applied sciences

MDPI

Article

GraphRAG Retrieval of Regulatory Knowledge: An Empirical Evaluation for Ecuadorian Tax Legislation Query Answering

Daniel Vinicio Cuatín Flores [1,*], Jorge Andrés Mosquera Sotomayor [1], Carlos Iván Díaz Hidalgo [1], Pablo Marcel Recalde Varela [1,*,†], Paúl Francisco Baldeón-Egas [1,†], Monserrate Intriago-Pazmiño [2,†] and Yadira Quiñonez [3,†]

[1] Departamento de Ciencias de la Ingeniería, Universidad Tecnológica Israel, Quito 170522, Ecuador; e1721715025@uisrael.edu.ec (J.A.M.S.); e1709431132@uisrael.edu.ec (C.I.D.H.); pbaldeon@uisrael.edu.ec (P.F.B.-E.)
[2] Departamento de Informática y Ciencias de la Computación, Escuela Politécnica Nacional, Quito 170525, Ecuador; monserrate.intriago@epn.edu.ec
[3] Facultad de Informática Mazatlán, Universidad Autónoma de Sinaloa, Sinaloa 82017, Mexico; yadiraqui@uas.edu.mx
* Correspondence: e1722864087@uisrael.edu.ec (D.V.C.F.); precalde@uisrael.edu.ec (P.M.R.V.)
† Academic Tutors.

Featured Application

The evaluated architecture is intended for deployment as a tax-consultation assistant for Ecuadorian accountants, tax practitioners, and taxpayers. Its modular design may be adapted to other dynamic regulatory domains in Latin America, such as public procurement, banking, or environmental compliance; however, the performance results reported here have not been validated in those domains.

Abstract

Tax regulation in Ecuador changes constantly, and the resulting information is scattered across laws, regulations, resolutions, forms, and annexes with strong ties across meaning and time. We compare four retrieval-augmented generation (RAG) strategies for question answering over Ecuadorian tax legislation (vector-only, GraphRAG, hybrid, and agentic retrieval) in a system that supports text, speech, and image inputs and combines a tax knowledge graph, dense vector search, and three agents that refine, validate, and plan each query's strategy. Following a Design Science Research approach informed by a SALSA literature review, the system was tested on a curated 51-document corpus using 100 text-only tax queries across the four retrieval modes and two cloud generator models (gemma4:31b-cloud and gpt-oss:20b-cloud), for 800 RAGAS-judged executions, plus a Likert-scale review by five domain specialists. Speech- and image-input performance was not evaluated in this benchmark. Graph-only retrieval was weakest on all five RAGAS metrics (Context Precision = 0.02), while hybrid gave the best balance (Faithfulness = 0.67, Answer Relevancy = 0.64, Context Precision = 0.52, Context Recall = 0.76) at the lowest latency (2.66 s). Faithfulness depended more on the generator model than on retrieval mode (0.78 vs. 0.47). No strategy won on every metric, so architecture choice in dynamic regulatory domains should follow operational priority rather than a single universally optimal setup.

Keywords: GraphRAG; agentic RAG; hybrid retrieval-augmented generation; knowledge graph; dynamic regulatory domains; tax legislation retrieval

Check for updates

Academic Editor: Petros Nicopolitidis

Received: 9 September 2026
Revised: 10 October 2026
Accepted: 20 October 2026
Published: 25 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Ecuador's fiscal and regulatory environment has become increasingly volatile, and its resources continue to be dispersed among many legal instruments. During the fiscal years 2024 and 2025, a number of economic reforms changed the rental tax regulations, retention regulations, investment incentives, Impuesto al Valor Agregado (IVA) provisions, and temporary contributions [1]. These changes demonstrate the dynamic nature of Ecuadorian fiscal regulation, which allows laws, regulations, resolutions, formulas, and appendices to be updated, replaced, or supplemented throughout time. Therefore, determining which dispositions apply to a particular consultation may necessitate relating many documents and taking into account the period during which a disposition was in effect.

The time dimension of regulatory knowledge also makes information retrieval more difficult. According to [2], knowledge is not static and can change with the availability of new information. Therefore, retrieval systems operating in dynamic regulatory environments must be able to take into account changes in the underlying knowledge base, otherwise they will return provisions which are no longer relevant or fail to return information introduced by subsequent updates. This challenge is especially important in high-stakes domains such as law and finance, where outdated information can directly affect the reliability of system outputs [3].

Large language models (LLMs) have shown great capabilities for natural-language understanding and generation. However, their parametric knowledge may become outdated and insufficient for recent or domain-specific knowledge. They may also hallucinate information that is not supported, which is particularly problematic in regulated applications where accuracy, traceability and time validity are key [3,4]. Retrieval-Augmented Generation (RAG) addresses these limitations by incorporating external evidence into the generation process, reducing the dependence on the model's parametric knowledge [5,6].

Nevertheless, regulatory inquiries may necessitate more than mere semantic similarity between a query and specific documents. They may involve explicit relationships among entities, provisions, documents, and temporal conditions. Graph-based, hybrid, and agentically orchestrated retrieval methods have been suggested as a solution to the complementary limitations of conventional vector retrieval. However, their computational requirements and effectiveness may vary significantly. Consequently, it is still imperative to conduct an empirical investigation into the comparative effectiveness of these strategies in the context of dynamic regulatory question answering.

This study, therefore, compares the performance of Vector RAG, GraphRAG, Hybrid RAG, and Agentic RAG to answer questions about Ecuadorian tax legislation. The evaluation considers answer faithfulness, answer relevance, contextual retrieval, computational latency, and the influence of the generator model. The main objective is to characterize the performance trade-offs among these retrieval strategies and determine when the additional complexity of graph-based or agentic retrieval is justified in a dynamic regulatory domain.

To support reproducibility, the source code and technical documentation were released in a public repository, so the technical community can install, run, and audit the artifact directly, which also helps reproducibility (see the Data Availability Statement).

This study makes three contributions. First, a comparative experimental evaluation that quantifies four RAG paradigms over a real tax corpus, measuring the trade-off between normative faithfulness (Faithfulness), answer pertinence (Answer Relevancy), and processing latency. Second, an exploratory, descriptive account of graph activation by manually assigned query category; because the complex-query subgroup is small and imbalanced, this analysis is not used to establish routing accuracy or a general association with query complexity. Third, an empirical characterization of the trade-offs among RAG architectures according to operational priorities such as speed, traceability, or adaptive retrieval.

2. Related Work

2.1. Retrieval-Augmented Generation (RAG)

RAG combines information retrieval with large language models (LLMs) and integrates externally retrieved evidence into the generation process. This decreases dependency on parametric knowledge and allows for the inclusion of updated information without retraining the underlying model [4,5]. This has been helpful for knowledge-intensive and rapidly evolving domains, in particular by improving grounding and reducing unsupported generation [6].

Adaptive retrieval and answer verification mechanisms have been proposed in several extensions, including Corrective RAG (CRAG), which assesses retrieved information prior to generation [7], and Self-RAG, which uses self-reflection to decide when retrieval is needed and to critique the generated answers [8]. However, traditional RAG depends heavily on similarity-based retrieval on independent document fragments, which may not be sufficient for answering queries that require explicit relationships among entities, documents or heterogeneous sources [9]. Such limitations motivate the use of structured representations, multimodal retrieval and adaptive orchestration mechanisms [9].

2.2. Multimodal Retrieval-Augmented Generation (MRAG)

Multimodal Retrieval-Augmented Generation (MRAG) extends conventional RAG by incorporating heterogeneous information sources such as text, images, tables, and other visual content into retrieval and generation processes [10]. This capability is particularly relevant for specialized domains in which information is distributed across heterogeneous document formats.

Recent approaches have explored structured representations to improve multimodal retrieval. For example, multimodal knowledge graphs have been employed to explicitly model the relations between textual and visual information, resulting in better retrieval coherence as compared to embedding-based approaches [11]. Other work has explored complex document understanding via hierarchical multimodal retrieval [12] and specialized retrieval mechanisms for high-resolution visual information [13]. While these advances have been made, multimodal RAG still encounters challenges in interpreting information from different modalities together through shared entities and relationships.

This limitation is of particular importance for regulatory documents. In such documents the answer can be found by legal provisions, tables, forms, diagrams and supporting documents taken together. This means that multimodal retrieval alone might not have enough structural context to support complex regulatory reasoning.

2.3. Knowledge Graphs and GraphRAG

Knowledge Graphs are explicit representations of entities and their relations. They cluster information not by text segments but by semantic and structural dependencies. RAG architectures employ graph-based retrieval preserving entity-document relations and allowing reasoning over inter-linked evidence [11]. GraphRAG improves retrieval using graph structures for retrieval and generation. This is useful for multi-step reasoning or when the information is spread over multiple documents [9].

Recently, graph-based retrieval has been advanced from different aspects. Papageorgiou et al. [14] proposed a hybrid multi-agent GraphRAG architecture for e-government applications, combining graph-based retrieval and agent collaboration to improve reliability and traceability. Similarly, Ref. [15] proposed a dynamic-subgraph retrieval mechanism to preserve the semantic relationships during retrieval and minimize the contextual information loss. These approaches show the potential of graph-based representations for complex knowledge-intensive tasks.

However, existing GraphRAG systems have major limitations. Their performance is susceptible to the quality, completeness and coverage of the underlying knowledge graph. Dynamic update and multimodal information integration still remain challenging. Moreover, graph retrieval is often based on predefined or relatively static retrieval procedures, limiting the ability to adapt the retrieval strategy to different query complexities. These limitations motivate architectures capable of dynamically planning retrieval, selecting appropriate information sources, and iteratively verifying the evidence used for generation.

2.4. Agentic RAG and Agentic Orchestration

Agentic RAG extends traditional retrieval architectures by incorporating planning and autonomous decision-making in the retrieval-generation process. Rather than a single retrieval operation, agentic systems can break up complicated queries, choose retrieval strategies or tools, gather further evidence iteratively, and verify intermediate or final results [16]. This architecture is particularly relevant when the information required to answer a query is distributed across heterogeneous sources or when different retrieval mechanisms are appropriate for different types of questions.

Recent studies illustrate the potential of agentic orchestration in specialized domains. Hu et al. [17] proposed a self-correcting agentic GraphRAG architecture for clinical decision support, which applies iterative retrieval and validation to improve answer consistency. In e-government, Papageorgiou et al. [14] used multiple agents to coordinate query planning, graph retrieval and answer validation. Similarly, reinforcement-learning-based coordination has been explored for multimodal retrieval and reasoning [18]. These studies indicate that agentic mechanisms can complement structured retrieval by adapting the reasoning and retrieval process to task requirements.

Nevertheless, agentic RAG introduces additional computational and architectural costs. Iterative retrieval, agent coordination, tool selection, and repeated verification may increase latency and resource consumption. Furthermore, most existing evaluations have focused on specific domains such as medicine, pathology, or government services, leaving limited empirical evidence regarding the trade-off between retrieval quality, faithfulness, and computational cost in dynamic regulatory environments. This gap motivates the comparative evaluation of agentic and non-agentic RAG architectures in the present study.

