Arquitetura RAG Híbrida para Consulta em Linguagem Natural a Dados de Emendas Parlamentares Brasileiras

André Ferreira Santana[1], **Laura Costa Sarkis**[1], **Raoni Simões Ferreira**[2]

[1]CCET – Centro de Ciências Exatas e Tecnológicas – Universidade Federal do Acre (UFAC)
Rio Branco – AC – Brasil

andre.santana@sou.ufac.br, laura.sarkis@ufac.br

[2]Instituto de Computação – Universidade Federal do Amazonas (UFAM)
Manaus – AM – Brasil

raoni@icomp.ufam.edu.br

Abstract. *Transparency in the execution of Brazilian parliamentary amendments is limited by budget vocabulary complexity, data fragmentation across government sources, and the lack of accessible query interfaces. This work proposes a hybrid RAG architecture combining structured SQL search with semantic vector retrieval (pgvector/HNSW) via weighted Reciprocal Rank Fusion (RRF) with similarity gate and adaptive per-query strategy. Evaluated with 120 queries across four complexity levels, the hybrid approach achieved P@5 of 73.33%, NDCG@5 of 64.55%, and MRR of 75.00%, outperforming pure SQL and pure vector baselines on key metrics. Paired t-tests ($n = 120$, $\alpha = 0.05$) confirmed statistical significance for P@5, F1, and NDCG@5, with the largest gain in semantic queries (+15.7 pp NDCG@5). Entity extraction accuracy reached 95.4% overall. The system operates with data from all 27 Brazilian states (2020–2024), comprising 32,787 amendments, and is publicly available as an open-source platform.*

Resumo. *A transparência na execução de emendas parlamentares brasileiras é limitada pela complexidade dos vocabulários orçamentários, fragmentação dos dados entre múltiplas fontes e ausência de interfaces acessíveis. Este trabalho propõe uma arquitetura RAG híbrida que combina busca SQL com recuperação vetorial semântica (pgvector/HNSW) via Reciprocal Rank Fusion (RRF) ponderada com gate de similaridade e estratégia adaptativa por consulta. Avaliada com 120 consultas em quatro níveis de complexidade, a abordagem híbrida alcançou P@5 de 73,33%, NDCG@5 de 64,55% e MRR de 75,00%, superando as abordagens SQL puro e vetorial puro nas principais métricas. Testes t pareados ($n = 120$, $\alpha = 0{,}05$) confirmaram significância estatística para P@5, F1 e NDCG@5, com maior ganho em consultas semânticas (+15,7 pp NDCG@5). A acurácia de extração de entidades alcançou 95,4% no geral. O sistema opera com dados dos 27 estados brasileiros (2020–2024), totalizando 32.787 emendas, e encontra-se disponível como plataforma de código aberto.*

1. Introdução

As emendas parlamentares constituem um dos principais mecanismos de alocação de recursos públicos no Brasil, movimentando bilhões de reais anualmente. Em 2025, o Tribunal de Contas da União (TCU), em nota técnica encaminhada ao Supremo Tribunal

Federal no âmbito da ADPF 854, identificou 964 planos de trabalho com rastreabilidade comprometida, totalizando R$ 694,7 milhões em recursos cuja destinação final não pôde ser adequadamente verificada [Supremo Tribunal Federal 2025]. A Lei Complementar nº 210/2024 [Brasil 2024] e as decisões do Supremo Tribunal Federal na ADPF 854 [Supremo Tribunal Federal 2024] reforçaram a necessidade de transparência e rastreabilidade nesses processos.

O acesso aos dados de emendas parlamentares, embora tecnicamente disponível através de portais como o Portal da Transparência (CGU) [Controladoria-Geral da União 2025] e a API de Dados Abertos da Câmara dos Deputados [Câmara dos Deputados 2025], permanece restrito a usuários com conhecimento técnico. Vocabulários orçamentários especializados, esquemas de dados fragmentados entre múltiplas fontes e a ausência de interfaces intuitivas de consulta criam barreiras significativas para o exercício do controle social [Michener et al. 2018, Tejedo-Romero et al. 2025].

Sistemas baseados em Geração Aumentada por Recuperação (RAG) [Lewis et al. 2020] oferecem uma abordagem promissora ao permitir consultas em linguagem natural sobre bases de dados estruturadas. Embora os dados de emendas parlamentares sejam estruturados (registros JSON via APIs REST [Controladoria-Geral da União 2025] e tabelas relacionais), sua utilização apresenta desafios decorrentes da fragmentação entre fontes com esquemas heterogêneos. Por exemplo, o Portal da Transparência codifica áreas temáticas como funções orçamentárias numéricas (código “10” para Saúde), enquanto o usuário consulta por termos como “saúde”; os dados de beneficiários finais são obtidos a partir de *endpoints* distintos, sem vínculo direto com os registros de emendas.

