alta na resposta final, essa por si só não passará pela politica de *Zero-Trust* caso a resposta não esteja devidamente documentada e leva o juiz a penalizar o modelo gerador nesses casos. Esse rigor garante que a precisão da fonte primária sobreponha-se à capacidade inferencial isolada do LLM, assegurando a integridade técnica estrita exigida no suporte de hardware.

6. Conclusão

Este trabalho apresentou uma abordagem de RAG temporal para suporte técnico de hardware em ambiente local com restrição de 4GB de VRAM. A combinação entre re-ranking temporal, trava de SKU e auditoria híbrida permitiu reduzir respostas sem ancoragem documental no cenário avaliado. Em comparação com a variante baseline, a versão proposta apresentou ganhos relevantes de precisão contextual e fidelidade, sugerindo que mecanismos simples de sensibilidade temporal e validação estruturada podem aumentar a confiabilidade de pipelines RAG compactos. Esses resultados devem ser interpretados no escopo de uma base curada de notebooks gamer, constituindo uma prova de conceito para investigações futuras em outros domínios de suporte técnico.

A arquitetura proposta distingue-se em seus componentes de domínio. O motor de canonização de SKUs (Seção 3) é intrinsicamente acoplado ao domínio de notebooks, enquanto o algoritmo LinearV3 e o protocolo de auditoria V8 são genéricos, operando sobre qualquer base temporalmente indexada. Contudo, os resultados aqui reportados baseiam-se em uma suíte de 13 casos de teste, um número adequado como prova de conceito, porém pequeno para generalizações estatísticas. A validação em larga escala com testes de significância, bem como a aferição do viés do juiz automatizado por meio de avaliação humana, constituem etapas necessárias para consolidar a robustez do framework.

Como trabalhos futuros, prospecta-se a investigação de mecanismos para a resolução de intenções temporais implícitas (e.g., expressões relativas a atualizações de software), a modelagem da evolução de falhas por meio de grafos de conhecimento temporais (*Temporal Knowledge Graphs*) e a adaptação da arquitetura para a ingestão e indexação contínua de dados (*streaming*) sem comprometer o teto computacional estabelecido.

Agradecimentos

Os autores agradecem a FAPERGS (projetos números 24/2551-0000645-1 e 24/2551-0000669-9) e ao projeto CNPq Universal número 402086/2023-6 e a CAPES.

Referências

Chu, C., Jeong, Y., Cho, H., Kim, J., Lee, J., Bang, B., Lee, J., Song, U., and Kim, S. B. (2026). Cats-rag: Contextual augmented triplet synthesis for rag in technical qa. *Expert Systems with Applications*, page 131491.

Gade, A. and Jetcheva, J. (2024). It's about time: Incorporating temporality in retrieval augmented language models.

Han, J., Cheung, A., Wei, Y., Yu, Z., Wang, X., Zhu, B., and Yang, Y. (2025). Rag meets temporal graphs: Time-sensitive modeling and retrieval for evolving knowledge.

Hill, A. (2026). Retrieval-augmented generation for enterprise search: A comparative user study of rag-based and bm25-based information retrieval systems.

Ip, J. and Vongthongsri, K. (2026). deepeval. Open-source LLM evaluation framework. Source code available at: https://github.com/confident-ai/deepeval.

Meta (2024). Llama 3.2 3B Instruct Model Card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

Microsoft (2024). Phi-3.5 technical report: A highly capable language model.

Ouyang, J., Pan, T., Cheng, M., Yan, R., Luo, Y., Lin, J., and Liu, Q. (2025). Hoh: A dynamic benchmark for evaluating the impact of outdated information on retrieval-augmented generation.

Park, S. (2021). Quality management practices of dell and hp.

Piryani, B., Abdallah, A., Mozafari, J., Anand, A., and Jatowt, A. (2025). It's high time: A survey of temporal question answering.

Sarmah, B., Li, M., Lyu, J., Frank, S., Castellanos, N., Pasquali, S., and Mehta, D. (2024). How to choose a threshold for an evaluation metric for large language models.

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., and Wei, F. (2024). Improving text embeddings with large language models.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. (2025). Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.