2.5. Knowledge Retrieval in Dynamic Regulatory Domains

Regulatory knowledge differs from many specialized domains because its validity depends on temporal, hierarchical, and cross-document relationships. Laws, regulations, resolutions, technical annexes, forms, and interpretive provisions may be modified, replaced, or explicitly referenced by other regulatory instruments. Consequently, reliable retrieval requires not only identifying semantically relevant information but also determining whether the retrieved provision is applicable and sufficiently supported by its regulatory context [14,19]. RAG is well suited to dynamic regulatory environments because external knowledge can be incorporated at inference time without continuously retraining the underlying language model. When the document contains cross-references, amendments, or different versions of the same regulation, similarity-based retrieval may return fragmented evidence or fail to distinguish provision relationships [20]. Graphs can explicitly represent regulatory entity-document relations, helping solve this problem. Hybrid retrieval uses structured or lexical evidence and semantic similarity.

Existing research has explored these approaches in legal and regulatory contexts. ChatLaw, for example, combines vector and keyword-based retrieval with knowledge-graph-enhanced language modeling in a multi-agent legal assistant [21]. Other studies have investigated graph-based approaches for regulatory documents and enterprise knowl-

edge [19]. Nevertheless, the available literature provides limited comparative evidence on how vector, graph, hybrid, and agentic retrieval strategies perform under the same regulatory workload, particularly when the evaluation considers both answer quality and computational cost.

Ecuadorian tax regulation represents a relevant case for this problem because its knowledge base includes laws, regulations, administrative resolutions, forms, technical annexes, circulars, and interpretive provisions that are periodically updated and interdependent. Therefore, an effective retrieval architecture must combine current evidence, structural relationships, multimodal information, and mechanisms for adaptive retrieval. This motivates the experimental comparison of Vector RAG, GraphRAG, Hybrid RAG, and Agentic RAG conducted in this study.

Overall, the literature shows a gradual evolution from traditional vector-based RAG to multimodal, graph-based, hybrid and agentic retrieval architectures. Vector RAG provides a computationally efficient baseline, whereas GraphRAG introduces explicit relationships and supports structured reasoning. Hybrid approaches combine complementary retrieval signals, while agentic architectures add adaptive planning and iterative verification. Despite these advances, few studies provide a controlled comparison of these paradigms within a dynamic regulatory environment while jointly considering answer faithfulness, relevance, and computational latency. The present study addresses this gap by evaluating the four architectures over a common Ecuadorian tax-regulation corpus and under a common evaluation framework.

3. Materials and Methods

3.1. Research Design

This study took an experimental, empirical approach: it combines designing and building a multimodal regulatory-knowledge-retrieval system with a quantitative evaluation of its performance and a qualitative check on its effectiveness for Ecuadorian tax legislation. The work is applied, because it builds a technological artifact meant to solve the problem of accessing complex regulatory information. It is also descriptive, since it characterizes system behavior across different retrieval architectures within the tax-regulatory domain. The experimental evaluation compares retrieval and answer-quality metrics and execution latency across four retrieval strategies using text-only queries; speech- and image-input performance was not evaluated. Validation combined three things: automated benchmarking with the RAGAS framework alongside computational-performance metrics and expert review of the legal precision of the generated content.

The study follows the Design Science Research (DSR) paradigm, which centers on designing, building, and validating technological artifacts that solve real problems in specialized domains. That approach lets us do more than build a working prototype: it lets us empirically show that the prototype actually improves knowledge retrieval in Ecuadorian tax regulation. Peffers et al. [22] describe DSR as systematically tying together problem identification, design, implementation, and evaluation, which keeps both scientific rigor and practical usefulness in view; Hevner [23] adds that validating an artifact has to rest on experimental evidence gathered in real scenarios.

For the literature review we used the SALSA methodology (Search, Appraisal, Synthesis and Analysis), a structured way to identify, evaluate, and synthesize scientific evidence without the full exhaustiveness of a systematic review [24,25]. It is a common choice in engineering and information-systems reviews precisely because it balances methodological rigor with flexibility, which suits building a conceptual framework rather than running an exhaustive systematic review. In the Search phase, systematic literature queries were conducted between June 28 and 30 July 2026, using SciSpace as an integrated AI search

engine querying SciSpace Full Text, Google Scholar, and arXiv. The primary query target was 'Multimodal GraphRAG Framework with Agentic Orchestration for Knowledge Retrieval in Dynamic Regulatory Domains', supported by Boolean search strings such as ("multimodal GraphRAG" OR "multimodal graph RAG") AND ("agentic orchestration" OR "knowledge retrieval"). The initial retrieval yielded 200 candidates from SciSpace metadata/full-text searches, alongside targeted retrievals from Google Scholar and arXiv. After automated deduplication and cross-database reranking, 89 unique candidate papers were compiled. Initial title and abstract screening filtered this set down to 48 potentially relevant publications. In the Appraisal phase, full texts were evaluated against predefined criteria: (i) published between 2021 and 2026, (ii) direct alignment with multimodal RAG, GraphRAG, or agentic orchestration in specialized/regulatory domains, and (iii) indexing in Scopus or Web of Science within Q1 or Q2 quartiles. Ultimately, seven articles met all criteria.

Additionally, one IEEE-indexed study was retained as an explicit thematic exception because it empirically demonstrates that combining Knowledge Graphs with agentic filtering outperforms large vision-language models (e.g., LLaVA-Med) in structured precision. In the Synthesis and Analysis phases, information from the selected studies was organized around the main thematic axes in the literature, which let us compare approaches, spot technological trends, recognize the limits of existing research, and pin down the scientific gaps this study addresses (Section 2).

3.2. Procedure

The research procedure ran through six sequential phases: (i) identifying the problem and analyzing the Ecuadorian tax domain; (ii) collecting and preparing the documentary corpus; (iii) building the knowledge graph and generating multimodal vector representations; (iv) developing the multimodal GraphRAG system with agentic orchestration; (v) experimentally evaluating it with tax queries and specialized metrics; and (vi) analyzing and interpreting the results. Each phase fed the next one its inputs, which kept the research process traceable end to end. Validation drew on both domain experts and end users, since the two give complementary kinds of evidence: experts check technical precision, regulatory consistency, and answer reliability, while end users judge usefulness, ease of use, and how acceptable the solution is. Combining both angles strengthens the validity of the results and gives a more complete picture of the artifact, in line with methodological recommendations for validating DSR artifacts and with recent work evaluating RAG systems in specialized domains [26,27].

3.3. Population and Sample

Five specialists with backgrounds in taxation, accounting, and software architecture participated in the evaluation. Participants were selected purposively based on their professional experience and relevance to the system assessment. All had at least five years of professional experience, ranging from five to thirteen years (Table 1). This criterion refers to professional experience and should not be interpreted as indicating that all participants had equivalent expertise in Ecuadorian tax regulations.

Table 1. Professional profile of the five specialists who validated the system.

Specialist	Field/Academic Background	Professional Role	Years
Specialist 1	Accounting (costing and taxation)	Portfolio analyst	5
Specialist 2	Accounting/business management	Manager	13
Specialist 3	Accounting and auditing (Ph.D.)	Internal auditor	10
Specialist 4	Finance and auditing (CPA)	Accounting assistant	10
Specialist 5	Software engineering and information systems (M.Sc.)	Software architect	7

3.4. Techniques and Instruments

The technology stack was chosen for interoperability, computational efficiency, and multimodal-data support (Table 2), covering document processing, knowledge-graph management, agentic routing, and system evaluation. Implementation and experimental evaluation ran on Azure for Students, across two virtual machines: an ingestion VM (Standard_E4s_v3, 4 vCPU/32 GB RAM, 64 GB disk) that built the corpus, vector store, and knowledge graph via MinerU, and a deployment VM (Standard_B2s, 2 vCPU/4 GB RAM) that served the Gradio interface and ran the retrieval pipeline, including the 800-execution RAGAS benchmark. LLM text generation ran through Ollama Cloud rather than locally, keeping the deployment VM's resource footprint small. Both models were called through Ollama's native chat API with the same generation settings for every query: temperature 0.1 and a 400-token output cap (`num_predict`). Top-p was not set explicitly in either case, so each model used its own default on Ollama Cloud rather than a value we controlled.

Table 2. Technology stack used for system implementation and evaluation.

Category	Technology	Purpose	Rationale
Programming language	Python 3.12	System development	Broad AI/ML ecosystem support
Interface	Gradio 6.x	User interaction	Multimodal UI without a custom frontend
LLM (generation)	gpt-oss:20b-cloud and gemma4:31b-cloud	Answer generation	Native tool calling and strong reasoning
Computer vision	Moondream	Image processing	Lightweight, runs locally via Ollama
Speech recognition	Faster-Whisper	Speech-to-text	Low latency on GPU-free hardware
Speech synthesis	Piper TTS	Text-to-speech	Fully local, no cloud dependency
Embeddings	sentence-transformers (paraphrase-multilingual-MiniLM-L12-v2)	Vector representation of the corpus (text and tables)	Tuned for Spanish semantic similarity
Vector database	ChromaDB	Semantic retrieval	Simple, embedded deployment
Knowledge graph	NetworkX	Representation of regulatory relationships	In-memory and auditable, no server needed
Hybrid retrieval	Hybrid Retriever	RAG + GraphRAG integration	Best balance of faithfulness and relevance
Document processing	MinerU and PyMuPDF	Text and table extraction	MinerU primary, PyMuPDF as fallback
DOCX processing	python-docx	Document reading	Reads the question bank directly
Multi-agent system	Planner Agent, Query Refiner Agent, and Query Validator Agent	Agentic orchestration	Tool calling improves small-model reliability
Evaluation	RAGAS	System performance evaluation	Correlates well with human judgment
Infrastructure	Azure for Students (ingestion VM + deployment VM) + Ollama Cloud	Pipeline execution and experimental evaluation	Separates heavy ingestion from light serving

3.5. System Architecture and Corpus

To support the comparative evaluation and handle the complexity and pace of change in Ecuadorian tax regulation, we designed a hybrid multimodal system built around agentic orchestration and relational knowledge structures. The architecture brings together an iterative refinement cycle, domain-guardrail validation, hybrid (vector and relational) retrieval, and a configurable response pipeline (Figure 1).

The knowledge base used to evaluate the artifact is a curated set of 51 PDF documents covering Ecuadorian tax legislation and guidance published by the Servicio de Rentas Internas (SRI). This unified corpus supersedes the earlier, preliminary development versions and is the single source used for ingestion, knowledge-graph representation, and the empirical evaluation. By document type, it comprises 12 consolidated legal texts (laws, codes, and

regulations), 10 SRI resolutions, 1 SRI circular, 2 SRI bulletins, 16 guides or tables published on the SRI portal, and 10 article excerpts.