Dessa forma, os desafios identificados são: (i) a necessidade de combinar recuperação semântica com consultas SQL determinísticas para mapear vocabulário coloquial à classificação orçamentária formal; (ii) a integração de dados de beneficiários finais, dispersos em múltiplos endpoints governamentais; e (iii) a ausência de uma interface unificada que abstraia essa complexidade para usuários sem conhecimento técnico.

O objetivo deste trabalho é projetar, implementar e avaliar uma arquitetura RAG híbrida que permita a consulta em linguagem natural a dados de emendas parlamentares brasileiras, combinando recuperação SQL determinística com busca vetorial semântica para superar as barreiras de vocabulário, fragmentação e acessibilidade identificadas. Diante desses desafios, este trabalho apresenta três contribuições: (i) **Arquitetura RAG híbrida** que combina busca SQL parametrizada com recuperação vetorial semântica (pgvector/HNSW), fusionadas via Reciprocal Rank Fusion (RRF) ponderada [Cormack et al. 2009] com *gate* de similaridade e estratégia adaptativa por consulta; (ii) **Avaliação experimental** com 120 consultas em quatro níveis de complexidade, comparando três abordagens (Text-to-SQL puro, busca vetorial pura e híbrido) por meio de sete métricas de recuperação de informação; (iii) **Plataforma de código aberto** capaz de processar dados em escala nacional (27 estados, 2020–2024), abstraindo a complexidade de vocabulários orçamentários e esquemas fragmentados em uma interface de linguagem natural acessível.

As seções seguintes apresentam os trabalhos relacionados (§2), a arquitetura pro-

posta (§3), a avaliação experimental (§4), a discussão (§5) e as considerações finais (§6).

2. Trabalhos Relacionados

RAG para Dados Estruturados. O paradigma RAG [Lewis et al. 2020], originalmente concebido para textos, foi estendido a dados tabulares e semi-estruturados. [Biswal et al. 2025] introduziram o TAG, argumentando que Text-to-SQL isolado é insuficiente para raciocínio semântico; [Gao et al. 2024] identificam a hibridização como fronteira promissora; [Li et al. 2024] demonstram ganhos com múltiplas representações (StructRAG). Esses trabalhos fundamentam a decisão de combinar modalidades de recuperação.

Text-to-SQL com LLMs. *Surveys* recentes [Liu et al. 2025] mapearam o estado da arte em Text-to-SQL, e [Hong et al. 2025] propuseram uma taxonomia com quatro estágios. [Pourreza and Rafiei 2023] demonstraram que a decomposição contextualizada (DIN-SQL) melhora a acurácia em *benchmarks* como Spider e BIRD [Li et al. 2023]. Diferentemente dessas abordagens generativas, este trabalho adota geração parametrizada determinística, priorizando segurança e reprodutibilidade.

Transparência Governamental e IA. A Operação Serenata de Amor [Op. Serenata de Amor 2016] utilizou ML para identificar gastos suspeitos, demonstrando o potencial da automação no controle social. [Tejedo-Romero et al. 2025] analisaram a usabilidade de portais de dados abertos brasileiros, identificando lacunas em qualidade e acessibilidade; a OECD [OECD 2017] estabeleceu diretrizes para transparência orçamentária. A lacuna identificada é a ausência de sistemas que combinem RAG moderno com dados governamentais brasileiros estruturados.

Avaliação de Sistemas RAG. [Es et al. 2024] propuseram o RAGAS com métricas para respostas fundamentadas (Faithfulness, Answer Relevancy); o CRAG [Yang et al. 2024] mostrou que mesmo as melhores soluções industriais respondem apenas 63% das consultas sem alucinação.

A Tabela 1 sintetiza o posicionamento deste trabalho.

Tabela 1. Comparação com trabalhos relacionados.

Trabalho	Dados	Recuperação	Domínio	Métr. RI
TAG [Biswal et al. 2025]	Tabular	LLM direto	Genérico	–
DIN-SQL [Pourreza and Rafiei 2023]	Tabular	Text-to-SQL	Genérico	Acur.
StructRAG [Li et al. 2024]	Híbrido	Estrut.+LLM	Genérico	EM/F1
CRAG [Yang et al. 2024]	Textual	RAG	Genérico	8 métr.
RAGAS [Es et al. 2024]	Textual	Vetorial	Genérico	Próprias
Serenata [Op. Serenata de Amor 2016]	Tabular	ML	Gov. BR	–
Este trabalho	Híbrido	SQL+Vec.+RRF	Gov. BR	7 métr.

Os trabalhos existentes concentram-se em domínios genéricos ou em uma única modalidade de recuperação. StructRAG é o mais próximo, ao combinar múltiplas representações, mas em domínio genérico e sem fusão ponderada; a Serenata atua no domínio governamental brasileiro, mas via ML supervisionado, sem interface em linguagem natural. Este trabalho diferencia-se ao combinar recuperação SQL determinística

com busca vetorial via fusão RRF ponderada, aplicada a dados governamentais brasileiros com avaliação por sete métricas de RI.

3. Arquitetura Proposta

A arquitetura da plataforma está organizada em quatro camadas, conforme ilustrado na Figura 1: Ingestão, Indexação Híbrida, RAG Híbrido e Interface, sendo descrita em detalhes nas seções 3.1 a 3.4.

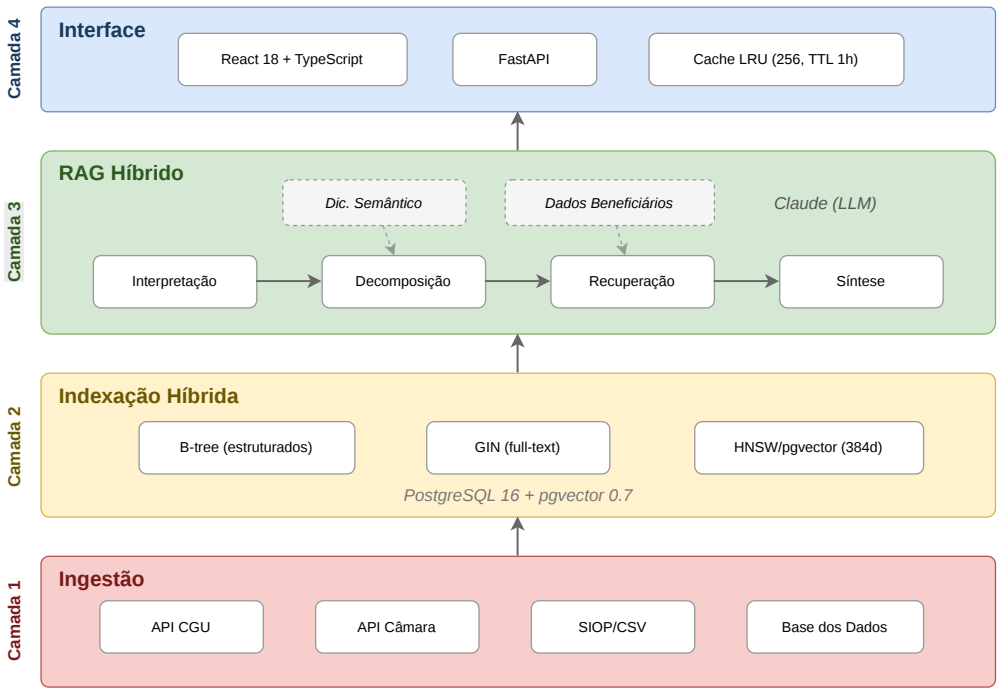

Figura 1. Arquitetura em quatro camadas da plataforma de transparência fiscal.

Para ilustrar o fluxo, considere a consulta "Qual o valor total das emendas individuais da deputada Fernanda Melchionna em 2024?". A **Interpretação** extrai autor, tipo e ano; a **Decomposição** converte essas entidades em filtros SQL parametrizados (`WHERE autor = :autor AND ano = :ano AND tipo = :tipo`); a **Recuperação** aciona `sql_only` por alta especificidade e retorna as 14 emendas sem *fallback* vetorial; a **Síntese** produz uma resposta em linguagem natural com valor empenhado, quantidade de emendas e links para o Portal da Transparência.

3.1. Camada de Ingestão

A camada de ingestão integra e normaliza dados de quatro fontes governamentais heterogêneas — Portal da Transparência/CGU, Câmara dos Deputados, SIOP/SIGA Brasil e Base dos Dados — em um modelo relacional unificado PostgreSQL com 23 tabelas. Os detalhes das fontes, do processo de integração e do esquema resultante são apresentados na Seção 4.3.

3.2. Camada de Indexação Híbrida

Os dados são indexados em PostgreSQL 16 com extensão pgvector [Kane 2024] via três estratégias complementares: (i) B-tree para campos estruturados (ano, UF,

autor, função, tipo); (ii) GIN para busca *full-text* em português; (iii) HNSW [Malkov and Yashunin 2020] (`m=16`, `ef_construction=200`) para similaridade de cosseno em *embeddings* de 384 dimensões (modelo intfloat/multilingual-e5-small [Wang et al. 2024]). Os *embeddings* cobrem quatro entidades — emendas, classificações orçamentárias, documentos de emenda e favorecidos — gerados por concatenação dos campos descritivos e normalizados para cosseno.