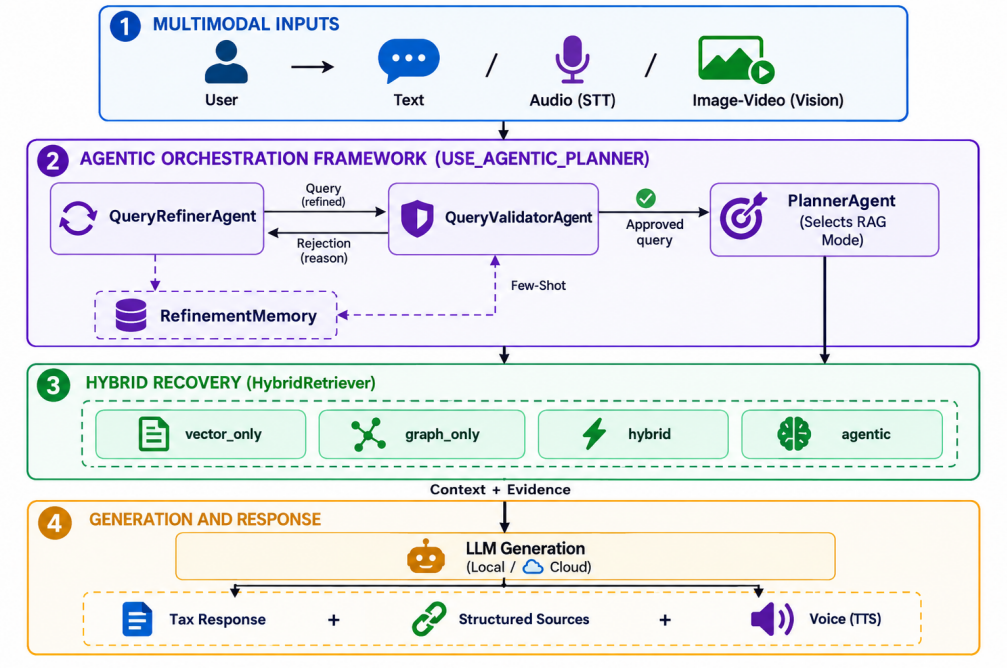


Figure 1. Architecture of the evaluated multimodal system with agentic orchestration.

Supplementary Material S2 provides a manifest of all 51 files, listing each file's thematic folder, file name, document type, identifier or title, number of pages, signature or publication date, last amendment as stated in the document, repository-addition date, and validity note. Where a document does not state a date or version (e.g., portal guides printed to PDF), the manifest reports this rather than inferring it. The evaluated corpus snapshot is dated 13 August 2026, the date of the last commit to the corpus folder; the 51 files in the evaluated version were committed on 7 and 13 August 2026, and benchmark executions ran after this snapshot. This date identifies the corpus snapshot, not a comprehensive legal-status audit.

The source files and metadata were collected from the SRI portal, but the documents do not all provide a complete consolidated version or validity period: for example, the LRTI file reports amendments only through 20 June 2023, its Regulation through 24 November 2023, and the corpus also retains historical material such as the SRI's RISE guide for historical queries (Supplementary Material S2). The system does not model validity periods or succession between norms (all norms share a single node type, as described below); accordingly, it cannot determine temporal validity at retrieval time, and later reforms are represented only where covered by other included documents. These limitations are considered in Section 5.

Query processing runs through three specialized agents that interact sequentially and iteratively (their exact configuration is summarized in Table 3). The Query Refiner Agent (Figure 2) restructures and sharpens the incoming query, whatever its source (text, speech, or visual analysis), so its terminology matches tax-regulation vocabulary. It pulls in conversational context to resolve ambiguous references on the first interaction, and keeps a refinement-memory mechanism that retrieves lessons from earlier corrections via semantic-embedding similarity search, applying in-context (few-shot) learning without retraining the model's weights. This memory is shared across the whole benchmark run rather than reset between executions or between generator models, so a lesson learned while refining one query can resurface while refining a later one; we treat this as a limitation of the experimental design, discussed in Section 5.

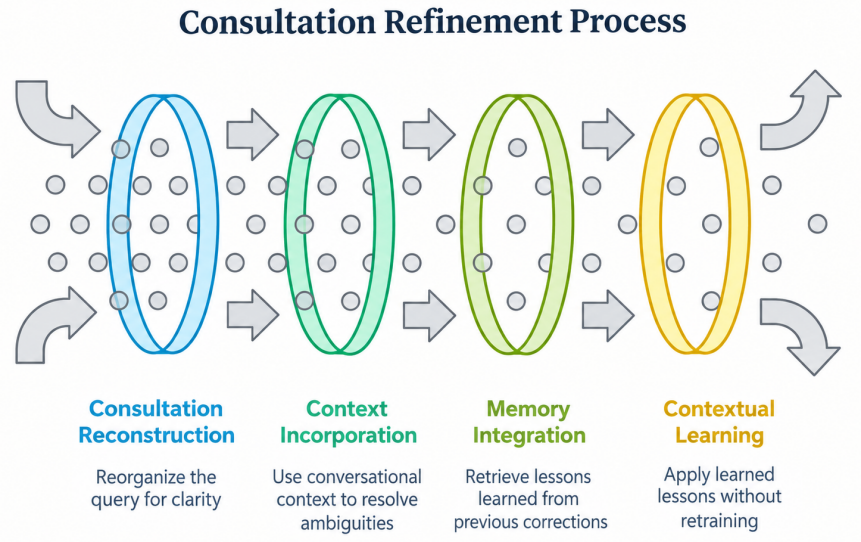


Figure 2. Behavior of the query refinement agent.

Table 3. Exact configuration of the three orchestration agents. Full tool definitions and the loop's stopping logic are in Supplementary Material S1.

Agent/Step	Decision	Tool(s)	Timeout	On Failure
Validator, domain pre-check	in-domain vs. off-topic	`pregunta_fuera_de_dominio`	30 s	defaults to in-domain
Refiner ↔ Validator loop	approve vs. reject-and-retry, up to 2 rounds, tested against a real `vector_only` retrieval each round	`rechazar_pregunta`, `pregunta_fuera_de_dominio`	30 s per call	Refiner keeps the query unchanged; Validator defaults to approve
Planner	vector-only vs. vector + graph	`buscar_relaciones_grafo`	30 s	defaults to vector-only

The Query Validator Agent (Figure 3) checks, through a preliminary test search, whether the refined query has enough information and is actually tax-relevant, under two decision schemes. Rejection and feedback sends an imprecise or ambiguous tax-domain query back to the refiner with a specific reason, inside an iteration-capped cycle. A domain-relevance guardrail catches out-of-domain queries from the start and stops the pipeline immediately, avoiding wasted rewriting.

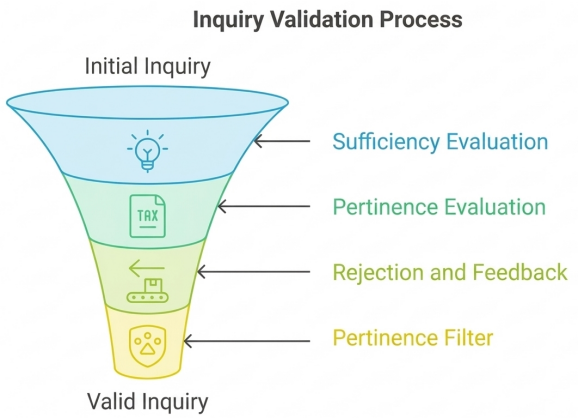


Figure 3. Behavior of the query validation agent.

The Planner Agent (Figure 4) uses LLM tool calling to decide whether a validated query should also be sent to the knowledge graph. Its tool description gives explicit query-level

cues: call the graph tool when a question explicitly asks about a relationship, obligation, or connection between tax concepts, and do not call it for a simple definition, a specific rate, or the text of an article. This is an instruction-guided decision with lexical/task cues, not a separately validated measure or classifier of semantic complexity.

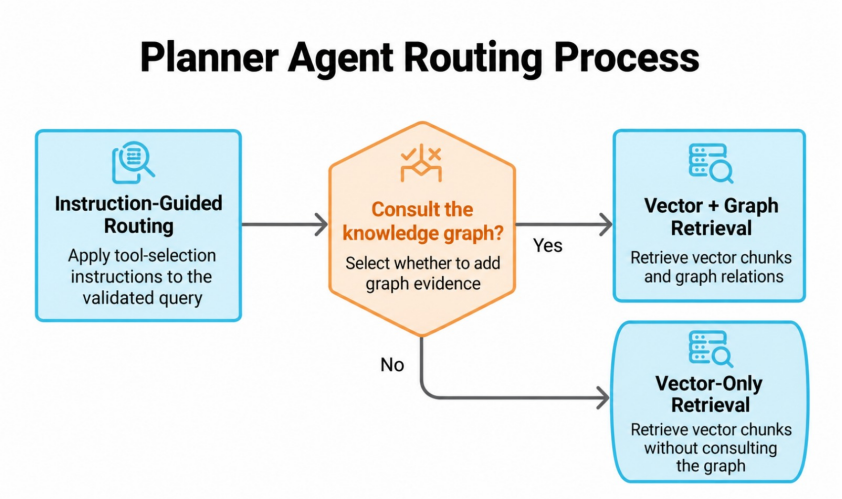


Figure 4. Planner agent routing between vector-only retrieval and combined vector and graph retrieval.

If a query is still not approved after the two refinement rounds, the pipeline does not block or return an error: it proceeds with the last refined query and the chunks retrieved in that last round, the same fail-open principle used throughout the pipeline. The retrieval subsystem is a hybrid module that switches between four acquisition modes for direct comparison: exclusive vector retrieval (dense embedding–similarity search), relational graph retrieval (structured search over entities and their explicit relations), hybrid retrieval (fusing vector and graph evidence in parallel), and adaptive agentic retrieval (dynamic mode selection driven by the planner agent). A configurable generation engine synthesizes the final answer. Speech processing, visual analysis, vector storage, and knowledge-graph construction all run locally, while the generative language model can switch between local and cloud-deployed instances, keeping query processing adaptable. A traceability mechanism, built on a secondary communication channel, supports reproducibility and continuous evaluation: it logs structured events at key pipeline stages in real time and tracks latency, resource use, and the other performance indicators needed for experimental validation.

Both the vector store and the knowledge graph were built from the same document fragments the ingestion pipeline produced (MinerU as the primary extractor, PyMuPDF as a fallback), though each one processes that information its own way. For the vector store, each document was split into text chunks that kept track of the page and source article, so the exact source could always be cited; tables were vectorized from a plain-text version of their content, without HTML tags, since the full HTML is kept for the generator model to display but dilutes the semantic signal if used for the vector itself, and images were folded in the same way as text, using MinerU's generated caption or, failing that, OCR-recognized text.

For the knowledge graph, entities were identified by matching text against a curated list of tax-domain terms (taxes, taxpayers, obligations, forms, regulations, concepts, and similar categories), keeping the longer match whenever two candidates overlapped; we chose this rule-based approach over a trained named-entity-recognition model because Ecuadorian tax vocabulary is bounded and known ahead of time, which gives a deterministic, auditable result. Relations were pulled out sentence by sentence: whenever two or more entities appeared together, the system looked for a verb or phrase signaling the relation type between

them (*must file*, *is exempt*, *a rate applies*, and so on), taking the first entity in the sentence as the relation's origin and the rest as its targets.