3.3. Camada RAG Híbrido

O pipeline RAG processa consultas em quatro etapas:

Etapa 1 — Interpretação: Um LLM (Claude Sonnet 4 [Anthropic 2024], $T = 0$) extrai entidades estruturadas: autor, partido, UF, ano, área temática, tipo, beneficiário e operação.

Etapa 2 — Decomposição: As entidades são mapeadas para filtros SQL e/ou busca vetorial. Um dicionário semântico de termos orçamentários resolve termos como "saúde"→"10"; termos não mapeados acionam busca vetorial na tabela de classificação como *fallback*. Os filtros SQL usam `WHERE` parametrizado com *named placeholders* (`:autor`, `:ano`, `:uf`), sem geração livre de SQL por LLM — diferentemente de DIN-SQL [Pourreza and Rafiei 2023], isso garante determinismo e elimina riscos de injeção SQL.

Etapa 3 — Recuperação Híbrida: Esta etapa compreende dois componentes: o *query planner* e a fusão RRF estendida.

Query planner. Um componente baseado em LLM seleciona a estratégia de recuperação por consulta entre três modalidades: `sql_only`, para consultas com entidades completas (e.g., autor + ano), nas quais filtros SQL são suficientes; `sql_first`, para consultas parcialmente especificadas, com *fallback* vetorial ativado quando o número de resultados SQL é inferior a um limiar de suficiência θ; e `rrf`, para consultas semânticas e de beneficiários, nas quais ambas as buscas são executadas em paralelo e fusionadas. O limiar θ é definido pelo *planner* por consulta ($\theta = 1$ para consultas específicas com autor e ano/UF; $\theta = 20$ para consultas exploratórias). Para consultas de beneficiários, JOINs com `documentos_emenda` e `favorecidos` identificam os favorecidos finais.

Fusão RRF estendida. A fusão híbrida estende a formulação original da RRF [Cormack et al. 2009] em três aspectos: (a) *pesos assimétricos* — a formulação adotada utiliza $w_{\text{sql}} = 0{,}7$ e $w_{\text{vec}} = 0{,}3$, priorizando a precisão dos filtros SQL sobre a abrangência vetorial, dado que os dados são predominantemente estruturados; (b) *gate* de similaridade — candidatos vetoriais com similaridade de cosseno inferior a 0,55 são descartados antes da fusão, evitando que resultados de baixa confiança contaminem o *ranking* final; e (c) *estratégia adaptativa* — a fusão RRF é ativada apenas quando necessário (modo `rrf`), preservando a eficiência SQL em consultas bem-estruturadas. A fórmula resultante é:

$$\text{RRF}(d) = \frac{w_{\text{sql}}}{k + r_{\text{sql}}(d)} + \frac{w_{\text{vec}}}{k + r_{\text{vec}}(d)} \tag{1}$$

com $k = 60$. Os pesos foram definidos empiricamente; a otimização automática é indicada como trabalho futuro.

Etapa 4 — Síntese: O LLM gera uma resposta em linguagem natural acessível,

fundamentada exclusivamente nos dados recuperados, com citação de fontes verificáveis e formatação em markdown.

O Algoritmo 1 formaliza o fluxo: em `sql_only`/`sql_first` a busca SQL é executada primeiro, com *fallback* vetorial e fusão RRF apenas quando o número de resultados é inferior a θ; em `rrf`, ambas as buscas executam em paralelo, com *gate* de similaridade filtrando candidatos vetoriais antes da ordenação por RRF ponderada.

Algorithm 1 Pipeline RAG híbrido com estratégia adaptativa.

Require: consulta q em linguagem natural, base D
Ensure: resposta r em linguagem natural com fontes
$E \leftarrow \textsc{Interpretar}(q)$; $F \leftarrow \textsc{Decompor}(E, D)$; $s \leftarrow \textsc{Planejar}(q, E)$
if $s \neq$ `rrf` **then** ▷ sql_only ou sql_first
  $R_{\text{sql}} \leftarrow \textsc{BuscaSQL}(F, D)$
  if $|R_{\text{sql}}| < \theta$ **then**
    $R_{\text{vec}} \leftarrow \textsc{BuscaVetorial}(q, D)$
    $R \leftarrow \textsc{FusaoRRF}(R_{\text{sql}}, R_{\text{vec}})$
  else $R \leftarrow R_{\text{sql}}$
  end if
else ▷ rrf: buscas em paralelo
  $R_{\text{sql}}, R_{\text{vec}} \leftarrow \textsc{BuscaSQL}(F, D) \parallel \textsc{BuscaVetorial}(q, D)$
  Filtrar R_{vec}: remover d com $\text{sim}(d, q) < 0{,}55$
  $\forall d$: $\text{RRF}(d) = \frac{0{,}7}{60+r_{\text{sql}}} + \frac{0{,}3}{60+r_{\text{vec}}}$
  $R \leftarrow \textsc{Ordenar}(R_{\text{sql}} \cup R_{\text{vec}}, \text{RRF})$
end if
$r \leftarrow \textsc{Sintetizar}(q, R)$ ▷ LLM gera resposta
return r