Each entity became a graph node and each relation an edge (NetworkX); norms (laws, regulations, resolutions) were all represented as a single node type without an explicit hierarchy among them, and every edge kept the source document and the exact text fragment that produced it, so the origin of any queried relation could be traced back.

The graph that comes out of this process is small by design: 36 entity nodes, linked by 3898 extracted relations. Three out of every four of those relations (75%, 2939 of 3898) end up under the generic `relacionado_con` ("related to") label, which the extractor falls back on whenever it cannot pin down a more specific verb-based pattern; only the remaining 25% carry a sharper relation type such as `establece`, `aplica_tarifa`, or `debe_presentar`. We built the taxonomy around the regimes that dominate our query bank (income tax, VAT, withholding, RIMPE, and electronic invoicing), so the 19 lower-frequency categories in Table 4 (2–3 queries each, among them RIVUT, ZEDE, the banana-sector tax, and inheritance tax) have no entities of their own, and graph-based retrieval has little to offer those queries on its own.

Table 4. Categorization of the 100-query evaluation bank used for the RAGAS benchmark.

Category	N
Income tax	14
Withholding at source	10
VAT	9
RIMPE	8
Other	8
Electronic invoicing	7
Vehicle taxes	4
ICE (special consumption tax)	3
Foreign trade	3
International taxation	3
Capital gains	3
Transfer pricing	2
Rural land tax	2
Redeemable tax on plastic bottles	2
Inheritance, bequests, and donations	2
RUC (taxpayer registry)	2
Tax benefits—disability	2
Tax benefits—tourism	2
Tax on assets held abroad	2
ZEDE (special economic development zones)	2
Social-interest housing	2
RIVUT (regularization of assets held abroad)	2
Tax benefits—older adults	2
Banana sector	2
VAT refund to exporters' suppliers	2
Total	100

Overlaps between aliases are resolved by keeping the longest match: when two candidate entities cover the same span of text, the extractor keeps only the one that captures the more specific term, on the idea that a longer match carries more information than a shorter one nested inside it. A concrete case from the taxonomy itself: the phrase "declaración anual de impuesto a la renta" matches three different entities at once, `declaración`, `impuesto a la renta`, and `declaración de impuesto a la renta` (the longest and most specific of the three), and the extractor keeps only the last one. That choice has a cost: nested entities are dropped rather than kept as secondary matches, so a relation involving `impuesto a la`

`renta` on its own is never recorded for that sentence, even though the sentence is clearly about income tax.

We accepted this trade-off for a simpler, deterministic extraction rule, but it is a real source of missed relations that a scheme allowing nested entities could recover. The relation extractor carries a related risk: within a sentence, it always treats the first detected entity as the relation's source and every other entity in that sentence as a target, all under the same relation type. A real case from the graph shows how this can misfire: the page-footer text "103 de 128" ("page 103 of 128") was matched as the entity `formulario 103` (Ecuador's withholding-tax form, whose alias list includes the bare number "103" with no surrounding context required), and because it happened to be the first entity detected in that sentence, it became the source of four unrelated edges, to `contribuyente`, `SRI`, `obligación tributaria`, and `Código Tributario`, in a sentence that has nothing to do with Form 103 at all.

We have not run a systematic audit of extraction precision and recall, since that would require manually annotating a sample of sentences with the relations a human reader would extract, including the ones the system misses, which was beyond the scope of this evaluation; we flag it here as an open limitation rather than attach an unverified accuracy figure to it.

Chunking and retrieval used a fixed, shared configuration across all four modes. Text chunks were 500 characters long with a 60-character overlap between consecutive chunks. At query time, the vector store returned chunks with a cosine similarity of at least 0.30, re-ranked by a lightweight keyword-overlap score (a boost based on how many query terms appear in the chunk's text, its source document's name, and its regulatory category, not a trained reranker) before keeping the top 12. The knowledge graph, independently, returned up to 10 triples within two hops of the entities detected in the query. These two budgets are independent and additive: `hybrid`, and the graph-activated branch of `agentic`, can forward up to 12 chunks and 10 triples to the generator at the same time, while `vector_only` and `graph_only` draw only on their own budget.

3.6. Data Collection and Evaluation Instruments

Data collection and evaluation relied on four main instruments: a documentary corpus of official Ecuadorian tax regulation pulled from public sources such as the SRI portal; a question bank built from representative consultation scenarios and needs in the tax domain; the RAGAS evaluation framework, which objectively measures retrieval and generation quality through five indicators: Faithfulness, Answer Relevancy, Answer Correctness, Context Precision, and Context Recall; and a structured, Likert-scale questionnaire that captures users' perception of the system's usefulness and reliability.

Evaluating a RAG architecture properly means checking retrieval quality, answer faithfulness, and context relevance all at once, and RAGAS has become one of the most widely used frameworks for exactly that reason. Its specific metrics (Faithfulness, Answer Relevancy, Answer Correctness, Context Precision, and Context Recall; defined in Table 5) give a more representative picture than indicators based purely on textual similarity, and they correlate well with human-expert judgment in specialized domains [28]. We used RAGAS version 0.2.15, pinned deliberately to avoid a `langchain-core` dependency conflict with the rest of the pipeline. Answer Correctness, specifically, combines an LLM-judged factual-overlap score against the question bank's expected answer with the cosine similarity of their embeddings, computed with the same `paraphrase-multilingual-MiniLM-L12-v2` sentence-transformer used elsewhere in the system.

Table 5. Terminology used in the RAGAS evaluation tables.

Column	Meaning
F (Faithfulness)	RAGAS score (0–1): whether the answer is grounded in the retrieved context, or the LLM asserted something unsupported by it.
AR (Answer Relevancy)	RAGAS score (0–1): whether the answer addresses the question asked, without digressing.
AC (Answer Correctness)	RAGAS score (0–1): compares the answer against the question bank's expected answer, combining factual overlap and semantic similarity.
CP (Context Precision)	RAGAS score (0–1): of the retrieved context fragments, the proportion that was actually relevant to answering the question.
CR (Context Recall)	RAGAS score (0–1): of everything the expected answer required, the proportion that was actually found in the retrieved context.
%Doc	Share of vector-retrieval queries in which the true source document was retrieved; not applicable to `graph_only`, which never queries the vector store.
Time	Average total response time per query.
Graph%	Share of queries for which the planner agent activated GraphRAG; only meaningful for the agentic mode.
Tokens	Average tokens consumed generating the answer (excludes the separate RAGAS judge's tokens).

The 800 experimental executions were judged automatically by a natural-language model accessed via the Anthropic API (`langchain-anthropic`'s `ChatAnthropic` client, model identifier `claude-haiku-4-5`), run between August 24 and 27 August 2026, at a total cost of roughly USD 37 (about USD 0.045 per evaluated row). This identifier is the model alias we requested and logged; the API call did not return a dated snapshot pinning it to a specific underlying checkpoint, and we did not record one separately, so we report the alias actually used rather than guess at a more precise version.

For the RAGAS evaluation we used 100 queries about Ecuadorian tax regulation, designed to represent different retrieval needs across twenty-five thematic areas (RIMPE, electronic invoicing, withholding at source, VAT, income tax, and others), and covering factual queries, queries conditioned on taxpayer characteristics, queries that require relating multiple regulatory elements, and queries tied to regulatory amendments or validity periods (Table 4). Although the deployed system accepts text, speech, and image inputs, all 100 benchmark questions were submitted as text; speech- and image-input pathways were not evaluated in this experiment.

Each query came with an expected answer, a source regulatory document, and the corresponding provision, which together served as the reference against which generated answers were judged. Building this question bank (Appendix A) followed a fixed process, run between August 4 and 13 August 2026, across five sequential batches of roughly 16 questions each (the first 20 questions carried over unchanged from an earlier 20-question pilot bank, without re-verifying their validity status in this expansion).

An LLM-based coding-assistant subagent, prompted to role-play an Ecuadorian certified public accountant, drafted each candidate question the way a practicing accountant would phrase a real day-to-day question, navigated live to the SRI's official portal to find the relevant source document, and quoted the verbatim, currently-in-force text of the cited article or resolution as the expected answer, only after actively checking whether that provision had since been amended or repealed. An external certified public accountant, not one of the authors and not one of the five specialists who later validated the system in Section 4.4, reviewed every candidate question before it was added to the bank; none of it was integrated automatically. The full set was run against all four retrieval strategies

(vector_only, graph_only, hybrid, and agentic) and both generator models, for a total of 800 experimental executions.

Data analysis combined quantitative and qualitative approaches. RAGAS results went through both descriptive statistics (central tendency and dispersion) and inferential comparisons: since each of the 100 queries was run under all four retrieval strategies and both generator models, strategies were compared pairwise with the Wilcoxon signed-rank test on matched (question, model) pairs rather than treating the 800 executions as independent observations, with 95% confidence intervals from 10,000 bootstrap resamples of the paired differences and Holm–Bonferroni correction within each metric's family of six pairwise comparisons. The single `vector_only/gpt-oss:20b-cloud` execution that hit the generation timeout (Section 4.3) was excluded from these paired tests, since its RAGAS scores are an imputed average, not a measurement.

For an exploratory post hoc analysis of planner activation, the author Jorge Mosquera manually classified the 100 unique questions after the benchmark runs. Questions requiring relationships across two or more distinct tax regimes were classified as complex; the remaining questions were classified as simple, yielding 9 complex and 91 simple questions. This operational definition was used for the routing analysis and should not be interpreted as a comprehensive measure of question difficulty. No independent second annotation was performed, and inter-annotator agreement was therefore not assessed. Each question was evaluated with two generator models, yielding 18 executions for complex questions and 182 for simple questions.

These executions represent repeated evaluations of the same questions, rather than additional independent questions. Given the small number of complex questions and the class imbalance, the class-stratified activation rates are reported descriptively and do not establish the planner's ability to assess semantic complexity. The specialists' qualitative observations were organized through thematic analysis to pull out strengths, limitations, and opportunities for improvement in tax-domain knowledge retrieval.

In line with journal policy on disclosing generative-AI (GenAI) use, we note that a general-purpose LLM-based coding assistant helped build the RAG/GraphRAG system, its agents, and its interface, and helped organize the official tax regulation used to construct the 100-question evaluation bank and analyze the experimental benchmark results; the system's own experimental design also uses an LLM (`claude-haiku-4-5`) as an automated judge of generated answers. A role-playing coding-assistant subagent, prompted to act as an expert Ecuadorian accountant, also helped draft candidate RAGAS question–answer pairs and locate up-to-date SRI source documents for corpus curation. All AI-assisted output was reviewed and edited by the authors, who take full responsibility for the content of this publication.

4. Results

4.1. RAGAS Evaluation by Retrieval Mode

To keep the tables readable, Table 6 uses the short column headers defined in Table 5.

Table 6 and Figure 5 summarize the RAGAS results by retrieval mode. In the evaluated implementation, `graph_only` obtained the lowest mean values across all five metrics, including context precision of 0.02 and context recall of 0.04. These results indicate limited retrieval of relevant evidence under this configuration. However, the benchmark does not isolate whether this outcome resulted primarily from graph coverage, entity and relation extraction, the retrieval procedure, or a combination of these factors.

On the remaining four metrics, hybrid and vector_only post similar means, and formal pairwise testing (Table 7) confirms they are not statistically distinguishable on Faithfulness, Context Precision, Context Recall, or Answer Correctness; their one real, if small, difference

is Answer Relevancy, where hybrid edges ahead. Against agentic, the picture splits by metric: Faithfulness and Context Precision show no significant difference from either hybrid or vector_only, while Answer Relevancy, Context Recall, and Answer Correctness are where agentic clearly falls behind both, consistent with the planner's graph activations not translating into measurably better grounded or more precise context in this sample.

Table 6. RAGAS results by retrieval mode (mean over 200 executions per mode: 100 queries × 2 generator models). Column headers are defined in Table 5.

Mode	F	AR	AC	CP	CR	%Doc	Time	Graph%
graph_only	0.51	0.31	0.27	0.02	0.04	—	2.78 s	—
hybrid	0.67	0.64	0.56	0.52	0.76	93%	2.66 s	—
vector_only	0.66	0.62	0.55	0.52	0.77	93%	8.60 s	—
agentic	0.66	0.45	0.46	0.50	0.69	89.5%	36.15 s	48%

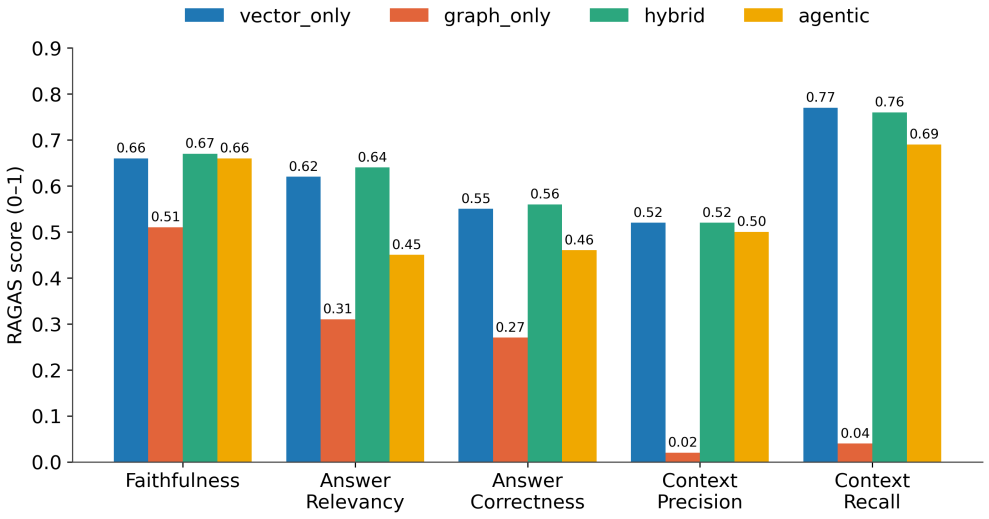


Figure 5. Comparison of RAGAS metrics by retrieval mode.

Table 7. Holm-corrected Wilcoxon signed-rank tests, paired by (question, model), excluding the imputed timeout execution. Comparisons against `graph_only` are omitted here since all are significant at $p < 0.0001$ (see Table 6 for its much lower raw means).

Metric	Hybrid vs. Vector_Only	Hybrid vs. Agentic	Vector_Only vs. Agentic
Faithfulness	$p = 0.99$ (n.s.)	$p = 0.93$ (n.s.)	$p = 0.88$ (n.s.)
Context Precision	$p = 0.46$ (n.s.)	$p = 0.95$ (n.s.)	$p = 1.00$ (n.s.)
Answer Relevancy	$p = 0.017$ * (+0.021)	$p < 0.001$ * (+0.193)	$p < 0.001$ * (+0.174)
Context Recall	$p = 0.37$ (n.s.)	$p = 0.0015$ * (+0.076)	$p < 0.001$ * (+0.083)
Answer Correctness	$p = 0.17$ (n.s.)	$p < 0.001$ * (+0.103)	$p < 0.001$ * (+0.091)

* $p < 0.05$ after Holm–Bonferroni correction; n.s. = not significant. Parenthetical values show the mean difference (first mode minus second) in the metric's own 0–1 scale, reported only for significant comparisons.

As for response time (Figure 6), hybrid (2.66 s) and graph_only (2.78 s) come out fastest, since both run a single retrieval pass with no extra reasoning steps; vector_only (8.60 s) is slower because of the size of the context it forwards to the LLM, and agentic (36.15 s) is by far the most expensive, mostly because of the iterative refinement cycle (20.18 s) and planning (12.93 s) that happen before retrieval even starts. Given that agentic offers no quality edge over hybrid or vector_only either, that extra cost is hard to justify on its own in this scenario.

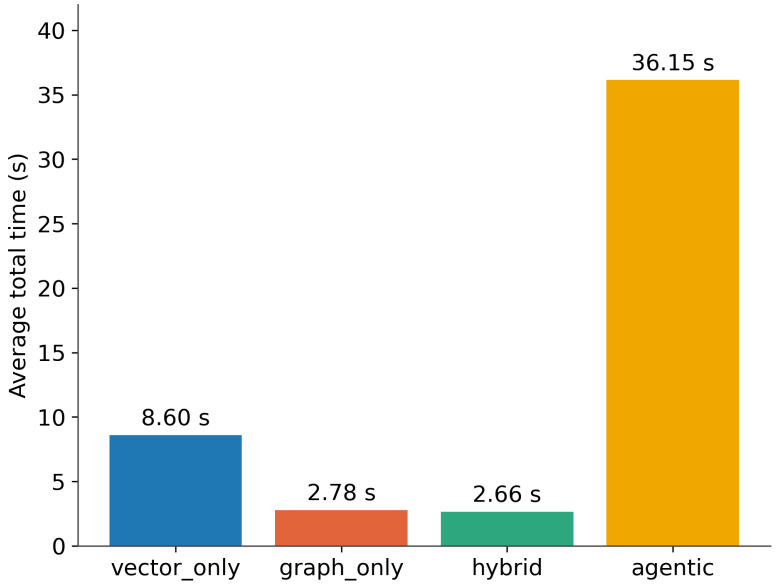


Figure 6. Average total response time by retrieval mode.

Response time is right-skewed in every mode, since a handful of slow queries pull the mean up, so Table 8 reports the median, interquartile range, and 95th percentile alongside it. The typical (median) query runs noticeably faster than the mean suggests for vector_only (5.46 s vs. 8.60 s) and graph_only (1.83 s vs. 2.78 s); agentic, by contrast, stays slow throughout the distribution (median 34.60 s, P95 65.94 s), so its mean is a fair summary. vector_only's long tail (P95 18.39 s, max 180.77 s) is partly driven by the single query that hit the 180-second generation timeout discussed in Section 4.3.

Table 8. Response-time dispersion by retrieval mode, over the 200 executions per mode (total response time, in seconds).

Mode	Median	IQR (Q1–Q3)	P95	Max	Mean
graph_only	1.83 s	0.68–4.13 s	8.12 s	17.44 s	2.78 s
hybrid	2.57 s	1.65–3.49 s	4.48 s	6.93 s	2.66 s
vector_only	5.46 s	1.51–13.94 s	18.39 s	180.77 s *	8.60 s
agentic	34.60 s	21.61–46.79 s	65.94 s	118.17 s	36.15 s

* This maximum is the single query that hit the 180-second generation timeout described in Section 4.3; its latency is a real measurement, and only its five RAGAS scores were backfilled.

4.2. RAGAS Evaluation by Generator Model

Table 9 reports quality, response time, and token consumption separately because these dimensions represent different operational priorities. gemma4:31b-cloud achieved higher Faithfulness and lower mean output-token consumption, whereas gpt-oss:20b-cloud achieved higher Answer Relevancy and lower mean response time. These results do not establish a single overall model ranking.

Table 9. RAGAS results by generator model (mean over 400 executions per model: 100 queries × 4 retrieval modes). Column headers are defined in Table 5.

Model	F	AR	Time	Tokens
gemma4:31b-cloud	0.78	0.37	13.43 s	2369
gpt-oss:20b-cloud	0.47	0.64	11.67 s	2674

Both models run on the same free Ollama Cloud infrastructure, so the difference comes down to the models themselves rather than the hardware. Under our ranking heuristic (50% quality, 30% speed, 20% token cost), gemma4:31b-cloud scores 0.70 against gpt-oss:20b-cloud's 0.30; its higher quality and lower cost (an average of 2369 tokens versus 2674) make up for a slightly slower response time (13.43 s vs. 11.67 s).

Figure 7 combines both aggregations into a single heatmap, confirming that retrieval mode drives context quality (hybrid and vector-only are the most robust, graph-only the weakest) while the generator model governs the faithfulness-relevance trade-off. The generator models exhibited different trade-offs between faithfulness and answer relevance. gemma4:31b-cloud achieved higher faithfulness but lower answer relevance, suggesting that stronger alignment with the retrieved context did not necessarily translate into a more direct response to the question. Conversely, gpt-oss:20b-cloud obtained higher answer relevancy but lower faithfulness. These metrics therefore capture distinct aspects of answer quality and should be interpreted separately.

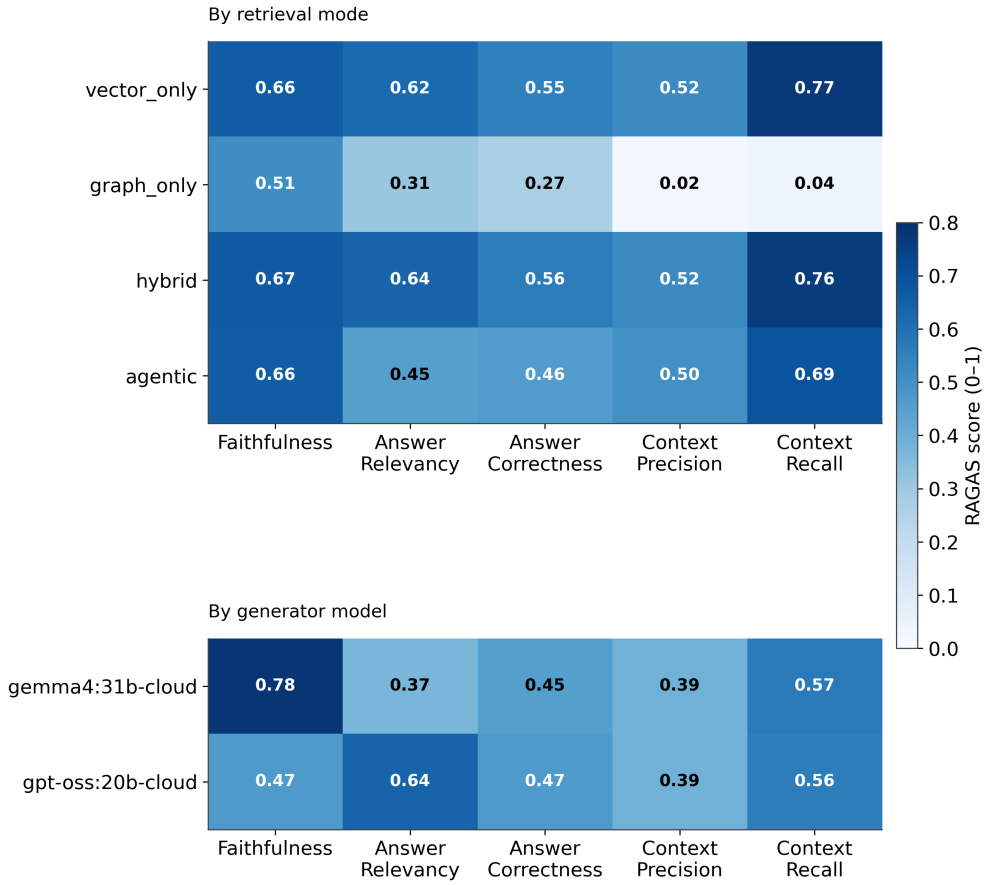


Figure 7. RAGAS heatmap by retrieval mode and by generator model.