3.4. Camada de Interface

A interface é uma SPA (React 18 + TypeScript + Vite) com *backend* FastAPI, com dois modos: (i) consulta em linguagem natural, com resposta em *markdown*, indicador de completude ("dados completos" ou "amostra de N de M"), links verificáveis para o Portal da Transparência e a Câmara, sugestões de perguntas e metadados do *pipeline*; abaixo da resposta, uma tabela interativa exibe os registros (parlamentar, partido, UF, função, valores, ano); (ii) painel fiscal exploratório com indicadores, mapa do Brasil, *treemap* de funções, *ranking* de parlamentares e Sankey de fluxo de recursos. Uma paleta de comandos (Ctrl+K) permite navegação ou consulta em linguagem natural de qualquer tela.

4. Avaliação Experimental

4.1. Metodologia

A avaliação seguiu a metodologia Design Science Research (DSR) [Peffers et al. 2007, Hevner et al. 2004]. O artefato foi avaliado com 120 consultas distribuídas em quatro tipos: **A — Factual Direto (30)**, com filtros simples (e.g., "Total de emendas do deputado X em 2024?"); **B — Cruzamento Semântico (30)**, com mapeamento de área temática para classificação orçamentária e/ou busca em documentos de despesa (e.g., "Emendas

para saúde em SP em 2023?"); **C — Comparação/Tendência (30)**, com raciocínio temporal (e.g., "Evolução das emendas para educação 2020–2024?"); **D — Beneficiários (30)**, sobre destinatários finais (e.g., "Quais instituições receberam recursos da emenda X?").

A classificação funcional (A–D) difere da taxonomia por complexidade SQL adotada no Spider [Li et al. 2023]: como a arquitetura não gera SQL livre, a complexidade relevante reside na modalidade de recuperação requerida (campos diretos, mapeamento semântico, raciocínio temporal ou integração de beneficiários), o que aciona caminhos distintos do *pipeline*.

Cada consulta foi executada em três abordagens: (i) Text-to-SQL puro, (ii) busca vetorial pura, e (iii) híbrido proposto, totalizando 360 execuções.

4.2. Métricas

Sete métricas foram utilizadas: P@5 ($K = 5$); R@5 contra o total de correspondentes na base; F1; NDCG@5 com IDCG em nível de coleção; MRR; acurácia de extração de entidades; e latência *end-to-end*. A relevância adapta o *framework* RAGAS [Es et al. 2024] via soma ponderada de campos (UF e autor: 0,4; partido, área temática e beneficiário: 0,3; ano e tipo: 0,2), com limiar $\geq 0{,}4$ e normalização Unicode. Consultas de agregação foram avaliadas pela acurácia de extração.

4.3. Dados e Infraestrutura de Ingestão

A camada de ingestão integra dados de quatro fontes governamentais: Portal da Transparência/CGU (emendas, documentos de despesa e beneficiários finais via API REST), Câmara dos Deputados (dados de parlamentares via API REST), SIOP/SIGA Brasil (classificação funcional-programática via CSV) e Base dos Dados (dados complementares via BigQuery). Os dados de beneficiários finais são obtidos por consulta sequencial a três *endpoints* da API do Portal da Transparência, integrando documentos de despesa, favorecidos por documento e detalhes complementares. Esse processo, realizado previamente, permite que consultas sobre destinatários finais sejam respondidas a partir dos dados pré-integrados na base relacional.

A normalização unificou vocabulários e esquemas heterogêneos (JSON/REST, CSV, BigQuery) em um modelo PostgreSQL com 23 tabelas em quatro grupos: (i) centrais (parlamentares, emendas, execução orçamentária); (ii) beneficiários (documentos de emenda, favorecidos por CPF/CNPJ e junção documento–favorecido); (iii) classificação orçamentária (com *embeddings* vetoriais, funções, subfunções, programas, ações); e (iv) integridade (cadastros de sanções CEIS/CNEP/CEPIM/CEAF e convênios).

Em escala nacional (2020–2024, 27 estados), o sistema processa 915 parlamentares, 32.787 emendas com valores de empenho, liquidação e pagamento, 730.571 documentos de despesa e 41.883 beneficiários finais. A disponibilidade de beneficiários depende da execução orçamentária: apenas emendas com pagamento registrado geram registros na tabela.

4.4. Resultados

A Tabela 2 sumariza os resultados comparativos entre as três abordagens, obtidos a partir da execução de 360 avaliações (120 consultas $\times$ 3 abordagens).