4.3. Error Analysis

Looking across the 800 executions, three recurring failure types stood out (Table 10). These three categories do not overlap with each other: each comes from a disjoint subset of modes, and the counts below are executions, not distinct questions, since the same question can fail this way under more than one model or retrieval mode.

Table 10. Error analysis by failure type, computed over the 800 benchmark executions.

Failure Type	Executions	Denominator	Example	Possible Explanation
No relevant context retrieved (Context Precision = 0)	196 (98 of 100 questions, either model)	200 `graph_only` executions (100 queries × 2 models)	A query about VAT refund limits for older taxpayers returned no graph triples on that topic.	The graph's 36-entity taxonomy, built around the five most-represented regimes, does not cover most of the 25 topics in the corpus (Section 3.5).
Source document matched, but Faithfulness = 0	21 (17 distinct questions)	551 executions with a matched source document, across `vector_only`, `hybrid`, and `agentic`	For a question about agricultural land declared an ecological reserve, the model named a specific resolution that RAGAS could not verify against the retrieved fragment.	Either an unverifiable or slightly off citation, or an abstention answer ("the SRI does not establish…") that RAGAS cannot confirm as grounded either way; concentrated in `gpt-oss:20b-cloud` (19 of 21 cases).
Generation timeout	1	200 `vector_only` executions	One `gpt-oss:20b-cloud` query exceeded the 180-second generation limit.	An infrastructure limit, not a retrieval or generation failure; its five RAGAS scores were backfilled with the mean of the other 199 `vector_only` executions.

4.4. Specialist Validation

To complement the automated RAGAS evaluation, five specialists in taxation, accounting, and software architecture tested the deployed system with their own queries and rated it on a five-point Likert scale (1 = very poor to 5 = excellent) across five indicators: accuracy, relevance, correct identification of the applicable regulation, completeness, and clarity. All five had at least five years of professional experience, ranging from five to thirteen years (Table 11). It is important to emphasize that this specialist review was designed as a complementary qualitative layer rather than a direct calibration of the RAGAS judge, while RAGAS evaluated pipeline consistency over a fixed 100-query benchmark, domain experts assessed practical usability, real-world applicability, and nuance on ad hoc queries without a shared question bank or fixed retrieval mode. Individual and mean scores appear in Table 11 and Figure 8.

Table 11. Specialist validation results (five-point Likert scale).

Specialist	Accuracy	Relevance	Applicable Regulation	Completeness	Clarity
Specialist 1 (5 yrs)	5	3	3	5	4
Specialist 2 (13 yrs)	2	3	2	2	2
Specialist 3 (10 yrs)	3	2	2	3	3
Specialist 4 (10 yrs)	4	3	4	3	4
Specialist 5 (7 yrs)	3	4	3	3	4
Mean	3.40	3.00	2.80	3.20	3.40

All five specialists rated the same five indicators on queries of their own choosing; no indicator was assigned by profile. Judgments that lean most heavily on tax-law knowledge (accuracy, applicable regulation, and completeness) should be read with that in mind for Specialist 5, the software architect, whose main contribution was the technical finding discussed below rather than a tax-domain judgment. For transparency, Table 11's means change only slightly if Specialist 5 is excluded: 3.50 (Accuracy), 2.75 (Relevance),

2.75 (Applicable Regulation), 3.25 (Completeness), and 3.25 (Clarity), for an overall mean of 3.10 versus 3.16 with all five included. None of this study's conclusions depend on which of the two means is used.

Table 12 reports the frequency of each Likert response by indicator, providing the response distribution underlying the means in Table 11.

Table 12. Frequency distribution of specialist Likert ratings by indicator (five responses per indicator).

Indicator	1	2	3	4	5
Accuracy	0	1	2	1	1
Relevance	0	1	3	1	0
Applicable regulation	0	2	2	1	0
Completeness	0	1	3	0	1
Clarity	0	1	1	3	0
All indicator ratings	0	6	11	6	2

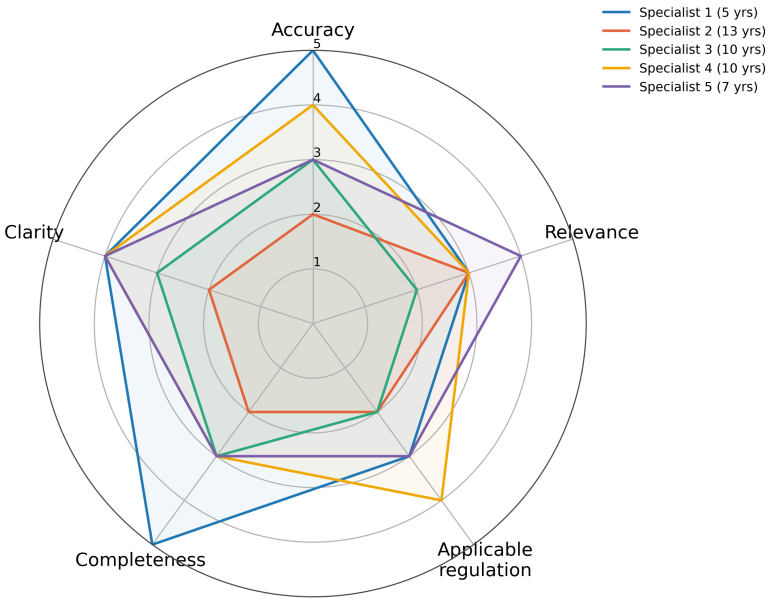


Figure 8. Comparison across specialists on the five evaluation indicators.

Specialist 1 gave the highest per-specialist mean (4.00), while Specialists 2 and 3 gave the lowest (2.20 and 2.60, respectively). Across all 25 specialist ratings, a score of 3 occurred 11 times, scores of 2 and 4 occurred six times each, a score of 5 occurred twice, and no rating of 1 was recorded (Table 12). The overall mean was 3.16 out of 5. These descriptive results summarize the five specialists' responses and are not assigned qualitative categories.

The applicable-regulation indicator scored lowest on average (2.80), which fits with the specific errors the reviewers documented: Specialist 2 (mean 2.20) found that the system missed the current 15% VAT rate and left out Article 97.4 of the LRTI (Ley de Régimen Tributario Interno); Specialist 3 (mean 2.60) flagged inaccurate information on dematerializing credit notes and an incorrect VAT-refund limit for older adults, both consistent with the low graph_only and agentic RAGAS scores on Context Precision and Context Recall reported above. Specialist 5, a software architect with no tax background, rated the system about the same as the others overall (mean 3.40) but offered the sharpest technical critique: working independently from the benchmark published inside the system, this specialist noticed that the trace shown in the interface displays the refined query cut off at 150 characters, something that happens in 97.8% of queries, and flagged it as a possible cause of incomplete answers.

We traced this back in the code: the 150-character cut only happens in the log strings used to build that visible trace (`agents/query_refiner_agent.py`, `agents/coordinator.py`, `agents/rag_agent.py`); the full, untruncated query is what actually reaches the retriever and the generator model, and it is also what `scripts/run_benchmark.py` uses for all 800 experimental executions. So this particular limit does not explain the RAGAS results reported here, even though Specialist 5's underlying observation still stands as useful interface feedback: the trace should make that distinction clearer, so it does not read as if the system were reasoning over a truncated question.

5. Discussion

The low retrieval scores of `graph_only` should be interpreted as evidence about the specific graph construction and retrieval pipeline evaluated in this study, rather than as a general limitation of GraphRAG. A curated entity vocabulary and sentence-level relation extraction may have limited the evidence represented in the graph or introduced inaccurate relations. These are plausible explanations that require further validation. The present experiment did not separate the effects of graph construction quality from those of graph-based retrieval.

Hybrid obtained the highest mean Answer Relevancy (0.64), followed by `vector_only` (0.62), agentic (0.45), and `graph_only` (0.31). These descriptive results should be considered alongside the paired comparisons reported in Table 7 and the methodological limitations discussed below.

The planner's graph-use decision is made by LLM tool calling under explicit instructions to call the graph tool when a query asks for a relationship, obligation, or connection between tax concepts, and not to call it for a simple definition, a specific rate, or the text of an article (Section 3.5). It should therefore not be interpreted as a validated semantic-complexity assessment. In an exploratory post hoc classification of the 100 unique bank questions (91 simple and 9 complex; operational definitions and procedure in Section 3.6), graph activation occurred in 11 of 18 complex question/model executions (61.1%) and 85 of 182 simple question/model executions (46.7%), or 96 of 200 executions overall (48.0%). These execution-level rates are descriptive: the complex subgroup comprises only nine unique questions, no independent second-annotator validation or inter-annotator agreement statistic is reported, and the analysis does not provide ground-truth labels for whether graph retrieval was necessary. The observed percentages therefore do not establish a reliable association between query complexity and planner behavior, nor do they measure routing accuracy.

Separately, on the full benchmark, agentic obtained lower mean Context Precision (0.50) and Context Recall (0.69) than both hybrid (0.52 and 0.76) and `vector_only` (0.52 and 0.77). It also had the highest mean latency (36.15 s, compared with 8.60 s for `vector_only`, 2.78 s for `graph_only`, and 2.66 s for hybrid), with the refinement cycle (20.18 s) and planning (12.93 s) accounting for most of its execution time. These benchmark results describe the evaluated configuration; the exploratory class-stratified analysis does not establish why those performance differences occurred.

Papageorgiou et al. [14] reported improvements in factual grounding using a framework that combined embedding-based retrieval, GraphRAG, and agentic mechanisms. The present study did not observe a corresponding improvement from the evaluated agentic configuration. However, these findings are not directly comparable because the studies differ in their corpora, graph construction, retrieval configurations, generator models, and evaluation procedures. In particular, claim-based factual-grounding measures and RAGAS scores should not be treated as interchangeable. The observed differences therefore do not

establish a contradiction between retrieval paradigms or identify a single explanation for their performance.

For the evaluated configuration, hybrid appears to be a reasonable practical default when balancing answer quality and system latency. It had the lowest observed mean latency among the four retrieval modes (2.66 s), and the paired comparisons reported in Table 7 detected an advantage in Answer Relevancy over `vector_only` and agentic. For Faithfulness and Context Precision, the comparisons reported in Table 7 did not detect statistically significant differences among hybrid, `vector_only`, and agentic. However, the absence of a statistically significant difference does not establish equivalence or non-inferiority.

`vector_only` remains a reasonable alternative when architectural simplicity is a priority. Its mean Faithfulness (0.66) and Context Recall (0.77) were close to those of hybrid, with a mean latency of 8.60 s. By contrast, the low Context Precision (0.02) and Context Recall (0.04) of `graph_only` do not support recommending the evaluated graph configuration as a standalone retrieval method for this corpus. Whether a more carefully constructed graph could provide useful relational evidence or improve traceability requires separate evaluation. The agentic configuration likewise did not demonstrate an advantage in mean Context Precision or Context Recall over hybrid or `vector_only`, while incurring substantially higher latency. Its use would therefore require a task-specific justification for dynamic tool selection, whose benefits were not established by the present benchmark.

The reported latencies measure system execution time, not end-to-end professional task completion. An automated assistant may reduce the effort required to locate candidate evidence and draft an initial response, but the present study did not measure manual consultation time or the time specialists needed to verify generated answers. Consequently, the results do not establish that AI-assisted consultation is faster than manual consultation or quantify any productivity gain. Such gains would depend on retrieval quality, answer accuracy, and the verification effort required, and should be assessed through a dedicated user-in-the-loop study.

Several limitations restrict the interpretation and generalizability of these findings. The experimental corpus contains 51 PDF documents of Ecuadorian SRI regulation, so its size and coverage may limit how well the results extend to a broader and more dynamic regulatory setting. The corpus is a snapshot fixed at 13 August 2026 (Supplementary Material S2), and the system does not represent validity periods or the succession between norms. Several consolidated texts in the corpus predate later reforms: the LRTI file reports its last amendment on 20 June 2023, and its Regulation on 24 November 2023. Later reforms are therefore available to the system only where another document in the corpus covers them. This limitation may be relevant to the applicable-regulation errors reported by Specialists 2 and 3 (Section 4.4) and to applicable regulation being the lowest-rated specialist indicator (2.80), although the evaluation does not isolate it as their cause.

The evaluation covered 100 queries across four retrieval strategies and two generator models, for 800 experimental executions. Applying the configurations to the same question bank supports within-benchmark comparisons, but these executions do not constitute 800 independent questions. A larger and more varied question bank would provide stronger evidence about performance across different retrieval needs. The small number of complex questions also limits the interpretation of the exploratory planner analysis. Furthermore, testing only two generator models limits conclusions about how consistently the observed retrieval patterns would hold across other models.

The automated RAGAS judge (`claude-haiku-4-5`) was not formally validated against human ratings on identical question–response pairs. Domain specialists instead evaluated the system independently using Likert scales and their own queries. RAGAS scores and specialist assessments must therefore be interpreted as complementary perspectives, rather

than directly comparable measurements or evidence of judge–human agreement. Future work should include an alignment protocol in which specialists evaluate a subset of the benchmark responses using criteria mapped to RAGAS dimensions, allowing the assessment of inter-annotator agreement and automated-judge calibration.

The relation-extraction step was not independently audited for precision or recall. As the example in Section 3.5 illustrates, its lexical, first-entity-as-source rule can produce spurious relations, but the frequency of such errors across the full graph was not measured. An independent extraction audit is needed to assess graph quality and investigate its contribution to downstream retrieval performance.

The agentic condition also has a cross-execution dependency: the query refiner's memory was shared across the benchmark and was not reset between executions or between generator models (Section 3.5). A correction stored during one execution could therefore influence the refinement of a later query, including across model runs. The condition was not repeated with isolated memory, so the magnitude and direction of this effect remain unknown. This limits the interpretation of the agentic results as outcomes of independent trials and warrants caution when interpreting inferential comparisons involving that condition. A repeat evaluation with memory reset for each execution would help separate the effect of the agentic pipeline from accumulated execution history.

Another limitation is the absence of a purely lexical retrieval baseline, such as BM25. The low scores of `graph_only` do not establish how lexical retrieval would perform, because relational graph traversal and term-based document retrieval use different mechanisms. Future comparisons should include BM25 and a combined BM25–vector configuration to assess whether the proposed architecture offers benefits over conventional lexical and hybrid information retrieval approaches.

The design principles of this architecture may be adaptable beyond Ecuador's tax domain. Public procurement, mining, and banking regulation across Latin America also involve resolutions, laws, and technical annexes [16]. However, transferability has not been empirically established. Evaluation in a new domain would require domain-specific entity and relation definitions, recalibration of the QueryValidatorAgent's domain boundaries, and testing on larger document collections with temporal amendments and cross-references. Comparisons across additional generator models and against lexical baselines would also be needed to determine whether the observed findings extend to those settings.

The specialists' overall score of 3.16 out of 5 reflects their Likert-scale assessment on ad hoc queries and is distinct from the automated RAGAS results in Section 4. The benchmark provides a reason to evaluate hybrid as the default retrieval mode in place of the current `auto` fallback, given its observed balance of RAGAS scores and latency. However, the study did not test whether this change would improve the specialists' ratings, and an increase in the human evaluation score cannot be inferred from the retrieval-mode comparison alone.

A separate proposed improvement concerns temporal validity. Applicable regulation was the specialists' lowest-rated indicator (2.80), with reported problems including Specialist 2's missed 15% VAT rate and omitted Article 97.4 of the LRTI, and Specialist 3's incorrect VAT-refund limit for older taxpayers. Missing validity-period metadata and incomplete coverage of later reforms are plausible contributors to these errors, but the study did not establish their causes or rule out retrieval and generation failures. Adding effective dates, amendment links, and supersession relationships to the knowledge representation, together with retrieval logic that uses this information, is therefore a concrete direction for future evaluation. Its effect on applicable-regulation accuracy and specialist ratings remains to be measured.

Future work will expand the corpus to cover additional validity periods, reforms, repealed provisions, cross-references, and complementary documents. This expansion should

be accompanied by an updated and independently reviewed question bank, a systematic graph-quality audit, and experiments with isolated agent memory. Testing additional local and cloud-based generator models would further clarify how retrieval performance depends on the generator and how consistently the findings hold across configurations.

6. Conclusions

This study evaluated four RAG paradigms over a fixed snapshot of Ecuadorian tax legislation. Rather than making a broad claim of technical viability, our empirical findings demonstrate specific performance trade-offs that directly inform architectural choices in regulatory knowledge retrieval. First, relying exclusively on knowledge-graph relations (`graph_only`) proved insufficient, yielding the lowest RAGAS metrics across the board due to severe context omission (Context Precision 0.02, Context Recall 0.04). Second, hybrid and `vector_only` obtained similar mean faithfulness scores (0.67 and 0.66, respectively), with no statistically significant difference detected by the reported paired test. This does not establish statistical equivalence. Hybrid nevertheless had a lower observed mean execution latency (2.66 s vs. 8.60 s), supporting its consideration as a practical default for the evaluated configuration. Third, the evaluated agentic configuration activated the graph in 96 of 200 query/model executions (48.0%) and introduced substantial latency (36.15 s) without improving context precision (0.50) or recall (0.69) compared to non-agentic hybrid retrieval. Consequently, agentic orchestration requires more refined routing criteria and memory management before its operational complexity is justified in real-world tax consultation systems.

A complementary validation with five specialists in taxation, accounting, and software architecture yielded an overall mean rating of 3.16 out of 5, with applicable regulation as the lowest-rated indicator; the full response distribution is reported in Table 12. The software-architecture specialist also noted that the agentic mode did not deliver a quality benefit that offsets its higher computational cost.

The findings point to the choice of generator model as a decisive factor in overall system performance. The differences we saw between the two models tell us that picking an LLM should consider not just generation quality but also infrastructure constraints, latency, and available resources; the architectures examined here would benefit from choosing the generator model based on context, especially where efficient computational-resource use and local deployment matter most.

The study's two main limitations are the limited number of regulatory documents in the experimental corpus and the size of the query sample used for evaluation, so the results are best read as evidence for this specific case rather than a generalization across the tax-regulatory domain as a whole. The expert validation was also limited to each specialist's own queries, so extending it to the full 100-question bank and to more specialists remains future work. This specialist review should also be read as a small, independent, non-standardized assessment rather than a formal legal-correctness audit: each specialist tested different queries of their own choosing, so no inter-rater agreement can be computed from these scores.

In summary, this work offers empirical evidence on how retrieval quality, answer relevance, and computational cost relate to each other in RAG systems used for regulatory domains. Designing such systems, the findings suggest, should not chase a single metric but instead aim for a flexible design that lets the generator model and retrieval strategy be picked based on query complexity, the faithfulness level required, and resource constraints. That perspective sets up future work aimed at improving agentic orchestration and validating the system on broader, more complex regulatory corpora.

https://doi.org/10.3390/app162110570

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/app162110570/s1, Supplementary Material S1: exact, verbatim configuration (prompts, tools, thresholds, and stopping logic) of the three orchestration agents (Query Refiner, Query Validator, Planner); Supplementary Material S2: manifest of the 51 corpus documents (thematic folder, file name, document type, identifier or title, number of pages, signature or publication date, last amendment, date added to the repository, and validity note).

Author Contributions: Conceptualization, D.V.C.F., J.A.M.S., C.I.D.H. and P.M.R.V.; methodology, D.V.C.F., J.A.M.S., C.I.D.H. and P.F.B.-E.; software, D.V.C.F., J.A.M.S. and C.I.D.H.; validation, D.V.C.F., J.A.M.S., C.I.D.H., M.I.-P., Y.Q. and P.F.B.-E.; formal analysis, D.V.C.F.; investigation, D.V.C.F., J.A.M.S. and C.I.D.H.; resources, D.V.C.F., J.A.M.S. and C.I.D.H.; data curation, J.A.M.S. and C.I.D.H.; writing—original draft preparation, D.V.C.F., J.A.M.S. and C.I.D.H.; writing—review and editing, D.V.C.F., J.A.M.S., C.I.D.H., P.M.R.V. and P.F.B.-E.; visualization, J.A.M.S.; supervision, P.M.R.V. and P.F.B.-E.; project administration, D.V.C.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable. This study did not involve any experimental intervention on human or animal subjects. The participation of the five domain specialists consisted of a voluntary, anonymous professional assessment of system outputs using a Likert-scale questionnaire, and did not involve the collection of personal or sensitive data.

Informed Consent Statement: Informed consent was obtained from all specialists who participated in the qualitative validation of the system.