Tabela 2. Resultados comparativos por tipo de consulta e abordagem.

Tipo	Abordagem	P@5	R@5	F1	NDCG@5	Lat.(s)
A	SQL Puro	0,17	0,06	0,07	**0,16**	12,5
	Vetorial	0,17	0,06	0,07	0,14	13,0
	Híbrido	0,17	0,06	0,07	**0,16**	12,9
B	SQL Puro	0,70	0,37	0,35	0,75	14,5
	Vetorial	0,81	0,72	0,49	0,82	14,6
	Híbrido	**0,84**	**0,74**	**0,51**	**0,91**	26,2
C	SQL Puro	**0,96**	**0,14**	**0,18**	**0,77**	15,7
	Vetorial	0,89	0,11	0,14	0,70	16,1
	Híbrido	**0,96**	**0,14**	**0,18**	**0,77**	15,6
D	SQL Puro	**0,97**	**0,14**	**0,21**	**0,75**	15,8
	Vetorial	0,90	0,12	0,19	0,54	15,5
	Híbrido	**0,97**	**0,14**	**0,21**	**0,75**	17,1

Os resultados revelaram comportamentos distintos por tipo. No Tipo A (factual), as três abordagens apresentaram métricas uniformemente baixas (P@5≈0,17), com acurácia de extração perfeita (1,000). O Tipo B (semântico) apresentou a maior diferenciação: a híbrida alcançou NDCG@5 de 0,91 frente a 0,75 do SQL puro (+15,7 pp). Nos Tipos C e D, a abordagem híbrida obteve resultados idênticos ou próximos ao SQL puro (P@5 de 0,96/0,97). O *planner* distribuiu as consultas entre `sql_only` (38%), `sql_first` (30%) e `rrf` (32%). As latências variaram entre 12,9 s e 26,2 s. A Seção 4.6 analisa esses padrões.

A Tabela 3 apresenta a acurácia de extração de entidades e a taxa de consultas com resultados para a abordagem híbrida.

Tabela 3. Acurácia de extração de entidades e taxa de consultas com resultados por tipo. A etapa de interpretação é compartilhada entre as três abordagens.

Tipo	Acur. Entidades	c/ resultados	s/ resultados
A — Factual	1,000	30/30	0/30
B — Semântico	0,944	30/30	0/30
C — Comparativo	1,000	30/30	0/30
D — Beneficiário	0,872	30/30	0/30
Geral	0,954	120/120	0/120

Os Tipos A e C alcançaram acurácia perfeita; o Tipo B (0,944) apresentou erros em áreas temáticas compostas; o Tipo D (0,872) refletiu a complexidade de nomes de beneficiários não-canônicos. Todas as 120 consultas retornaram resultados.

A Tabela 4 apresenta a comparação agregada entre as três abordagens, consolidando as métricas gerais para as 120 consultas.

[1]Médias aritméticas sobre as 120 consultas individuais; incluem o Tipo A, cujas métricas de recuperação são inerentemente baixas (Seção 4.6).

Tabela 4. Comparação agregada: desempenho médio sobre as 120 consultas individuais, por modalidade de recuperação.[1]

Métrica	SQL Puro	Vetorial	Híbrido
P@5 média (%)	69,83	69,00	**73,33**
R@5 média (%)	17,78	25,12	**27,01**
F1 médio (%)	20,44	22,26	**24,39**
NDCG@5 médio (%)	60,63	54,82	**64,55**
MRR (%)	74,17	74,03	**75,00**
Latência média (s)	14,6	14,8	21,0
Consultas c/ resultado	120/120	120/120	120/120

Testes pareados (t-student bicaudal, $n = 120$, $\alpha = 0{,}05$, *bootstrap* 95%/10.000 reamostras) confirmaram diferenças significativas do híbrido sobre o vetorial para P@5 ($p = 0{,}001$), F1 ($p = 0{,}016$) e NDCG@5 ($p < 0{,}001$), e sobre o SQL puro para P@5 ($p = 0{,}015$) e NDCG@5 ($p = 0{,}029$); MRR não atingiu significância em nenhuma comparação. As diferenças concentraram-se no Tipo B ($p = 0{,}013$ para P@5, $p = 0{,}027$ para NDCG@5); nos demais tipos, o *planner* direcionou para `sql_only`/`sql_first`, produzindo resultados idênticos.

4.5. Análise de Latência

O *overhead* do híbrido varia por tipo: em A, C e D, o roteamento para `sql_only`/`sql_first` mantém latências comparáveis às puras (12,5–17,1 s). O Tipo B atinge 26,2 s (+81% sobre SQL puro), por acionar fusão RRF completa, mas o acréscimo de ≈12 s é compensado pelo ganho de +15,7 pp em NDCG@5. A latência é dominada pelas chamadas ao LLM (interpretação e síntese, ≈70%), enquanto as buscas contribuem com ≈30%; a substituição por modelos locais é a principal oportunidade de otimização.