Data Availability Statement: The source code, technical documentation, and the 100-question evaluation bank (with expected answers and cited source provisions) supporting the results of this study are openly available in the GitHub repository at https://github.com/jorge1150/SRI_IA_Multimodal (accessed on 6 September 2026). The underlying Ecuadorian tax regulations analyzed are public documents issued by Ecuador's Servicio de Rentas Internas (SRI).

Acknowledgments: The authors thank the five domain specialists in taxation, accounting, and software architecture who voluntarily evaluated the system, and Universidad Tecnológica Israel (UISRAEL) for its academic support. During the preparation of this manuscript, the authors used a general-purpose LLM-based coding assistant (Claude Opus 5, Anthropic, San Francisco, CA, USA) to help build the RAG/GraphRAG system, its agents, and its interface; to help organize the official tax regulation used to construct the 100-question evaluation bank; to support analysis of the experimental benchmark results; and, in a role-playing configuration prompted to act as an expert Ecuadorian accountant, to help draft candidate RAGAS question–answer pairs and locate up-to-date SRI source documents. An LLM (`claude-haiku-4-5`) was also used as the automated RAGAS judge within the system's own experimental design. The authors reviewed and edited all AI-assisted output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RAG	Retrieval-Augmented Generation
MRAG	Multimodal Retrieval-Augmented Generation
KG	Knowledge Graph
LLM	Large Language Model
SRI	Servicio de Rentas Internas (Ecuador's tax authority)
VAT	Value-Added Tax
RIMPE	Régimen Simplificado para Emprendedores y Negocios Populares
LRTI	Ley de Régimen Tributario Interno
STT	Speech-to-Text
TTS	Text-to-Speech

OCR	Optical Character Recognition
DSR	Design Science Research
SALSA	Search, Appraisal, Synthesis and Analysis
RAGAS	Retrieval-Augmented Generation Assessment

Appendix A. Sample of the Expert-Validated Tax Question Bank

Table A1 reproduces a representative sample of the 100-question evaluation bank built jointly with a tax-regulation specialist to identify the model's shortcomings when answering; the full bank, together with the expected answer and cited source provision for every query, is available in the GitHub repository referenced in the Data Availability Statement.

Table A1. Representative sample of the expert-validated Ecuadorian tax-regulation evaluation bank.

ID	Category	Query (Original Spanish)	Source Provision
q01	RIMPE	Tengo un cliente que es negocio popular dentro del RIMPE y me pregunta si puede seguir emitiendo notas de venta preimpresas o si ya está obligado a facturar electrónicamente. ¿Qué le respondo?	Art. 6, Resolución Nro. NAC-DGERCGC24-00000027
q05	Electronic invoicing	Emito facturas electrónicas y hasta hace poco sabía que tenía hasta 4 días hábiles para transmitirlas al SRI. Un colega me dijo que eso cambió. ¿Cuál es el plazo vigente para transmitir mis comprobantes electrónicos?	Disposición Reformatoria Primera, num. 2, Resolución NAC-DGERCGC25-00000014
q09	Withholding at source	Voy a pagar honorarios a un abogado independiente (persona natural) por asesoría legal. ¿Qué porcentaje de retención en la fuente de Impuesto a la Renta debo aplicarle?	Art. 2, num. 7, lit. a), Resolución NAC-DGERCGC26-00000009
q13	VAT	Mi negocio se dedica exclusivamente a la venta de productos agrícolas en estado natural, gravados con tarifa 0% de IVA. ¿Debo declarar el IVA todos los meses o puedo hacerlo semestralmente?	Art. 158, Reglamento para la Aplicación de la LRTI
q17	Income tax	Para la declaración del Impuesto a la Renta del ejercicio fiscal 2026, ¿hasta qué monto de ingresos una persona natural no paga impuesto (fracción básica exenta)?	Tabla de Impuesto a la Renta Personas Naturales, Año 2026, Resolución NAC-DGERCGC25-00000043

All appendix sections are cited in the main text (see Section 3.6).

References

1. Servicio de Rentas Internas. *Resolución Nro. NAC-DGERCGC24-00000013*; SRI: Quito, Ecuador, 2024. Available online: https://www.sri.gob.ec/o/sri-portlet-biblioteca-alfresco-internet/descargar?id=6b8588f2-a4bf-44bb-ac40-085391ba2aed&nombre=NAC-DGERCGC24-00000013.pdf (accessed on 10 August 2026).
2. Li, D.; Niu, Y.; Ai, Y.; Zou, X.; Qi, B.; Liu, J. T-GRAG: A Dynamic GraphRAG Framework for Resolving Temporal Conflicts and Redundancy in Knowledge Retrieval. In Proceedings of the ACM International Conference on Information and Knowledge Management, Dublin, Ireland, 27–31 October 2025; pp. 11880–11889. [CrossRef]
3. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Dong, Z.; Hou, Y.; Zhang, B.; Min, Y.; Zhang, J.; Liu, P.; et al. A Survey of Large Language Models. *Front. Comput. Sci.* **2026**, *20*, 12. [CrossRef]
4. Bubeck, S.; Chadrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y.T.; Li, Y.; Lundberg, S.; et al. Sparks of Artificial General Intelligence: Early Experiments with GPT-4. *arXiv* **2023**, arXiv:2303.12712.
5. Gao, Y.; Xiong, Y.; Zhong, Y.; Bi, Y.; Xue, M.; Wang, H. Synergizing RAG and Reasoning: A Systematic Review. *arXiv* **2025**, arXiv:2504.15909.
6. Gao, Y. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv* **2026**, arXiv:2312.10997.
7. Yan, S.-Q.; Gu, J.-C.; Zhu, Y.; Ling, Z.-H. Corrective Retrieval Augmented Generation. *arXiv* **2024**, arXiv:2401.15884.
8. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *arXiv* **2023**, arXiv:2310.11511. [CrossRef]

9. Peng, B.; Zhu, Y.; Liu, Y.; Bo, X.; Shi, H.; Hong, C.; Zhang, Y.; Tang, S. Graph Retrieval-Augmented Generation: A Survey. *ACM Trans. Inf. Syst.* **2026**, *44*, 2. [CrossRef]
10. Mei, L.; Chen, C.; Mo, S.; Yang, Z.; Cloud Bu, H. A Survey on Multimodal Retrieval-Augmented Generation. *arXiv* **2025**, arXiv:2504.08748.
11. How, S.K.; Ong, L.Y.; Leow, M.C. Enhance Multimodal Retrieval-Augmented Generation Using Multimodal Knowledge Graph. *Emerg. Sci. J.* **2025**, *9*, 3349–3361. [CrossRef]
12. Dang, Q.V.; Nguyen, N.S.A.; Vo, T.B.D. HierFinRAG—Hierarchical Multimodal RAG for Financial Document Understanding. *Informatics* **2026**, *13*, 30. [CrossRef]
13. Zhang, Z.; Shen, H.; Zhao, T.; Guan, Z.; Chen, B.; Wang, Y.; Xu, J.; Cai, Y.; Shang, Y.; Yin, J. ImageRAG: Enhancing Ultra High-Resolution Remote Sensing Imagery Analysis with ImageRAG. *IEEE Geosci. Remote Sens. Mag.* **2025**, *13*, 369–394. [CrossRef]
14. Papageorgiou, G.; Sarlis, V.; Maragoudakis, M.; Tjortjis, C. Hybrid Multi-Agent GraphRAG for E-Government: Towards a Trustworthy AI Assistant. *Appl. Sci.* **2025**, *15*, 6315. [CrossRef]
15. Hu, Y.; Lei, Z.; Zhang, Z.; Pan, B.; Ling, C.; Zhao, L. GRAG: Graph Retrieval-Augmented Generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*; Association for Computational Linguistics: Albuquerque, NM, USA, 2025; pp. 4145–4157. [CrossRef]
16. Singh, A.; Ehtesham, A.; Kumar, S.; Khoei, T.T.; Vasilakos, A.V. Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG. *arXiv* **2026**, arXiv:2501.09136.
17. Hu, Y.; Xuan, W.; Zhou, Q.; Li, Z.; Li, Y.; Hu, J.; Fang, F. A Self-Correcting Agentic Graph RAG for Clinical Decision Support in Hepatology. *Front. Med.* **2025**, *12*, 1716327. [CrossRef] [PubMed]
18. Lin, J.; Jiang, C.; Lin, X.; Zhang, R.; Zhu, X.; Liu, J.; Tang, C.; Du, Y.; Gao, S.; Ning, J.; et al. EvoGraph-R1: Self-Evolving Multimodal Knowledge Hypergraphs for Agentic Retrieval. *arXiv* **2026**, arXiv:2607.12764.
19. Hebbar, K.S.; Patel, P.; Sepuri, S. Graph-Augmented AI Systems for Deep Reasoning in Enterprise Codebases and Regulatory Documents. In *Proceedings of the Global Conference on Sustainable Energy Systems, Smart Electronics and Intelligent Computing (GCSESEIC 2025), Chennai, India, 11–13 November 2025*; Vijayaragavan, S.P., Ed.; Atlantis Press: Dordrecht, The Netherlands, 2026; Volume 297, pp. 708–722. [CrossRef] [PubMed]
20. Huang, Y.; Huang, J.X. A Survey on Retrieval-Augmented Text Generation for Large Language Models. *ACM Comput. Surv.* **2026**, *58*, 1–38. [CrossRef]
21. Cui, J.; Ning, M.; Li, Z.; Chen, B.; Yan, Y.; Li, H.; Ling, B.; Tian, Y.; Yuan, L. ChatLaw: A Multi-Agent Collaborative Legal Assistant with Knowledge Graph Enhanced Mixture-of-Experts Large Language Model. *arXiv* **2024**, arXiv:2306.16092.
22. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [CrossRef]
23. Hevner, A.R. A Three Cycle View of Design Science Research. *Scand. J. Inf. Syst.* **2007**, *19*, 87–92.
24. Snyder, H. Literature Review as a Research Methodology: An Overview and Guidelines. *J. Bus. Res.* **2019**, *104*, 333–339. [CrossRef]
25. Grant, M.J.; Booth, A. A Typology of Reviews: An Analysis of 14 Review Types and Associated Methodologies. *Health Inf. Libr. J.* **2009**, *26*, 91–108. [CrossRef] [PubMed]
26. Mbabu, A.; Underwood, J.; Munir, M.; Saad, A. A Comparative Analysis of Methodological Adoption, Adaptation and Implementation of Design Science Research (DSR) across Engineering Disciplines. *Int. J. Ind. Eng. Oper. Manag.* **2026**, 1–19. [CrossRef]
27. Fathullah, M.A.; Subbarao, A.; Ahmad, A.; Muthaiyah, S. Merging Qualitative Design Methods with Design Science. *Int. J. Qual. Methods* **2025**, *24*, 1–11. [CrossRef]
28. Cho, Z.H.; Osvaldo, M.; Doloi, S.; Das, M.; Goh, J.C.; Tan, B.S.; Wang, J.; Li, Y.; Xiao, X.; Joshi, A.; et al. Large Language Models in Materials Science: Assessing RAG Evaluation Frameworks through Graphene Synthesis. *RSC Adv.* **2026**, *16*, 11306–11313. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.