4.6. Análise de Acertos e Falhas

Acertos. *Roteamento adaptativo:* nos Tipos C e D, o *planner* direcionou corretamente para `sql_only`/`sql_first`, igualando o SQL puro; a distribuição entre estratégias (38%/30%/32%) indica seleção efetiva. *Vantagem semântica no Tipo B:* a fusão RRF obteve ganho de 15,7 pp em NDCG@5 sobre o SQL puro, com a busca em documentos de despesa contribuindo para recuperação de emendas relacionadas a obras e serviços específicos. *Gradiente na extração:* a acurácia decresceu de 1,000 (A, C) para 0,944 (B) e 0,872 (D), concentrando a dificuldade em áreas temáticas compostas e nomes de beneficiários não-canônicos.

Falhas. *(i) Avaliação inadequada para agregação:* o Tipo A apresentou P@5≈0,17 nas três abordagens, apesar de extração perfeita — consultas que retornam linha agregada não são verificáveis por correspondência individual (limitação da metodologia, não do sistema). *(ii) Extração de beneficiários:* a menor acurácia (0,872) reflete a complexidade de nomes abreviados ou não-canônicos; a vetorial pura nesse tipo (NDCG@5=0,54) ficou bem abaixo do híbrido (0,75), indicando que *embeddings* isolados são insuficientes para consultas com alto grau de estruturação. *(iii) Ausência de corpus anotado:* mitigada parcialmente por *recall* verdadeiro e IDCG em nível de coleção.

5. Discussão

5.1. Contribuições em Relação aos Trabalhos Relacionados

Em relação ao TAG [Biswal et al. 2025], a contribuição residiu na aplicação a dados governamentais brasileiros, com requisitos de mapeamento semântico e rastreabilidade não contemplados por *benchmarks* como BIRD [Li et al. 2023]. A principal contribuição técnica é a extensão da RRF [Cormack et al. 2009] com pesos assimétricos, *gate* de similaridade e estratégia adaptativa, com vantagem estatisticamente significativa sobre as abordagens puras (Tabela 4). Em contraste com abordagens Text-to-SQL [Hong et al. 2025], a fusão RRF demonstrou ganhos expressivos em consultas semânticas (Tipo B), enquanto o *planner* preservou a eficiência SQL em consultas bem-estruturadas. A integração de dados de beneficiários finais oferece uma interface unificada que facilita a verificação da destinação de recursos, alinhando-se à LC 210/2024 [Brasil 2024].

5.2. Limitações

As principais limitações são: (i) dependência de LLM proprietário (Claude Sonnet 4, US$ 0,02–0,03/consulta) e ingestão em modo *batch*; (ii) pesos RRF e limiar empíricos, sem ablação por componente (custo das 360 reavaliações); (iii) avaliação primária por um dos autores; (iv) execução única por consulta, justificada pelo custo (US$ 7–10/rodada) e pelo determinismo (100% estável em teste 5×3); (v) avaliação inadequada para agregação (Tipo A) e ausência de avaliação de *faithfulness*.

5.3. Ameaças à Validade

Interna: Consultas autorais; mitigada por distribuição equilibrada (deputados, partidos, estados, áreas). **Construção:** Sem corpus anotado; mitigada por *recall* verdadeiro e IDCG em nível de coleção. **Conclusão:** Testes pareados confirmaram significância (Tabela 4); teste 5×3 confirmou determinismo. **Externa:** Resultados limitados a emendas parlamentares brasileiras; arquitetura generalizável.

6. Conclusão

Este trabalho propôs e avaliou uma arquitetura RAG híbrida baseada em fusão RRF ponderada [Cormack et al. 2009] com *gate* de similaridade e estratégia adaptativa, aplicada a emendas parlamentares brasileiras. A abordagem híbrida superou as abordagens puras (P@5 de 73,33%, NDCG@5 de 64,55%), com significância estatística ($n = 120$) concentrada no Tipo B (+15,7 pp NDCG@5). A acurácia de extração alcançou 95,4% e todas as 120 consultas retornaram resultados, validando a viabilidade da arquitetura para transparência fiscal em escala nacional.

Como trabalhos futuros, destacam-se: (i) estudo de ablação isolando a contribuição dos pesos assimétricos, do *gate* de similaridade e do *planner* adaptativo, seguido de otimização automática dos pesos RRF e do limiar; (ii) substituição do LLM proprietário por modelos abertos e locais (e.g., Llama, Mistral); (iii) avaliação de *faithfulness* via RAGAS [Es et al. 2024] com múltiplos avaliadores independentes; (iv) metodologia específica para consultas de agregação e construção de corpus anotado; e (v) comparação com *pipelines* Text-to-SQL generativos e sistemas baseados em agentes.

A plataforma foi disponibilizada como software de código aberto em `https://github.com/astrosoftbr/FiscalIA` [Santana et al. 2026], e encontra-se em

operação em `https://fiscalia.astrosoft.com.br`. O repositório inclui, além do código-fonte, o conjunto de 120 consultas de avaliação com os respectivos critérios de relevância, constituindo um *benchmark* reprodutível para o domínio de emendas parlamentares.

Declaração de Uso de IA

Ferramentas de inteligência artificial generativa (Claude, Anthropic) foram utilizadas como apoio nas seguintes etapas deste trabalho: (i) redação e revisão do texto do artigo; (ii) desenvolvimento do código-fonte da plataforma; (iii) geração e formatação de tabelas e figuras; e (iv) análise estatística dos resultados experimentais. Todo o conteúdo gerado com auxílio de IA foi revisado, validado e editado pelos autores, que assumem integral responsabilidade pelo trabalho apresentado.

Referências

Anthropic (2024). The Claude 3 model family: Opus, sonnet, haiku. Model Card.

Biswal, A., Patel, L., Jha, S., Kamsetty, A., Liu, S., Gonzalez, J. E., Guestrin, C., and Zaharia, M. (2025). Text2SQL is not enough: unifying AI and databases with TAG. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, Amsterdam, The Netherlands.

Brasil (2024). Lei complementar nº 210, de 25 de novembro de 2024: dispõe sobre a transparência e rastreabilidade das emendas parlamentares. Brasília: Diário Oficial da União.

Câmara dos Deputados (2025). Dados abertos da Câmara dos Deputados: documentação da API.

Controladoria-Geral da União (2025). Portal da transparência do Governo Federal: documentação da API de dados.

Cormack, G. V., Clarke, C. L., and Buettcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference*, pages 758–759. ACM.

Es, S., James, J., Espinosa-Anke, L., and Schockaert, S. (2024). RAGAS: automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–163. Association for Computational Linguistics.

Gao, Y. et al. (2024). Retrieval-augmented generation for large language models: a survey. *arXiv preprint arXiv:2312.10997*.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

Hong, Z. et al. (2025). Next-generation database interfaces: a survey of LLM-based text-to-SQL. *IEEE Transactions on Knowledge and Data Engineering*.

Kane, A. (2024). pgvector: open-source vector similarity search for PostgreSQL.

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

Li, J. et al. (2023). Can LLM already serve as a database interface? A BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems*, volume 36.

Li, Z. et al. (2024). StructRAG: boosting knowledge intensive reasoning of LLMs via inference-time hybrid information structurization. *arXiv preprint arXiv:2410.08815*.

Liu, X. et al. (2025). A survey of text-to-SQL in the era of LLMs: where are we, and where are we going? *arXiv preprint arXiv:2408.05109*. Revisado em dezembro de 2025.

Malkov, Y. A. and Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836.

Michener, G., Contreras, E., and Niskier, I. (2018). From opacity to transparency? Evaluating access to information in Brazil five years later. *Revista de Administração Pública*, 52(4):610–639.

OECD (2017). *OECD Budget Transparency Toolkit: practical steps for supporting openness, integrity and accountability in public financial management*. OECD Publishing, Paris.

Op. Serenata de Amor (2016). Inteligência artificial para controle social dos gastos públicos.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77.

Pourreza, M. and Rafiei, D. (2023). DIN-SQL: decomposed in-context learning of text-to-SQL with self-correction. In *Advances in Neural Information Processing Systems*, volume 36.

Santana, A. F., Sarkis, L. C., and Ferreira, R. S. (2026). FiscalIA: Plataforma de transparência fiscal com RAG híbrido. Repositório público contendo código-fonte completo da plataforma.

Supremo Tribunal Federal (2024). Decisão monocrática do min. flávio dino na ADPF 854: suspensão de emendas parlamentares por falta de transparência. Brasília: STF.

Supremo Tribunal Federal (2025). STF determina que TCU identifique e envie à PF lista com emendas parlamentares sem plano de trabalho. Decisão monocrática do Min. Flávio Dino na ADPF 854. Brasília: STF.

Tejedo-Romero, F., Araujo, J. F. F. E., and Ribeiro, M. J. G. (2025). The usability of Brazilian government open data portals: ensuring data quality. *Humanities and Social Sciences Communications*, 12.

Wang, L. et al. (2024). Multilingual E5 text embeddings: a technical report. *arXiv preprint arXiv:2402.05672*.

Yang, X. et al. (2024). CRAG – comprehensive RAG benchmark. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*.